Undecidability of the Domino Problem

E. Jeandel and P. Vanier

Loria (Nancy), LACL (Créteil), France

November 20-24

Main problem

Given a surface S, a set of tiles τ and a tiling rule, is there a tiling of the surface using the tiles of τ ?

A lot of different variations depending on the surface (the plane, the discrete plane, the hyperbolic plane, a torus) and the tiling rules (which isometry can we apply to the tiles)

Most of these problems are undecidable: No algorithm can solve them

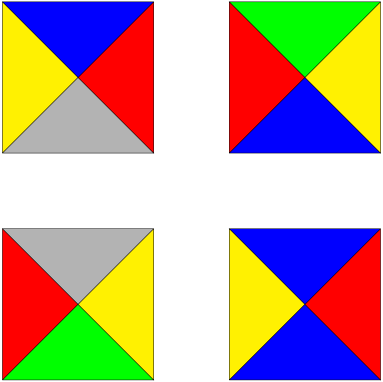
As a consequence of Berger's result.

In this (series of) talk

- What is the theorem of Berger ?
- What does it imply ?
- How to prove it ?

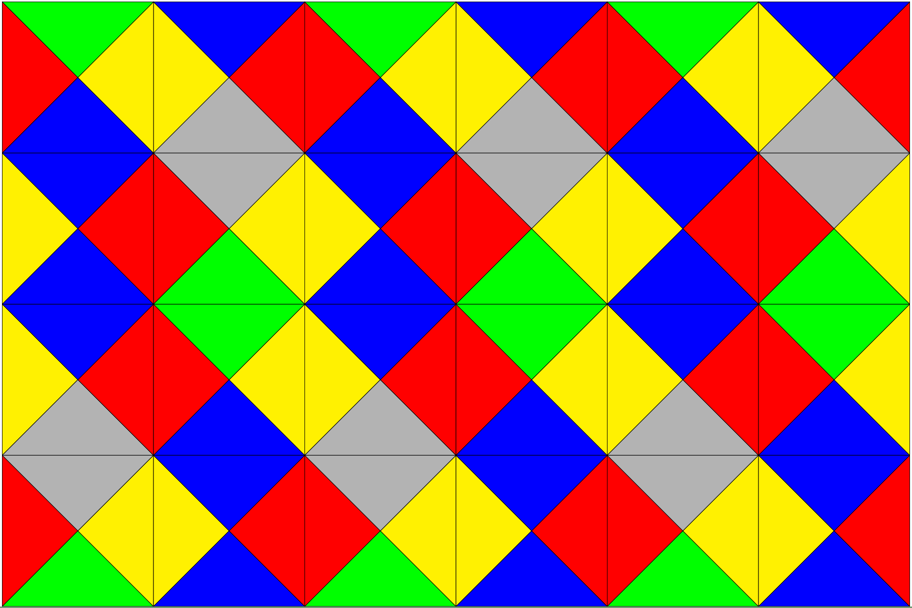
Four vastly different proofs, we will present at least 1 and at most 3 of them.

Wang tiles



Each tile can be used as much as you want.
The goal is to tile the entire plane, s.t. two adjacent tiles match on their common edge.

Example



Tilings with Wang tiles

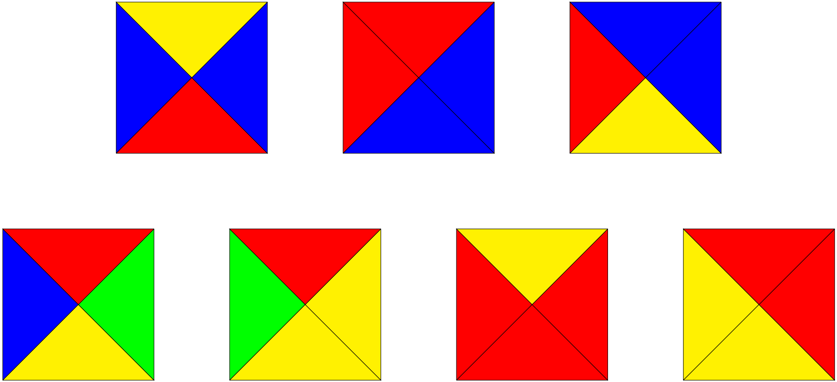
- A set of tiles τ will be called a *tileset*.
- We will be interested in tilings of the plane, but also tilings of finite regions, that respect the local constraints.
- Constraints: two adjacent tiles should match on their common edge, tiles cannot be rotated or reflected.

Domino Problem

How to know if there is a tiling of the plane by the tiles of τ ?

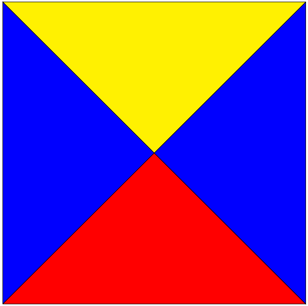
What could happen ?

Example 1



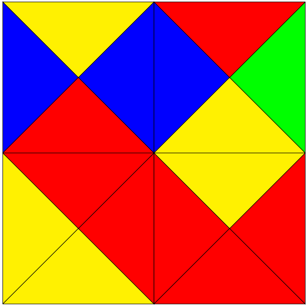
Example 1

A 1×1 square:



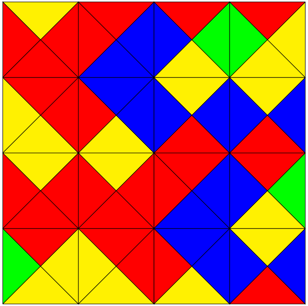
Example 1

A 2×2 square:



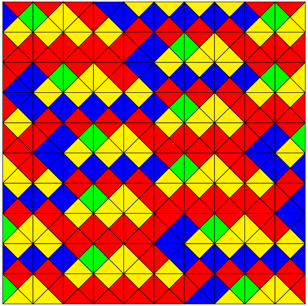
Example 1

A 4×4 square:



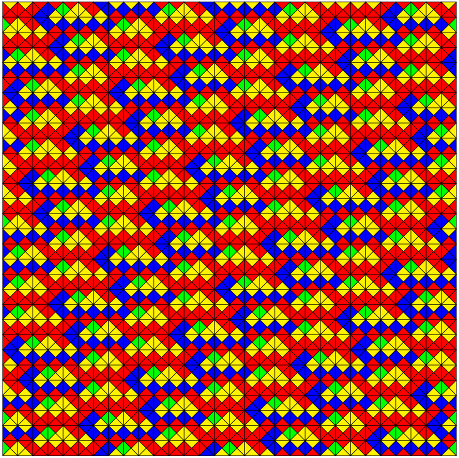
Example 1

A 10×10 square

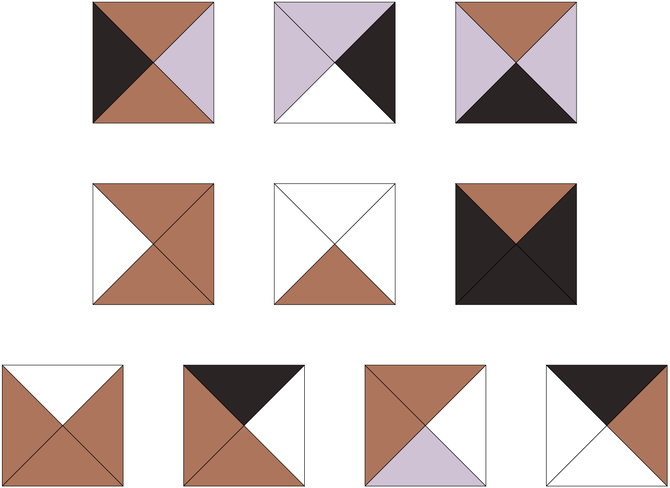


Example 1

A 30×30 periodic square

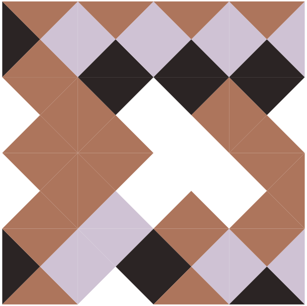


Example 2



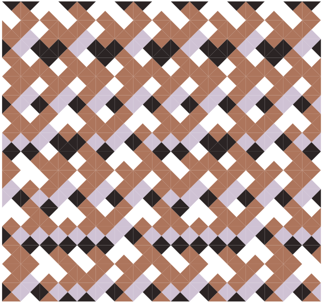
Example 2

A 4×4 square



Example 2

A 16×16 square:



Example 2

A 256×256 square:

(Actual result may differ from picture shown)

Example 2

No 512×512 square

Discussion

As the examples show

There is no easy bound N s.t. looking at tilings of a $N \times N$ square will enlighten us

Theorem (Berger 1964 (PhD), 1966 (Memoirs of the AMS))

There is no algorithm that decides, given a tileset τ, if τ tiles the plane.

Plan

1. What does it all mean ?
2. How bad is it ?
3. Algorithms
   - First (Naive) Algorithm
   - A better algorithm
   - Third Algorithm
4. Conclusion

The theorem (again)

Theorem (Berger 1964 (PhD), 1966 (Memoirs of the AMS))

There is no algorithm that decides, given a tileset τ, if τ tiles the plane.

The theorem somehow hints at the fact that there should be “hard” tilesets.
What are their properties ?

It's hard not to tile

Proposition

If τ does not tile the plane, there exists a size N s.t. τ does not tile a $N \times N$ square.

Contrast this with:

Proposition

Let f be any function that can be programmed on a computer. Then there exists a tileset τ of n tiles, with n arbitrary large, s.t. τ tiles a square of size $f(n) \times f(n)$ but does not tile the plane.

Note: up to recoloring, there are only finitely many tilesets with n tiles (at most $(4n)^{4n}$).

Compactness

Proposition

If τ does not tile the plane, there exists a size N s.t. τ does not tile a $N \times N$ square.

For all N, let X_N be a tiling by τ of the square $[-N, N] \times [-N, N]$ centered at the origin.

- As τ is finite, infinitely many of the X_N agree on the tile at the origin. We may extract from them a subsequence X_N^1 with the same tile at the origin.
- Infinitely many of the X_N^1 agree on the square $[-1, 1] \times [-1, 1]$. We may extract from them a subsequence X_N^2 with the same 3×3 square at the origin.
- ...
- The limit $\lim_N X_N^N$ is well defined and is a tiling of the plane.

It's hard not to tile

Proposition

Let f be any function that can be programmed on a computer.
Then there exists a tileset τ of n tiles, with n arbitrary large, s.t. τ tiles a square of size $f(n) \times f(n)$ but does not tile the plane.

Consider the following program on input τ:

- Let n be the number of tiles of τ
- Try all possible ways to tile a square of $f(n) \times f(n)$. If it succeeds, say "τ tiles the plane"
- Otherwise, say "τ does not tile the plane"

This program is incorrect. Otherwise, it would solve the Domino Problem, which isn't possible by Berger's theorem. Therefore there exists τ on which the program fails.

Another program that fails

Another program:

- For every N in $1, +\infty$:
  - Try to tile a square of size N
  - If it is not possible, say "τ does not tile the plane"
  - If there is a way to tile this square in a *periodic* manner, i.e. with the same colors on the borders, say "τ tiles the plane"
  - Otherwise, go to the next value of N.

Again, this program cannot succeed.

Trichotomy theorem

You see, in this world, there's three kinds of tilesets, my friend:

- Those who cannot tile a square of size *n* for some *n*
  - They do not tile the plane
- Those who can tile a square of size *n* for some *n* with the same colors on the borders
  - This gives a periodic tiling of the plane
- Those who tile the plane, but cannot do it periodically.
  - These are *aperiodic* tilesets.

Aperiodic tilesets

By Berger's theorem:

Aperiodic tilesets exist

In fact all known proofs of Berger theorem first build an aperiodic tileset.

Periodic vs aperiodic

Proposition

Let f be any function that can be programmed on a computer. Then there exists a tileset τ of n tiles, with n arbitrary large, s.t. τ tiles the plane periodically, but does not tile periodically a square of size less than $f(n) \times f(n)$.

Technically NOT a consequence of Berger's result, but of Gurevich-Koryakov (1972).

Plan

Good and bad news

Suppose I know τ tiles the plane. Does it mean, tilings by τ are hard to draw ?

Good news

Berger's theorem doesn't say anything about that. In fact, tilesets in Berger's proof are easy to draw (if they exist).

Bad news

Further theorems show bad things can happen

Worse things

Robinson 1971

There exists a tileset τ s.t. there is no algorithm that can decide, given a finite pattern P, if P can be extended into a tiling of the plane

The tileset τ is fixed.
Still doesn't rule out that there are easy tilings by τ. In fact in this example τ admits periodic tilings.
(This result will be proven next time).

The worst situation

Hanf-Myers 1974

There exists a tileset τ s.t. τ tiles the plane, but no tiling of τ can be obtained algorithmically.

Levin 2013

There exists a tileset τ s.t.

- τ tiles the plane
- No tiling of τ can be obtained algorithmically.
- There exists a randomized program that succeeds with probability $> 1 - \epsilon$.

Tilings by τ are “random”.

Plan

Algorithms

No algorithm can decide in all cases if a tileset τ tiles the plane.

But we can still try and answer in some particular cases!
If τ does not tile the plane, this can be shown easily: just find *N* s.t. no square of size *N* can be tiled by τ. Are there better algorithms ?

First Algorithm

For each N, try to find a way to tile a square of size $N \times N$.

How do to it correctly ?

First Algorithm

For each N, try to find a way to tile a square of size $N \times N$. Stop if it is not possible.

The algorithm will halt iff τ does not tile the plane.
How do write this algorithm efficiently ?

First remark

- If we found one way to tile a square of size $N \times N$, this particular tiling might not be extendable to a square of size $N+1 \times N+1$
- Instead of searching one way to tile a square of size $N \times N$, find *all ways* to tile a square of size $N \times N$.

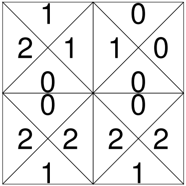
Let L_N be all the possible ways to tile a square of size $N \times N$. For each N, compute L_N starting from L_{N-1}.

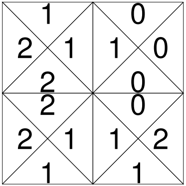
Complexity of the N-th step is $|L_N|2^{O(N)}$

Second remark

- Actually, we don't need to know all ways to tile a $N \times N$ square.
- We only need to know what are the possible *borders* of a $N \times N$ square.

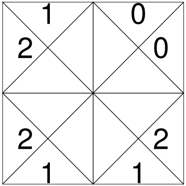
Example

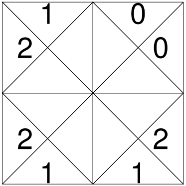




The first pattern is extendable to a 3×3 square iff the second pattern is extendable to a 3×3 square as they have the same border.

Example





The first pattern is extendable to a 3×3 square iff the second pattern is extendable to a 3×3 square as they have the same border.

New Algorithm

Let B_N be all the possible borders of squares of size $N \times N$.
For each N, compute B_N starting from B_{N-1}.

Complexity of the N-th step is $2^{O(N)}$.

One cannot do better unless $\mathbf{P} = \mathbf{NP}$.

(not an exact statement)

Strips

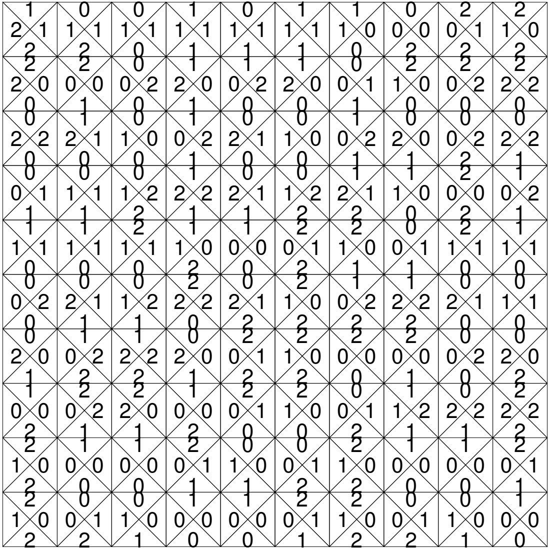
Instead of testing if τ tiles a square of size $N \times N$, test if τ tiles a horizontal (biinfinite) strip of height N.

Tradeoff

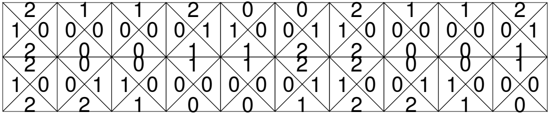
If τ tiles an horizontal strip of height N, it tiles a square of size $N \times N$.

Obvious.

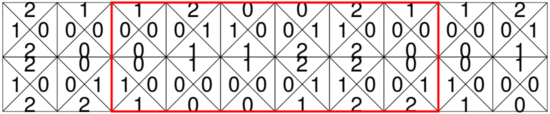
If τ tiles a square of size $N \times N$, it tiles an horizontal strip of height $\log_c N$ where c is the number of different colors in τ.



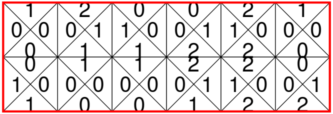
Here is a 10×10 square



We restrict our attention to a 10×2 rectangle



By pigeonhole ($10 > 3^2$), some border appear twice

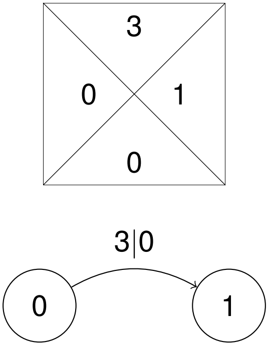


We can use this smaller rectangle to tile an horizontal strip periodically

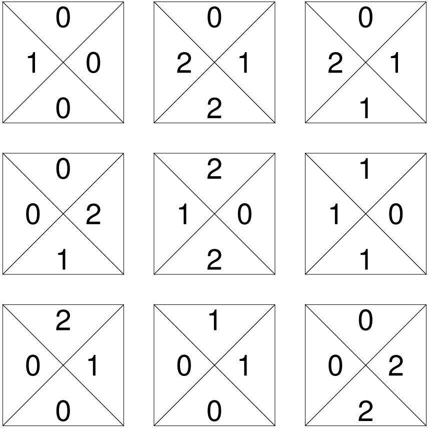
Question

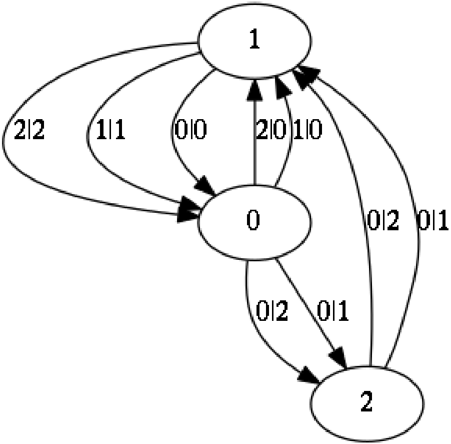
How to test efficiently how a tileset tiles a strip of height n ?

Main idea



Main idea


0 1 0 0
0 2 1 2
0 2 1 1
0 0 2 1
2 1 0 2
1 1 0 1
2 0 1 0
1 0 1 0
0 0 2 2


1
0
2
2|2
1|1
0|0
2|0
1|0
0|2
0|1
0|2
0|1

Main idea

A tileset is the same as a automaton (transducer.)

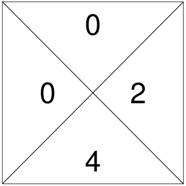
A tiling of an entire row is a biinfinite path on the automaton.

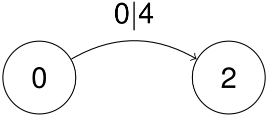
It exists iff the underlying graph contains a cycle

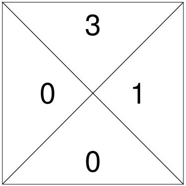
Strips

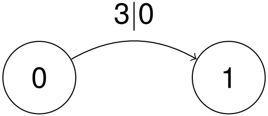
How do we interpret strips of height 2 ?

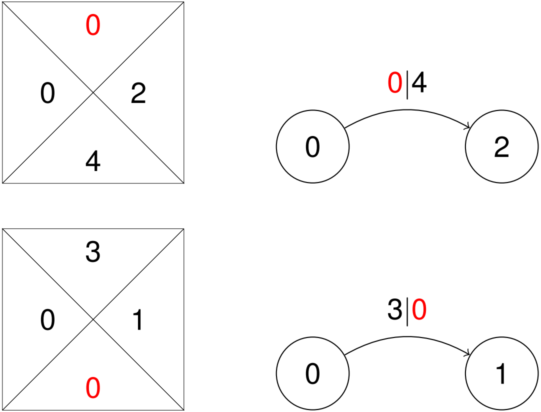
Strips



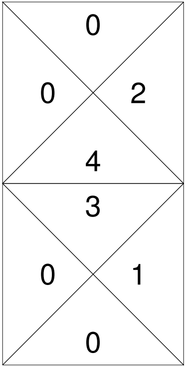


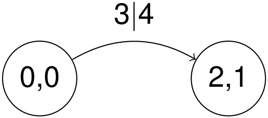





0
0
2
4
0|4
0
2
3
0
1
0
3|0
0
1

Strips





Strips

Strips of height 2 are obtained by *composing* the automaton with itself

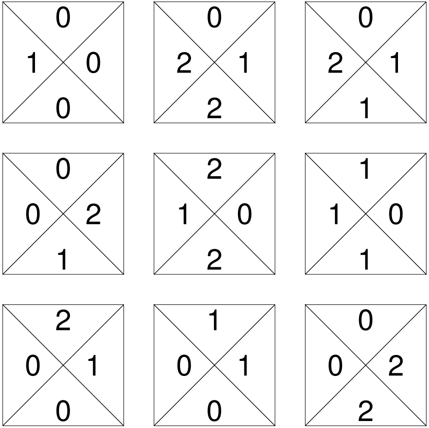
output of the first automaton must match input of the second one.

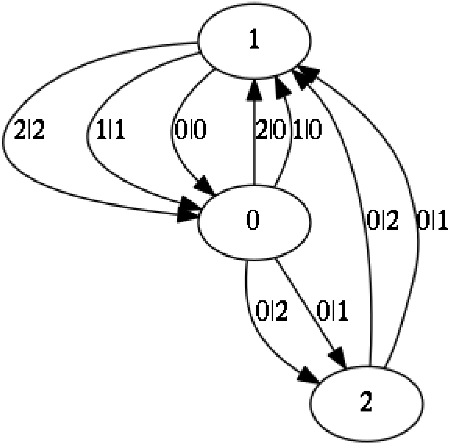
Algorithm

We see a tileset as a transducer T. For each n:

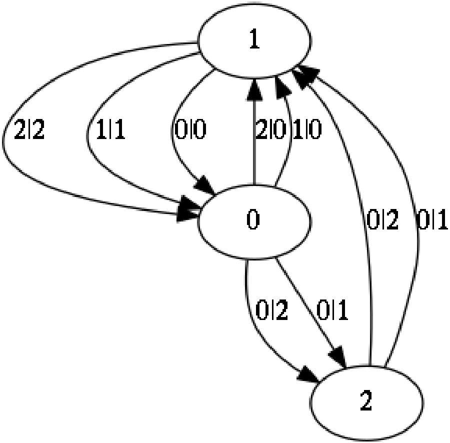
- We compute $T^n = T^{n-1} \circ T$
- If T^n does not contain any cycle, the tileset does not tile a strip of height n
- Otherwise, we test the next value of n

Main idea


0
1 0
0
0
2 1
2
0
2 1
1
0
0 2
1
2
1 0
2
1
1 0
1
2
0 1
0
1
0 1
0
0
0 2
2

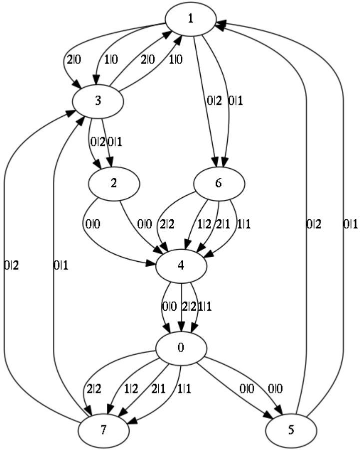

1
0
2
2|2
1|1
0|0
2|0
1|0
0|2
0|1
0|2
0|1

$n = 1$



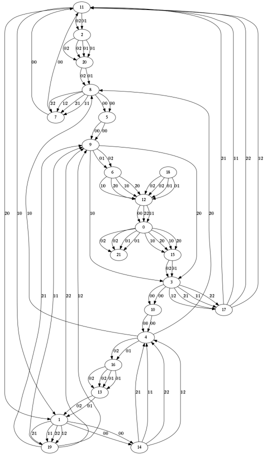
There is a cycle, a strip of height 1 can be obtained

$n = 2$



There is a cycle, a strip of height 2 can be obtained

$n = 3$



There is a cycle, a strip of height 3 can be obtained

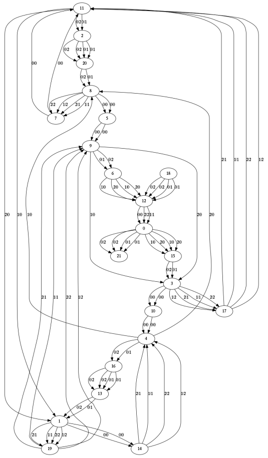
$n = 4$

etc, etc.

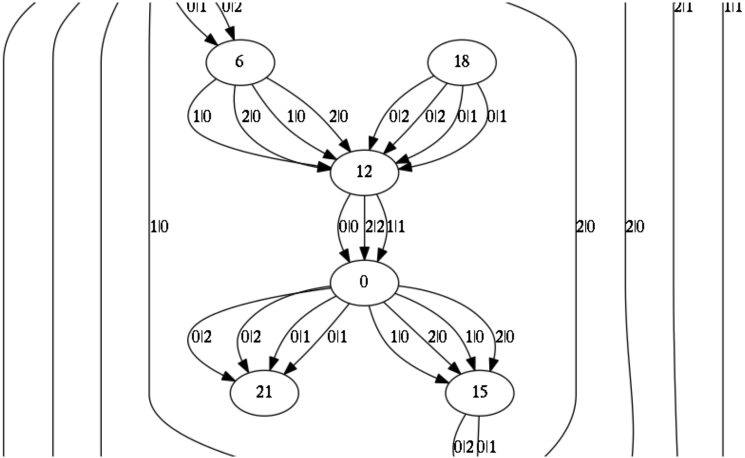
Optimizations

We can be smarter

$n = 3$



$n = 3$


0|1
0|2
2|1
1|1
6
18
1|0
2|0
1|0
2|0
0|2
0|2
0|1
0|1
12
1|0
0|0
2|2
1|1
2|0
2|0
0
0|2
0|2
0|1
0|1
1|0
2|0
1|0
2|0
21
15
0|2
0|1

How to be smarter

At each step, we can delete states that have no incoming (outgoing) edges.

At each step, we can delete edges that cannot be part of a cycle.

i.e. whose ends belong to different strongly connected components.

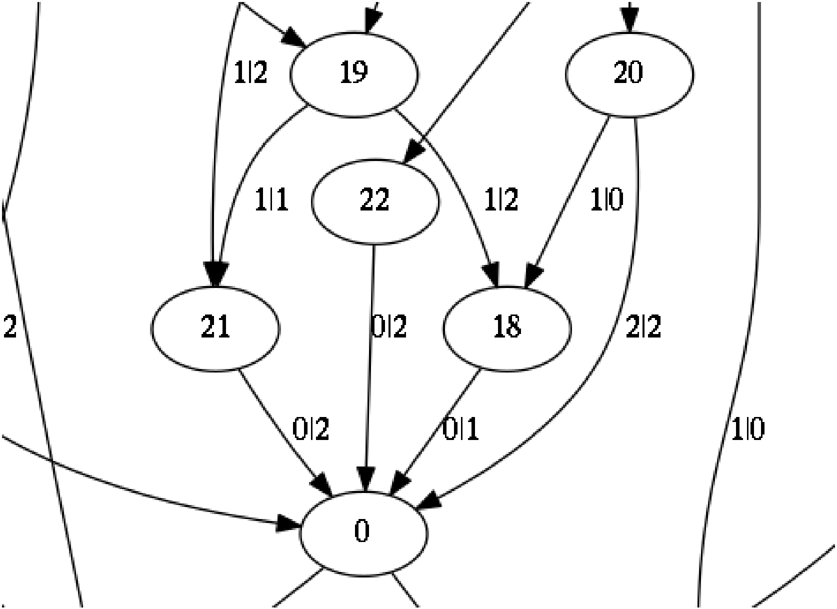
We can do better

The exact transducer does not matter, only the mapping (from input to output) it encodes

We can minimize the transducer at each step

Warning: minimizing a nondeterministic transducer is PSPACE-hard, we merely reduce it using bisimilarity.

Example



Second Algorithm, revisited

We see a tileset as a transducer T. For each n:

- We compute $T^n = T^{n-1} \circ T$
- We simplify T^n as much as possible
- If T^n does not contain any cycle, the tileset does not tile a strip of height n
- Otherwise, we test the next value of n

Same complexity as the previous algorithm, faster in practice.

Notes

- We can look at the vertical strips rather than the horizontal strips
- We are essentially looking at the tileset as a function(relation) that maps biinfinite sequences to biinfinite sequences
  - What functions can be obtained this way ?
  - More on this later

Third algorithm

- The third algorithm is only a basic test that can discard some tilesets without any computation.
- Obtained by Chazottes-Gambaudo-Gautero 2014, based on homology of branched surfaces

Idea

- Suppose we have a *periodic* tiling by τ, say a $p \times p$ square X that repeats periodically.
- Let c be a color and e_c (resp. w_c) the number of times that c appears in the east (resp. west) side of a tile in X.
- As X is periodic

$$e_c = w_c$$

How to compute e_c ?

Equation

Easy answer:

$$e_c = \sum_{t \in \tau \text{ with } c \text{ at the east side}} \# \text{ of times } t \text{ appears in} X$$

$$w_c = \sum_{t \in \tau \text{ with } c \text{ at the west side}} \# \text{ of times } t \text{ appears in} X$$

This gives us an equation.

System of equations

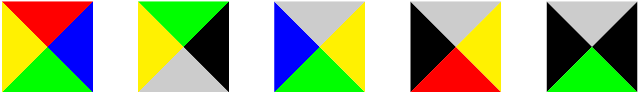
- To each tile $t_i \in \tau$ we associate a variable x_i which represents the number of times t_i appear in X
- For each color c we have an equation

$$\sum_{t_i \in E_\tau(c)} x_i = \sum_{t_i \in W_\tau(c)} x_i$$

  where $E_\tau(c)$ is the set of tiles of τ where c appear on the east side, similarly with $W_\tau(c)$
- We also have north/south equations.

Example

1 2 3 4 5



Horizontal equations

$$\begin{aligned} x_1 + x_2 &= x_3 + x_4 \\ x_4 + x_5 &= x_2 + x_5 \\ x_3 &= x_1 \end{aligned}$$

Vertical equations

$$\begin{aligned} x_1 &= x_4 \\ x_2 &= x_1 + x_3 + x_5 \\ x_3 + x_4 + x_5 &= x_2 \end{aligned}$$

Easy remark

Proposition

If τ tiles the plane periodically, then the system of equations has a nontrivial nonnegative solution.

(Note: $x_i = 0$ is always a trivial solution)

Corollary

The previous example does not tile the plane periodically

New equations

Instead of looking at the number of times each time appear, we will look at their density.
We now have a new equation:

$$\sum_i x_i = 1$$

Proposition

If τ tiles the plane periodically, then the system of equations has a nonnegative solution.

Theorem

If τ tiles the plane, then the system of equations has a nonnegative solution.

Proof

Suppose that τ tiles the plane. We now look at a tiling X of a $n \times n$ square.

- Let y_i be the number of times the tile t_i appear in X
- Let $x_i = y_i/n^2$ be the density of tile t_i in X

For each color c:

$$\sum_{t_i \in E_\tau(c)} y_i - \sum_{t_i \in W_\tau(c)} y_i = O(n)$$

Indeed, the difference is due to the border.

Proof

Suppose that τ tiles the plane. We now look at a tiling X of a $n \times n$ square.

- Let y_i be the number of times the tile t_i appear in X
- Let $x_i = y_i/n^2$ be the density of tile t_i in X

For each color c:

$$\sum_{t_i \in E_\tau(c)} x_i - \sum_{t_i \in W_\tau(c)} x_i = O(\frac{1}{n})$$

And of course

$$\sum x_i = 1$$

Proof

Suppose that τ tiles the plane. We now look at a tiling X of a $n \times n$ square.

- Let y_i be the number of times the tile t_i appear in X
- Let $x_i = y_i/n^2$ be the density of tile t_i in X

For each color c:

$$\sum_{t_i \in E_\tau(c)} x_i - \sum_{t_i \in W_\tau(c)} x_i = O(\frac{1}{n})$$

And of course

$$\sum x_i = 1$$

Now we let $n \to +\infty$ and extract a subsequence

The theorem

Theorem

If τ tiles the plane, then the system of equations has a nonnegative solution.

Corollary

The previous example does not tile the plane periodically.

Note: it tiles a 3×3 square.
Note: we can test whether a system of equations has a nonnegative solution using linear programming.

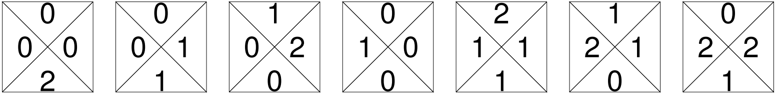
The theorem

Theorem

If τ tiles the plane, then the system of equations has a nonnegative solution.

What happens if τ does not tile the plane, but the system has solutions?

Example



$$x_1 + x_2 + x_3 = x_1 + x_4 \tag{1}$$

$$x_4 + x_5 = x_2 + x_5 + x_6 \tag{2}$$

$$x_6 + x_7 = x_3 + x_7 \tag{3}$$

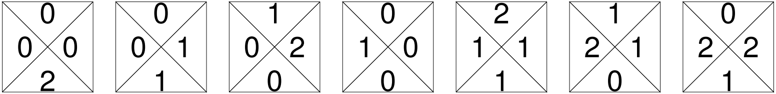
$$x_1 + x_2 + x_4 + x_7 = x_3 + x_4 + x_6 \tag{4}$$

$$x_3 + x_6 = x_2 + x_5 + x_7 \tag{5}$$

$$x_5 = x_1 \tag{6}$$

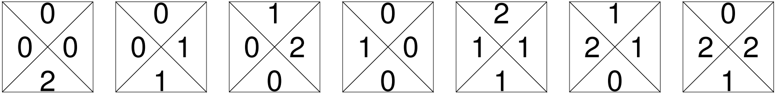
$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 1 \tag{7}$$

Example

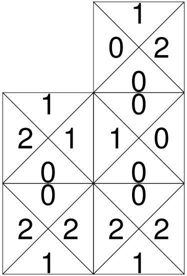


Possible solution: $x_1 = x_2 = x_5 = 0$, $x_3 = x_4 = x_6 = 1/5$ and $x_7 = 2/5$
How to interpret it ?

Example



Possible solution: $x_1 = x_2 = x_5 = 0$, $x_3 = x_4 = x_6 = 1/5$ and $x_7 = 2/5$
How to interpret it ?



More on the theorem

Theorem (Chazottes-Gambaudo-Gautero 2014)

If the system of equation has a solution, the tileset τ might be used to tile some surface.
However, if there is no tiling by τ, the genre of the surface has to grow linearly in the number of tiles.

(not an exact statement)

Plan

1. What does it all mean ?
2. How bad is it ?
3. Algorithms
   - First (Naive) Algorithm
   - A better algorithm
   - Third Algorithm
4. Conclusion

Conclusion

What we have seen today

- What are Wang tiles and tilesets τ.
- What is the Domino Problem (is there a tiling by τ ?)
- Consequences of the undecidability of the Domino Problem
- How to semi-test if there is a tiling of the plane by τ.

Next time:

- What does “undecidable” mean ?