104 tiles

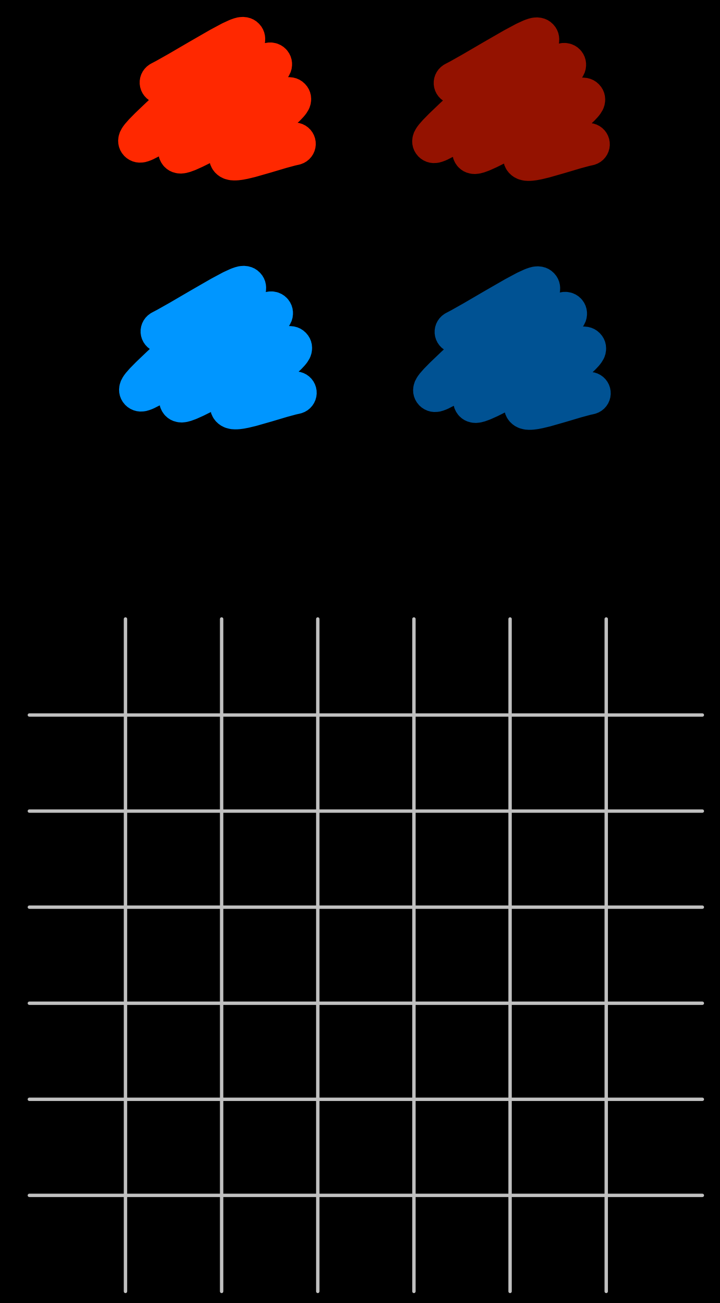
N. Ollinger (Escape, LIF)
in Frac d'automne 2007

104 tiles

N. Ollinger (Escape, LIF)
in Frac d'automne 2007

Foreword

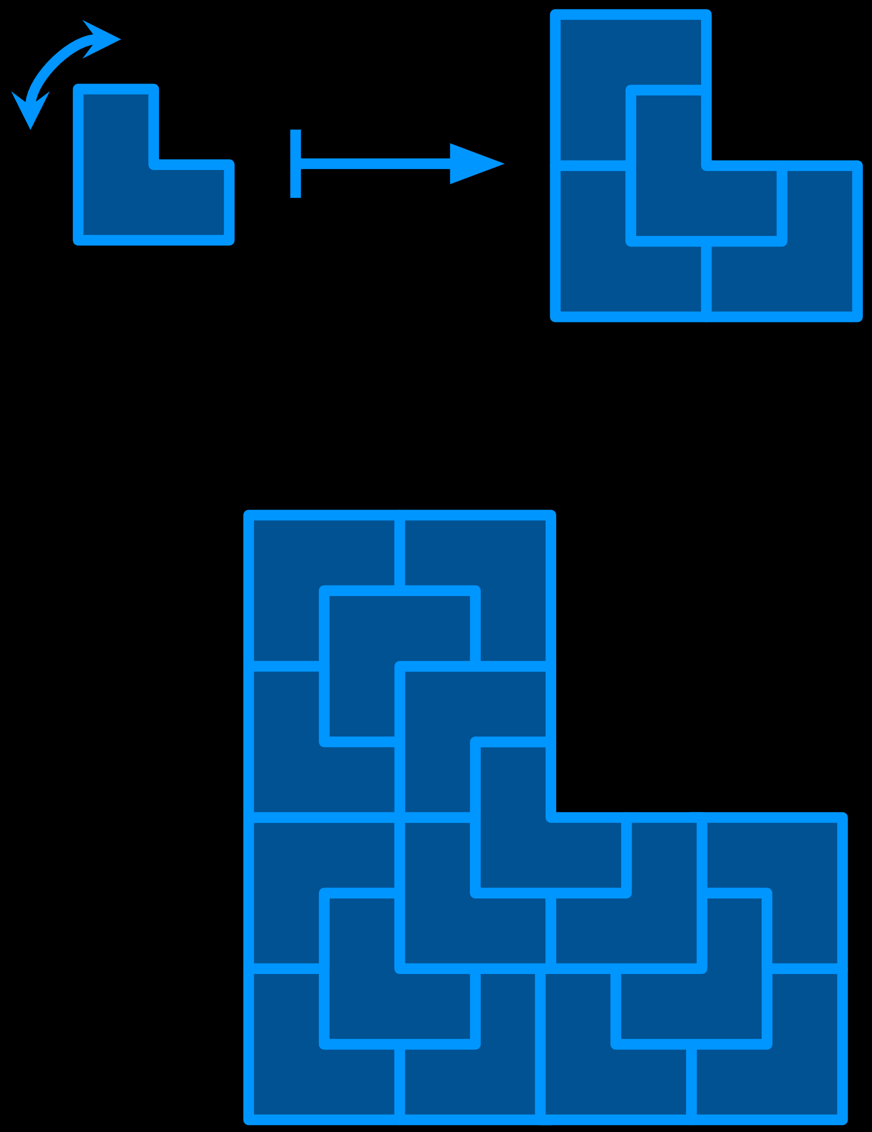
- Yet another presentation on the undecidability of the Domino Problem **[Berger 64]**
- Formal stuff (a few) stand on white background.
- Pictures (a lot of) stand on black background.
- This talk is very geometrical: we draw pictures on the Euclidian plane.



2×2 Substitutions

Substitutions

- Geometric substitutions provide a convenient recursive way to define aperiodic colorings of the plane.
- Subtle geometrical arguments are required: discuss dissection, inflation, scaling factor, etc.
- The classical L (or chair) substitution.

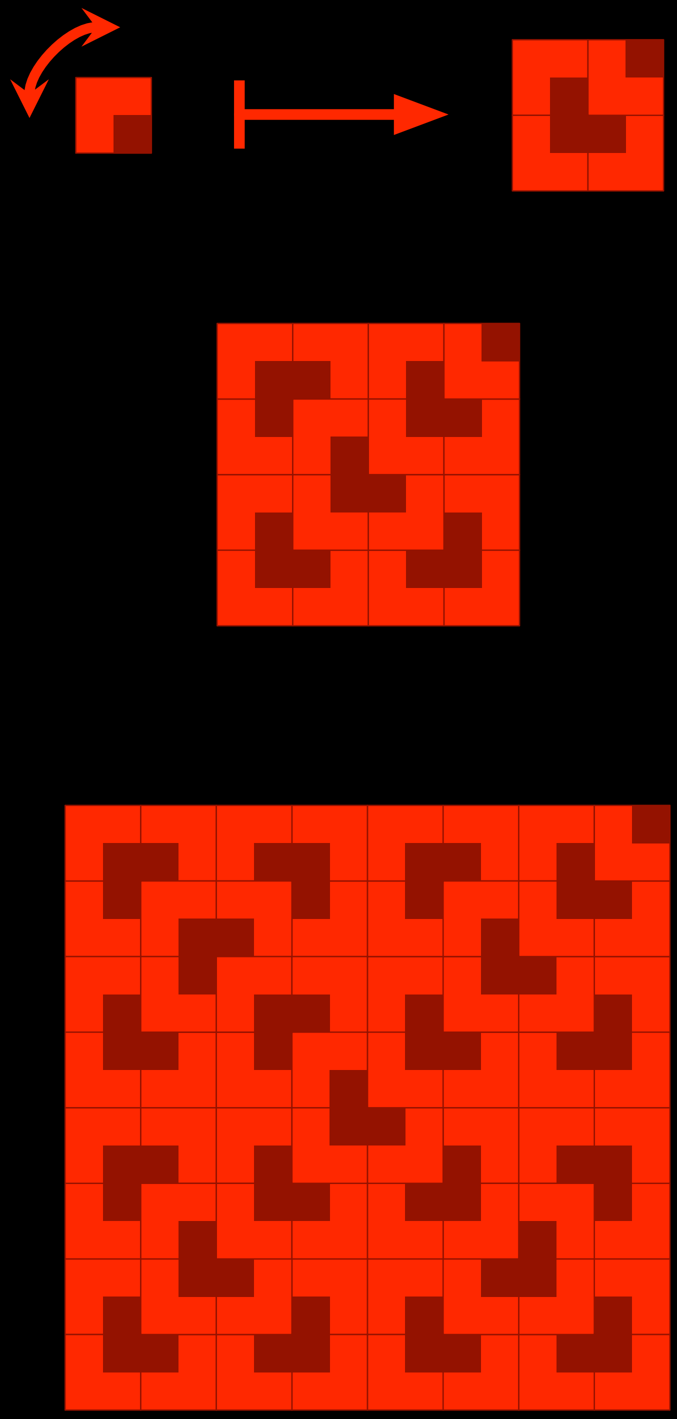


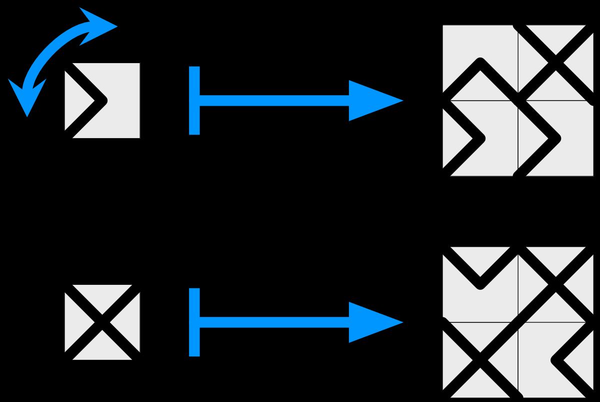
2×2 Substitutions

- A 2×2 substitution maps a finite alphabet to 2×2 squares of letters on that alphabet.
- The substitution is iterated to generate bigger squares.

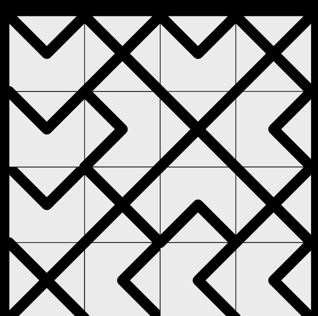
$$s : \Sigma \longrightarrow \Sigma^{\boxplus}$$

$$S \circ \sigma_u = \sigma_{2u} \circ S$$

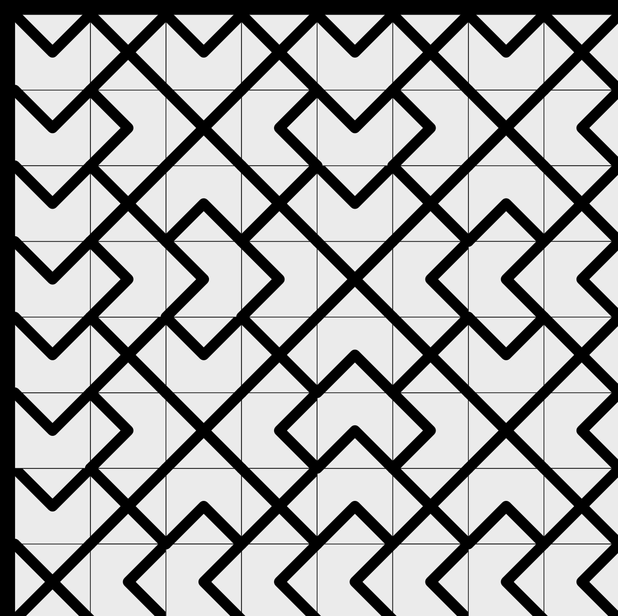




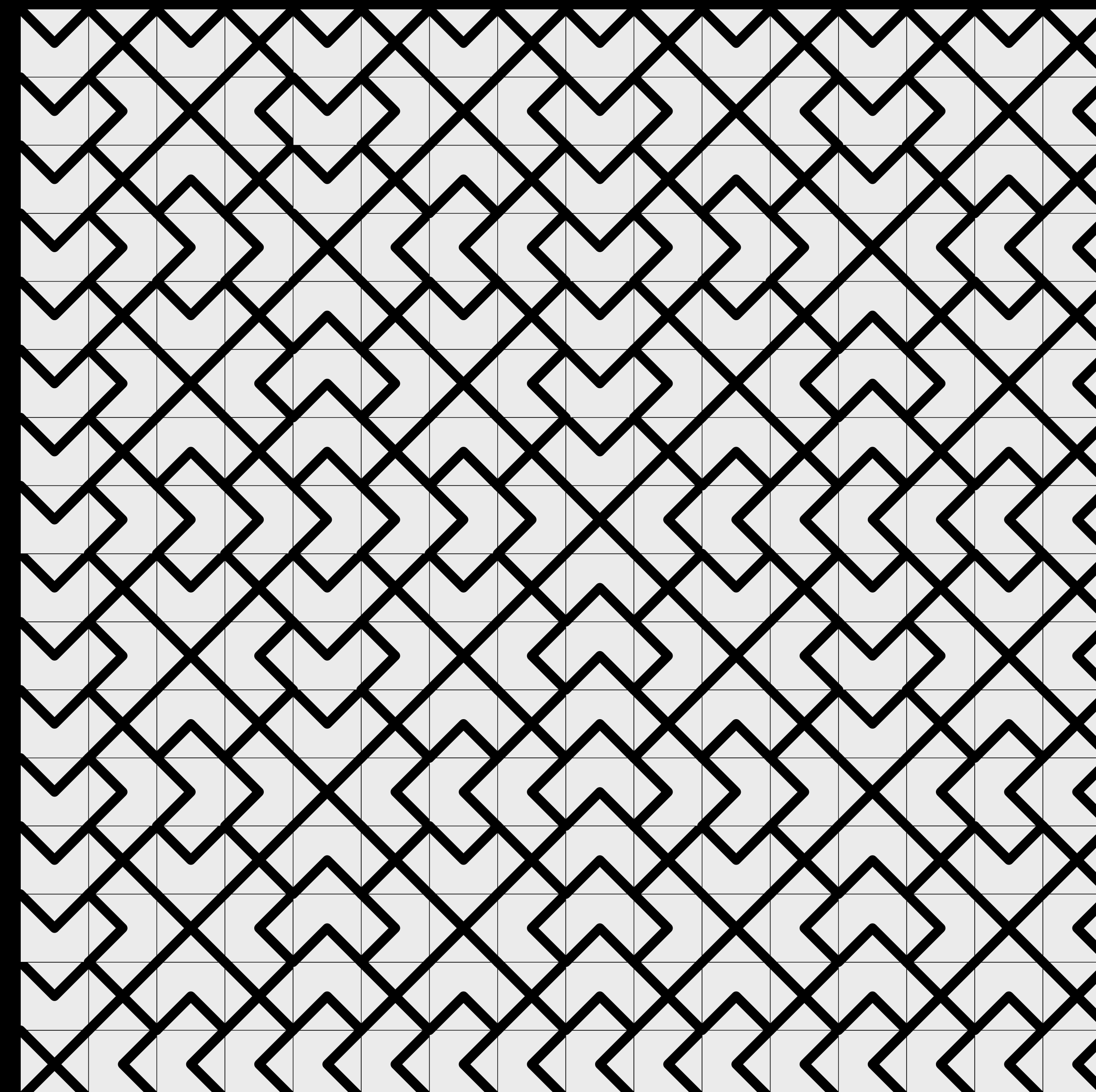
(5 letters)



$s^2(\boxtimes)$



$s^3(\boxtimes)$



$s^4(\boxtimes)$

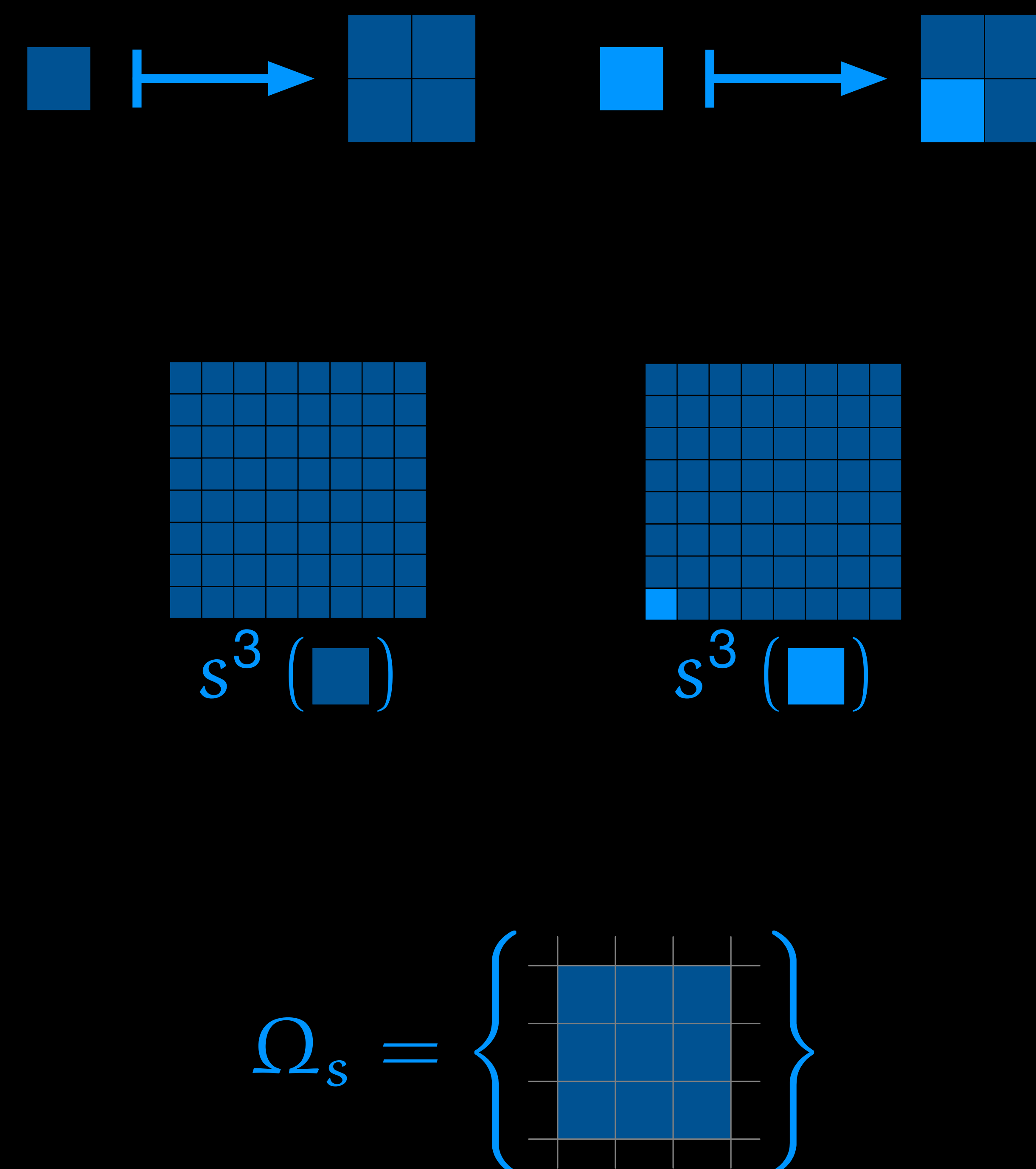
Another L-style substitution

Pattern closure?

- What is a coloring of the plane generated by a substitution?
- One possibility is to consider a *closure* of generated patterns up to translations.
- Difficult to check...

$C \in \Omega_s$ if

$$\forall P \prec_{\mathcal{F}} C, \quad \exists a, k, \quad P \prec s^k(a)$$

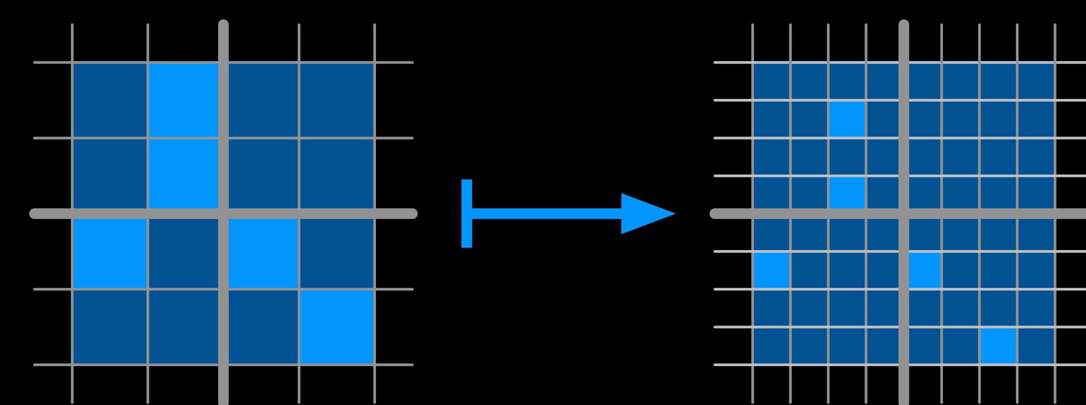


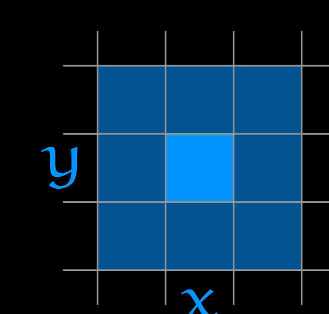
Limit set!

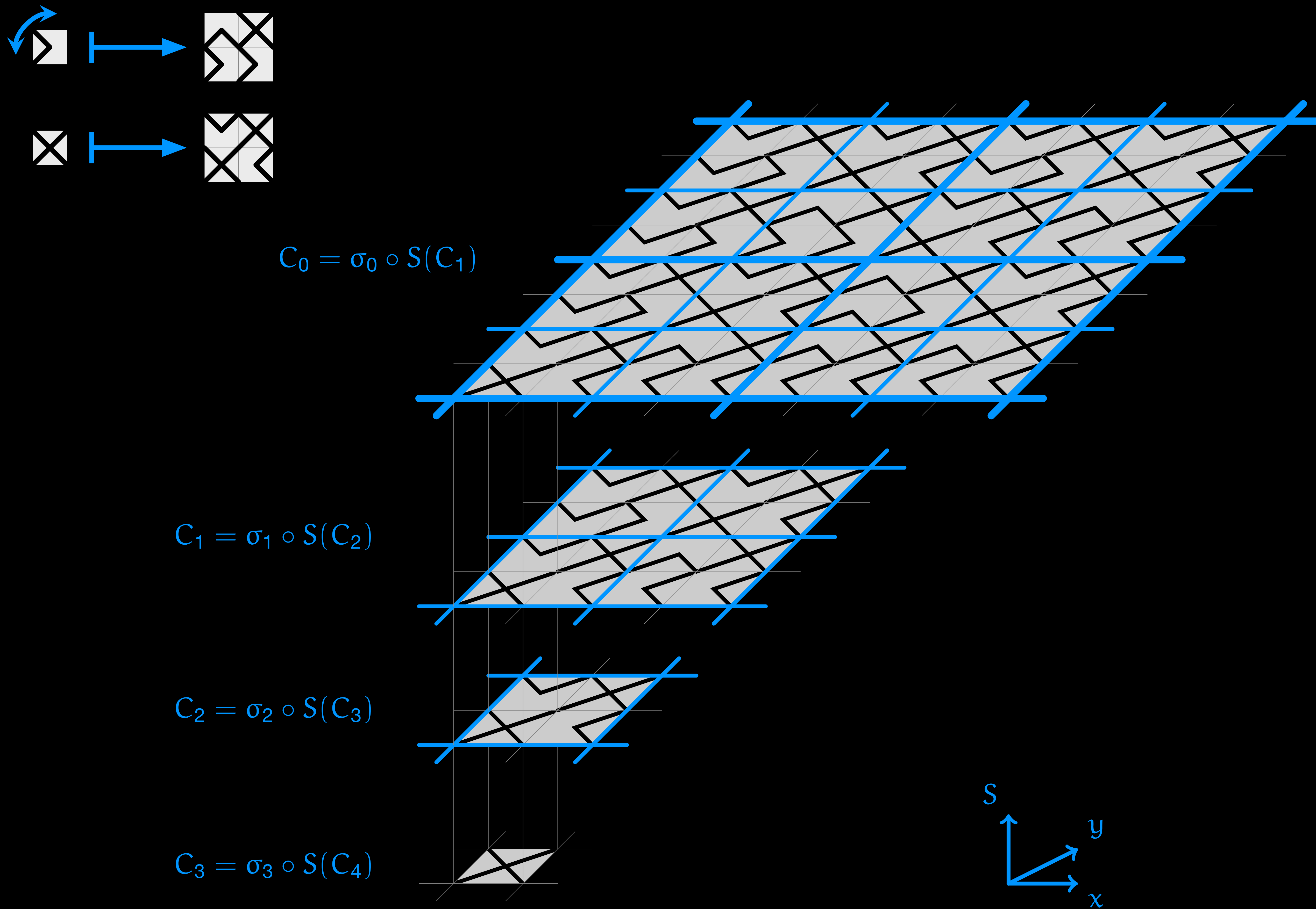
- Extend the substitution to a global map inflating around 0.
- Take the limit set of iterations of this map closed up to translations.
- More colorings!

$\Lambda_s = \bigcap_n \Lambda_s^{(n)}$ where

$$\Lambda_s^{(0)} = \Sigma^{\mathbb{Z}^2}$$
$$\Lambda_s^{(n+1)} = \left\langle S(\Lambda_S^{(n)}) \right\rangle_\sigma$$



$\Lambda_s = \Omega_s \cup \Bigg\{$  $\Bigg\}_{x,y \in \mathbb{Z}^2}$



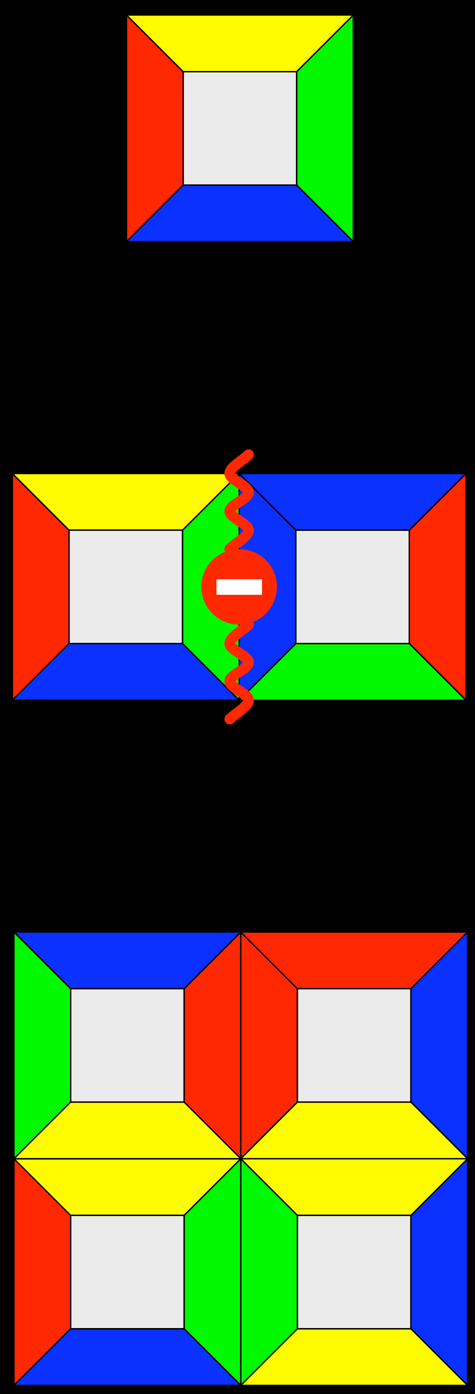
History of a coloring in the limit set

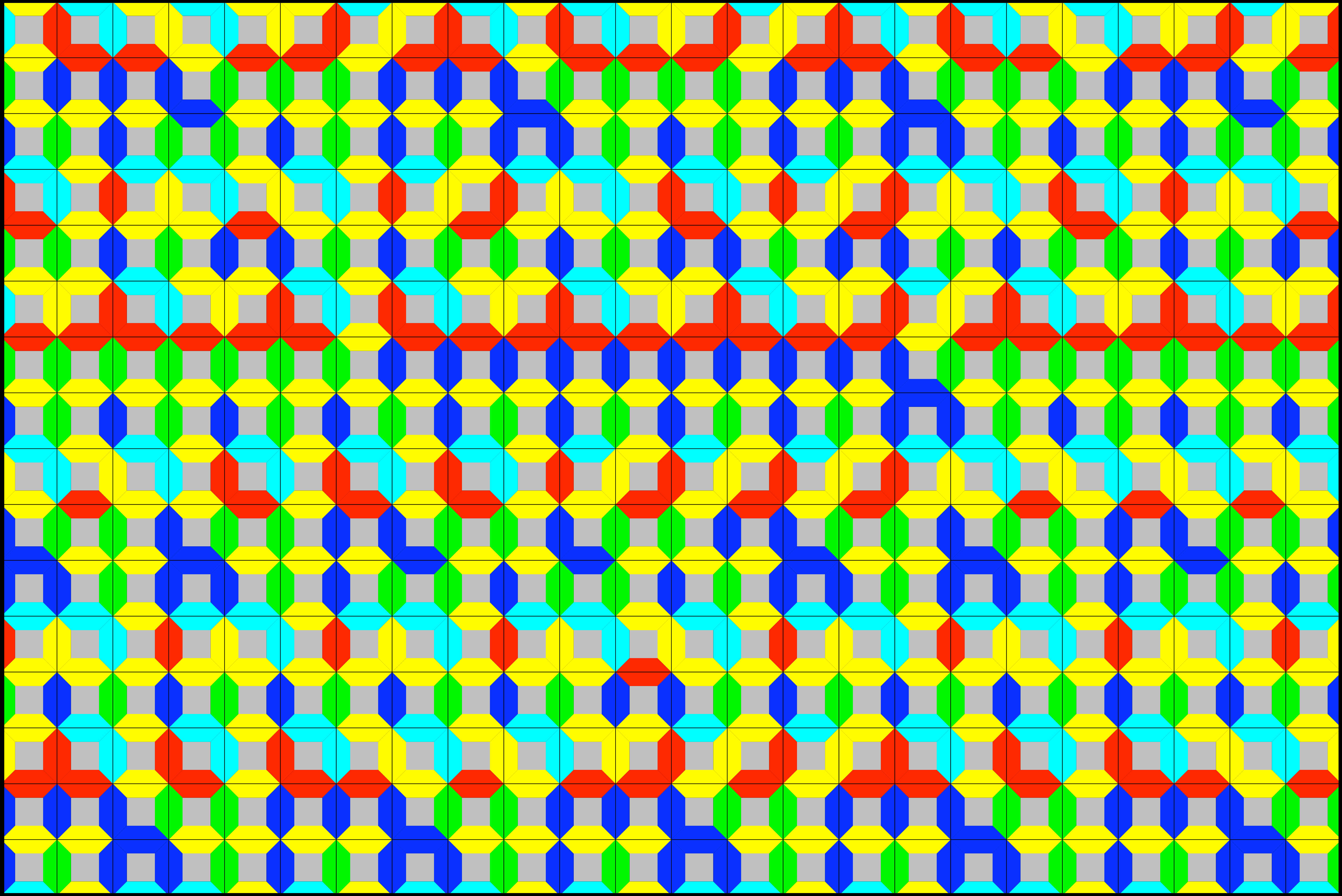
Tiling with Wang tiles

Wang tiles

- A Wang tile is a unit square with colored edges.
- A tileset is a finite set of Wang tiles.
- Tiling is done with translations only (no rotation) by matching colors along edges.

$$\tau \subseteq C^4$$

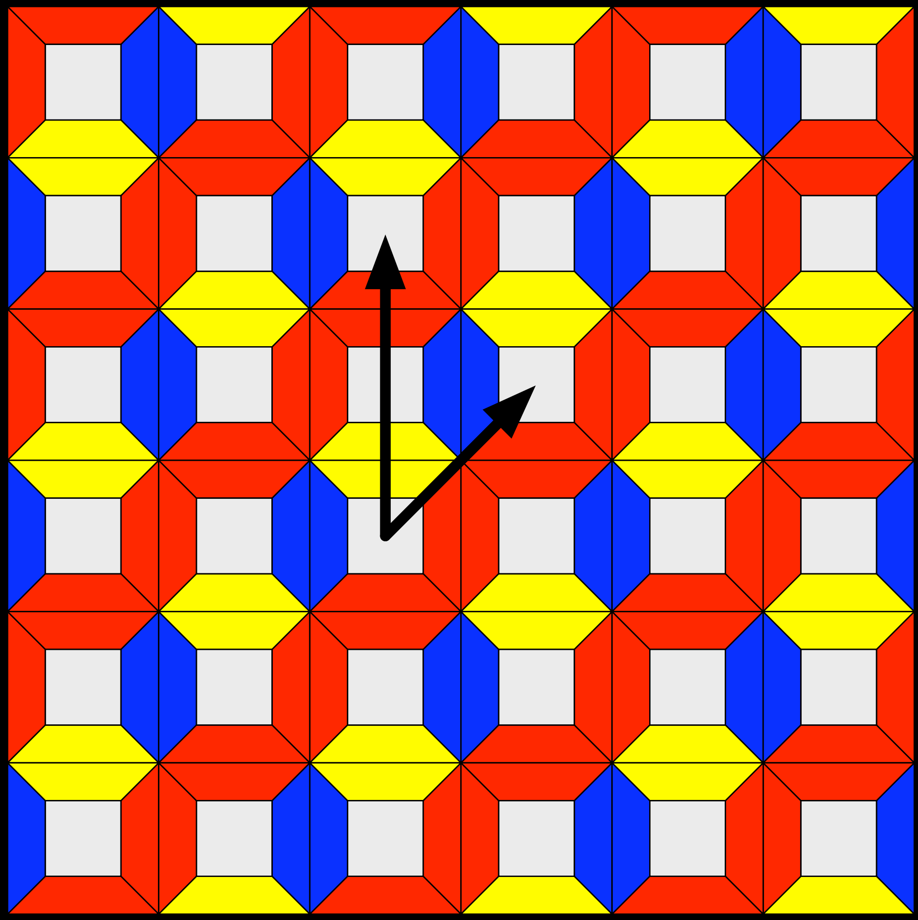




Tiling the whole plane with Wang tiles

Periodicity

- A tiling is periodic if it admits a periodicity vector.
- A tiling is biperiodic if it admits two non colinear periodicity vectors.
- If a tileset admits a periodic tiling then it admits a biperiodic tiling.
- A tileset is aperiodic if it admits no periodic tiling.



The Domino Problem

- **The Domino Problem:** given a tileset, decide if it admits a tiling.
- A long time ago, Wang conjectured there exists no aperiodic tileset and thus the Domino Problem is recursive.
- In 1964, his PhD student Berger proved that the problem is not recursive. To achieve this goal he encoded Turing machine computations inside aperiodic tilings.
- A whole talk could be devoted to the history of aperiodicity, the quest for small aperiodic tileset, the different proofs and technics to prove the non recursivity of the domino problem... **Not today!**

Au menu

- Define a particular set of 104 tiles;
- Prove in details its aperiodicity;
- Briefly explain how to complete the Domino Problem construction.
- This is a mix of different tools and ideas from:

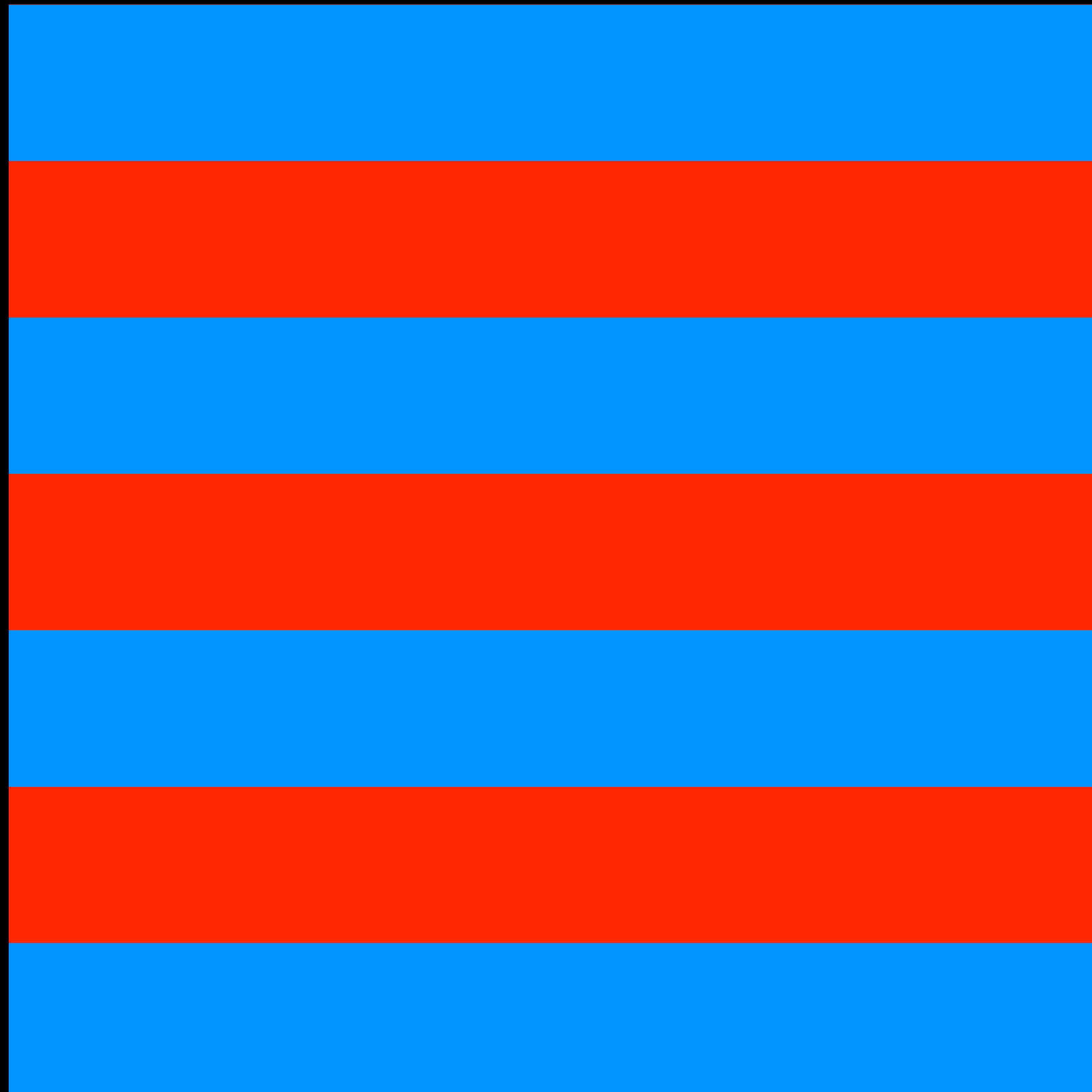
[Berger 64] *The Undecidability of the Domino Problem*

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

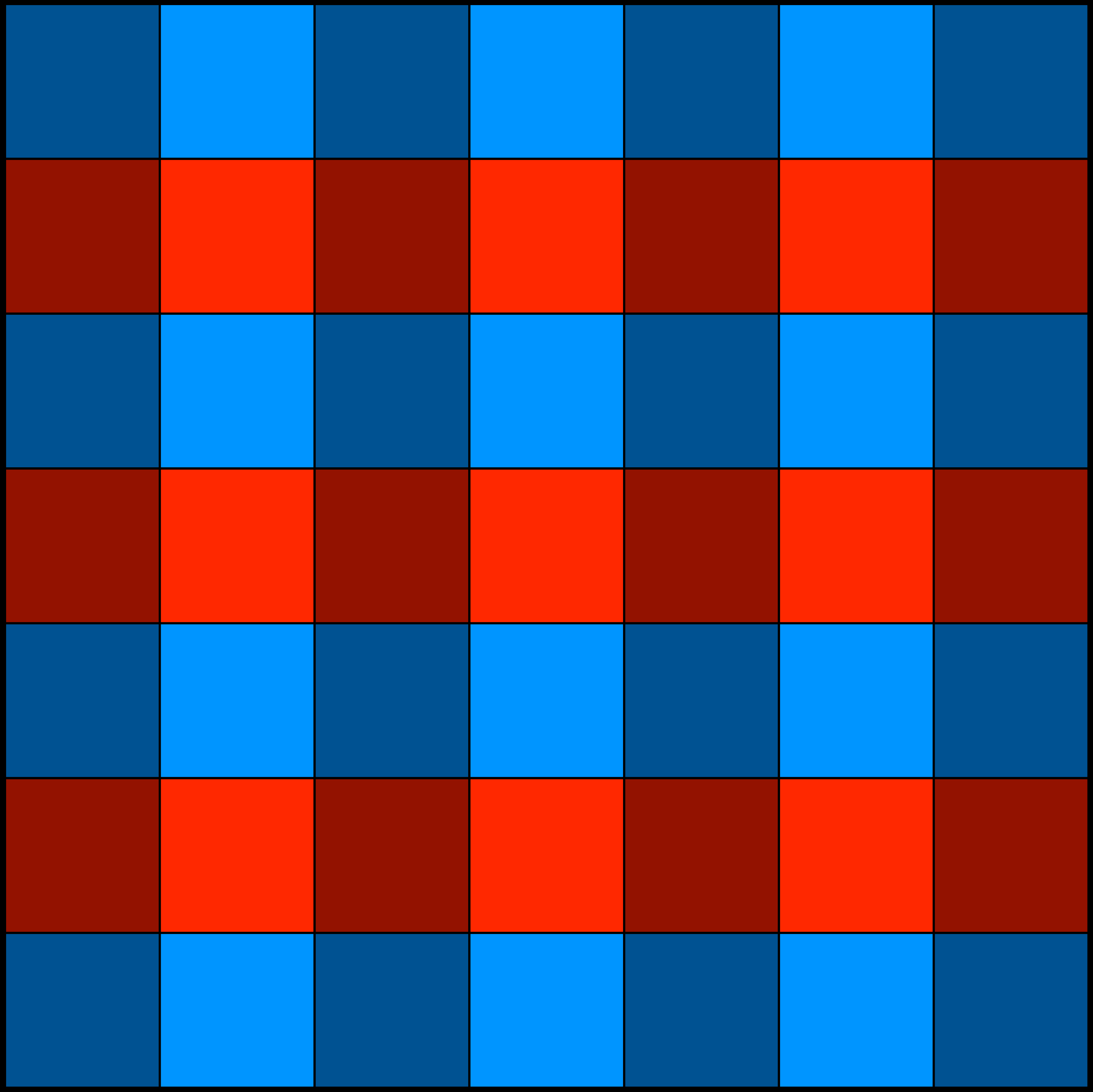
[Grünbaum Shephard 89] *Tilings and Patterns, an introduction*

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

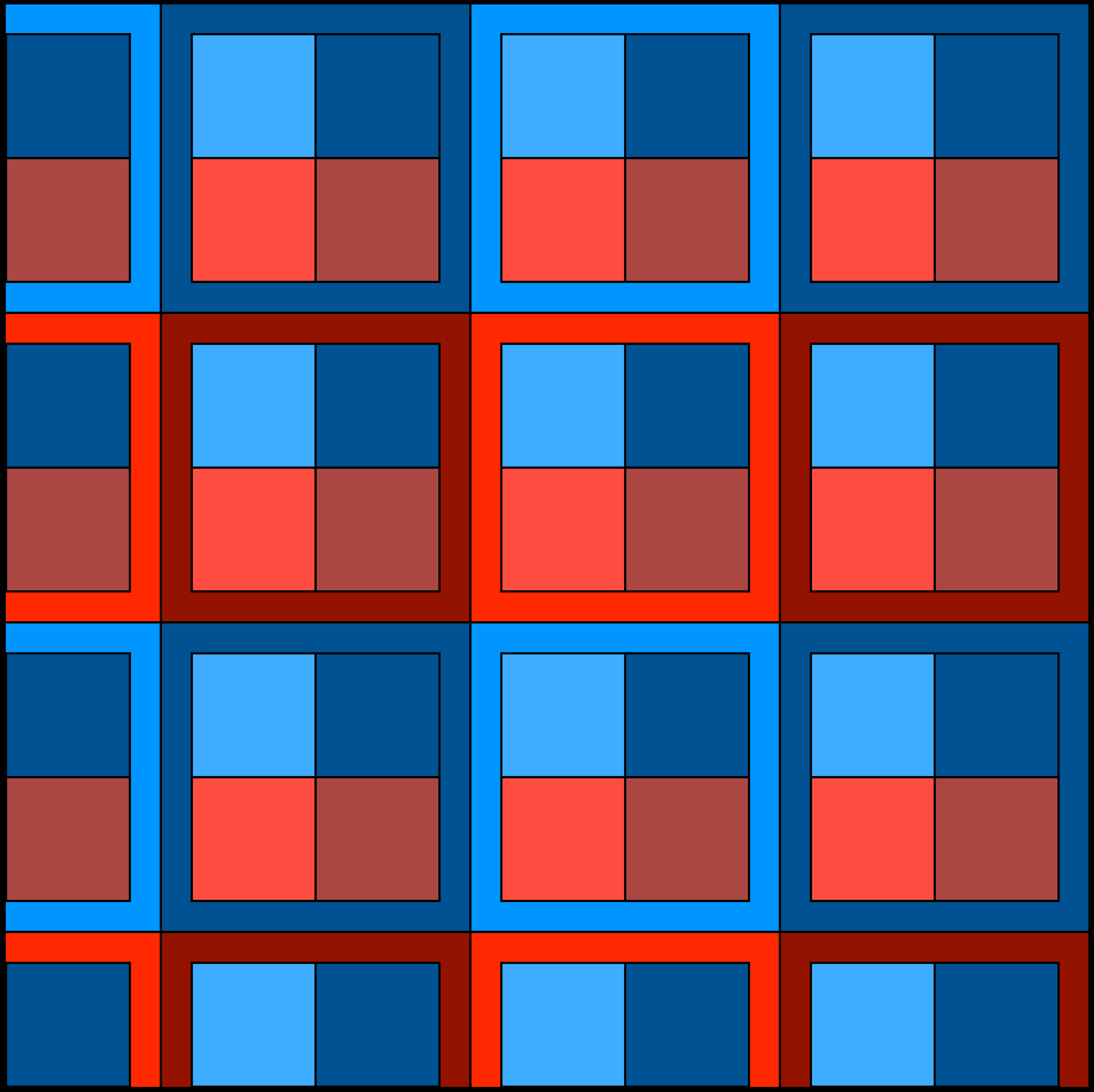
A set of 104 tiles



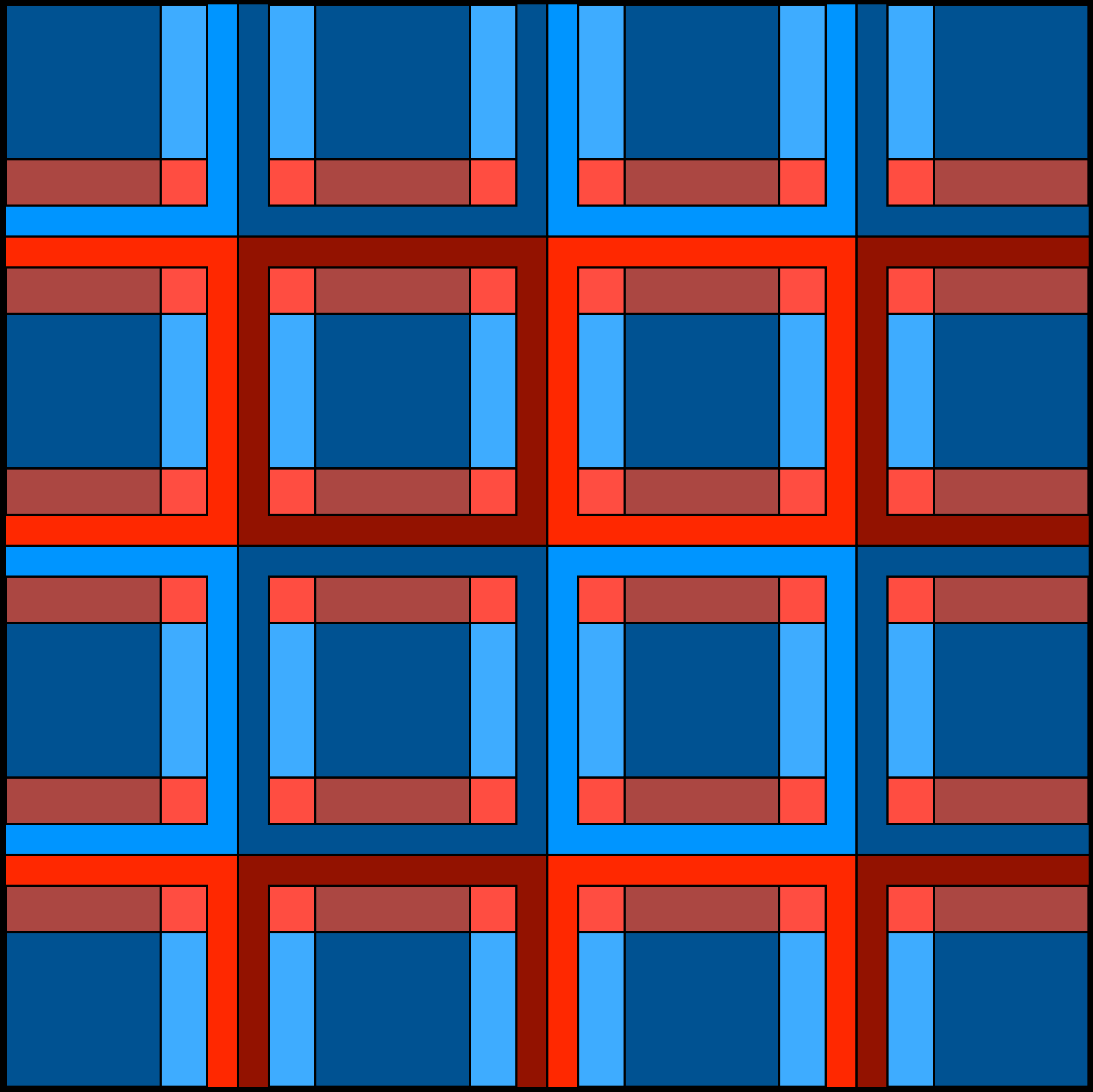
Encoding an infinite number of grids



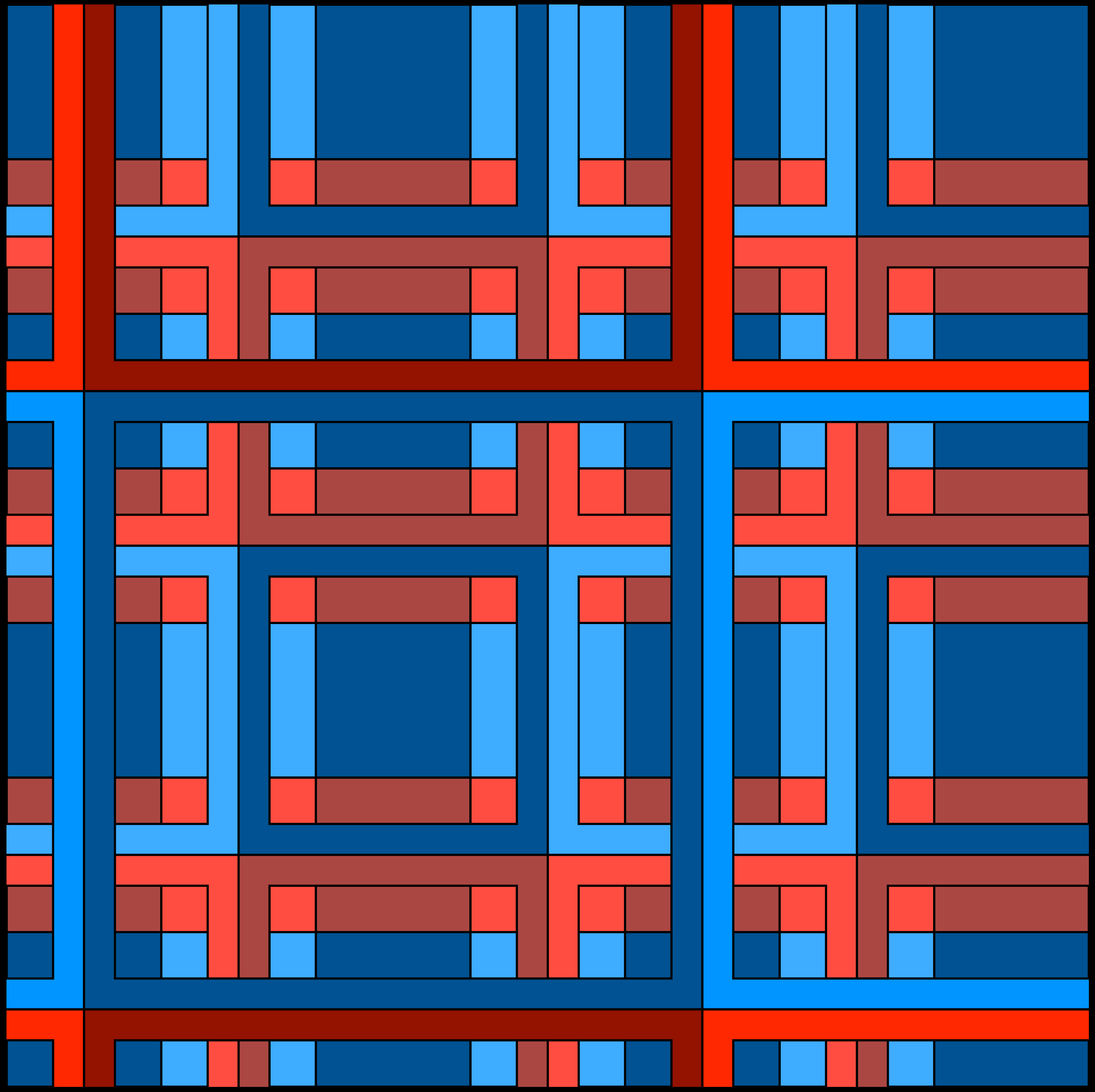
Encoding an infinite number of grids



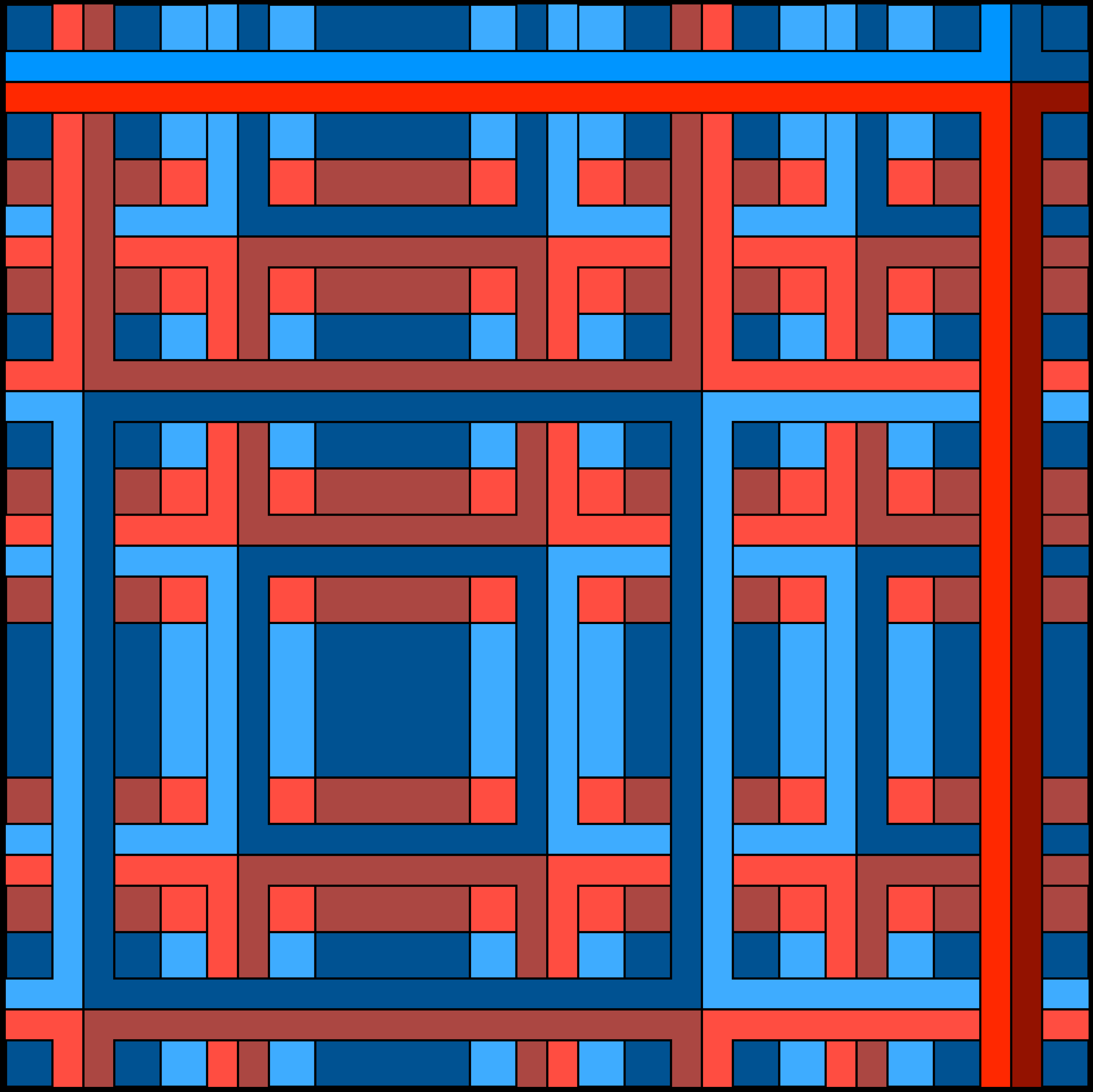
Encoding an infinite number of grids



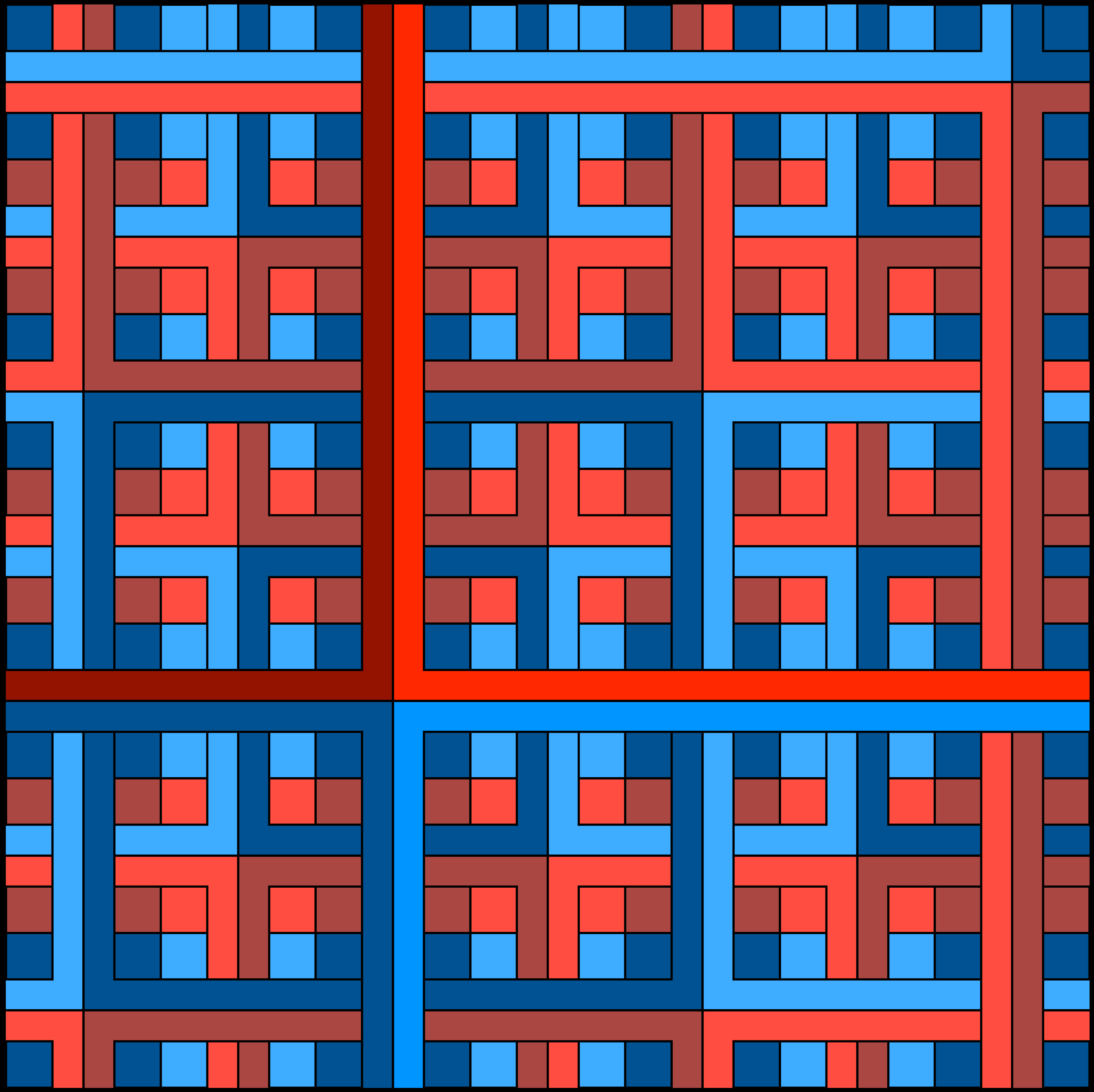
Encoding an infinite number of grids



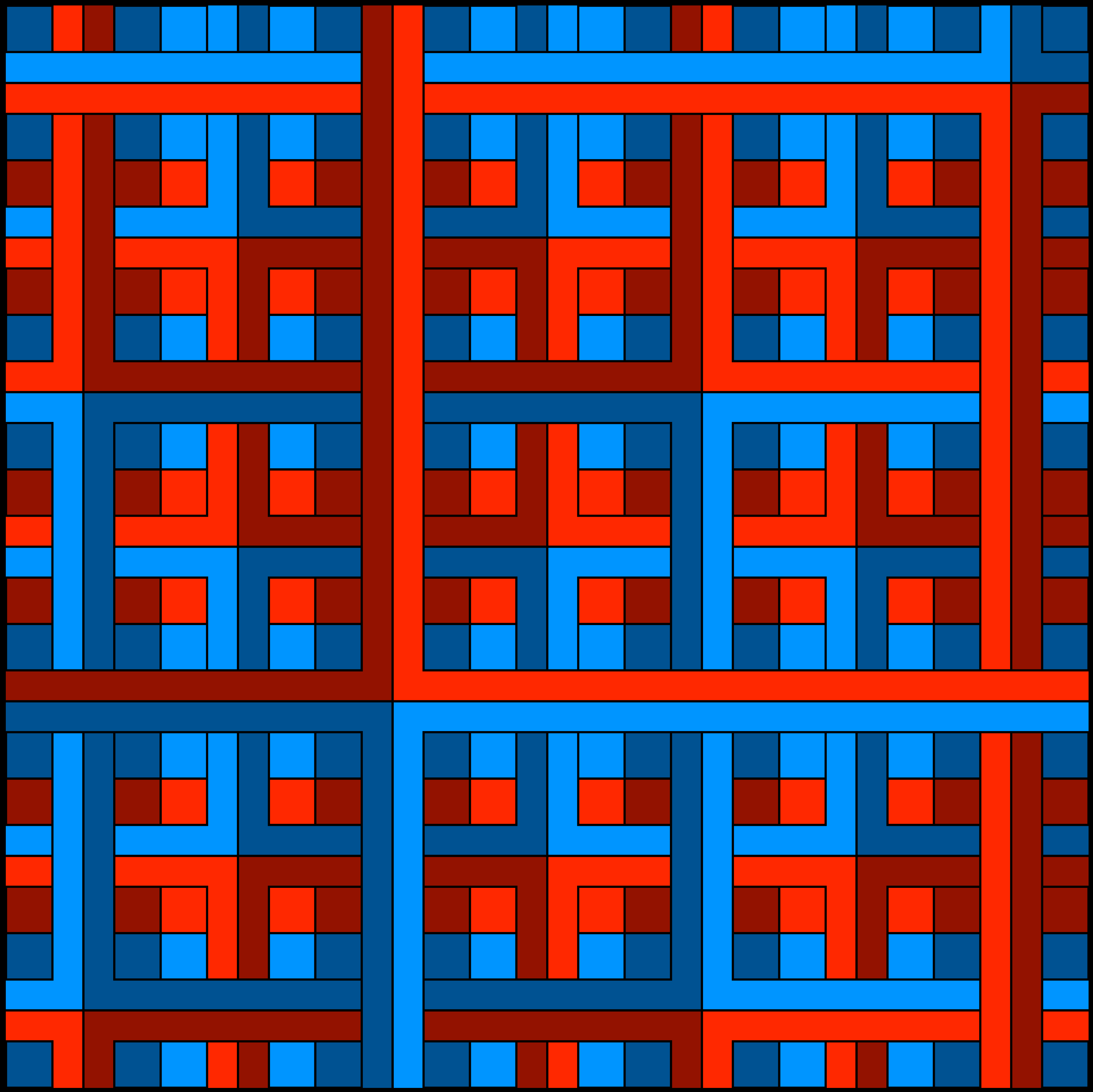
Encoding an infinite number of grids



Encoding an infinite number of grids



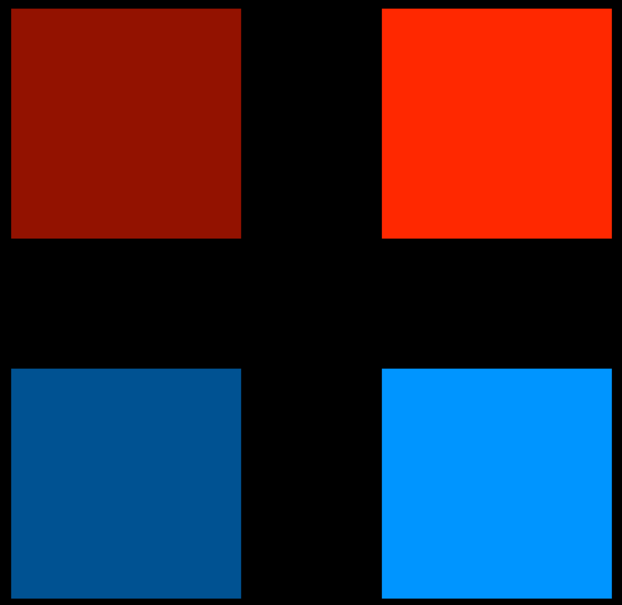
Encoding an infinite number of grids

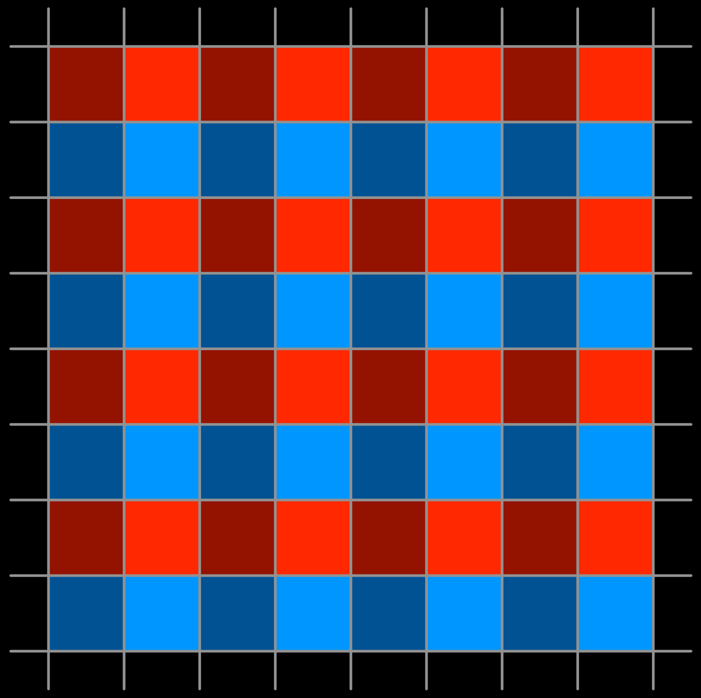


Encoding an infinite number of grids

Layer 1. initialization

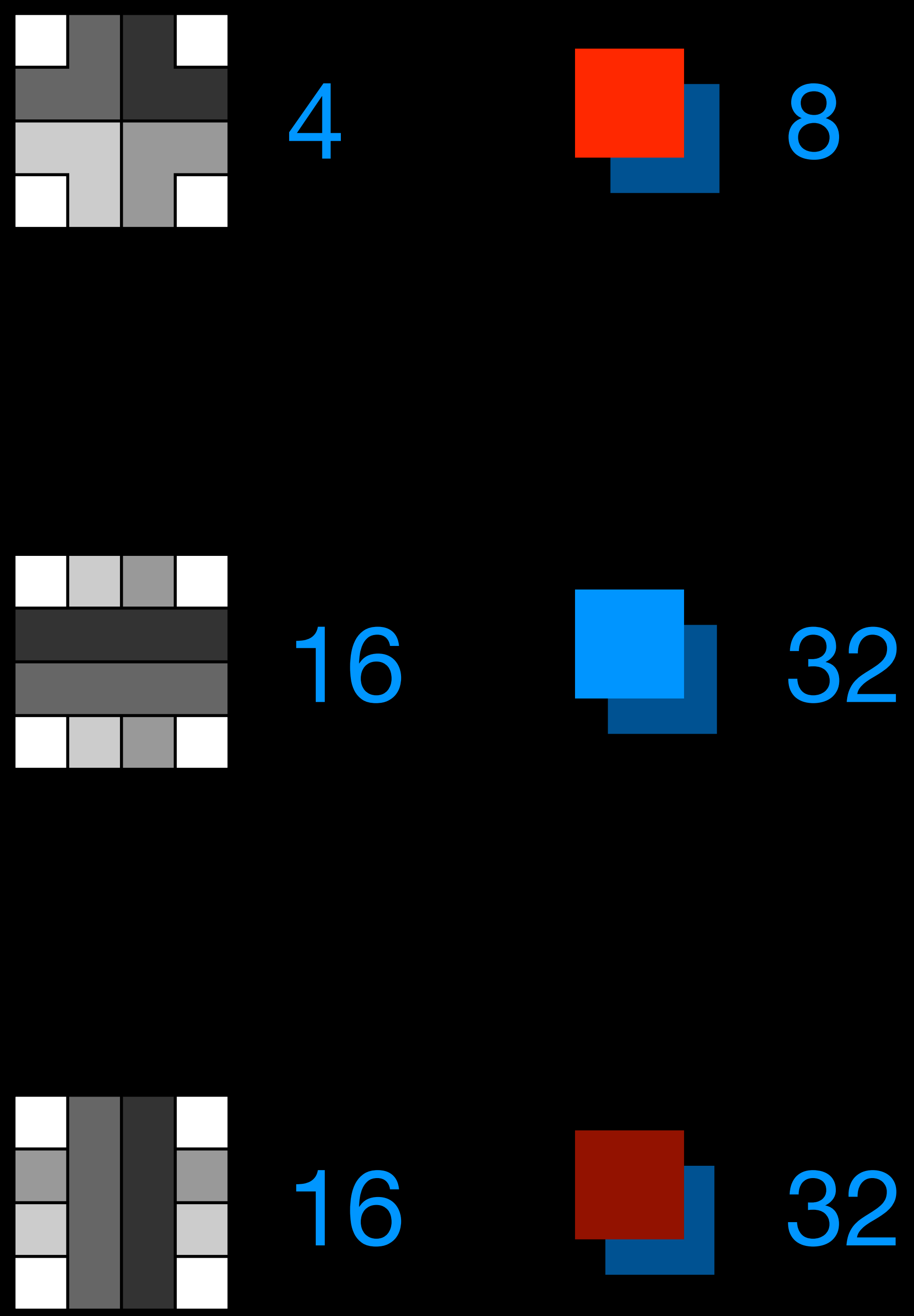
- 4 tiles to group tiles 2×2.
- Alternate red/blue vertically.
- Alternate light/dark horizontally.
- Simple and very constrained matching rule.





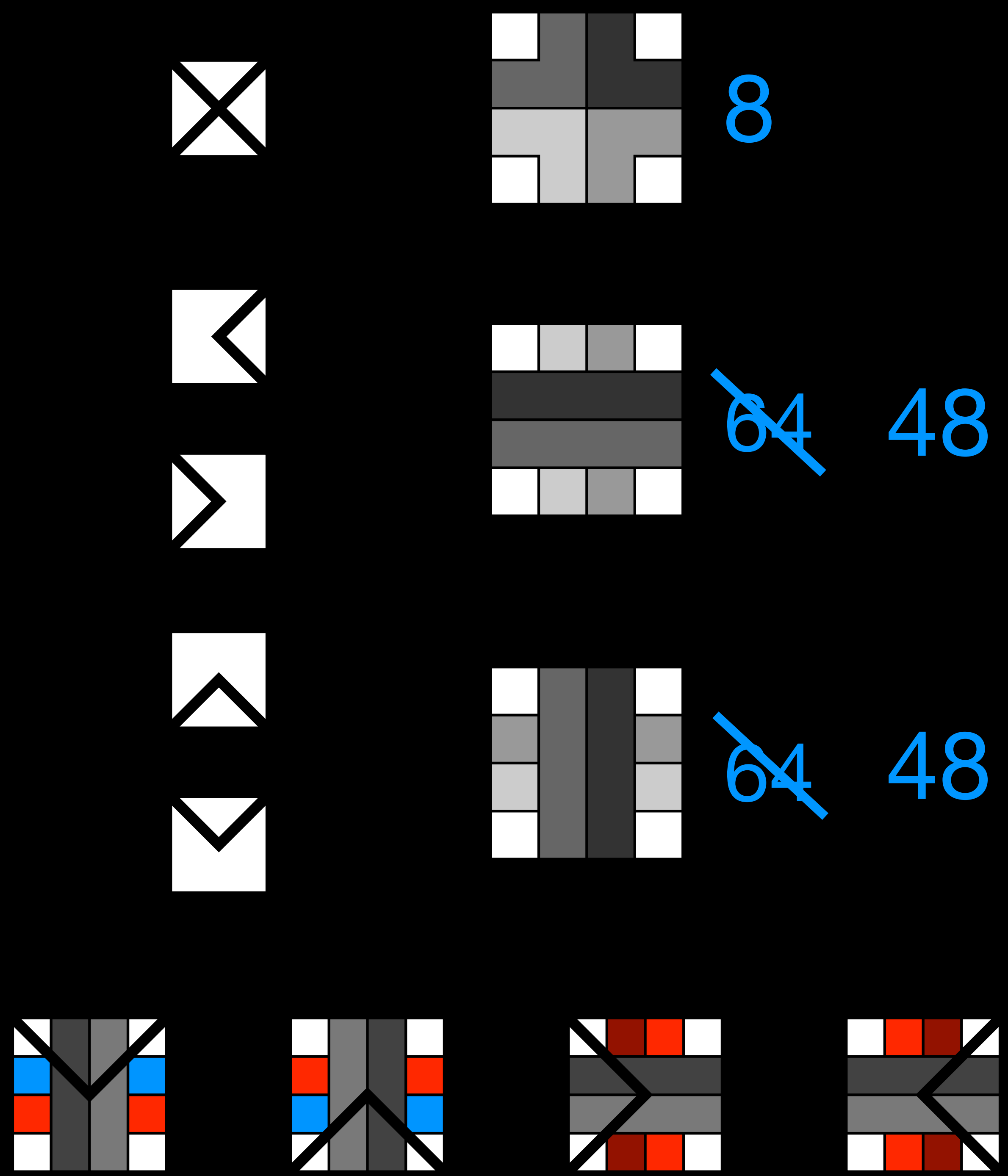
Layer 2. grids

- Tiles carry wires, two on each edge.
- Three kinds of tiles: X, H, V.
- Pairs of wires should be compatible.
- Matching rule: wire colors should match.
- Restrict layer 1 by kind.



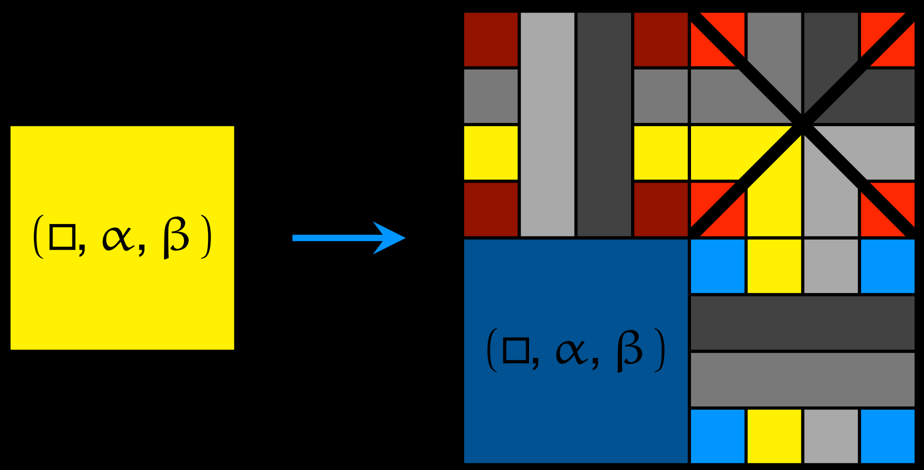
Layer 3. square-*ifier*

- One more bit of information per edge to force squares.
- 5 different tiles with arrows.
- Matching rule: arrows should match along edges.
- Restrict arrows by kind.
- **Important:** red wire is a one-way lane.

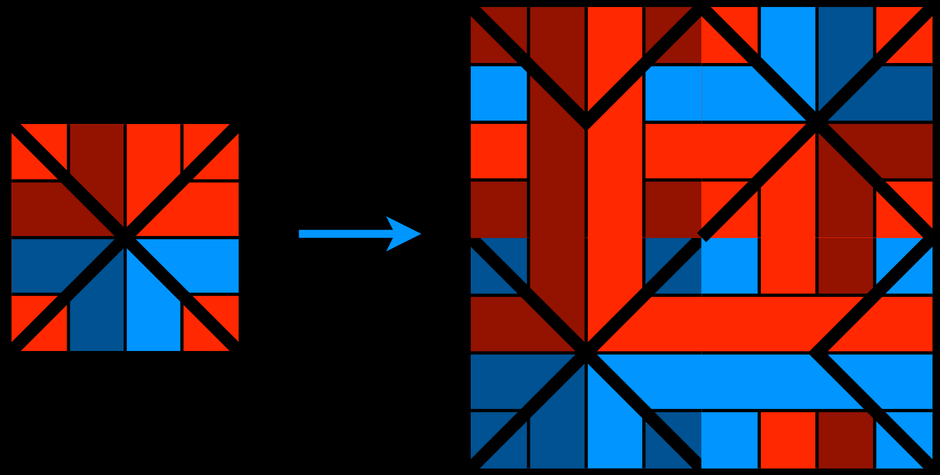


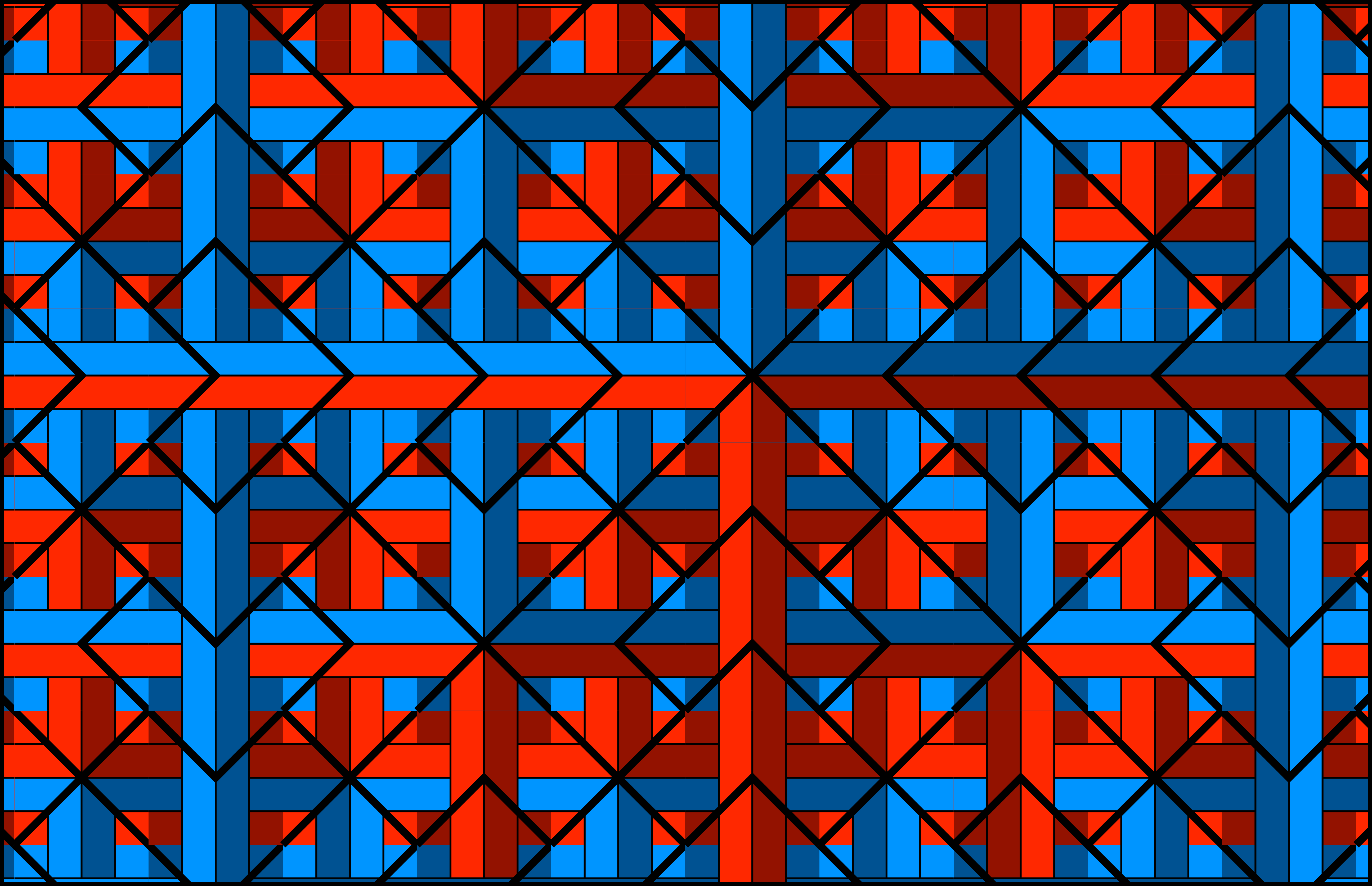
Canonical substitution

- 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.

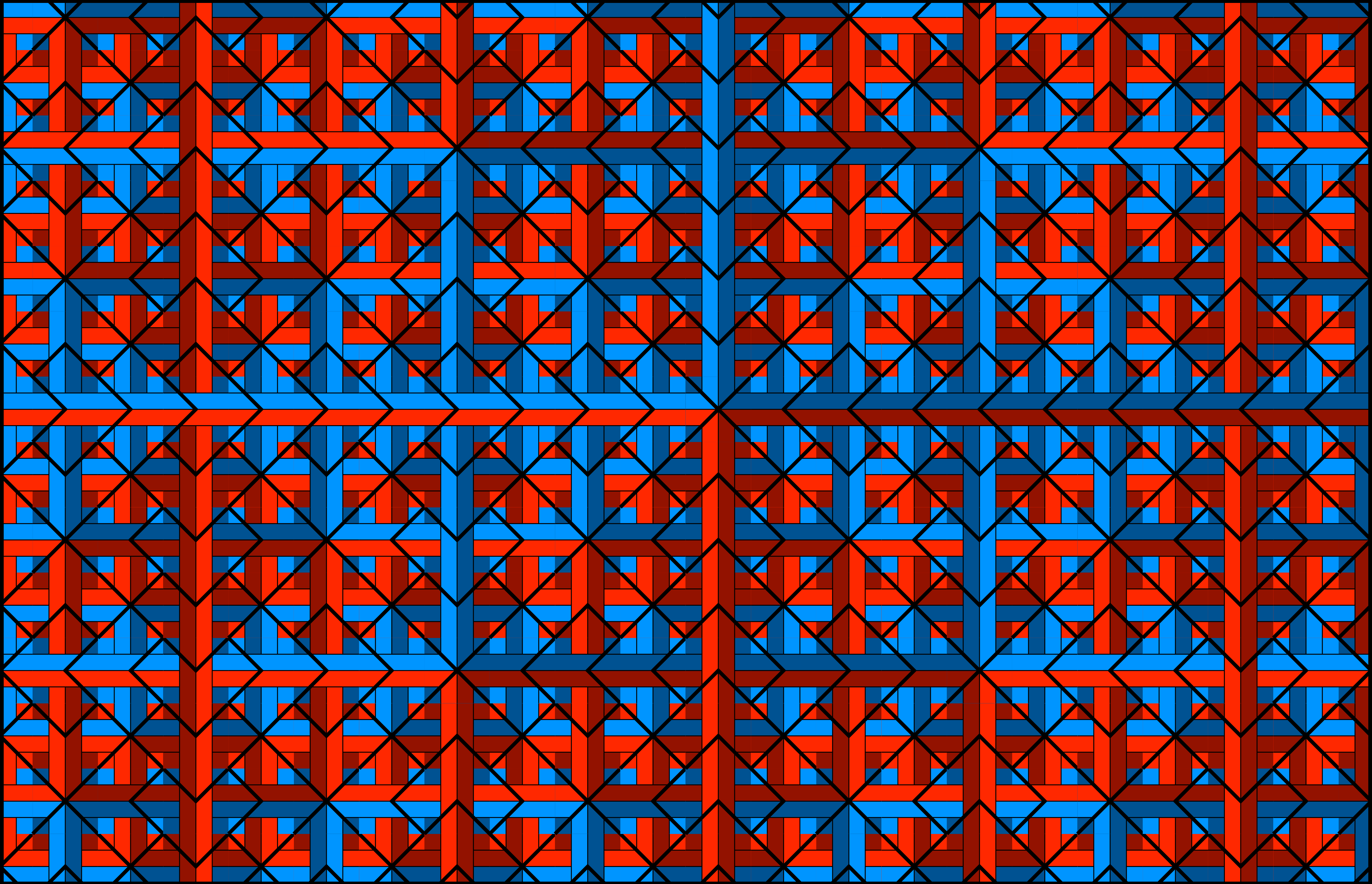


example:





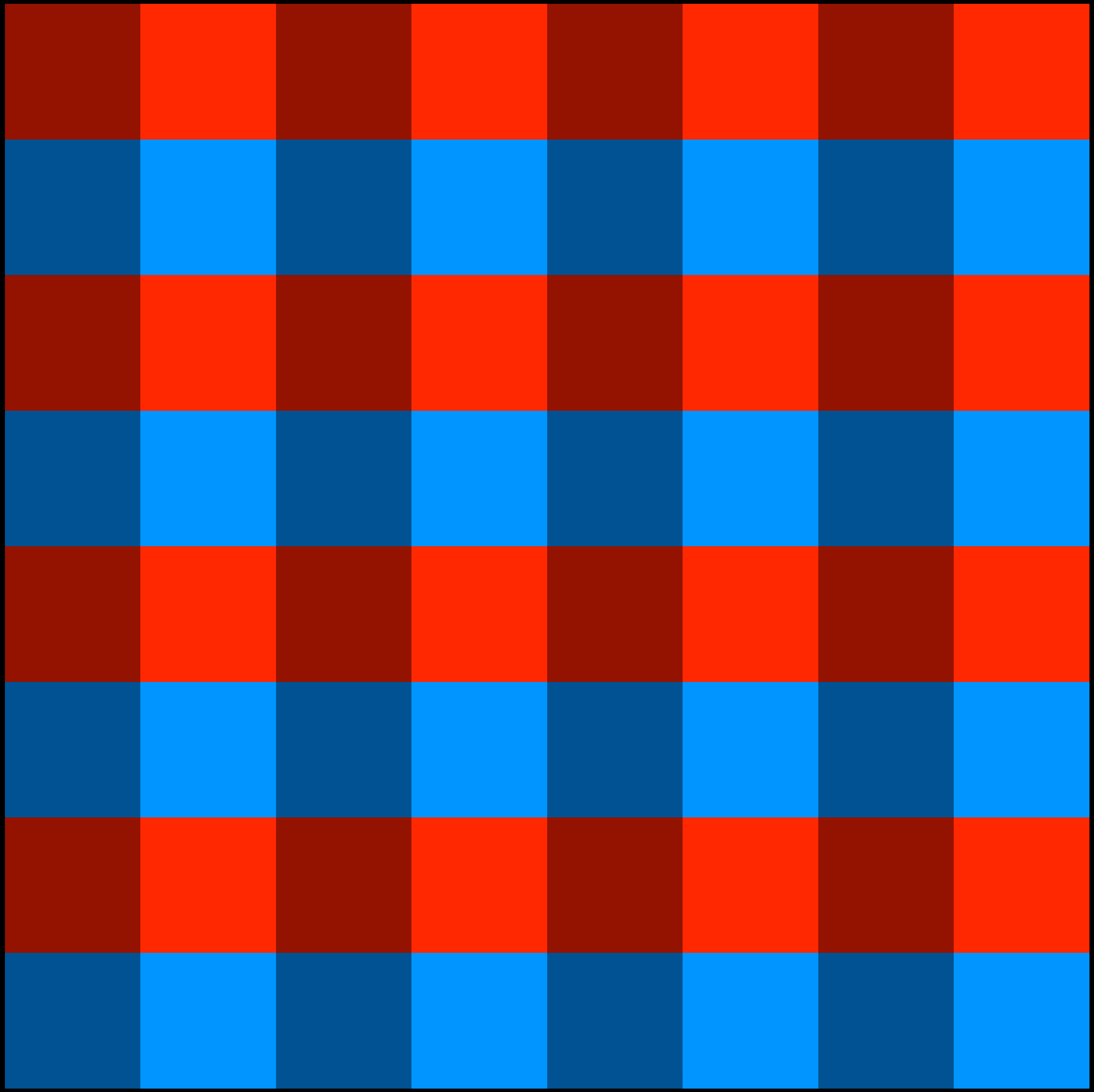
The tileset can tile the whole plane...



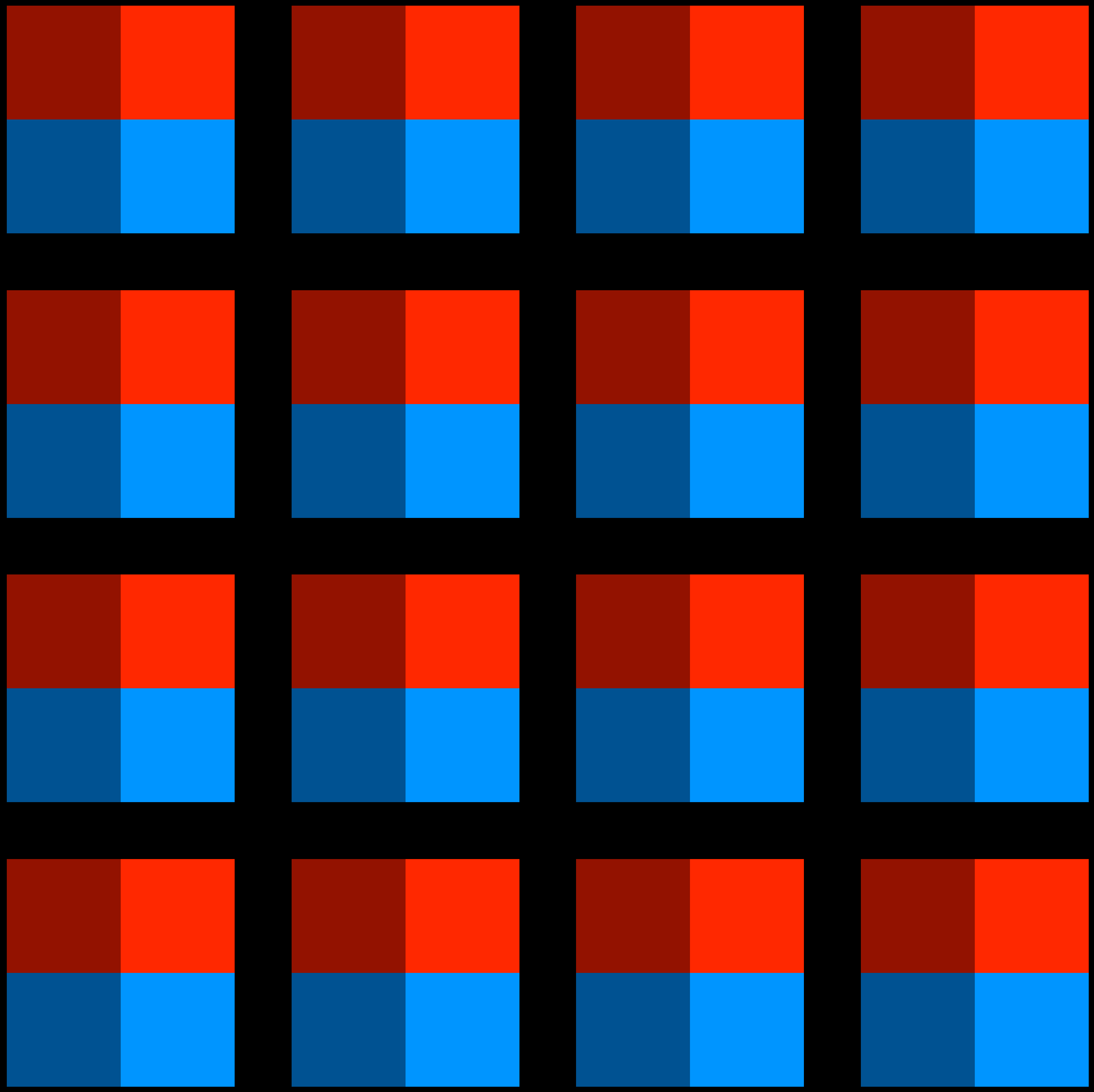
...extend *ad lib*!

This set of tiles is aperiodic

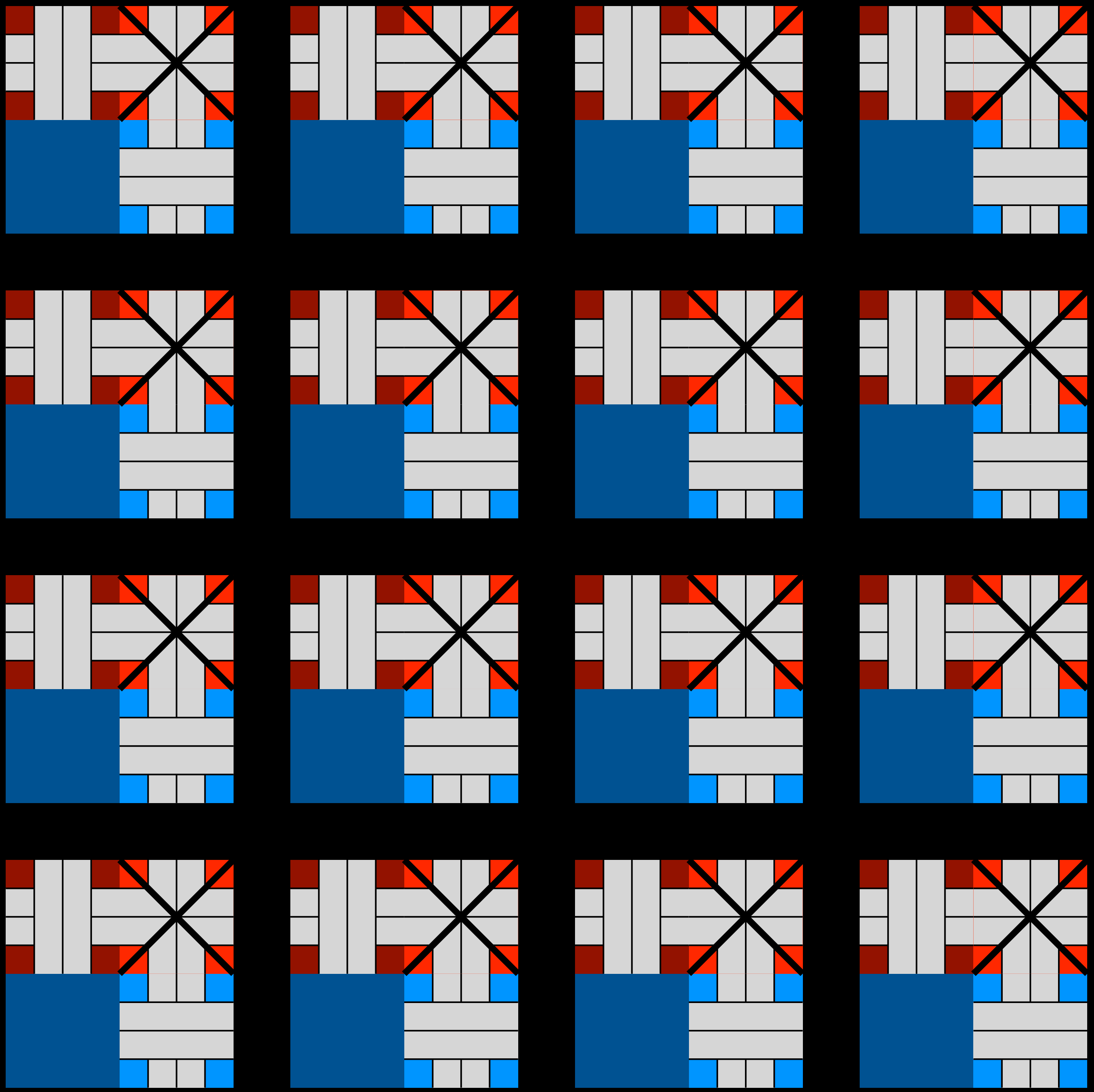
Lemma 1. each tiling from this set is an image of the canonical substitution.



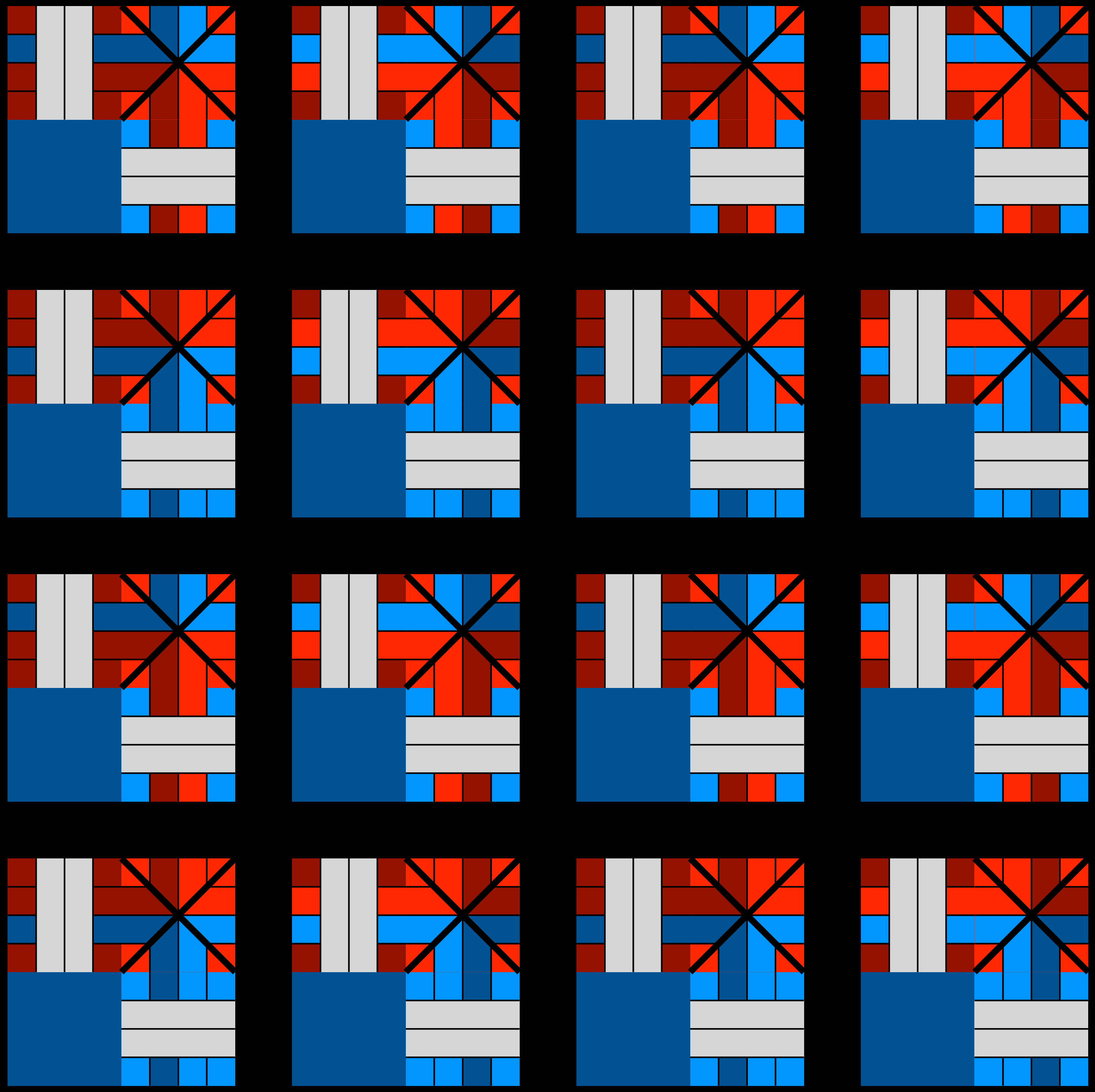
Lemma 1. each tiling from this set is an image of the canonical substitution.



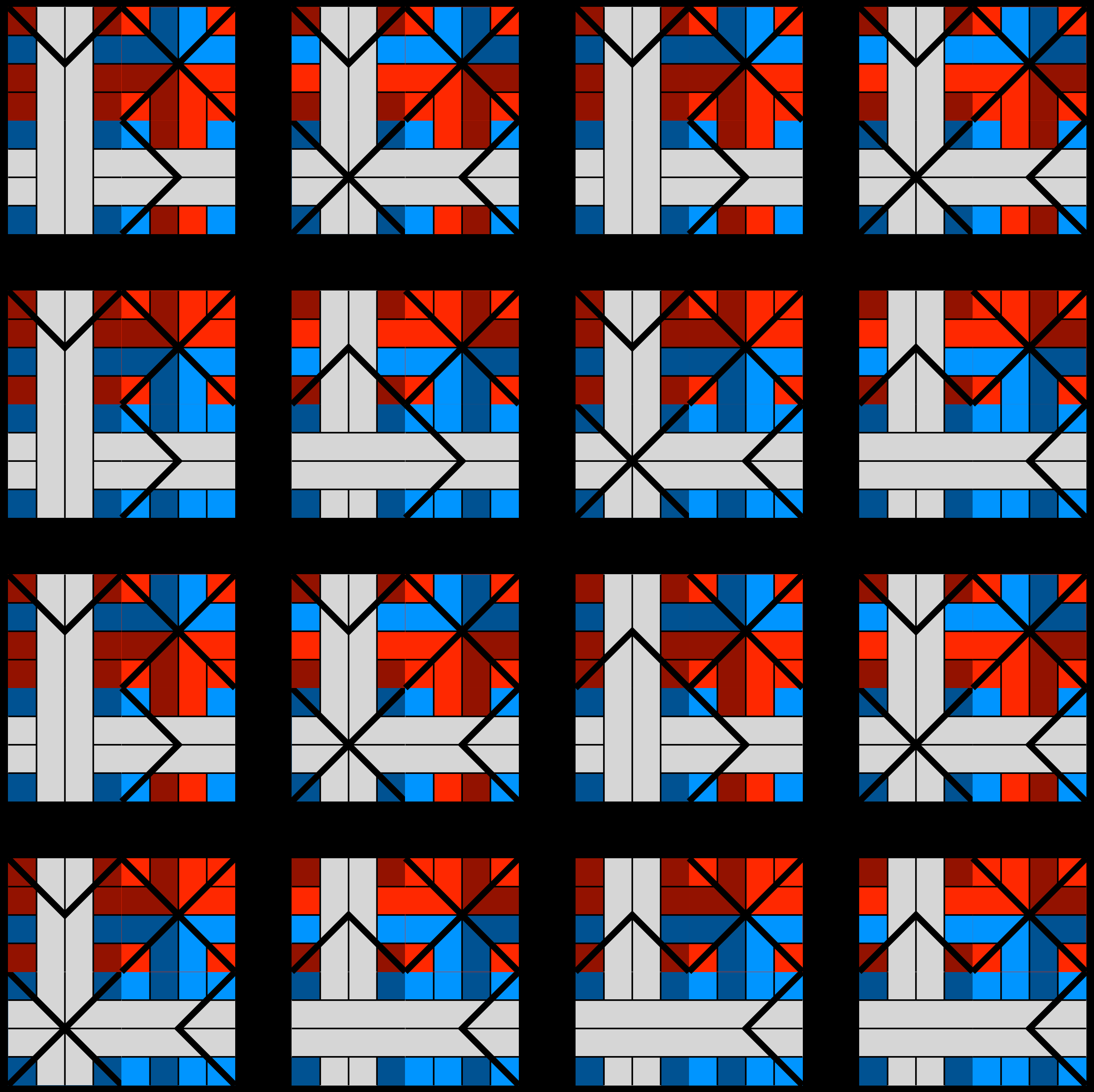
Lemma 1. each tiling from this set is an image of the canonical substitution.



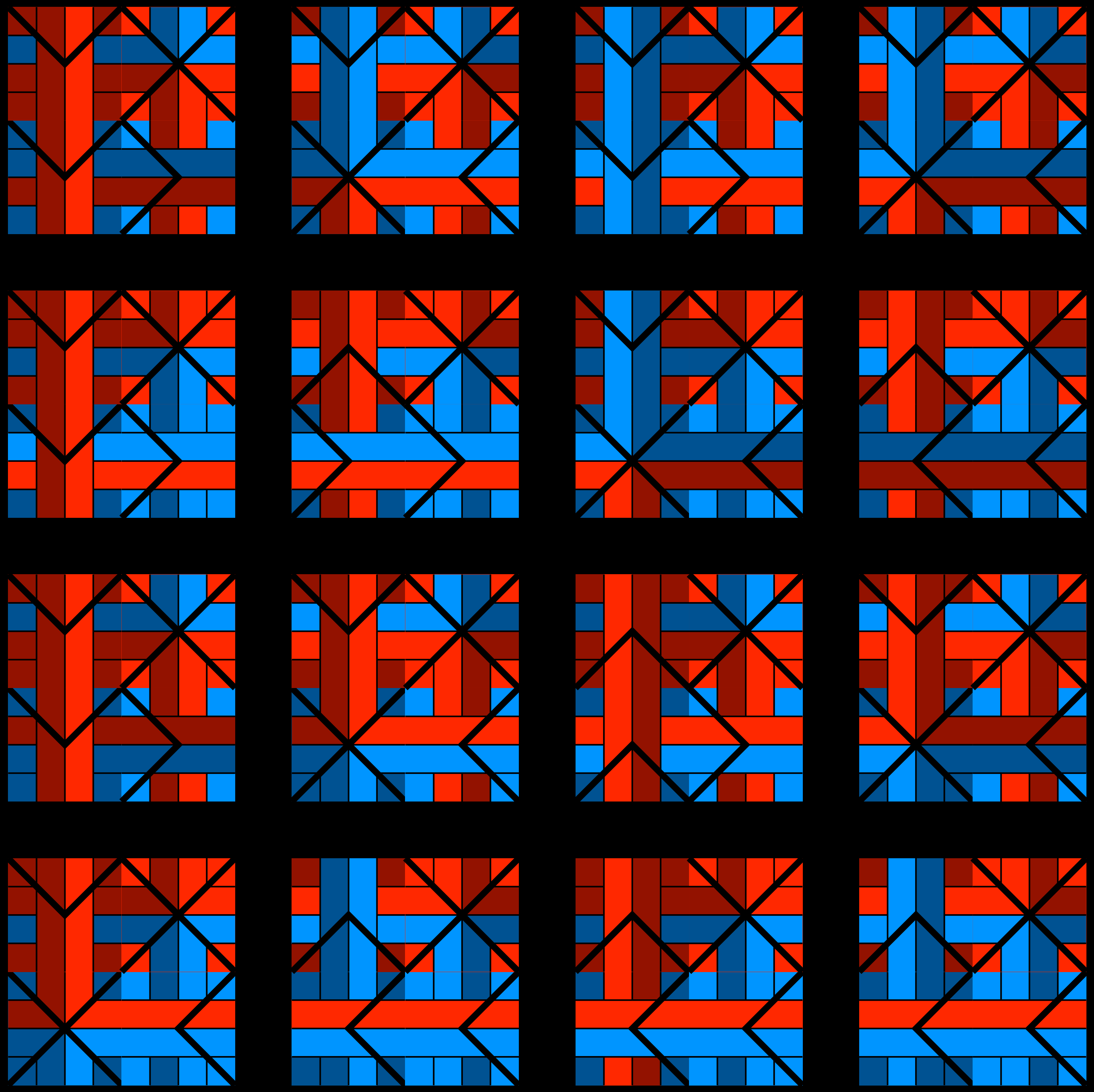
Lemma 1. each tiling from this set is an image of the canonical substitution.



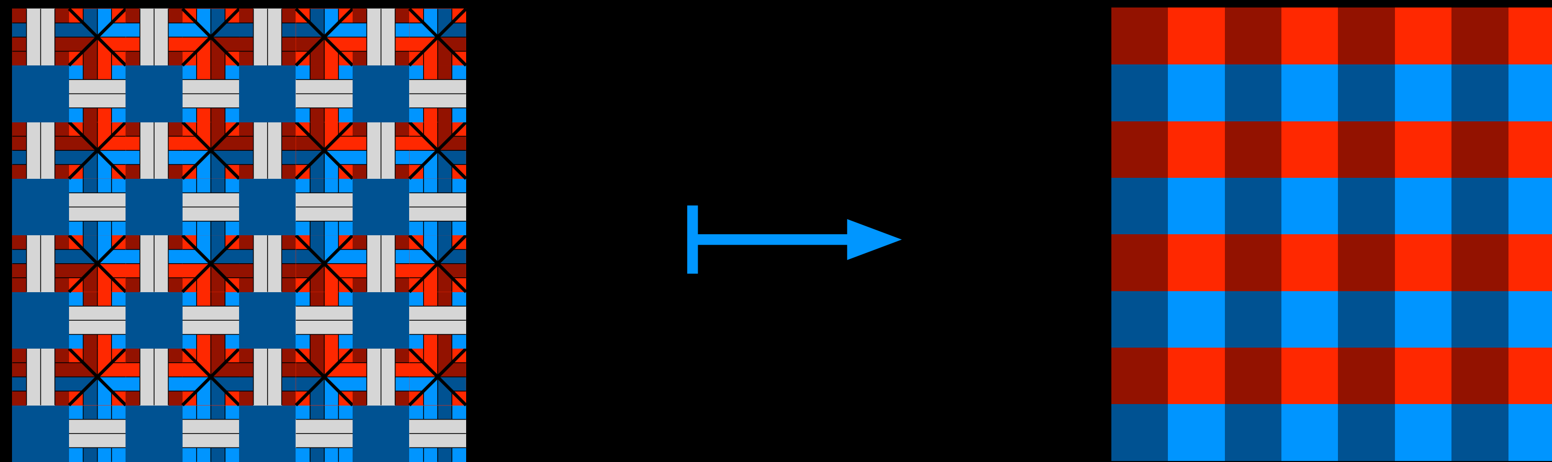
Lemma 1. each tiling from this set is an image of the canonical substitution.



Lemma 1. each tiling from this set is an image of the canonical substitution.

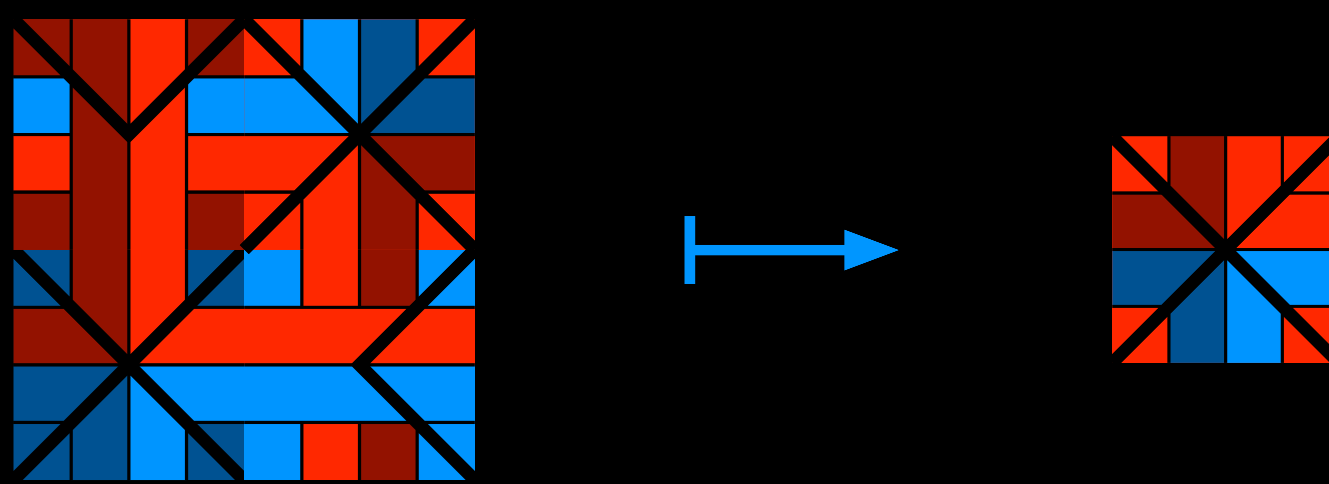


Lemma 1. each tiling from this set is an image of the canonical substitution.



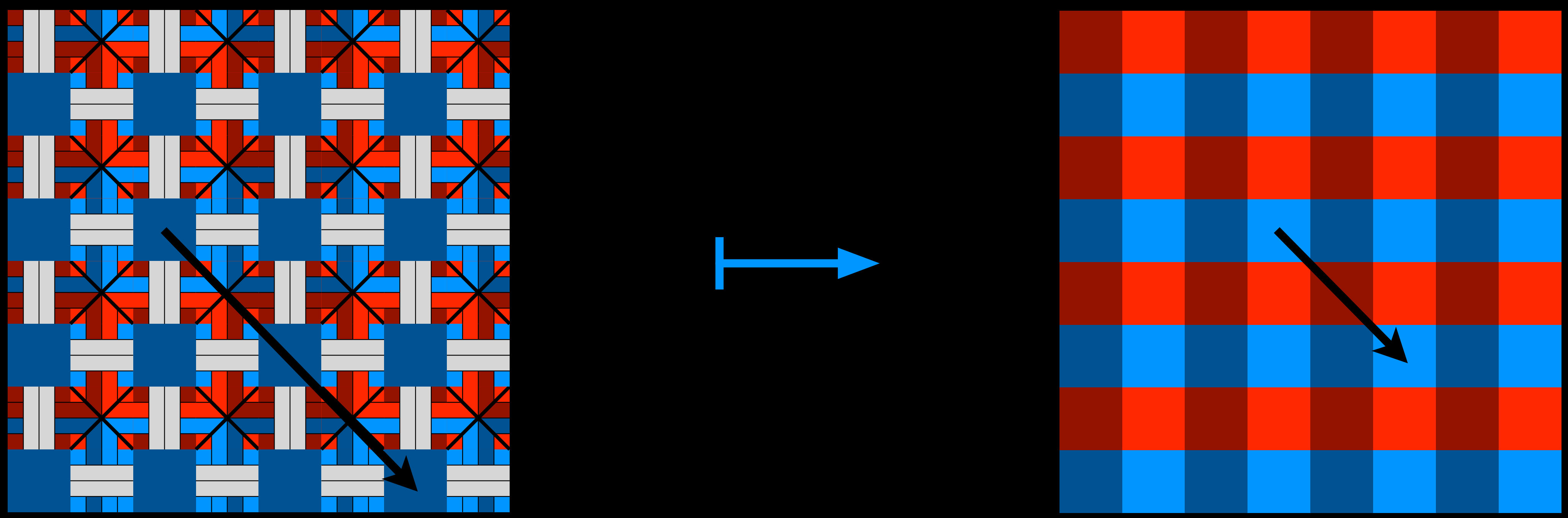
Layer 1

we just remove wires and squarifiers that propagate



Layers 2 & 3

Lemma 2. the preimage of each tiling is a (valid) tiling.

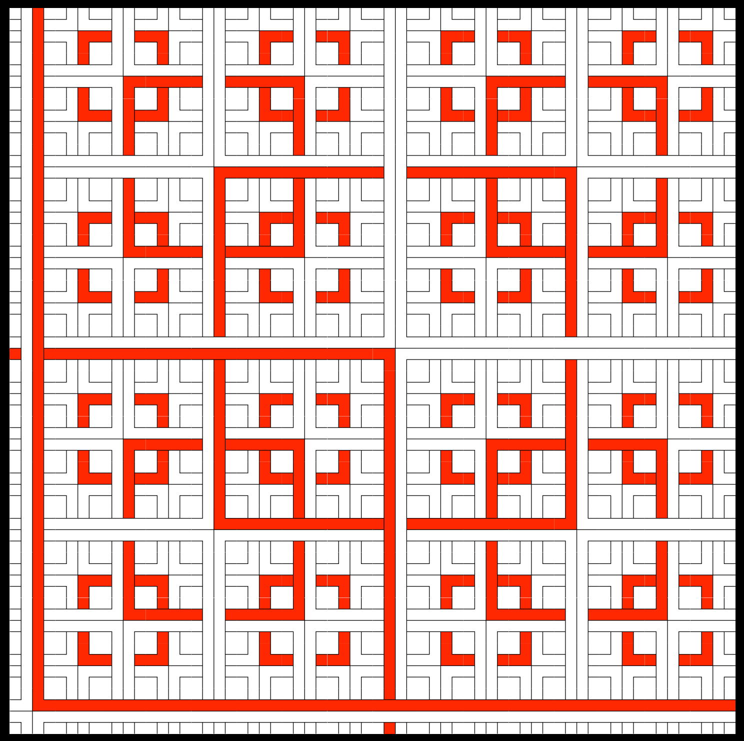


Lemma 3. a periodic tiling of period z has a preimage of period z/2.

Back to the Domino Problem

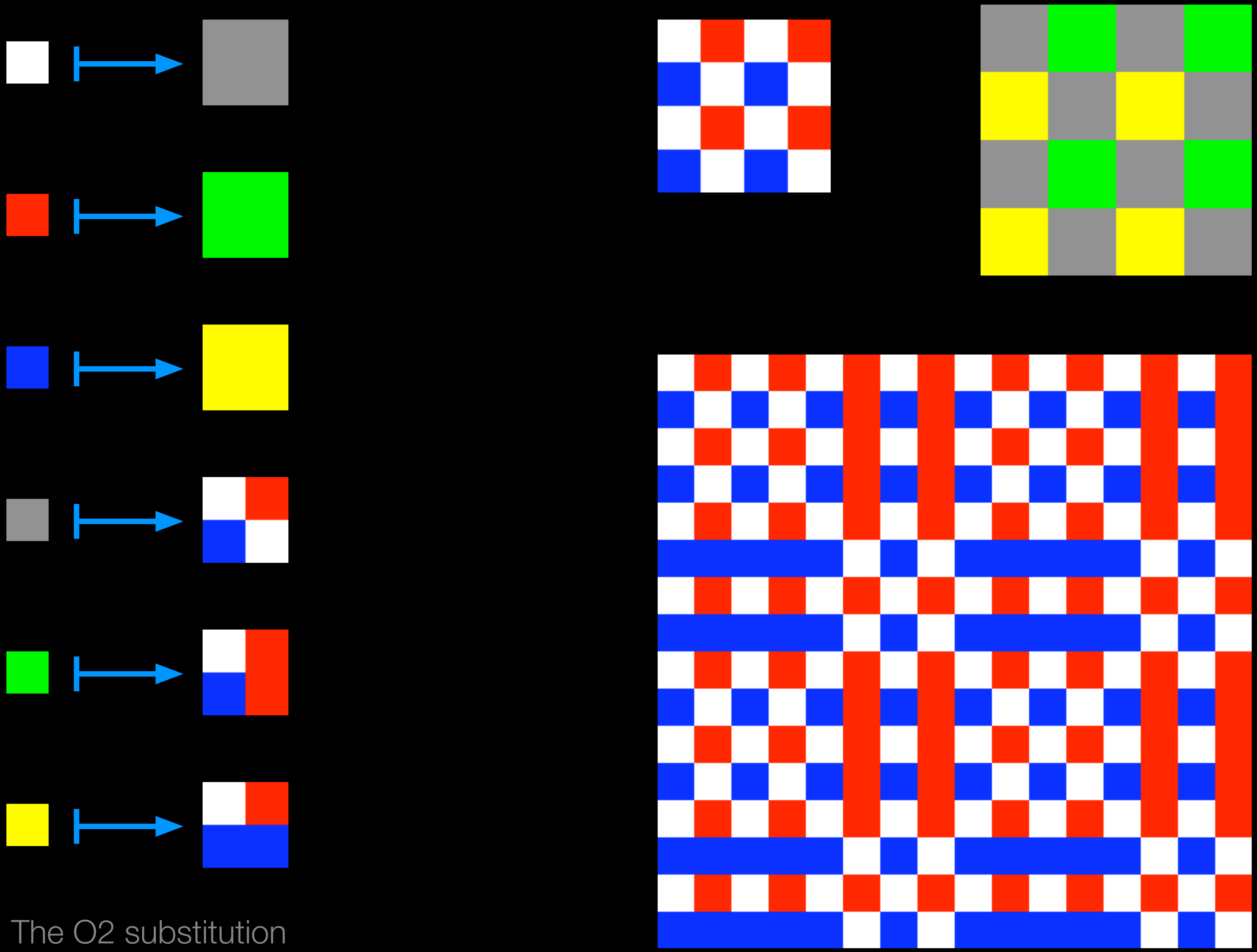
Encode substitutions

- Select one color of wires (say red): the sequence of squares encode a tree.
- Put a letter on each square.
- Apply the substitution rule to go from one square to the next one.
- Up to projection the substitution limit set is encoded in the set of tilings.

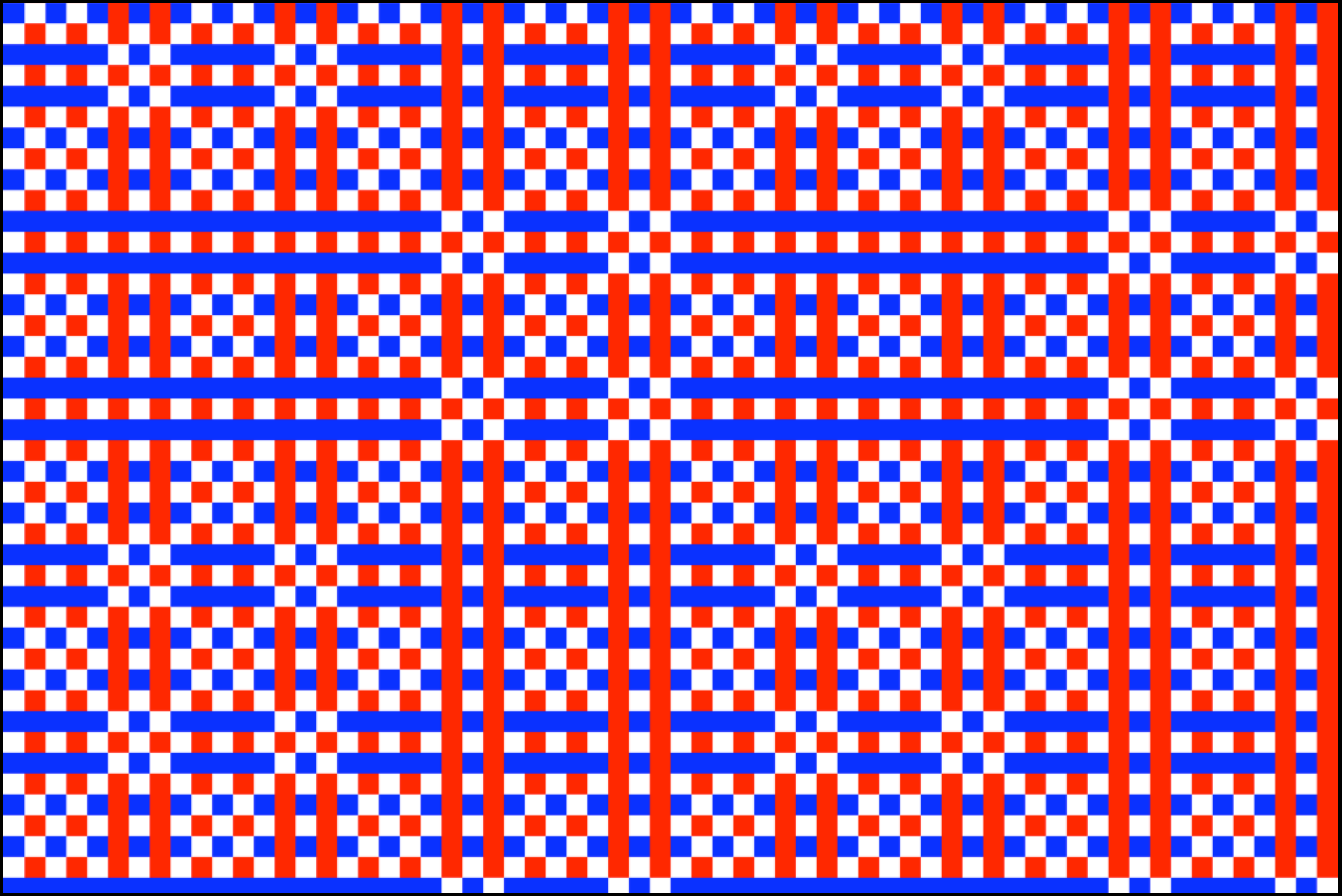


a

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



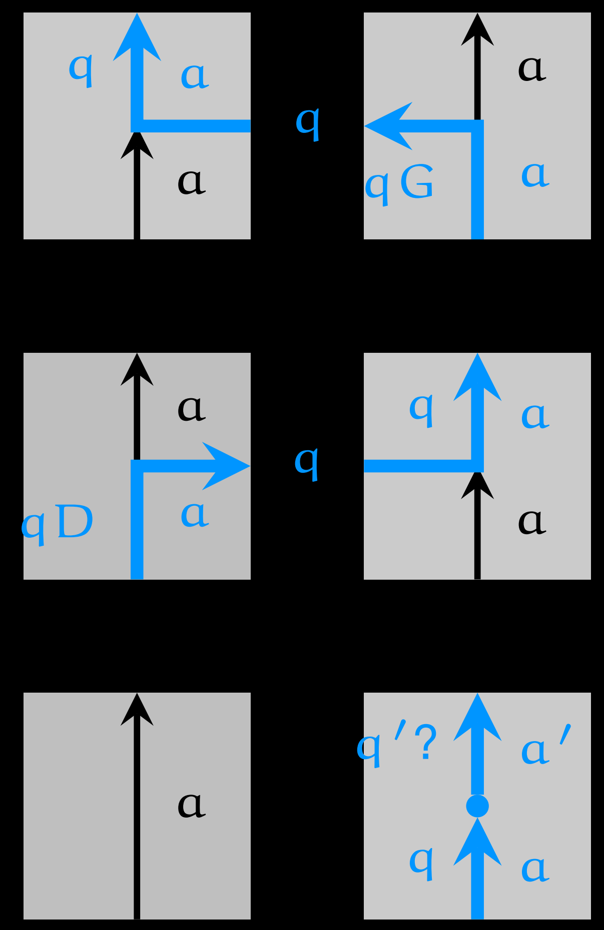
The O2 substitution



O2 iteration

Turing computation

- Turing machines can be simulated with tilesets.
- Put an initial computation at the SW corner of each O2 computation square.
- Remove the halting state.
- The tileset tiles the plane iff the Turing machine does not halt.



That’s all folks!