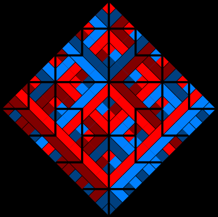
# Undecidability, Tilings and Polyominoes

Nicolas Ollinger (LIF, Aix-Marseille Université, CNRS)

GdT ARITH — **LIRMM**, 16 avril 2009



# Entscheidungsproblem: the $\forall\exists\forall$ case

**Hilbert's Entscheidungsproblem (semantic version)**
To find a method which for every sentence of elementary quantification theory yields a decision as to whether or not the sentence is satisfiable.

In the 60s, the **classical decision problem** is studied with respect to classes of quantification types.

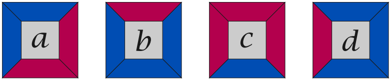
One big open class: the $\forall\exists\forall$ class. Wang and Büchi introduce in 1961 two decision problems in order to solve it.

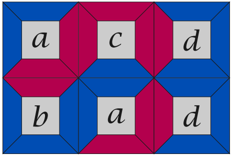
The problem is proved undecidable in 1962 by Kahr, Moore and Wang using a simpler reduction.

# The Domino Problem (DP)

*"Assume we are **given a finite set of square plates** of the same size with **edges colored**, each in a different manner. Suppose further there are infinitely many copies of each plate (plate type). We are **not permitted to rotate or reflect a plate**. The question is to find an effective procedure by which we can **decide**, for each given finite set of plates, **whether we can cover up the whole plane** (or, equivalently, an infinite quadrant thereof) **with copies of the plates subject to the restriction that adjoining edges must have the same color**."*

*(Wang, 1961)*



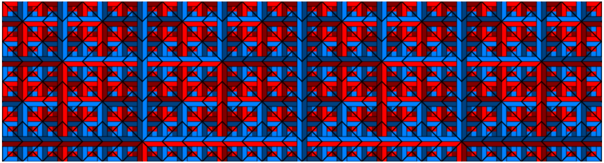
# The Domino Problem is undecidable

Tile sets without tilings are recursively enumerable.

A tile set with a periodic tiling admits a biperiodic tiling.

Tile sets with a biperiodic tiling are recursively enumerable.

Undecidability is to be found in **aperiodic tile sets**, tile sets that only admit aperiodic tilings.

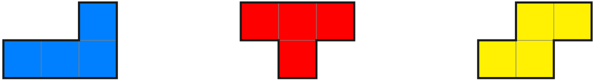


**Theorem [Berger 1964]** **DP** is undecidable.

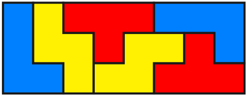
# 1. Tiling the Plane with a Fixed Number of Polyominoes

# Polyominoes

A **polyomino** is a simply connected tile obtained by gluing together rookwise connected unit squares.



A **tiling** of a region by a set of polyominoes is a partition of the region into images of the tiles by isometries.



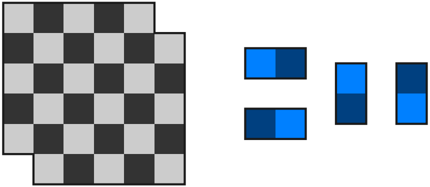
A **tiling by translation** is a tiling where isometries are restricted to translations.

# Tiling finite regions

The combinatorics of tilings of finite regions is challenging, polyominoes make great puzzles.

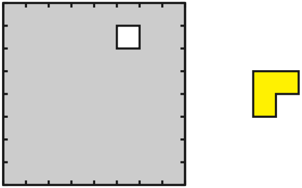
Can you tile with dominoes a $2m \times 2n$ rectangle with two opposite corners cut?

**[Golomb 1965]**



Can you tile with L-tiles a $2^n \times 2^n$ square with one cut unit square?

**[Golomb 1965]**



# Tiling the plane

In this talk, we consider tilings of **the whole Euclidian plane** by finite sets of polyominoes.

A tiling is **discrete** if all the unit squares composing images of the polyominoes are aligned on the grid $\mathbb{Z}^2$.

**Lemma** A tile set admits a tiling iff it admits a discrete tiling.

**Sketch of the proof** Non-discrete tilings have countably many infinite parallel fracture lines. By shifting along fracture lines, one constructs a discrete tiling from any non-discrete tiling.

# The $k$-Polyomino Problem

**Polyomino Problem**
Given a finite set of polyominoes, decide if it can tile the plane.

**$k$-Polyomino Problem**
Given a set of $k$ polyominoes, decide if it can tile the plane.

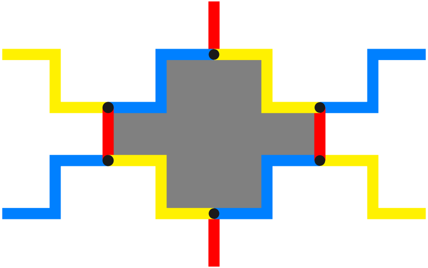
**Lemma** Finite sets of polyominoes tiling the plane are **co-re**.

**Sketch of the proof** Consider tilings of finite regions covering larger and larger squares. If the set does not tile the plane, by compacity, there exists a size of square it cannot cover with tiles.

# One polyomino by translation

**[Wijshoff and van Leeuwen 1984]** A **single polyomino** that tiles the plane **by translation** tiles it biperiodically. The problem is decidable.

**[Beauquier and Nivat 1991]** A single polyomino tiles the plane by translation iff it is a *pseudo-hexagon* (contour word $uvw\tilde{u}\tilde{v}\tilde{w}$).



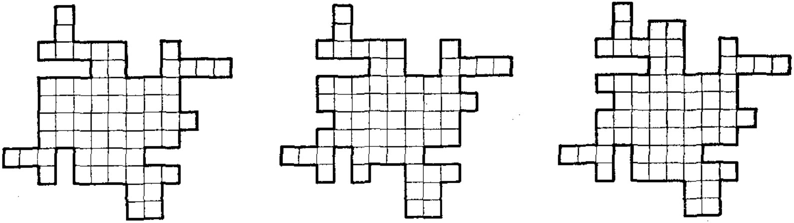
**[Gambini and Vuillon 2007]** This can be tested in $O(n^2)$.

# The Polyomino Problem is undecidable

Wang tiles are oriented unit squares with colors.

Colors can be encoded by **bumps and dents**.

A Wang tile can be **encoded** as a big pseudo-square polyomino with bumps and dents in place of colors.



**[Golomb 1970]** The Polyomino Problem is undecidable.

# Fixed number of polyominoes

The reduction of Golomb encodes $N$ Wang tiles into $N$ polyominoes.

What about the $k$-Polyomino Problem?

(1) either it is decidable for all $k$ and the family of algorithms is not itself recursive *(eg. set of Wang tiles with $k$ colors)*;

(2) either there exists a frontier between decidable and undecidable cases *(eg. Post Correspondence Problem)*.

We will show that (2) holds.

# Dented polyominoes

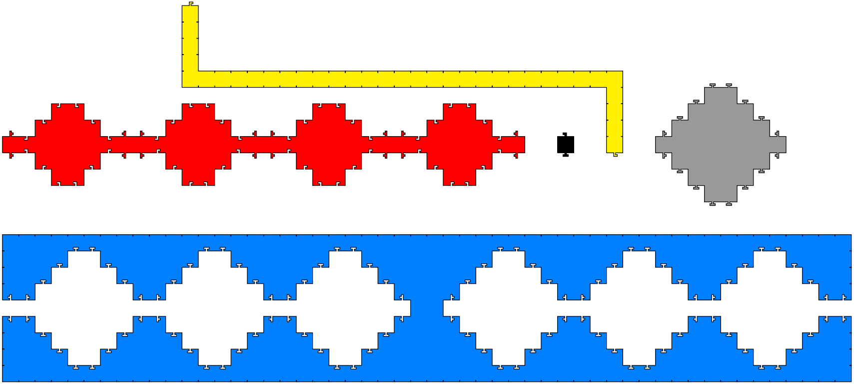
Computing with polyominoes relies on several levels of encoding. To lever the complexity of the tiles, we use dented polyominoes.

A **dented polyomino** is a polyomino with edges labeled by a **dent shape** and an **orientation**. When considering tilings, dents and bumps have to match.

**Lemma** Every set of $k$ dented polyominoes can be encoded as a set of $k$ polyominoes, preserving the set of tilings.

**Sketch of the proof** Scale each polyomino by a factor far larger than bumps, then add bumps and dents along edges.

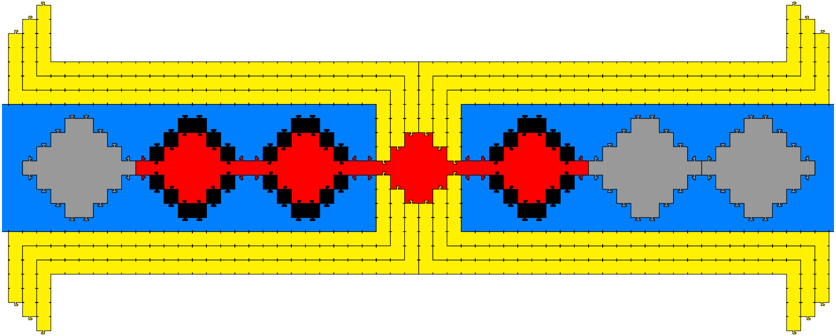
# 5 tiles



| | *blank* | *bit* | *marker* | *inside* |
|---|---|---|---|---|
| shape | | | | |
| bump | | **wire, tooth** | **meat, filler** | **tooth, filler** |
| dent | | **meat** | **jaw** | **jaw** |

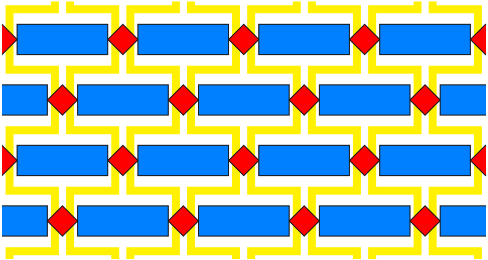
# Encoding Wang tiles

A **meat** is placed in between two **jaws** to select a tile. The gaps inside the **jaws** are filled by **fillers** and **teeth**.
**Wires** connect Wang tiles.



# Encoding a tiling by Wang tiles

Wang tiles are encoded and placed on a **regular grid**.
Tiles of a same diagonal are placed on a horizontal line sharing jaws.



# Every tiling is coding

It remains to show to **difficult part of the proof**.

Why does every tiling codes a tiling by Wang tiles?

(1) The polyominoes locally enforce Wang tiles coding;

(2) Details on the encoding of colors enforce a same orientation for all Wang tiles in the plane.

**Theorem** The 5-Polyomino Problem is undecidable.

# Tiling by translation

Previous encoding uses:

**1** meat, **1** jaw, **1** filler, **4** wires, **4** teeth.

**Theorem** The 11-Polyomino Translation Problem is undecidable.

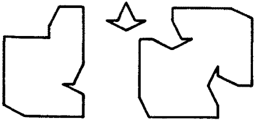
The problem is decidable for a single polyomino and undecidable for 11 polyominoes. What about $2 \leqslant k \leqslant 10$?

Even for $k = 2$, it seems that it is not trivial...

# Aperiodic set of polyominoes

A weaker property is the existence of **aperiodic** sets of polyominoes.

If all sets of polyominoes are biperiodic for a given $k$, the $k$-Polyomino Problem is decidable.



**[Ammann et al 1992]** There exists an aperiodic set of 3 polyominoes.

**[Ammann et al 1992]** There exists an aperiodic set of 8 polyominoes for tiling by translation.

# Open problem

**Tiling** Study $1 \leqslant k \leqslant 4$, aperiodicity for $1 \leqslant k \leqslant 2$.

**By translation** Study $2 \leqslant k \leqslant 10$, aperiodicity for $2 \leqslant k \leqslant 7$.

The following (old) problem is still open...

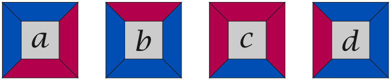
**Open Problem** Does there exist an aperiodic polyomino?

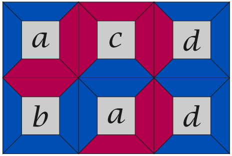
# 2. Undecidability of the Domino Problem

# The Domino Problem (DP)

*"Assume we are **given a finite set of square plates** of the same size with **edges colored**, each in a different manner. Suppose further there are infinitely many copies of each plate (plate type). We are **not permitted to rotate or reflect a plate**. The question is to find an effective procedure by which we can **decide**, for each given finite set of plates, **whether we can cover up the whole plane** (or, equivalently, an infinite quadrant thereof) **with copies of the plates subject to the restriction that adjoining edges must have the same color**."*

*(Wang, 1961)*



# Aperiodicity in DP

The set of tilings of a tile set $T$ is a compact subset of $T^{\mathbb{Z}^2}$.

By compacity, if a tile set does not tile the plane, there exists a square of size $n \times n$ that cannot be tiled.

Tile sets without tilings are recursively enumerable.

A set of Wang tiles with a periodic tiling admits a biperiodic tiling.

Tile sets with a biperiodic tiling are recursively enumerable.

Undecidability is to be found in **aperiodic tile sets**, tile sets that only admit aperiodic tilings.

# Undecidability of DP: a short history

**1964** Berger proves the undecidability of **DP**.

Two main type of related activities in the literature:

(1) construct aperiodic tile sets (small ones);
(2) give a full proof of the undecidability of **DP** (implies (1)).

From 104 tiles (Berger, 1964) to 13 tiles (Čulik, 1996) aperiodic sets.

Seminal self-similarity based proofs *(reduction from* ***HP****)*:

- Berger, 1964 *(20426 tiles, a full PhD thesis)*
- Robinson, 1971 *(56 tiles, 17 pages, long case analysis)*
- Durand et al, 2007 *(Kleene's fixpoint existence argument)*

Tiling rows seen as transducer trace based proof:
Kari, 2007 *(affine maps, short concise proof, reduces* ***IP****)*

# In this talk

A new self-similarity based construction building on classical proof schemes with concise arguments and few tiles:

1. two-by-two substitution systems and aperiodicity
2. an aperiodic tile set of 104 tiles
3. enforcing any substitution and reduction from **HP** *(sketch)*

This work combines tools and ideas from:

**[Berger 64]** *The Undecidability of the Domino Problem*

**[Robinson 71]** *Undecidability and nonperiodicity for tilings of the plane*

**[Grünbaum Shephard 89]** *Tilings and Patterns, an introduction*

**[Durand Levin Shen 05]** *Local rules and global order, or aperiodic tilings*

# Two-by-two substitution systems

A 2×2 substitution system maps a finite alphabet to 2×2 squares of letters on that alphabet.

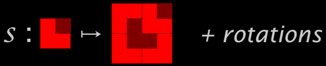
$$s : \Sigma \to \Sigma^{\boxplus}$$

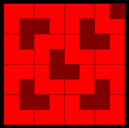
The substitution is iterated to generate bigger squares.

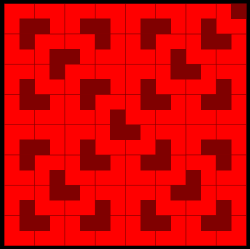
$$S : \Sigma^{\mathcal{P}} \to \Sigma^{\square(\mathcal{P})}$$

$$\forall z \in \mathcal{P}, \forall c \in \boxplus,$$
$$S(C)(2z + c) = s(C(z))(c)$$

$$S(u \cdot C) = 2u \cdot S(C)$$



# Coloring the whole plane *via* limit sets

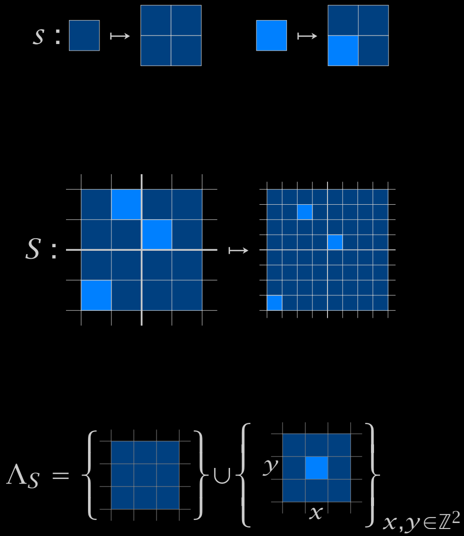
What is a coloring of the plane generated by a substitution?

With tilings in mind the set of colorings should be closed by translation and compact.

We take the limit set of iterations of the (continuous) global map closed up to translations.

$\Lambda_S = \bigcap_n \Lambda_S^n$ where $\Lambda_S^0 = \Sigma^{\mathbb{Z}^2}$

$\Lambda_S^{n+1} = \left\{ u \cdot S(C) \,\middle|\, C \in \Lambda_S^n, u \in \boxplus \right\}$



# Unambiguous substitutions are aperiodic

A substitution is **aperiodic** if its limit set $\Lambda_S$ is aperiodic.

A substitution is **unambiguous** if, for every coloring $C$ from its limit set $\Lambda_S$, there exists a **unique** coloring $C'$ and a **unique** translation $u \in \boxplus$ satisfying $C = u \cdot S(C')$.

**Proposition 3.** Unambiguity implies aperiodicity.

**Sketch of the proof.** Consider a periodic coloring with minimal period $p$, its preimage has period $p/2$. $\Diamond$

**Idea.** Construct a tile set whose tilings are in the limit set of an unambiguous substitution system.

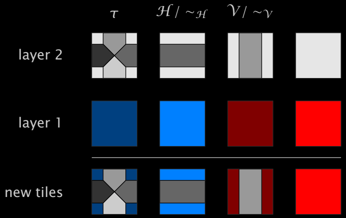
# Coding tile sets into tile sets

A tile set $\tau$ is a triple $(T, \mathcal{H}, \mathcal{V})$ where $\mathcal{H}$ and $\mathcal{V}$ define horizontal and vertical matching constraints.

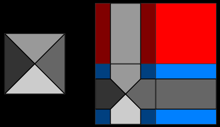
The set of tilings of $\tau$ is $X_\tau$.

A tile set $(T', \mathcal{H}', \mathcal{V}')$ **codes** a tile set $(T, \mathcal{H}, \mathcal{V})$, according to a **coding rule** $t : T \to T'^{\boxplus}$ if $t$ is injective and

$$X_{\tau'} = \{u \cdot t(C) | C \in X_\tau, u \in \boxplus\}.$$




*coding tile set*



*coding rule*

# Aperiodicity *via* unambiguous self-coding

A tile set $(T, \mathcal{H}, \mathcal{V})$ **codes** a substitution $s : T \to T^{\boxplus}$ if it codes **itself** according to the coding rule $s$.

**Proposition 4.** A tile set both admitting a tiling and coding an unambiguous substitution is aperiodic.

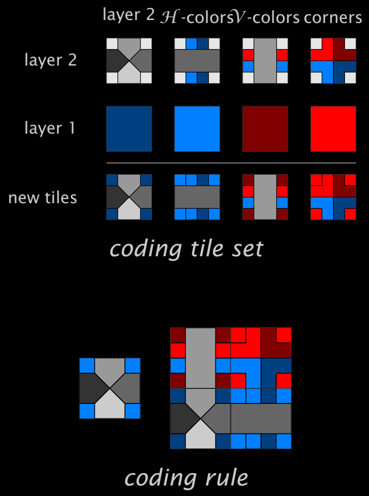
**Sketch of the proof.** $X_\tau \subseteq \Lambda_S$ and $X_\tau \neq \emptyset$. $\Diamond$

# A coding scheme with fixpoint?

Better scheme: not strictly increasing the number of tiles.

**Problem.** it cannot encode any layered tile set, constraints between layer 1 and layer 2 are checked edge by edge.

**Solution.** add a third layer with one bit of information per edge.



# Canonical substitution

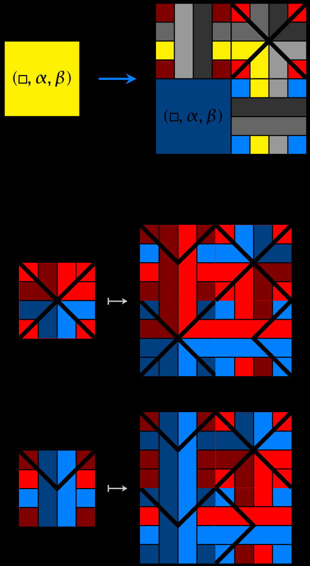
Copy the tile in the SW corner but for layer 1.

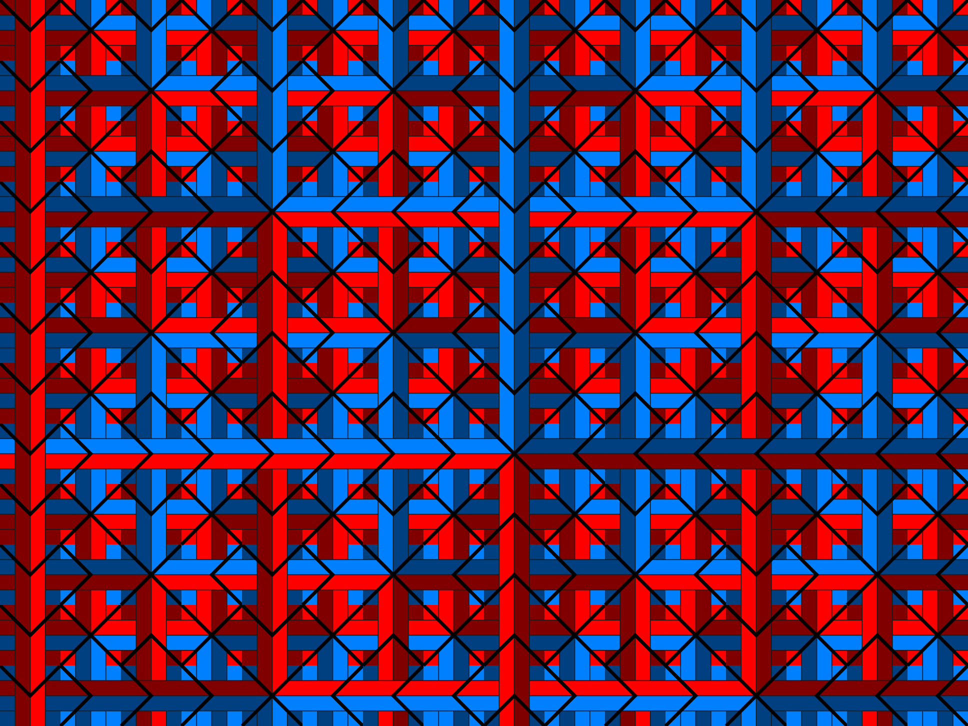
Put the only possible X in NE that carry layer 1 of the original tile on SW wire.

Propagate wires colors.

Let H et V tile propagate layer 3 arrows.

The substitution is injective.





# Aperiodicity: sketch of the proof

**1. The tile set admits a tiling:**
*Generate a valid tiling by iterating the substitution rule:* $X_\tau \cap \Lambda_S \neq \emptyset$.

**2. The substitution is unambiguous:**
*It is injective and the projectors have disjoined images.*

**3. The tile set codes the substitution:**

(a) each tiling is an image of the canonical substitution
*Consider any tiling, level by level, short case analysis.*

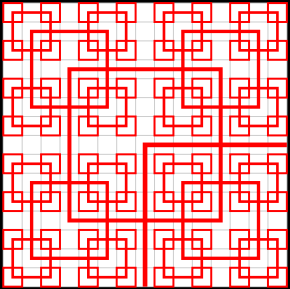
(b) the preimage of a tiling is a tiling
*Straightforward by construction (preimage remove constraints).*

# Enforcing substitutions *via* tilings

Let $\pi$ map every tile of $\tau(s')$ to $s'(a)(u)$ where $a$ and $u$ are the letter and the value of ⊞ on layer 1.

**Theorem 2.** Let $s'$ be any substitution system. The tile set $\tau(s')$ enforces $s'$: $\pi\left(X_{\tau(s')}\right) = \Lambda_{S'}$.

**Idea.** Every tiling of $\tau(s')$ codes an history of $S'$ and every history of $S'$ can be encoded into a tiling of $\tau(s')$.



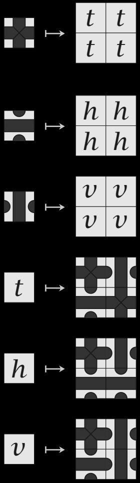
$b = s(a)\binom{1}{1}$

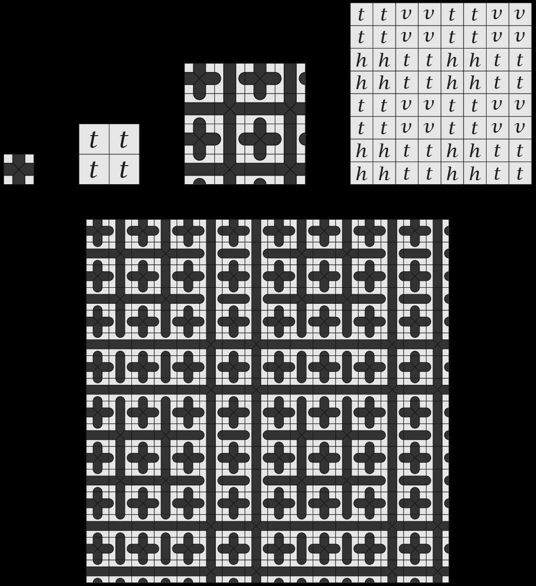
$a$

$b = s(a)\binom{0}{0}$

$a$

# Infinitely many squares of unbounded size



# Reducing HP to DP

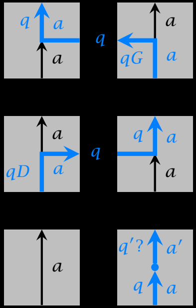
Any tiling by previous tile set contains infinitely many finite squares of unbounded size.

In each square, simulate the computation of the given Turing machine from an empty tape.

Initial computation is enforced in the SW corner.

Remove the halting state.

The tile set tiles the plan iff the Turing machine does not halt.



# Extensions and Open Problems

The construction extends to prove **soficity** of more **general substitutions**.

We need tools to prove that a set of colorings cannot be recognized by tilings (is **not sofic**).