Embedding computations in tilings
(a perspective of the course)

Andrei Romashchenko

30 May 2016

What is a **tile**?

What is a **tile**?

In **this** mini-course:

Color: an element of a finite set $C = \{\cdot, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot\}$

What is a **tile**?

In **this** mini-course:

Color: an element of a finite set $C = \{\cdot, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot\}$

Wang Tile: a unit square with colored sides.

What is a **tile**?

In **this** mini-course:

Color: an element of a finite set $C = \{\cdot, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot\}$

Wang Tile: a unit square with colored sides.

i.e, an element of C^4, e.g., 

What is a **tile**?

In **this** mini-course:

Color: an element of a finite set $C = \{\cdot, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot\}$

Wang Tile: a unit square with colored sides.

i.e, an element of C^4, e.g., 

Tile set: a set $\tau \subset C^4$

What is a **tile**?

In **this** mini-course:

Color: an element of a finite set $C = \{\cdot, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot\}$

Wang Tile: a unit square with colored sides.
i.e, an element of C^4, e.g., 

Tile set: a set $\tau \subset C^4$

Tiling: a mapping $f : \mathbb{Z}^2 \to \tau$
that respects the matching rules

A shift of finite type (SFT):

A shift of finite type (SFT):

- a finite set of letters τ

A shift of finite type (SFT):

- a finite set of letters τ
- a finite set of forbidden (finite) patterns $\mathcal{F}$
- SFT: the set of all configurations $f : \mathbb{Z}^2 \to \tau$ that does not contain forbidden patterns

A shift of finite type (SFT):

- a finite set of letters τ
- a finite set of forbidden (finite) patterns $\mathcal{F}$
- SFT: the set of all configurations $f : \mathbb{Z}^2 \to \tau$ that does not contain forbidden patterns

Remark: for every set of Wang tiles τ the set of all τ-tilings is an SFT

τ-tiling:
a mapping $f : \mathbb{Z}^2 \to \tau$ that respects the local rules.

τ-tiling:
a mapping $f : \mathbb{Z}^2 \to \tau$ that respects the local rules.

$T \in \mathbb{Z}^2$ is a **period** if $f(x + T) = f(x)$ for all x.

super-classic facts:

super-classic facts:

- SFT $\sim$ tilings

super-classic facts:

- SFT $\sim$ tilings
- if you can tile arbitrarily large square, than you can tile the infinite plane (**compacteness**)

super-classic facts:

- SFT $\sim$ tilings
- if you can tile arbitrarily large square, than you can tile the infinite plane (**compacteness**)
- if there exists a τ-tiling with *one* period T, then there exists another tiling with **two** non collinear periods T_1, T_2

super-classic facts:

- SFT $\sim$ tilings
- if you can tile arbitrarily large square, than you can tile the infinite plane (**compacteness**)
- if there exists a τ-tiling with *one* period T, then there exists another tiling with **two** non collinear periods T_1, T_2
- there exist tile sets τ s.t. all τ-tilings are **aperiodic**

super-classic facts:

- SFT $\sim$ tilings
- if you can tile arbitrarily large square, than you can tile the infinite plane (**compacteness**)
- if there exists a τ-tiling with *one* period T, then there exists another tiling with **two** non collinear periods T_1, T_2
- there exist tile sets τ s.t. all τ-tilings are **aperiodic**
- there exists a tile set τ s.t. all τ-tilings are **non-computable**

super-classic facts:

- SFT $\sim$ tilings
- if you can tile arbitrarily large square, than you can tile the infinite plane (**compacteness**)
- if there exists a τ-tiling with *one* period T, then there exists another tiling with **two** non collinear periods T_1, T_2
- there exist tile sets τ s.t. all τ-tilings are **aperiodic**
- there exists a tile set τ s.t. all τ-tilings are **non-computable**
- given a tile set τ we cannot algorithmically decide whether there exists a τ-tiling of $\mathbb{Z}^2$

super-classic facts:

- SFT $\sim$ tilings
- if you can tile arbitrarily large square, than you can tile the infinite plane (**compacteness**)
- if there exists a τ-tiling with *one* period T, then there exists another tiling with **two** non collinear periods T_1, T_2
- there exist tile sets τ s.t. all τ-tilings are **aperiodic**
- there exists a tile set τ s.t. all τ-tilings are **non-computable**
- given a tile set τ we cannot algorithmically decide whether there exists a τ-tiling of $\mathbb{Z}^2$

other super-classic facts:

super-classic facts:

- SFT $\sim$ tilings
- if you can tile arbitrarily large square, than you can tile the infinite plane (**compacteness**)
- if there exists a τ-tiling with *one* period T, then there exists another tiling with **two** non collinear periods T_1, T_2
- there exist tile sets τ s.t. all τ-tilings are **aperiodic**
- there exists a tile set τ s.t. all τ-tilings are **non-computable**
- given a tile set τ we cannot algorithmically decide whether there exists a τ-tiling of $\mathbb{Z}^2$

other super-classic facts:

- in any *reasonable* programming language you can write a program π that prints its own text

super-classic facts:

- SFT $\sim$ tilings
- if you can tile arbitrarily large square, than you can tile the infinite plane (**compacteness**)
- if there exists a τ-tiling with *one* period T, then there exists another tiling with **two** non collinear periods T_1, T_2
- there exist tile sets τ s.t. all τ-tilings are **aperiodic**
- there exists a tile set τ s.t. all τ-tilings are **non-computable**
- given a tile set τ we cannot algorithmically decide whether there exists a τ-tiling of $\mathbb{Z}^2$

other super-classic facts:

- in any *reasonable* programming language you can write a program π that prints its own text
- in any *reasonable* programming language you may assume that your program has an access to its own text

super-classic facts:

- SFT $\sim$ tilings
- if you can tile arbitrarily large square, than you can tile the infinite plane (**compacteness**)
- if there exists a τ-tiling with *one* period T, then there exists another tiling with **two** non collinear periods T_1, T_2
- there exist tile sets τ s.t. all τ-tilings are **aperiodic**
- there exists a tile set τ s.t. all τ-tilings are **non-computable**
- given a tile set τ we cannot algorithmically decide whether there exists a τ-tiling of $\mathbb{Z}^2$

other super-classic facts:

- in any *reasonable* programming language you can write a program π that prints its own text
- in any *reasonable* programming language you may assume that your program has an access to its own text
- any effective (polynomial time) real-life algorithm can be performed by a Turing machine in polynomial time

This mini-course:

This mini-course:

Two techniques

This mini-course:

Two techniques of *embedding a computation in a tiling*

- from **self-referential programs** to **self-similar** tilings

This mini-course:

Two techniques of *embedding a computation in a tiling*

- from **self-referential programs** to **self-similar** tilings
  [goes back to J. von Neumann]

This mini-course:

Two techniques of *embedding a computation in a tiling*

- from **self-referential programs** to **self-similar** tilings
  [goes back to J. von Neumann]
- from arithmetic in **Sturmian numeration system** to tilings

This mini-course:

Two techniques of *embedding a computation in a tiling*

- from **self-referential programs** to **self-similar** tilings
  [goes back to J. von Neumann]
- from arithmetic in **Sturmian numeration system** to tilings
  [J. Kari]

This mini-course:

Two techniques of *embedding a computation in a tiling*

- from **self-referential programs** to **self-similar** tilings [goes back to J. von Neumann]
- from arithmetic in **Sturmian numeration system** to tilings [J. Kari]

Very standard application:

- a construction of an aperiodic tile set

Less standard application:

- **aperiodicity** + **quasiperiodicity**

This mini-course:

Two techniques of *embedding a computation in a tiling*

- from **self-referential programs** to **self-similar** tilings
  [goes back to J. von Neumann]
- from arithmetic in **Sturmian numeration system** to tilings
  [J. Kari]

Very standard application:

- a construction of an aperiodic tile set

Less standard application:

- **aperiodicity + quasiperiodicity (and even minimality)**

Possible topics of this min-course

Possible topics of this min-course

Some applications of the self-simulating tilings:

Possible topics of this min-course

Some applications of the self-simulating tilings:

- the tiling problem is undecidable [Berger 1966]
- a tile set with only non computable tilings [Hanf & Myers 1974]
- a tile set with highly aperiodic tilings [?]
- robust (error-correcting) tilings [?]

Possible topics of this min-course

Some applications of the self-simulating tilings:

- the tiling problem is undecidable [Berger 1966]
- a tile set with only non computable tilings [Hanf & Myers 1974]
- a tile set with highly aperiodic tilings [?]
- robust (error-correcting) tilings [?]
- an effective shift is isomorphic to a subaction of a sofic shift [Hochman 2009, Aubrun & Sablik 2013]
- a *minimal* effective shift can be simulated by a *minimal* SFT [?]

Possible topics of this min-course

Some applications of the self-simulating tilings:

- the tiling problem is undecidable [Berger 1966]
- a tile set with only non computable tilings [Hanf & Myers 1974]
- a tile set with highly aperiodic tilings [?]
- robust (error-correcting) tilings [?]
- an effective shift is isomorphic to a subaction of a sofic shift [Hochman 2009, Aubrun & Sablik 2013]
- a *minimal* effective shift can be simulated by a *minimal* SFT [?]

Another remarkable result:

- Kari's technique gives non self-similar tilings [T. Monteil]

Possible topics of this min-course

Some applications of the self-simulating tilings:

- the tiling problem is undecidable [Berger 1966]
- a tile set with only non computable tilings [Hanf & Myers 1974]
- a tile set with highly aperiodic tilings [?]
- robust (error-correcting) tilings [?]
- an effective shift is isomorphic to a subaction of a sofic shift [Hochman 2009, Aubrun & Sablik 2013]
- a *minimal* effective shift can be simulated by a *minimal* SFT [?]

Another remarkable result:

- Kari's technique gives non self-similar tilings [T. Monteil]

Come to the lectures!