Sofic (and Effective) Subshifts on f.g. Groups

Lecture 2: Domino Problem, Part I: Wang tiles.

Nathalie Aubrun

LIP, ENS de Lyon, CNRS

December 16, 2014

Introduction

Mini-course divided into 4 lectures

- Lecture 1: SD on f.g. groups: a computational approach.
- **Lecture 2: Domino Problem, Part I: Wang tiles.**
- Lecture 3: Domino Problem, Part II: f.g. groups.
- Lecture 4: Effective subshifts.

Lecture 2: Domino Problem, Part I: Wang tiles.

FO Logic and the $\forall\exists\forall$ fragment

- Variables $(x, y, z, \dots)$, predicates $(P(x), Q(y, y), \dots)$.
- Quantify over ***variables***.
- Formula ψ : $\forall x \exists y, Q(x, y)$, $\exists x \forall y, P(y) \Rightarrow Q(y, x)$,...

FO Logic and the ∀∃∀ fragment

- Variables $(x, y, z, \dots)$, predicates $(P(x), Q(y, y), \dots)$.
- Quantify over ***variables***.
- Formula ψ : $\forall x \exists y, Q(x, y)$, $\exists x \forall y, P(y) \Rightarrow Q(y, x)$,...

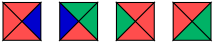
Study the unsolvability of the ∀∃∀-prefix class:

Satisfiability problem for ∀∃∀

Input: ψ a ∀∃∀-formula
Output: **Yes** if there exists a model $\mathfrak{M} \models \psi$, **No** otherwise.

Wang tiles model and the Domino Problem

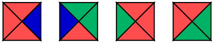
Finite set of Wang tiles τ (infinitely many copies of each tile)



Local adjacency rules

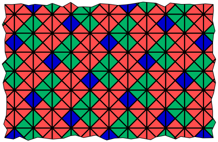
Wang tiles model and the Domino Problem

Finite set of Wang tiles τ (infinitely many copies of each tile)



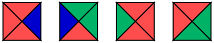
Local adjacency rules

Example of tiling by τ



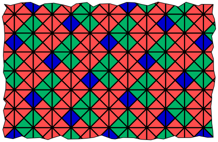
Wang tiles model and the Domino Problem

Finite set of Wang tiles τ (infinitely many copies of each tile)



Local adjacency rules

Example of tiling by τ



Domino Problem

Input: A finite set of Wang tiles τ
Output: **Yes** if there exists a valid tiling by τ, **No** otherwise.

Domino Problem and the ∀∃∀ fragment (I)

How to formalize tilings by τ in FO logics ? (= build a theory)

- FO variables: points in $\mathbb{Z}^2$
- Model $\mathfrak{M}$: configuration t in $\tau^{\mathbb{Z}^2}$
- Binary predicates $\{\mathbf{P}_i : i \in \tau\}$: $\mathbf{P}$  (x, y) is true iff $t_{(x,y)} =$ 

Domino Problem and the ∀∃∀ fragment (II)

Define H and V the subsets of $\tau \times \tau$ that code the horizontal and vertical allowed adjacencies.

Domino Problem and the ∀∃∀ fragment (II)

Define H and V the subsets of $\tau \times \tau$ that code the horizontal and vertical allowed adjacencies.

Let ψ_τ be the MSO formula $\forall x \exists x' \forall y \ \phi_\tau$ where $\phi_\tau(x, x', y)$ is

$$\underbrace{\bigwedge_{i \neq j} \neg \left(\mathbf{P}_i(x,y) \wedge \mathbf{P}_j(x,y)\right)}_{\text{at most one tile at each }(x,y)} \ \wedge$$

$$\underbrace{\bigvee_{(i,j)\in H} \left(\mathbf{P}_i(x,y) \wedge \mathbf{P}_j(x',y)\right)}_{\text{every column has a right neighbor}} \ \wedge \ \underbrace{\bigvee_{(i,j)\in V} \left(\mathbf{P}_i(y,x) \wedge \mathbf{P}_j(y,x')\right)}_{\text{every row has a top neighbor}}$$

Domino Problem and the ∀∃∀ fragment (II)

Define H and V the subsets of $\tau \times \tau$ that code the horizontal and vertical allowed adjacencies.

Let ψ_τ be the MSO formula $\forall x \exists x' \forall y\ \phi_\tau$ where $\phi_\tau(x, x', y)$ is

$$\underbrace{\bigwedge_{i \neq j} \neg (\mathbf{P}_i(x,y) \wedge \mathbf{P}_j(x,y))}_{\text{at most one tile at each } (x,y)} \wedge$$

$$\underbrace{\bigvee_{(i,j) \in H} (\mathbf{P}_i(x,y) \wedge \mathbf{P}_j(x',y))}_{\text{every column has a right neighbor}} \wedge \underbrace{\bigvee_{(i,j) \in V} (\mathbf{P}_i(y,x) \wedge \mathbf{P}_j(y,x'))}_{\text{every row has a top neighbor}}$$

Proposition

ψ_τ has a model $\Leftrightarrow \exists$ tiling of $\mathbb{N} \times \mathbb{N}$ by τ.

Remark: $\exists$ tiling of $\mathbb{N} \times \mathbb{N}$ by $\tau \Leftrightarrow \exists$ tiling of $\mathbb{Z} \times \mathbb{Z}$ by τ (by König's lemma).

Domino Problem and the $\forall\exists\forall$ fragment (III)

Putting everything together:

Comparison of Decidability

- If Satisfiability of $\forall\exists\forall$ is decidable, then Domino Problem is decidable.
- If Domino Problem is undecidable, then Satisfiability of $\forall\exists\forall$ is undecidable.

Periodicity in $\mathbb{Z}^2$ (I)

Reminder

- A configuration $x \in A^{\mathbb{Z}^2}$ is **weakly periodic** if its stabilizer is infinite.

  $\Leftrightarrow$ *x admits a non-trivial direction $\overrightarrow{u}$ of periodicity.*

- A configuration $x \in A^{\mathbb{Z}^2}$ is **strongly periodic** if its stabilizer is of finite index in $\mathbb{Z}^2$: $[\mathbb{Z}^2 : \mathrm{Stab}(x)] < \infty$.

  $\Leftrightarrow$ *x admits two non-colinear directions $\overrightarrow{u}, \overrightarrow{v}$ of periodicity.*

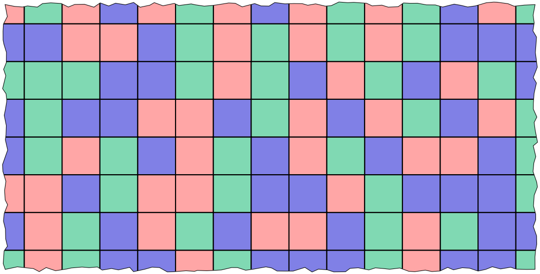
- A subshift $X \subset A^{\mathbb{Z}^2}$ is **weakly aperiodic** (resp. **strongly aperiodic**) if it contains no strongly periodic (resp. weakly periodic) configuration.

Periodicity in $\mathbb{Z}^2$ (II)

Proposition

Any $\mathbb{Z}^2$-SFT that contains a weakly periodic configuration also contains a strongly periodic configuration.

Proof: Let x be a configuration with period $\overrightarrow{u}$.

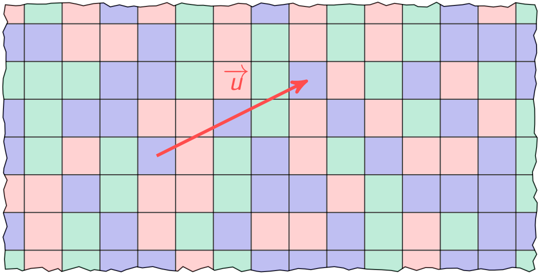


Periodicity in $\mathbb{Z}^2$ (II)

Proposition

Any $\mathbb{Z}^2$-SFT that contains a weakly periodic configuration also contains a strongly periodic configuration.

Proof: Let x be a configuration with period $\overrightarrow{u}$.

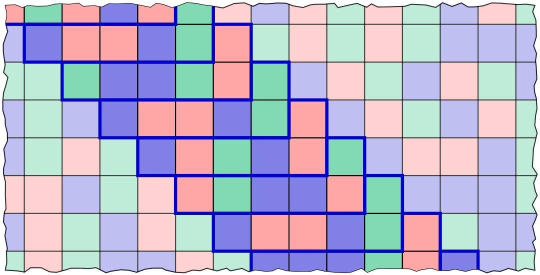


Periodicity in $\mathbb{Z}^2$ (II)

Proposition

Any $\mathbb{Z}^2$-SFT that contains a weakly periodic configuration also contains a strongly periodic configuration.

Proof: Let x be a configuration with period $\overrightarrow{u}$.

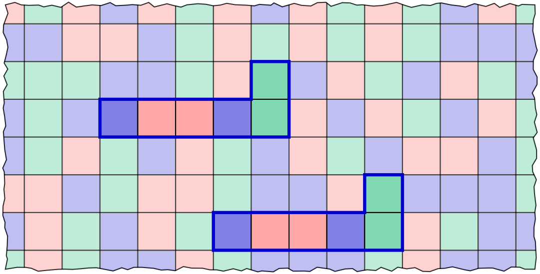


Periodicity in $\mathbb{Z}^2$ (II)

Proposition

Any $\mathbb{Z}^2$-SFT that contains a weakly periodic configuration also contains a strongly periodic configuration.

Proof: Let x be a configuration with period $\overrightarrow{u}$.

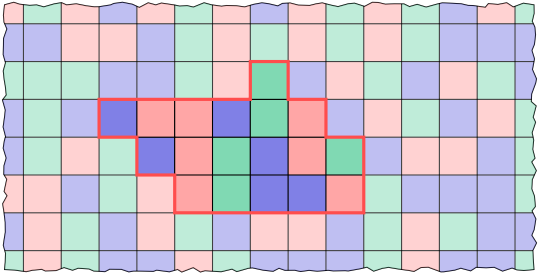


Periodicity in $\mathbb{Z}^2$ (II)

Proposition

Any $\mathbb{Z}^2$-SFT that contains a weakly periodic configuration also contains a strongly periodic configuration.

Proof: Let x be a configuration with period $\overrightarrow{u}$.

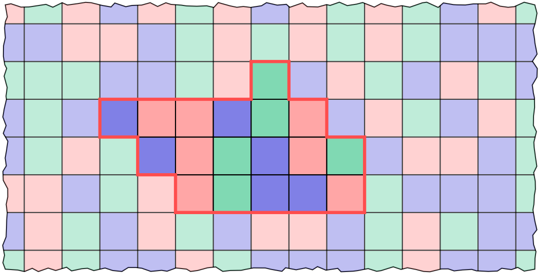


Periodicity in $\mathbb{Z}^2$ (II)

Proposition

Any $\mathbb{Z}^2$-SFT that contains a weakly periodic configuration also contains a strongly periodic configuration.

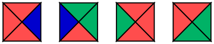
Proof: Let x be a configuration with period $\overrightarrow{u}$.



Consequence: On $\mathbb{Z}^2$, weakly aperiodic SFT are strongly aperiodic !

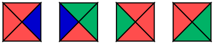
Semi-algorithm for periodicity (I)

Let τ be a finite set of Wang tiles.

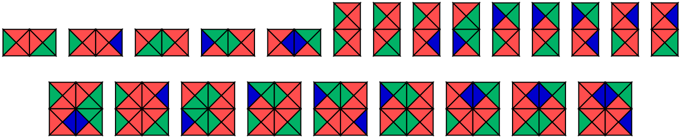


Semi-algorithm for periodicity (I)

Let τ be a finite set of Wang tiles.

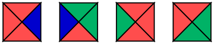


It is easy to generate, for every integers $n, m \in \mathbb{N}$, all locally admissible patterns of size $n \times m$.

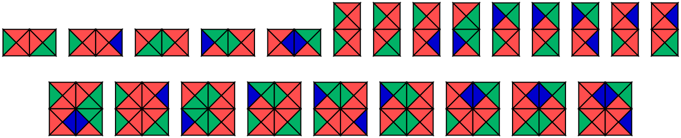


Semi-algorithm for periodicity (I)

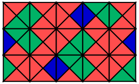
Let τ be a finite set of Wang tiles.



It is easy to generate, for every integers $n, m \in \mathbb{N}$, all locally admissible patterns of size $n \times m$.



If you find a locally admissible pattern with ***matching edges***, then τ tiles the plane periodically.



Semi-algorithm for periodicity (II)

Semi-algorithm:

1. gives a pattern that tiles the plane periodically if it exists
2. loops otherwise

Semi-algorithm for periodicity (II)

Semi-algorithm:

1. gives a pattern that tiles the plane periodically if it exists
2. loops otherwise

Questions:

- Can you check whether the locally admissible patterns are globally admissible ?
- Is it true that if τ admits no periodic pattern, then τ does not tile the plane ?

Wang's conjecture and the tiling problem

Wang's conjecture (1961)

If a set of Wang tiles can tile the plane, then they can always be arranged to do so periodically.

Wang's conjecture and the tiling problem

Wang's conjecture (1961)

If a set of Wang tiles can tile the plane, then they can always be arranged to do so periodically.

Suppose Wang's conjecture is true. Then you can decide the tiling problem !

Semi-algorithm 1:

1. gives a pattern that tiles the plane periodically if it exists
2. loops otherwise

Semi-algorithm 2:

1. gives an integer n so that $[1;n] \times [1;n]$ cannot be tiled if it exists
2. loops otherwise

Back to Wang's conjecture

Wang's conjecture (1961)

If a set of Wang tiles can tile the plane, then they can always be arranged to do so periodically.

Back to Wang's conjecture

Wang's conjecture (1961)

If a set of Wang tiles can tile the plane, then they can always be arranged to do so periodically.

- Refuted by Berger (Wang's student) in 1966: he exhibited an aperiodic set of 20426 Wang tiles.

Back to Wang's conjecture

Wang's conjecture (1961)

If a set of Wang tiles can tile the plane, then they can always be arranged to do so periodically.

- Refuted by Berger (Wang's student) in 1966: he exhibited an aperiodic set of 20426 Wang tiles.
- Robinson (1971): aperiodic set of 56 Wang tiles (32 square tiles)!

Back to Wang's conjecture

Wang's conjecture (1961)

If a set of Wang tiles can tile the plane, then they can always be arranged to do so periodically.

- Refuted by Berger (Wang's student) in 1966: he exhibited an aperiodic set of 20426 Wang tiles.
- Robinson (1971): aperiodic set of 56 Wang tiles (32 square tiles)!
- Kari (1996): aperiodic set of 14 Wang tiles !

Back to Wang's conjecture

Wang's conjecture (1961)

If a set of Wang tiles can tile the plane, then they can always be arranged to do so periodically.

- Refuted by Berger (Wang's student) in 1966: he exhibited an aperiodic set of 20426 Wang tiles.
- Robinson (1971): aperiodic set of 56 Wang tiles (32 square tiles)!
- Kari (1996): aperiodic set of 14 Wang tiles !
- Culik (1996): aperiodic set of 13 Wang tiles !

Back to Wang's conjecture

Wang's conjecture (1961)

If a set of Wang tiles can tile the plane, then they can always be arranged to do so periodically.

- Refuted by Berger (Wang's student) in 1966: he exhibited an aperiodic set of 20426 Wang tiles.
- Robinson (1971): aperiodic set of 56 Wang tiles (32 square tiles)!
- Kari (1996): aperiodic set of 14 Wang tiles !
- Culik (1996): aperiodic set of 13 Wang tiles !
- . . . suspicions about a set of 11 Wang tiles . . .

Back to Wang's conjecture

Wang's conjecture (1961)

If a set of Wang tiles can tile the plane, then they can always be arranged to do so periodically.

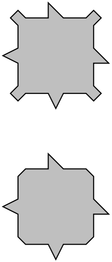
- Refuted by Berger (Wang's student) in 1966: he exhibited an aperiodic set of 20426 Wang tiles.
- Robinson (1971): aperiodic set of 56 Wang tiles (32 square tiles)!
- Kari (1996): aperiodic set of 14 Wang tiles !
- Culik (1996): aperiodic set of 13 Wang tiles !
- ...suspicions about a set of 11 Wang tiles ...

Remark

More than that, all these constructions actually show the undecidability of the tiling problem (from which you deduce the existence of an aperiodic tileset).

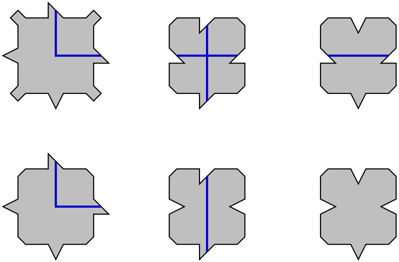
Robinson tileset

The Robinson tileset, where tiles can be rotated and reflected.



Robinson tileset

The Robinson tileset, where tiles can be rotated and reflected.



Existence of a valid tiling

Proposition

Robinson's tileset admits at least one valid tiling.

Existence of a valid tiling

Proposition

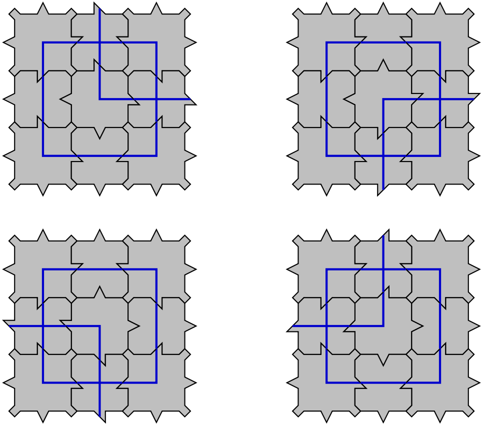
Robinson's tileset admits at least one valid tiling.

Proof:

- We can build arbitrarily large patterns (called macro-tiles) with the same structure.
- We thus conclude by compactness.

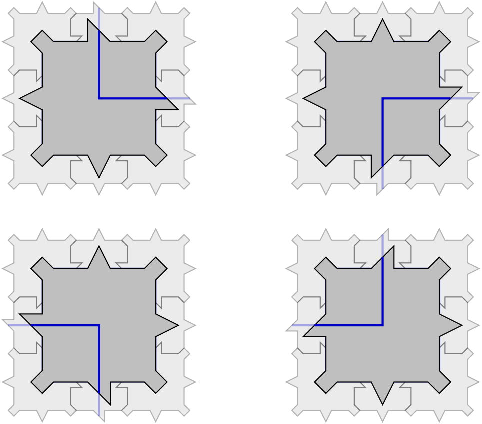
Macro-tiles of level 1

Macro-tiles of level 1.



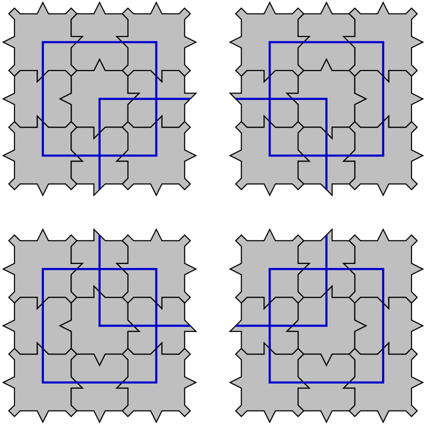
Macro-tiles of level 1

Macro-tiles of level 1.

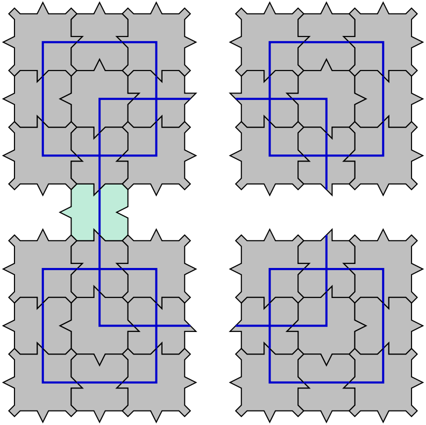


They behave like large .

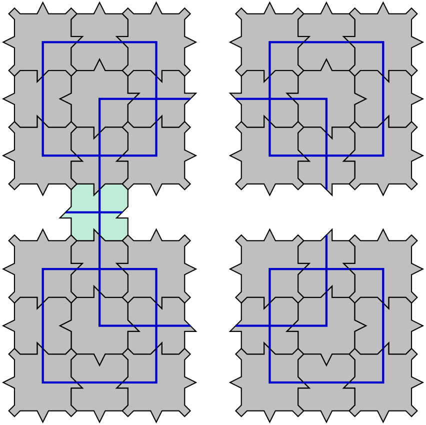
From macro-tiles of level 1 to macro-tiles of level 2



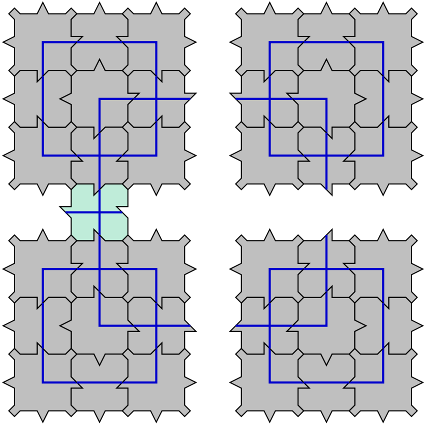
From macro-tiles of level 1 to macro-tiles of level 2



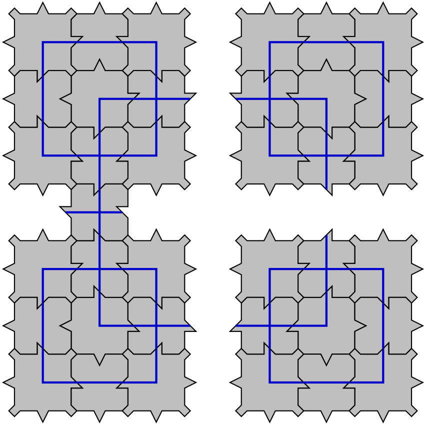
From macro-tiles of level 1 to macro-tiles of level 2



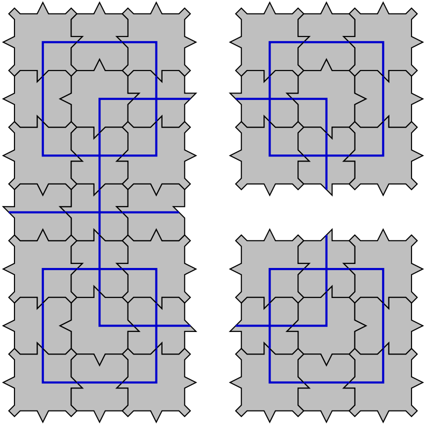
From macro-tiles of level 1 to macro-tiles of level 2



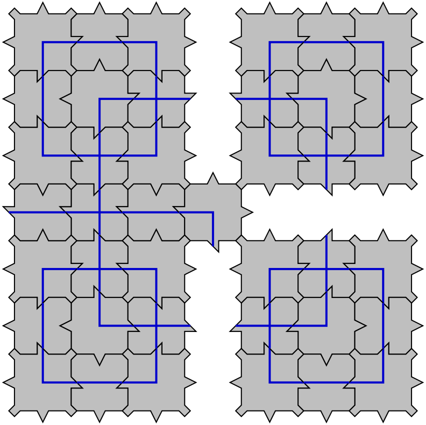
From macro-tiles of level 1 to macro-tiles of level 2



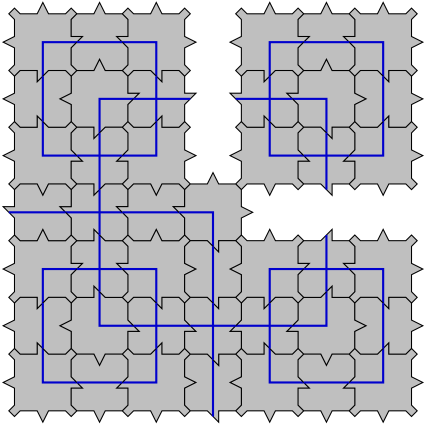
From macro-tiles of level 1 to macro-tiles of level 2



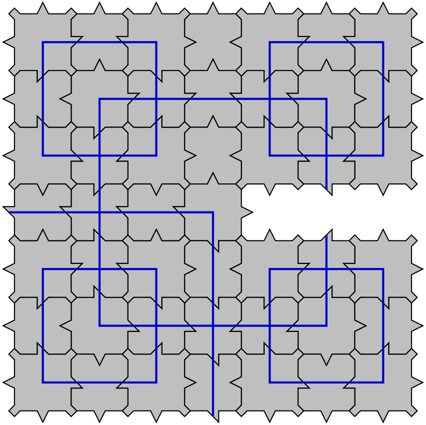
From macro-tiles of level 1 to macro-tiles of level 2



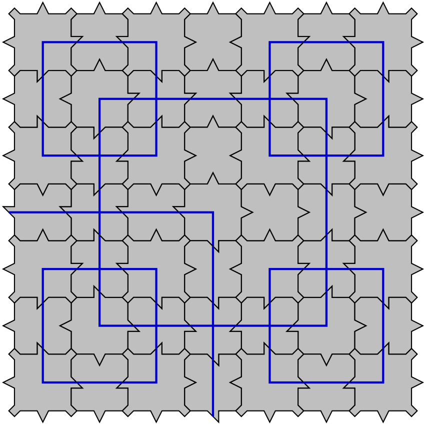
From macro-tiles of level 1 to macro-tiles of level 2



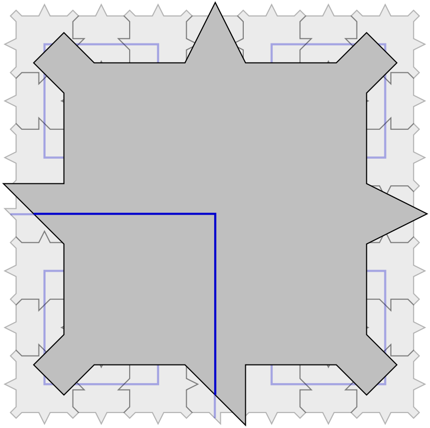
From macro-tiles of level 1 to macro-tiles of level 2



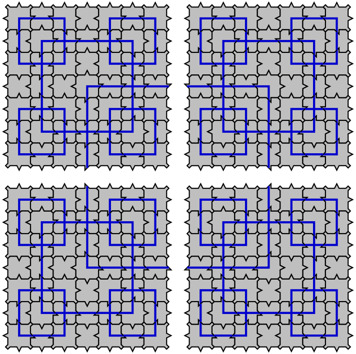
From macro-tiles of level 1 to macro-tiles of level 2



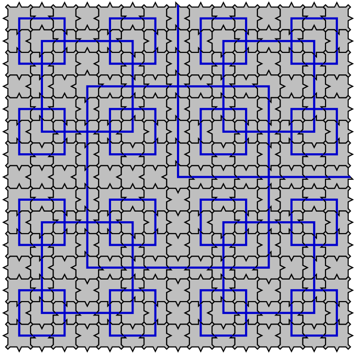
From macro-tiles of level 1 to macro-tiles of level 2



From macro-tiles of level n to macro-tiles of level $n+1$



$\Rightarrow$



This valid tiling is aperiodic

Proposition

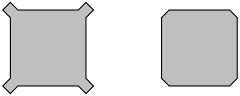
The valid tiling x obtained by compactness is aperiodic.

Proof:

- Centers of macro-tiles of level n are located on the lattice $2^{n+1}\mathbb{Z} \times 2^{n+1}\mathbb{Z}$.
- Suppose x admits a direction of periodicity $\overrightarrow{u}$.
- Then there exists an integer n s.t. $2^{n+1} > \|\overrightarrow{u}\|$.
- Thus a macro-tile of level n overlaps with its translation.
- $\Rightarrow$ contradiction.

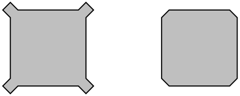
All valid tilings are aperiodic (I)

The two forms in Robinson tileset, cross (bumpy corners) and arms (dented corners).

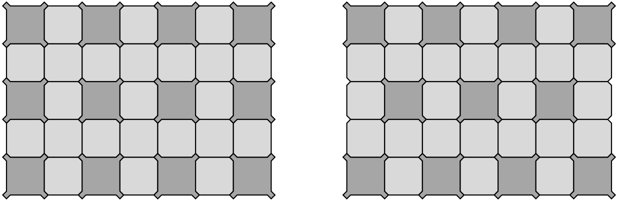


All valid tilings are aperiodic (I)

The two forms in Robinson tileset, cross (bumpy corners) and arms (dented corners).

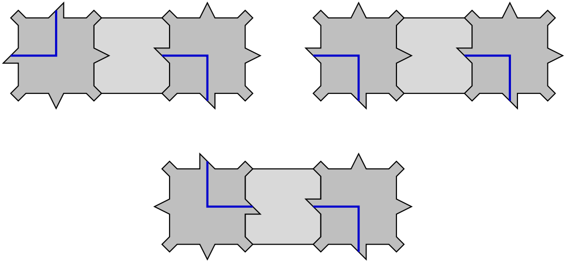


Obviously, two crosses cannot be in contact (neither through an edge nor a vertex) thus a cross must be surrounded by eight arms.



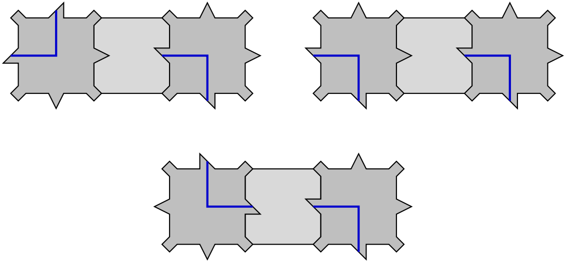
All valid tilings are aperiodic (II)

You cannot have things like

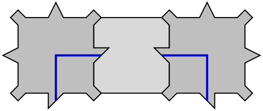


All valid tilings are aperiodic (II)

You cannot have things like

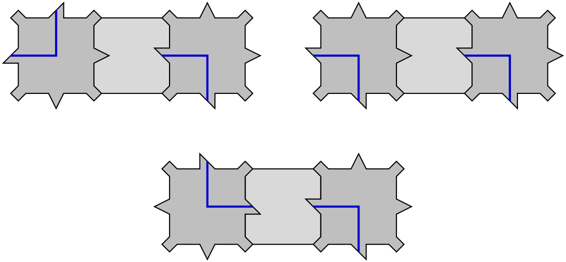


The only possibilities are thus

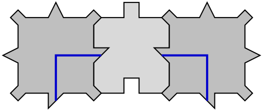


All valid tilings are aperiodic (II)

You cannot have things like

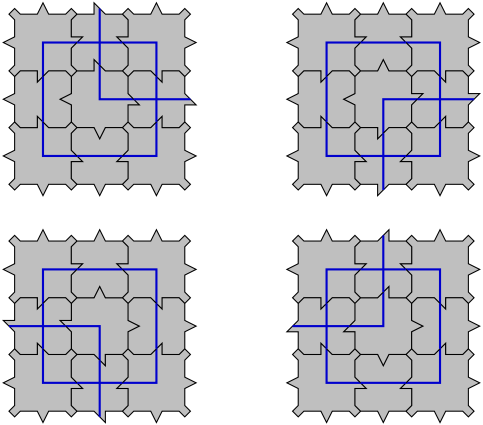


The only possibilities are thus



All valid tilings are aperiodic (III)

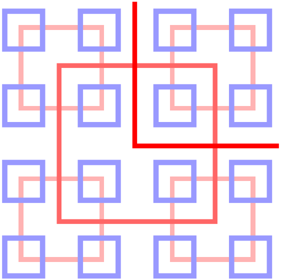
So each  is part of a macro tile of level 1



that behaves like a big , and so on. . .

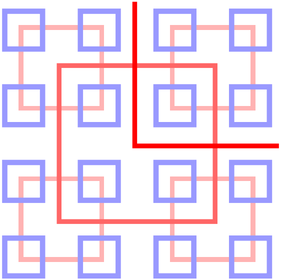
About Robinson's tiling structure

Hierarchy of squares: squares of level n are gathered by 4 to form a square of level $n+1$



About Robinson's tiling structure

Hierarchy of squares: squares of level n are gathered by 4 to form a square of level $n + 1$

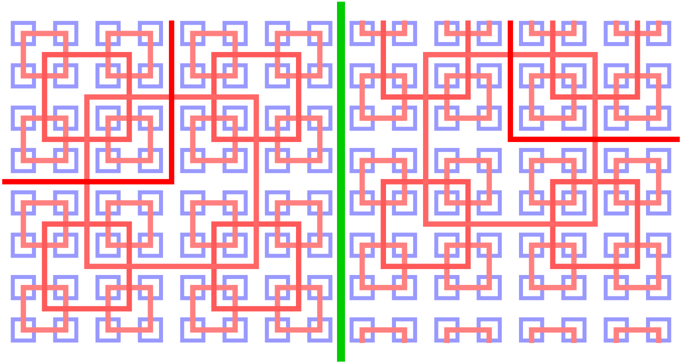


Proposition

The are uncountably many different valid tilings by the Robinson tileset.

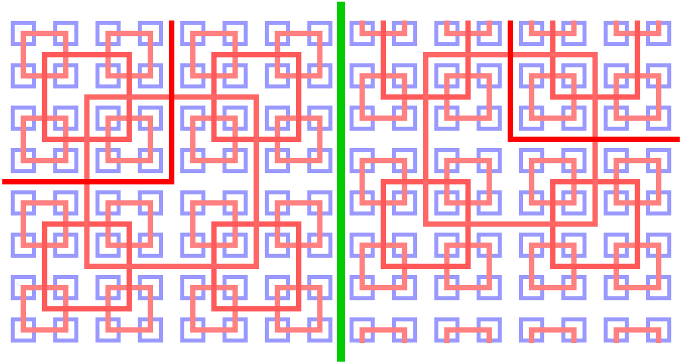
Fracture lines

Some sequences of choices (ultimately constant sequences) lead to



Fracture lines

Some sequences of choices (ultimately constant sequences) lead to



But it is possible to enrich the tiles to get rid of fracture lines ! (idea: synchronize squares of same level)

Lecture 2: Domino Problem, Part I: Wang tiles.

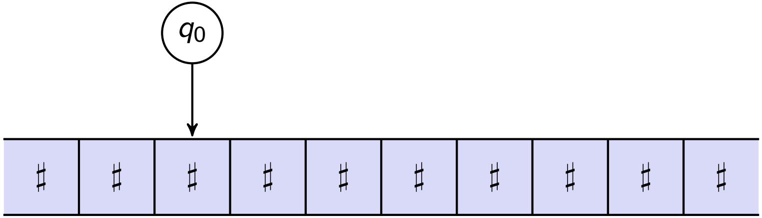
Turing machines: definition

A **Turing machine** is a tuple $\mathcal{M} = (Q, \Gamma, \sharp, q_0, \delta, Q_F)$ where

- Q is a finite set of **states**, $q_0 \in Q$ is the **initial state**,
- Γ is a finite alphabet,
- $\sharp \notin \Gamma$ is the **blank symbol**,
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \cdot, \to\}$ is the **transition function**,
- $Q_F \subset Q$ is the set of **final states**.

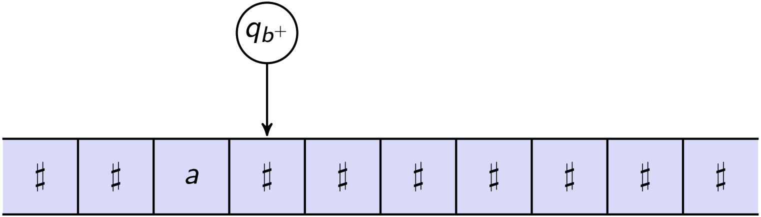
Turing machines: example

$\delta(q,x)$		Symbol x			
		a	b	$\|\|$	$\sharp$
State q	q_0	$\perp$	$\perp$	$\perp$	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	$\perp$	$(q_{b^{++}}, a, \rightarrow)$	$\perp$	$\perp$
	q_{b^+}	$\perp$	$\perp$	$\perp$	$(q_{\|\|}, b, \rightarrow)$
	$q_{b^{++}}$	$\perp$	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	$\perp$
	$q_{\|\|}$	$(q_{a^+}, a, \rightarrow)$	$(q_{\|\|}, b, \leftarrow)$	$(q_{\|\|}, \|\|, \leftarrow)$	$(q_{\|\|}, \|\|, \cdot)$



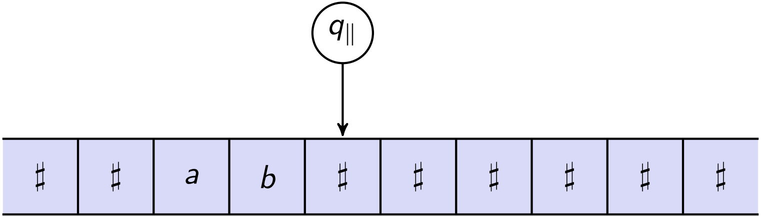
Turing machines: example

$\delta(q,x)$		Symbol x			
		a	b	$\|\|$	$\sharp$
State q	q_0	$\perp$	$\perp$	$\perp$	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	$\perp$	$(q_{b^{++}}, a, \rightarrow)$	$\perp$	$\perp$
	q_{b^+}	$\perp$	$\perp$	$\perp$	$(q_{\|\|}, b, \rightarrow)$
	$q_{b^{++}}$	$\perp$	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	$\perp$
	$q_{\|\|}$	$(q_{a^+}, a, \rightarrow)$	$(q_{\|\|}, b, \leftarrow)$	$(q_{\|\|}, \|\|, \leftarrow)$	$(q_{\|\|}, \|\|, \cdot)$



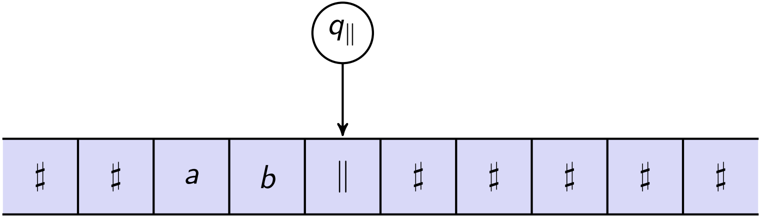
Turing machines: example

$\delta(q,x)$		Symbol x			
		a	b	$\|$	$\sharp$
State q	q_0	$\perp$	$\perp$	$\perp$	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	$\perp$	$(q_{b^{++}}, a, \rightarrow)$	$\perp$	$\perp$
	q_{b^+}	$\perp$	$\perp$	$\perp$	$(q_{\|}, b, \rightarrow)$
	$q_{b^{++}}$	$\perp$	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	$\perp$
	$q_{\|}$	$(q_{a^+}, a, \rightarrow)$	$(q_{\|}, b, \leftarrow)$	$(q_{\|}, \|, \leftarrow)$	$(q_{\|}, \|, \cdot)$



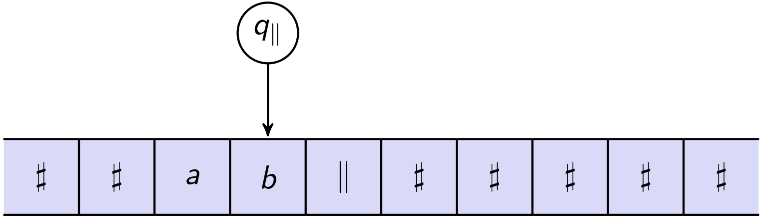
Turing machines: example

$\delta(q,x)$		Symbol x			
		a	b	$\|\|$	$\sharp$
State q	q_0	$\perp$	$\perp$	$\perp$	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	$\perp$	$(q_{b^{++}}, a, \rightarrow)$	$\perp$	$\perp$
	q_{b^+}	$\perp$	$\perp$	$\perp$	$(q_{\|\|}, b, \rightarrow)$
	$q_{b^{++}}$	$\perp$	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	$\perp$
	$q_{\|\|}$	$(q_{a^+}, a, \rightarrow)$	$(q_{\|\|}, b, \leftarrow)$	$(q_{\|\|}, \|\|, \leftarrow)$	$(q_{\|\|}, \|\|, \cdot)$



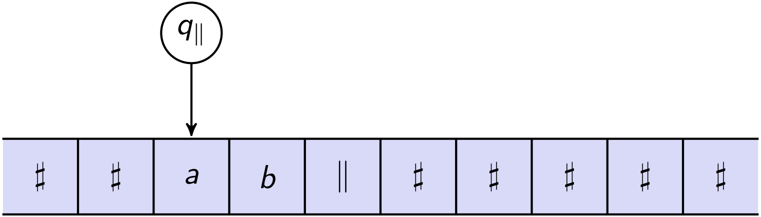
Turing machines: example

$\delta(q,x)$		Symbol x			
		a	b	$\|\|$	$\sharp$
State q	q_0	$\perp$	$\perp$	$\perp$	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	$\perp$	$(q_{b^{++}}, a, \rightarrow)$	$\perp$	$\perp$
	q_{b^+}	$\perp$	$\perp$	$\perp$	$(q_{\|\|}, b, \rightarrow)$
	$q_{b^{++}}$	$\perp$	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	$\perp$
	$q_{\|\|}$	$(q_{a^+}, a, \rightarrow)$	$(q_{\|\|}, b, \leftarrow)$	$(q_{\|\|}, \|\|, \leftarrow)$	$(q_{\|\|}, \|\|, \cdot)$



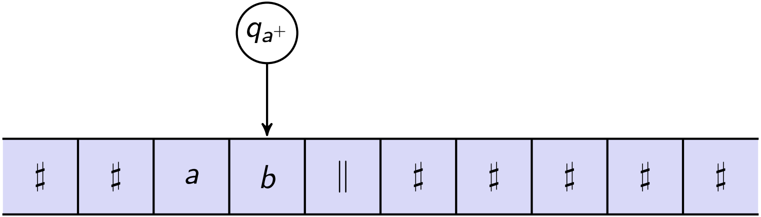
Turing machines: example

$\delta(q,x)$		Symbol x			
		a	b	$\|$	$\sharp$
State q	q_0	$\perp$	$\perp$	$\perp$	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	$\perp$	$(q_{b^{++}}, a, \rightarrow)$	$\perp$	$\perp$
	q_{b^+}	$\perp$	$\perp$	$\perp$	$(q_{\|}, b, \rightarrow)$
	$q_{b^{++}}$	$\perp$	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	$\perp$
	$q_{\|}$	$(q_{a^+}, a, \rightarrow)$	$(q_{\|}, b, \leftarrow)$	$(q_{\|}, \|, \leftarrow)$	$(q_{\|}, \|, \cdot)$



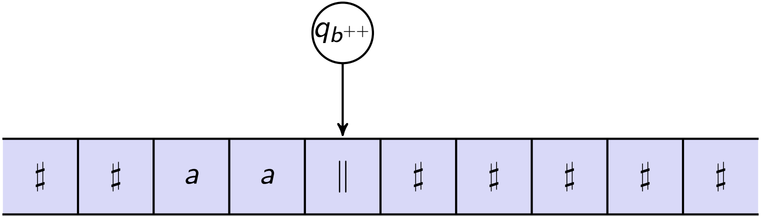
Turing machines: example

$\delta(q, x)$		Symbol x			
		a	b	$\|\|$	$\sharp$
State q	q_0	$\perp$	$\perp$	$\perp$	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	$\perp$	$(q_{b^{++}}, a, \rightarrow)$	$\perp$	$\perp$
	q_{b^+}	$\perp$	$\perp$	$\perp$	$(q_{\|\|}, b, \rightarrow)$
	$q_{b^{++}}$	$\perp$	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	$\perp$
	$q_{\|\|}$	$(q_{a^+}, a, \rightarrow)$	$(q_{\|\|}, b, \leftarrow)$	$(q_{\|\|}, \|\|, \leftarrow)$	$(q_{\|\|}, \|\|, \cdot)$



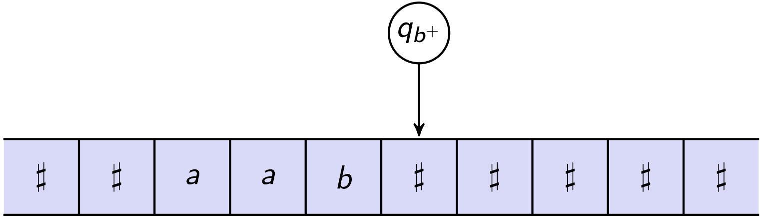
Turing machines: example

$\delta(q,x)$		Symbol x			
		a	b	$\|\|$	$\sharp$
State q	q_0	$\perp$	$\perp$	$\perp$	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	$\perp$	$(q_{b^{++}}, a, \rightarrow)$	$\perp$	$\perp$
	q_{b^+}	$\perp$	$\perp$	$\perp$	$(q_{\|\|}, b, \rightarrow)$
	$q_{b^{++}}$	$\perp$	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	$\perp$
	$q_{\|\|}$	$(q_{a^+}, a, \rightarrow)$	$(q_{\|\|}, b, \leftarrow)$	$(q_{\|\|}, \|\|, \leftarrow)$	$(q_{\|\|}, \|\|, \cdot)$



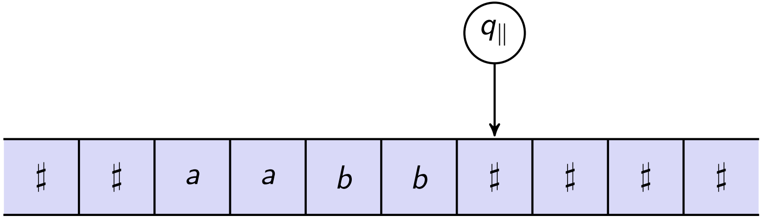
Turing machines: example

$\delta(q,x)$		Symbol x: a	b	$\|\|$	$\sharp$
State q	q_0	$\perp$	$\perp$	$\perp$	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	$\perp$	$(q_{b^{++}}, a, \rightarrow)$	$\perp$	$\perp$
	q_{b^+}	$\perp$	$\perp$	$\perp$	$(q_{\|\|}, b, \rightarrow)$
	$q_{b^{++}}$	$\perp$	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	$\perp$
	$q_{\|\|}$	$(q_{a^+}, a, \rightarrow)$	$(q_{\|\|}, b, \leftarrow)$	$(q_{\|\|}, \|\|, \leftarrow)$	$(q_{\|\|}, \|\|, \cdot)$



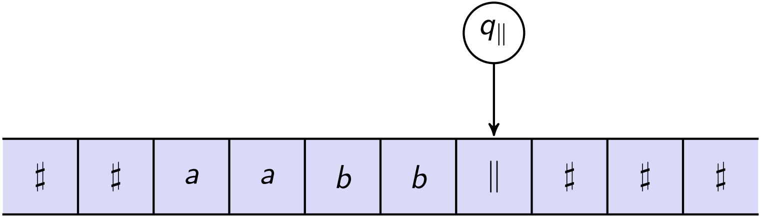
Turing machines: example

$\delta(q,x)$		Symbol x			
		a	b	$\|\|$	$\sharp$
State q	q_0	$\bot$	$\bot$	$\bot$	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	$\bot$	$(q_{b^{++}}, a, \rightarrow)$	$\bot$	$\bot$
	q_{b^+}	$\bot$	$\bot$	$\bot$	$(q_{\|\|}, b, \rightarrow)$
	$q_{b^{++}}$	$\bot$	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	$\bot$
	$q_{\|\|}$	$(q_{a^+}, a, \rightarrow)$	$(q_{\|\|}, b, \leftarrow)$	$(q_{\|\|}, \|\|, \leftarrow)$	$(q_{\|\|}, \|\|, \cdot)$



Turing machines: example

$\delta(q,x)$		Symbol x			
		a	b	$\parallel$	$\sharp$
State q	q_0	$\perp$	$\perp$	$\perp$	$(q_{b^+}, a, \rightarrow)$
	q_{a^+}	$\perp$	$(q_{b^{++}}, a, \rightarrow)$	$\perp$	$\perp$
	q_{b^+}	$\perp$	$\perp$	$\perp$	$(q_{\parallel}, b, \rightarrow)$
	$q_{b^{++}}$	$\perp$	$(q_{b^{++}}, b, \rightarrow)$	$(q_{b^+}, b, \rightarrow)$	$\perp$
	$q_{\parallel}$	$(q_{a^+}, a, \rightarrow)$	$(q_{\parallel}, b, \leftarrow)$	$(q_{\parallel}, \parallel, \leftarrow)$	$(q_{\parallel}, \parallel, \cdot)$



Turing machines: Halting Problem

Take any enumeration of Turing machines $(\mathcal{M}_i)_{i\in\mathbb{N}}$.

Halting Problem for Turing machines

Input: A Turing machine $\mathcal{M}_i$ and an input word w.
Output: **Yes** if $\mathcal{M}_i$ reaches a final state when computing on w, **No** otherwise.

Turing machines: Halting Problem

Take any enumeration of Turing machines $(\mathcal{M}_i)_{i\in\mathbb{N}}$.

Halting Problem for Turing machines

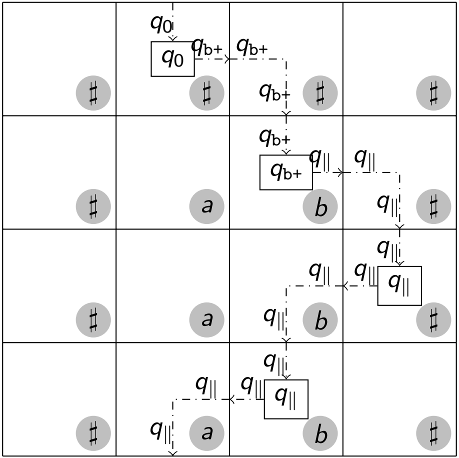
Input: A Turing machine $\mathcal{M}_i$ and an input word w.
Output: **Yes** if $\mathcal{M}_i$ reaches a final state when computing on w, **No** otherwise.

Theorem (Turing, 1936)

The Halting Problem for Turing machines is undecidable.

Proof: Diagonal argument.

TM inside Wang tilesets



Undecidability of the Domino Problem (I)

Can we reduce Domino Problem from Halting Problem ?

Idea

Build a finite tileset τ s.t. $X_\tau \neq \emptyset$ iff $\mathcal{M}$ halts on the empty input ${}^\infty\sharp^\infty$.

- Every Turing machine $\mathcal{M}$ can be associated with a finite tileset $\tau_\mathcal{M}$.
- If $\mathcal{M}$ never stops on the empty intput then $X_{\tau_\mathcal{M}}$ is non-empty.
- Unfortunately this SFT is always non-empty (blank configuration $\sharp^{\mathbb{Z}^2}$) independently from $\mathcal{M}$...

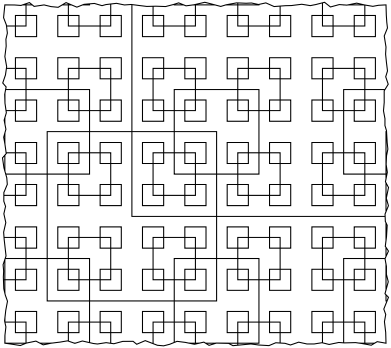
Problem

How to initialize computations ?

Undecidability of the Domino Problem (II)

Solution

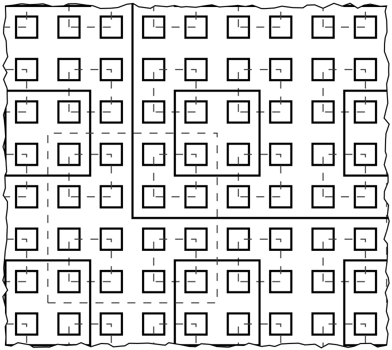
Embed Turing machine computations inside the hierarchy of squares given by Robinson's tiling.



Undecidability of the Domino Problem (II)

Solution

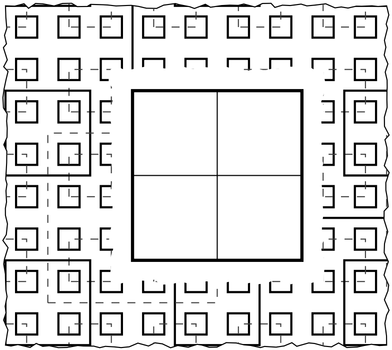
Embed Turing machine computations inside the hierarchy of squares given by Robinson's tiling.



Undecidability of the Domino Problem (II)

Solution

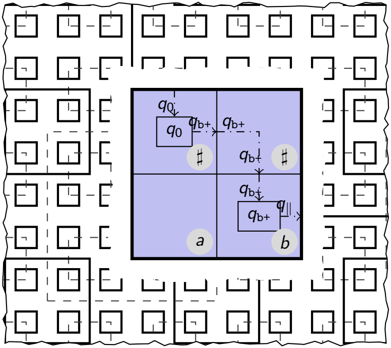
Embed Turing machine computations inside the hierarchy of squares given by Robinson's tiling.



Undecidability of the Domino Problem (II)

Solution

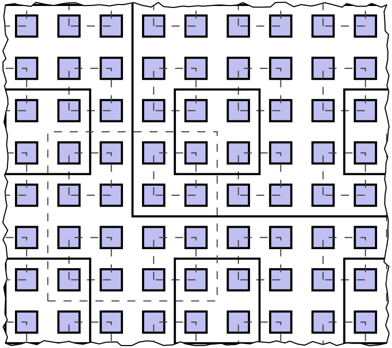
Embed Turing machine computations inside the hierarchy of squares given by Robinson's tiling.



Undecidability of the Domino Problem (II)

Solution

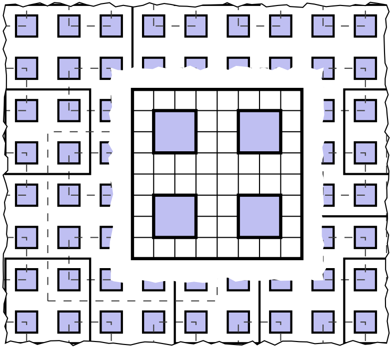
Embed Turing machine computations inside the hierarchy of squares given by Robinson's tiling.



Undecidability of the Domino Problem (II)

Solution

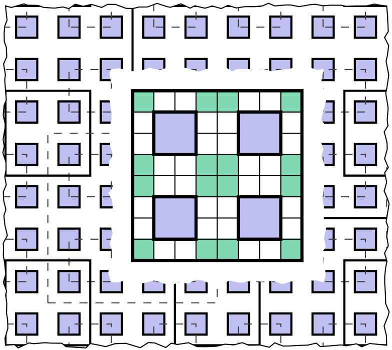
Embed Turing machine computations inside the hierarchy of squares given by Robinson's tiling.



Undecidability of the Domino Problem (II)

Solution

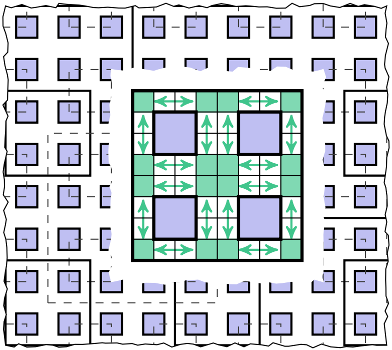
Embed Turing machine computations inside the hierarchy of squares given by Robinson's tiling.



Undecidability of the Domino Problem (II)

Solution

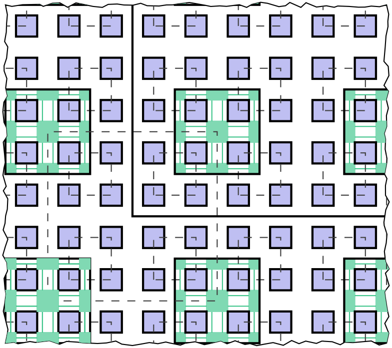
Embed Turing machine computations inside the hierarchy of squares given by Robinson's tiling.



Undecidability of the Domino Problem (II)

Solution

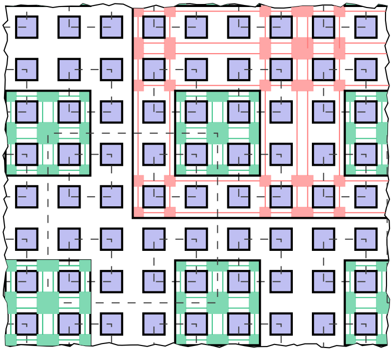
Embed Turing machine computations inside the hierarchy of squares given by Robinson's tiling.



Undecidability of the Domino Problem (II)

Solution

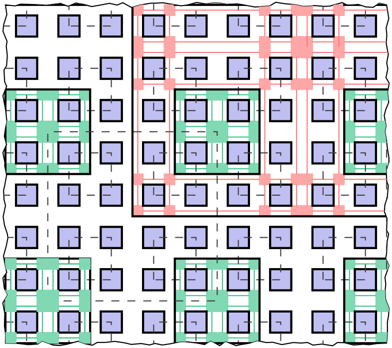
Embed Turing machine computations inside the hierarchy of squares given by Robinson's tiling.



Undecidability of the Domino Problem (II)

Solution

Embed Turing machine computations inside the hierarchy of squares given by Robinson's tiling.



Theorem (Berger, 1966)

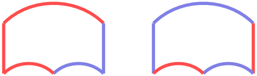
The Domino Problem is undecidable.

Lecture 2: Domino Problem, Part I: Wang tiles.

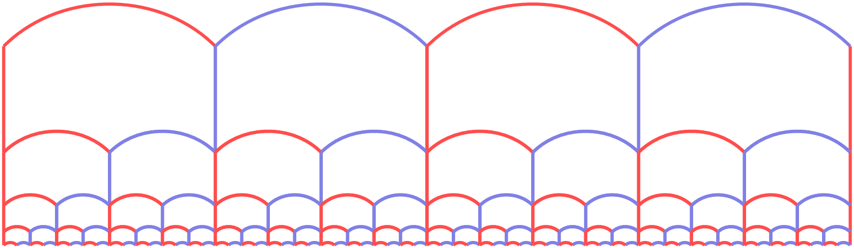
Tilings in $\mathbb{H}^2$

Wang tiles are replaced by  -tiles.

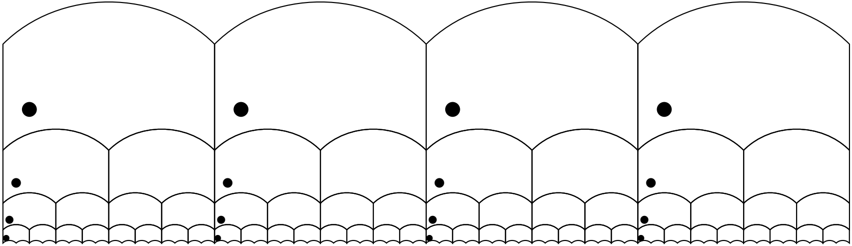
Example: Let τ be the finite tileset



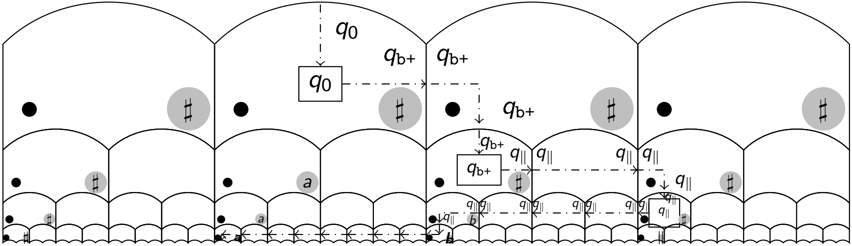
Then τ can produce tilings of $\mathbb{H}^2$



Turing machines inside  -tilesets (I)



Turing machines inside  -tilesets (II)



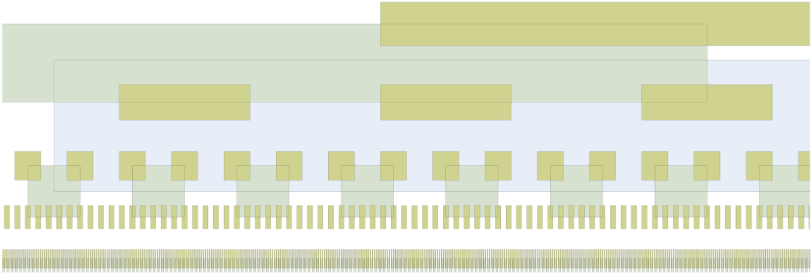
Undecidability of DP in $\mathbb{H}^2$

First proven by Kari (2007) **(see Lecture 3)** and Margenstern (2009).

Theorem

The Domino Problem is undecidable in the hyperbolic plane.

Idea: use Goodman-Strauss aperiodic hierarchical tiling of $\mathbb{H}^2$...



Conclusion

- Strong links between existence of aperiodic SFTs and Domino Problem.
- Undecidability comes from
  (i) the existence of aperiodic SFT
  (ii) encoding of Turing machines inside SFT
- Can be generalized to the hyperbolic plane.

On Thursday: what about Domino Problem on f.g. groups ?

Thank you for your attention !!