Fixed-point tile sets and their applications

Bruno Durand, Andrei Romashchenko, Alexander Shen

Fixed-point tile sets and their applications

Bruno Durand[*], Andrei Romashchenko[†], Alexander Shen[‡]

January 11, 2010

Abstract

An aperiodic tile set was first constructed by R. Berger while proving the undecidability of the domino problem. It turned out that aperiodic tile sets appear in many topics ranging from logic (the Entscheidungsproblem) to physics (quasicrystals).

We present a new construction of an aperiodic tile set that is based on Kleene's fixed-point construction instead of geometric arguments. This construction is similar to J. von Neumann self-reproducing automata; similar ideas were also used by P. Gács in the context of error-correcting computations.

This construction it rather flexible, so it can be used in many ways: we show how it can be used to implement substitution rules, to construct strongly aperiodic tile sets (any tiling is far from any periodic tiling), to give a new proof for the undecidability of the domino problem and related results, characterize effectively closed 1D subshift it terms of 2D shifts of finite type (improvement of a result by M. Hochman), to construct a tile set which has only complex tilings, and to construct a "robust" aperiodic tile set that does not have periodic (or close to periodic) tilings even if we allow some (sparse enough) tiling errors. For the latter we develop a hierarchical classification of points in random sets into islands of different ranks.

Finally, we combine and modify our tools to prove our main result: there exists a tile set such that all tilings have high Kolmogorov complexity even if (sparse enough) tiling errors are allowed.

Some of these results were included in the DLT extended abstract [8] and in the ICALP extended abstract [9].

[*]Laboratoire d'Informatique Fondamentale de Marseille, CNRS & Univ. Aix–Marseille. Supported in part by ANR Sycomore and NAFIT ANR-08-EMER-008-01grants.

[†]Laboratoire d'Informatique Fondamentale de Marseille, CNRS & Univ. Aix–Marseille. On leave from IITP RAS, Moscow. Supported in part by ANR Sycomore, NAFIT ANR-08-EMER-008-01, and RFBR 09-01-00709-a grants.

[‡]Laboratoire d'Informatique Fondamentale de Marseille, CNRS & Univ. Aix–Marseille. On leave from IITP RAS, Moscow. Supported in part by ANR Sycomore, NAFIT ANR-08-EMER-008-01, and RFBR 09-01-00709-a grants.

Contents

1 Introduction

In this paper, *tiles* are unit squares with colored sides. Tiles are considered as prototypes: we may place translated copies of the same tile into different cells of a cell paper (rotations are not allowed). Tiles in the neighbor cells should match (common side should have the same color in both).

Formally speaking, we consider a finite set C of *colors*. A *tile* is a quadruple of colors (left, right, top and bottom ones), i.e., an element of C^4. A *tile set* is a subset $\tau \subset C^4$. A *tiling* of the plane with tiles from τ (τ-*tiling*) is a mapping $U : \mathbb{Z}^2 \to \tau$ that respects the color matching condition. A tiling U is *periodic* if it has a *period*, i.e., a non-zero vector $T \in \mathbb{Z}^2$ such that $U(x+T) = U(x)$ for all $x \in \mathbb{Z}^2$. Otherwise the tiling is *aperiodic*. The following classical result was proved by Berger in a paper [2] where he used this construction as a main tool to prove *Berger's theorem*: the *domino problem* (to find out whether a given tile set has tilings or not) is undecidable.

Theorem 1. *There exists a tile set τ such that τ-tilings exist and all of them are aperiodic.* [2]

The first tile set of Berger was rather complicated. Later many other constructions were suggested. Some of them are simplified versions of the Berger's construction ([26], see also the expositions in [1, 6, 20]). Some others are based on polygonal tilings (including famous Penrose and Ammann tilings, see [13]). An ingenious construction suggested in [17] is based on the multiplication in a kind of positional number system and gives a small aperiodic set of 14 tiles (in [4] an improved version with 13 tiles is presented). Another nice construction with a short and simple proof (based explicitly on ideas of self-similarity) was recently proposed by N. Ollinger [24].

In this paper we present yet another construction of aperiodic tile set. It does not provide a small tile set; however, we find it interesting because:

- The existence of an aperiodic tile set becomes a simple application of a classical construction used in Kleene's fixed point (recursion) theorem, in von Neumann's self-reproducing automata [23] and, more recently, in Gács' reliable cellular automata [10, 11]; we do not use any geometric tricks. The construction of an aperiodic tile set is not only an interesting result but an important tool (recall that it was invented to prove that domino problem is undecidable); our construction makes this tool easier to use.

- The construction is rather general, so it is flexible enough to achieve some additional properties of the tile set. We illustrate this flexibility providing new proof for several known results and proving new results; these new results add robustness (resistance to sparse enough errors) to known results about aperiodic tile sets and tile sets that have only complex tilings.

It is not clear whether this kind of robustness can be achieved for previously known constructions of tile sets; on the other hand, robustness properties look important. For example, a mathematical model for processes like quasicrystals' growth or DNA-computation should take errors into account. Note that our model (independent choice of places where errors are allowed) has no direct physical meaning; it is just a simple mathematical model that can be used as a playground to develop tools for estimating the consequences of tiling errors.

The paper is organized as follows. In Section 2 we present the fixed-point construction of an aperiodic tile set (new proof of Berger's theorem). Then we illustrate the flexibility of this construction by several examples:

- In Section 3 we show that any 'uniform' substitution rule can be implemented by a tile set (thus providing a new proof for this rather old result). Then in Section 4 we use substitutions to show that there are strongly aperiodic tile sets (this means that any tiling is strongly aperiodic, i.e., any shift changes at least some fixed fraction of tiles).

- Fixed-point construction of Section 2 provides a self-similar tiling: blocks of size $n \times n$ ("macro-tiles") behave exactly as individual tiles, so on the next level we have $n^2 \times n^2$ blocks made of $n \times n$ macro-tiles that have the same behavior, etc. In Section 5 we make some changes in our construction that allow us to get variable zoom factors (the numbers of tiles in macro-tiles increases as the level increases).

  Variable zoom factor tilings can be used for simulating computations (higher levels perform more computation steps); we use them to give a simple proof of the undecidability of the domino problem (main technical difficulty in the standard proof was to synchronize computations on different levels, now this is not needed at all); we show also that other undecidability results can be obtained in this way.

- This technique can be used to push the strong aperiodicity to its limits: the distance between any tiling and any periodic one (or between any tiling and its nontrivial shift) can be made arbitrarily close to 1, not only separated from 0. This is done in Section 6 using an additional tool: error-correcting codes.

- In [5] a tile set was constructed such that every tiling has maximal Kolmogorov complexity of fragments ($\Omega(n)$ for $n \times n$ squares); all tilings for this tile set are non-computable (so we get a classical result of Hanf [15] and Myers [22] as a corollary). The construction was rather complicated and was based on a classical construction of an aperiodic tile set. In Section 7 we provide another proof of the same result that uses variable zoom factors. It is simpler in some respects and can be generalized to produce robust tile sets with complex tiling, which is our main result (Section 13).

  Further in Section 8 we use the same technique to give a new proof ofor some results by S. Simpson [28] and M. Hochman [16] about effectively closed subshifts: every 1-dimensional effectively closed subshift can be obtained as a projection of configurations of some 2-dimensional subshift of finite type (in an extended alphabet). Our construction provides a solution of Problem 9.1 from [16].

- To prove the robustness of tile sets against sparse errors we use a hierarchical classification of the elements of random sets into islands of different levels (a method that goes back to Gács [11, 12]). This method is described in Section 9.1.

- In Section 9.2 we give definitions and establish some probabilistic results about islands that are used to prove robustness: we show that a sparse random set on $\mathbb{Z}^2$ with probability 1

(for Bernoulli ditribution) can be represented as a union of 'islands' of different ranks. The higher is the rank, the bigger is the size of an island; the island are well isolated from each other (in some neighborhood of an island of rank k there is no other islands of rank $\geq k$). Then in Section 9.3 we illustrate these tools using standard results of percolation theory as a model example. In Section 11 we generalize results of Section 9.4 and prove similar results for weaker restrictions for the involved parameters. To achieve this generalization, a more technically advanced island classification is needed. In this classification two islands of the same rank can be close to each other (but not more than two).

- In Section 10 we use fixed-point construction to get an aperiodic tile set that is robust in the following sense: if a tiling has a "hole" of size n, then this hole can be patched by changing only $O(n)$-size zone around it. Moreover, an $O(n)$ zone (with bigger constant in O-notation) around the hole is enough for this (we don't need to have the entire plane covered). In Section 11 we explain how to get a robust aperiodic tile sets with variable zoom factors.

- In Section 12 we combine the techniques developed to establish one of our main results: there exists a tile set such that every tiling of a plane except a sparse set of random points is far from every periodic tiling.

- Finally, the Section 13 contains our most technically difficult result: a robust tile set such that all tilings, even with a sparsely placed errors, have linear complexity of fragments. To this end we need all our techinque: fixed-point construction with variable zoom factors, splitting of a random set into doubled islands, and robustification with filling of doubled holes.

2 Fixed-point aperiodic tile set

2.1 Macro-tiles

Fix a tile set τ and an integer $N > 1$ (*zoom factor*). A *macro-tile* is an $N \times N$ square tiled by matching τ-tiles (i.e., a square block of N^2 tiles). Every side of a macro-tile consists of a sequence of N colors called a *macro-color*.

Let ρ be a set of τ-macro-tiles. We say that τ *simulates* ρ if (a) τ-tilings exist, and (b) for every τ-tiling there exists a unique grid of vertical and horizontal lines that cuts this tiling into $N \times N$ macro-tiles from ρ.

Example 1. Assume that we have only one ('white') color and τ consists of a single tile with 4 white sides. Fix some N. There exists a single macro-tile of size $N \times N$. Let ρ be a singleton that contains this macro-tile. Then every τ-tiling can be cut into macro-tiles from ρ. However, τ does not simulate ρ, since the placement of cutting lines is not unique.

Example 2. In this example a set ρ that consists of exactly one macro-tile (that has the same macro-colors on all four sides) is simulated by some tile set τ. The tile set τ consists of N^2 tiles indexed by pairs (i, j) of integers modulo N. A tile from τ has colors on its sides as shown on Fig. 1. This figure also shows the macro-tile of ρ that has colors $(0,0),\ldots,(0,N-1)$ and $(0,0),\ldots,(N-1,0)$ on its borders.

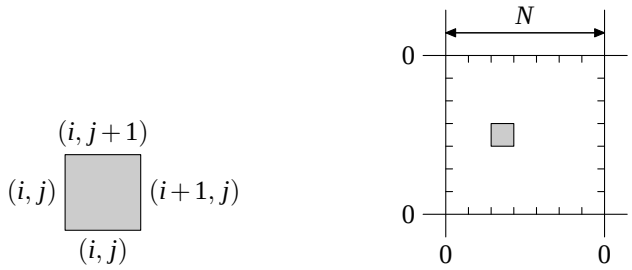

Figure 1: Tiles and macrotiles for Example 2

If a tile set τ simulates some set ρ of τ-macro-tiles with zoom factor $N > 1$ and ρ is isomorphic to τ, the set τ is called *self-similar*. Here an *isomorphism* between τ and ρ is a bijection that respects the relations "one tile can be placed on the right of another one" and "one tile can be placed on the top of another one". (An isomorphism induces two bijections: between horizontal/vertical colors of τ and horizontal/vertical macro-colors of ρ.)

The idea of self-similarity is used (more or less explicitly) in most constructions of aperiodic tile sets ([17, 4] are exceptions). However not all of these constructions provide literally self-similar tile set. We find the following explicit formulation useful.

Proposition 1. *All self-similar tile sets τ have only aperiodic tilings.*

Proof. Every τ-tiling U can be uniquely cut into $N \times N$-macro-tiles from ρ. So every period T of U is a multiple of N (since the T-shift of a cut is also a cut). Then T/N is a period of ρ-tiling, which is isomorphic to a τ-tiling, so T/N is again a multiple of N. Iterating this argument, we conclude that T is divisible by N^k for every k, so $T = 0$. □

So to prove the existence of aperiodic tile sets it is enough to construct a self-similar tile set.

Theorem 2. *There exists a self-similar tile sets τ.*

Theorem 2 was explicitly formulated and proven by N. Ollinger [24]; in his proof one particular self-similar tile set is constructed (it consists of 104 tiles). This tile set is used in [24] to embed in tilings any substitution rule (we explain this in more detail below, see Theorem 3). A very similar statement was proven in [6], though the definition of self-similarity was slightly stronger (it was required there that one tile cannot occur at different positions of macro-tiles, i.e., for each tile t there exists one single position (i, j) such that t can occur at position (i, j) of some macro-tiles). We prefer a less specific and more flexible argument, which is based on the fixed-point idea. Our proof works for a vast class of tile sets (though we cannot provide explicitly an aperiodic tile set of a reasonably small size). The rest of this section is devoted to our proof of Theorem 2. Before we prove this result, we explain some technique used in our construction: how to simulate a given tile set by embedding computations.

2.2 Simulating a tile set

For brevity we say that a tile set τ simulates a tile set ρ when τ simulates some set of macro-tiles $\tilde{\rho}$ isomorphic to ρ (e.g., we say that a self-similar tile set simulates itself).

Let us start with some informal discussion. Assume that we have a tile set ρ whose colors are k-bit strings ($C = \{0,1\}^k$) and the set of tiles $\rho \subset C^4$ is presented as a predicate $R(c_1, c_2, c_3, c_4)$ of four k-bit arguments. Assume that we have some Turing machine $\mathscr{R}$ that computes R. Let us show how to simulate ρ using some other tile set τ.

This construction extends Example 2, but simulates a tile set ρ that contains not a single tile but many tiles. We keep the coordinate system modulo N embedded into tiles of τ; these coordinates guarantee that all τ-tilings can be uniquely cut into blocks of size $N \times N$ and every tile "knows" its position in the block (as in Example 2). In addition to the coordinate system, now each tile in τ carries supplementary colors (from a finite set specified below) on its sides. These colors form a new "layer" superimposed with the old one, i.e., the set of colors is now a Cartesian product of the old one and the set of colors used in this layer.

On the border of a macro-tile (i.e., when one of the coordinates is zero) only two supplementary colors (say, 0 and 1) are allowed. So the macro-color encodes a string of N bits (where N is the size of macro-tiles). We assume that $N \geq k$ and let k bits in the middle of macro-tile sides represent colors from C. All other bits on the sides are zeros (this is a restriction on tiles: each tile "knows" its coordinates so it also knows whether non-zero supplementary colors are allowed).

Now we need additional restrictions on tiles in τ that guarantee that macro-colors on the sides of each macro-tile satisfy the relation R. To achieve this, we ensure that bits from the macro-tile sides are transferred to the central part of the tile where the checking computation of $\mathscr{R}$ is simulated (Fig. 2).

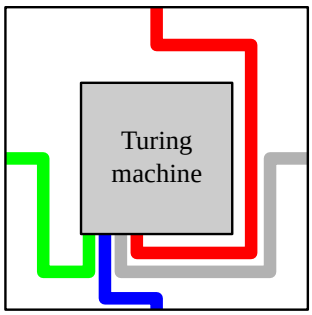

Figure 2: Wires and processing zones; wires appear quite narrow since $N \gg k$

For that we need to fix which tiles in a macro-tile form "wires" (this can be done in any reasonable way; let us assume that wires do not cross each other) and then require that each of these tiles carries equal bits on two sides (so some bit propagates along the entire wire); again it is easy to arrange since each tile knows its coordinates.

Then we check R by a local rule that guarantees that the central part of a macro-tile represents a time-space diagram of $\mathscr{R}$'s computation (the tape is horizontal, time goes up). This is done in a standard way. We require that computation terminates in an accepting state: if not, the tiling cannot be formed.

To make this construction work, the size of macro-tile (N) should be large enough: we need enough space for k bits to propagate and enough time and space (=height and width) for all accept-

ing computations of $\mathscr{R}$ to terminate.

In this construction the number of supplementary colors depends on the machine $\mathscr{R}$ (the more states it has, the more colors are needed in the computation zone). To avoid this dependency, we replace $\mathscr{R}$ by a fixed universal Turing machine $\mathscr{U}$ that runs a *program* simulating $\mathscr{R}$. Let us agree that the tape of the universal Turing machine has an additional read-only layer. Each cell carries a bit that is not changed during the computation; these bits are used as a program for the

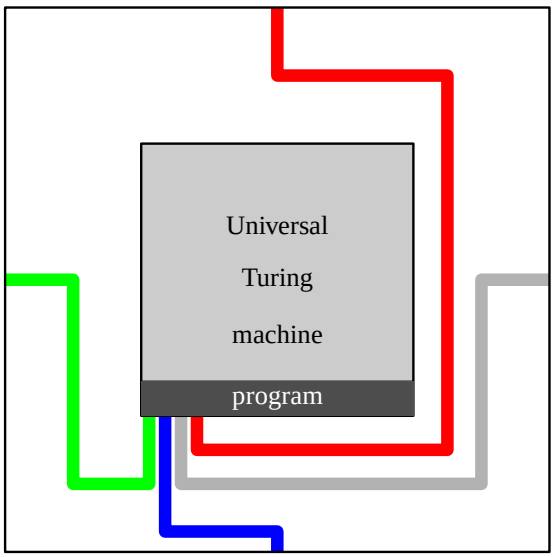

Figure 3: Checking tiles with a universal TM

universal machine U (Fig. 3). In terms of our simulation, the columns of the computation zone carry unchanged bits (considered as a program for U), and the tile set restrictions guarantee that the central zone represents the protocol of an accepting computation of U (with this program). In this way we get a tile set τ that simulates ρ with zoom factor N using $O(N^2)$ tiles. (Again we need N to be large enough, but the constant in $O(N^2)$ does not depend on N.)

2.3 Simulating itself

We know how to simulate a given tile set ρ (represented as a program for the universal TM) by another tile set τ with a large enough zoom factor N. Now we want τ to be isomorphic to ρ (then Proposition 1 guarantees aperiodicity). For this we use a construction that follows Kleene's recursion (fixed-point) theorem[1] [18].

[1] A reminder: Kleene's theorem says that for every transformation π of programs one can find a program p such that p and $\pi(p)$ produce the same output. Proof sketch: since the statement is language-independent (use translations in both directions before and after π), we may assume that the programming language has a function `GetText()` that returns the text of the program and a function `Exec(string s)` that replaces the current process by execution of a program `s`. (Think about an interpreter: surely it has an access to the program text; it can also recursively call itself with another program.) Then the fixed point is `Exec(`π`(GetText()))`.

Note that most rules of τ do not depend on the program for $\mathscr{R}$, dealing with information transfer along the wires, the vertical propagation of unchanged program bits, and the space-time diagram for the universal TM in the computation zone. Making these rules a part of ρ's definition (we let $k = 2\log N + O(1)$ and encode $O(N^2)$ colors by $2\log N + O(1)$ bits), we get a program that checks that macro-tiles behave like τ-tiles in this respect.

The only remaining part of the rules for τ is the hardwired program. We need to ensure that macro-tiles carry the same program as τ-tiles do. For that our program (for the universal TM) needs to access the bits of its own text. (This self-referential action is in fact quite legal: the program is written on the tape, and the machine can read it.) The program checks that if a macro-tile belongs to the first line of the computation zone, this macro-tile carries the correct bit of the program.

How should we choose N (hardwired in the program)? We need it to be large enough so the computation described (which deals with $O(\log N)$ bits) can fit in the computation zone. The computation is rather simple (polynomial in the input size, i.e., $O(\log N)$), so for large N it easily fits in $\Omega(N)$ available time.

This finishes the construction of a self-similar aperiodic tile set.

3 Substitution rules implemented

The construction of self-similar tiling is rather flexible and can be easily augmented to get a self-similar tiling with additional properties. Our first illustration is the simulation of substitution rules.

Let A be some finite alphabet and $m > 1$ be an integer. A *substitution rule* is a mapping $s\colon A \to A^{m\times m}$. A substitution rules defines a mapping on A-configurations. By *A-configuration* we mean an integer lattice filled with letters from A, i.e., a mapping $\mathbb{Z}^2 \to A$ considered modulo translations. A substitution rule s applied to a configuration X produces another configuration $s(X)$ where each letter $a \in A$ is replaced by an $m \times m$ matrix $s(a)$.

We say that a configuration X is *compatible* with substitution rule s if there exists an infinite sequence

$$\ldots \xrightarrow{s} X_3 \xrightarrow{s} X_2 \xrightarrow{s} X_1 \xrightarrow{s} X,$$

where X_i are some configurations. This definition was proposed in [24]. The classical definition (used, in particular, in [21]) is slightly different: configuration $X : \mathbb{Z}^2 \to A$ is said compatible with a substitution rule s if every finite part of X occurs inside of some $s^{(n)}(a)$ (for some $n \in \mathbb{N}$ and some $a \in A$). We prefer the first approach since it looks more natural in the context of tiling. However, all our results can reformulated and be proven (with minimal technical efforts) for another version of the definition.

Example 3. Let $A = \{0,1\}$,

$$s(0) = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right), \quad s(1) = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right).$$

It is easy to see that the only configuration compatible with s is the chess-board coloring.

Example 4 (Fig. 4). Let $A = \{0,1\}$,

$$s(0) = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right), \quad s(1) = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right).$$

One can check that all configurations that are compatible with this substitution rule (called *Thue – Morse configurations* in the sequel) are aperiodic. One way to prove this fact (see Proposition 3 in [24]) is to note that every configuration compatible with this substitution rule can be uniquely decomposed into disjoint 2×2 blocks $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$. Then we can apply the argument from Proposition 1 (with $N = 2$).

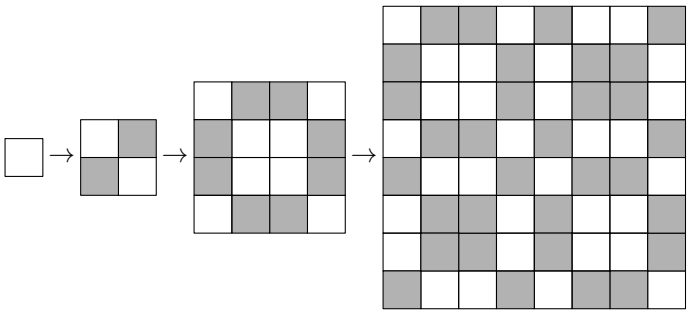

Figure 4: Three steps of Thue–Morse substitution

The following theorem goes back to Mozes [21]. It says that every substitution rule can be enforced by a tile set.

Theorem 3. *Let A be an alphabet and let s be a substitution rule over A. Then there exists a tile set τ and a mapping $e\colon \tau \to A$ such that*

(a) *s-image of any τ-tiling is an A-configuration compatible with s;*

(b) *every A-configuration compatible with s can be obtained in this way.*

A nice proof of this result for 2×2-substitutions was given in [24], where an explicit construction of a tile set τ for every substitution rule s was provided. We prove this theorem using our fixed-point argument, avoiding explicit construction of τ (the tile sets that can be extracted from our proof contain a huge number of tiles).

Proof. We modify the construction of the tile set τ (with zoom factor N) taking s into account. Let us first consider the very special case when

- the substitution rule maps each A-letter into an $N \times N$-matrix (i.e., $m = N$).
- the substitution rule is easy to compute: given a letter $u \in A$ and (i, j), we can compute the (i, j)-th letter of $s(u)$ in time $\mathrm{poly}(\log |A|) \ll N$.

In this case we proceed as follows. In our basic construction every tile knows its coordinates in the macro-tile and some additional information needed to arrange "wires" and simulate calculations of the universal TM. Now in addition to this basic structure each tile keeps two letters of A: the first is the label of a tile itself, and the second is the label of the $N \times N$-tile it belongs to. This means that we keep additional $2 \log |A|$ bits in each tile, i.e., multiply the number of tiles by $|A|^2$. It remains to explain how the local rules work. We add two requirements:

(a) the second letter is the same for neighbor tiles (unless they are separated by a border of some $N \times N$ macro-tile);

(b) the first letter in a tile is determined by the second letter and the coordinates of the tile inside the macro-tile, according to the substitution rule.

Both requirements are easy to integrate in our construction. The requirement (a) can be easily enforced; to achieve (b) a macro-tile should check that its first letter appears in $s([\text{second letter}])$ at the required position. It is possible when s is easy to compute (knowing the coordinates and the second letter, the program computes the required value of the first letter and then compares it with the actual value).

The requirements (a) and (b) ensure that configuration is an s-image of some other configuration. Also (due the self-similarity) we have the same at the level of macro-tiles. But this is not all: we need to guarantee that the first letter on the level of macro-tiles is identical to the second letter on the level of tiles. This is also achievable: the first letter of a macro-tile is encoded by bits on its border, and we can require that these bits match the second letter of the tiles at that place (recall that second letter is the same across the macro-tile). It is easy to see that now τ has the required properties (each tiling projects into a configuration compatible with τ and vice versa).

However, this construction assumes that N (the zoom factor) is equal to the matrix size in the substitution rule, which is usually not the case (m is given, and N we have to choose, and it needs to be large enough). To overcome this difficulty, we let N be equal to m^k for some k, and use the substitution rule s^k, i.e., the kth iteration of s (a configuration is compatible with s^k if and only if it is compatible with s). Now we do not need s to be easily computed: for large k the computation of s^k will fit into the space available (exponential in k). □

Remark. We use this "anthropomorphic" language (the tile "knows" something etc.), because the formal description would be too obscure (it would include writing a rather complicated program for an explicitly described universal TM). When we say that "tile knows its coordinates", we mean that this coordinates are determined by the colors of its sides (on the base level); when we say that tile additionally knows two A-letters, it means that the set of tiles is the product of the old one and $A \times A$. The condition (a) means that the second letter is also reflected in the colors of tile sides so that matching condition implies that neighbor tiles have equal labels (unless they are separated by a border). The condition (b) says which part of all combinations of letters and coordinates is allowed in tiles.

Of course, all these rules should be enforced on the next level, for macro-tiles. This means that macro-tile, in addition to the bits that represent its coordinates (and are sent to the borders according to the scheme of Fig. 1), should have some other bits that represent two A-labels. The bits of the second label should be sent to all borders having non-zero coordinates, since they should match for neighbor macro-tiles. The bits of the first label should be also present in (a known place of) the input of the computational zone, so the program can check the condition (b). Moreover, we require that if a tile is in this place, then its first internal A-label is consistent with the bit of the A-label in a macro-tile (and this again is translated into some part of the checking program) etc. Note that the first internal A-label is not reflected in macro-tile boundary colors directly since there is no need to compare these labels for neighbor macro-tiles. Our fixed-point construction can be easily adapted to such labels.

4 Thue–Morse lemma and strongly aperiodic tile sets

Let $\alpha > 0$ be a real number. A configuration $U\colon \mathbb{Z}^2 \to A$ is *α-aperiodic* if for every nonzero vector $T \in \mathbb{Z}^2$ there exists N such that in every square whose side is at least N the fraction of points x such that $U(x) \neq U(x+T)$ exceeds α.

Remark. If U is α-aperiodic, then Besicovitch distance between U and any periodic pattern is at least $\alpha/2$. (The Besicovitch distance is defined as $\limsup_N d_N$ where d_N is the fraction of points where two patterns differ in the $N \times N$ centered square. It is easy to see that it does not depend on the choice of the center point.)

Theorem 4. *There exists a tile set τ such that τ-tilings exist and every τ-tiling is α-aperiodic for every $\alpha < 1/4$.*

Proof. This is obtained by applying Theorem 3 to Thue–Morse substitution rule T (Example 4). Let C be a configuration compatible with T. We have to show that C is α-aperiodic for every $\alpha < 1/4$. (In fact we use only Thue–Morse bits here.)

Big squares in C obtained by iterating substitution rules can be represented as a `xor`-sum of two one-dimensional Thue–Morse sequences obtained using the substitution rules $0 \to 01$ and $1 \to 10$. More formally, *i*-*j*-bit of such a square is a `xor` of ith and jth bits in a_n or b_n, where we let $a_0 = 0$, $b_0 = 1$, $a_{n+1} = a_n b_n$, $b_{n+1} = b_n a_n$. (For example, $a_3 = a_2 b_2 = a_1 b_1 b_1 a_1 = 01101001$.) Evidently, $|a_i| = |b_i| = 2^i$ and b_i is the bitwise negation of a_i. To prove the required bound, we start with an estimate for (one-dimensional) aperiodicity of a_n and b_n:

Lemma 1 (folklore)**.** *For any integer $u > 0$ and for any n such that $u \leq |a_n|/4$ the shift by u steps to the right changes at least $|a_n|/4$ positions in a_n and leaves unchanged at least $|a_n|/4$ positions. (Formally, in the range $1 \ldots 2^n - u$ there is at least $(1/4)2^n$ positions i such that ith and $(i+u)$th bits in a_n coincide and at least $(1/4)2^n$ positions where these bits differ.*)

Proof of the Lemma: a_n can be represented as *abbabaab* where $a = a_{n-3}$ and $b = b_{n-3}$. One may assume without loss of generality that $u \geq |a|$ (otherwise we apply Lemma separately to the two halves of a_n). Note that *ba* appears in the sequence twice and once it is preceded by *a* and once by *b*. Since *a* and *b* are opposite, the shifted bits match in one of the cases and do not math in the other one. The same is true for *ab* that appears preceded both by *a* and *b*. □

Now let α be one-dimensional Thue–Morse infinite sequence; our two-dimensional configuration C is defined by $C_{ij} = \alpha_i \oplus \alpha_j$. Let T be any shift. If T is horizontal, then α_j is unchanged and the Lemma is enough (the lemma is stated for the intervals of some special form, but for large enough squares the boundary effects are compensated by the difference between $1/4$ and α). The same argument works for vertical shifts. If both coordinates of a shift are non-zero integers, the the fraction is questions is the probability of an event that is an xor-combination of two events with probabilities in $(1/4, 3/4)$. It is easy to check that such an event also has probability in $(1/4, 3/4)$ (in fact, in $(3/8, 5/8)$, but this stronger bound is not needed).

Theorem 4 is proved. □

In fact, the bound $1/4$ can be replaced by $1/3$ if we use more professional analysis of Thue–Morse sequence (see, e.g., [29]). But if we want to get a most strong result of this form and make

the bound close to 1, this substitution rule does not work. We can use some other rule (in a bigger alphabet) as Pritykin and Ulyashkina have shown [25], but we prefer to give another construction with variable zoom factors, see Section 6.

5 Variable zoom factor

The fixed point construction of aperiodic tile set is flexible enough and can be used in other contexts. For example, the "zoom factor" N could depend (recursively) on the level k (number of grouping steps). For this each macro-tile should have k encoded at its sides; this labeling should be consistent when switching to the next level. Using the anthropomorphic terminology, we say that each macro-tile "knows" its level, i.e., the sequence of bits that form a binary representation of this level, is transferred from the sides to the tape and the computation checks that all these numbers (level bits for all four sides) are the same. This is, so to say, a "conscious" information processed by a computation in the computation region of the macro-tile. One may say also that a macro-tile of any level contains "subconscious" information ("existing in mind but not immediately available to consciousness", as the dictionary says): this is the information that is conscious for the sub-tiles that form a macro-tile, and their sub-tiles (all the way down to the ground level).

Using this terminology, we can say that each macro-tile knows its coordinates in the macro-tile of the next level: for a tile of level k these coordinates are integers modulo N_{k+1}, so in total $\log k + O(\log N_{k+1})$ bits are required for keeping both the level and these coordinates. Note that N_k steps should be enough to perform increment operation modulo N_{k+1}; we assume that both $\log k$ and $\log N_{k+1}$ are much less than N_k. This means that N_k should not increase too fast or too slow (say, $N_k = \log k$ is too slow and $N_{k+1} = 2^{N_k}$ is too fast). Also we need to compute N_{k+1} when k is known, so we assume that not only the size of N_{k+1} (i.e., $\log N_{k+1}$) but also the time needed to compute this given k are small compared to N_k. These restrictions still allow many possibilities, say, $N_k = \sqrt{k}$, $N_k = k$, $N_k = 2^k$, $N_k = 2^{(2^k)}$, $N_k = k!$ etc.

There is one more important point that needs to be covered. How do we guarantee that the bits representing the level k (on the tape of a macro-tile) are correct? In other terms, we need to ensure that the levels known to a macro-tile and to one of its tiles differ by one. (In psychoanalytic terms we need to check that conscious and subconscious information in a tile match each other.) This is done as follows. The tile knows its level and also knows its position in the macro-tile it belongs (its father). So it knows whether it is in the place where father should keep level bits, and can check whether indeed the level bit that father keeps in this place is consistent with the level information the tile has. (In fact we have the same problem when simulating substitution rule: a check that the father letter of a tile coincides with the letter of the father tile, is done in the same way.)

This "self-similar" structure with variable zoom factor can be useful in some cases. Though it is not a self-similar according to our definition, one can still easily prove that any tiling is aperiodic. Note that now the computation time for the TM simulated in the central part increases with level, and this can be used for a simple proof of undecidability of domino problem. The problem in the standard proof (based on the self-similar construction with fixed zoom factor) is that we need to place computations of unbounded size into this self-similar structure, and for that we need special geometric tricks (see [2, 1]). With our new construction, if we want to reduce an instance of the

halting problem (some machine M) to the domino problem, we add to the program embedded in our construction the parallel computation of M on the empty tape; if it terminates, this destroys the tiling.

In a similar way we can show that the existence of a periodic tiling is an undecidable property of a tile set, and, moreover, the tile sets that admit periodic tilings and tile sets that have no tilings form two inseparable sets (another classical result, see [14]).

Here is an example of a more exotic version of the latter result (that has probably no interest in itself, just an illustration of the technique). We say that a tile set τ is *m-periodic* if τ-tilings exist and for each of them the set of periods is the set of *all* multiples of m (this is equivalent to the fact that both vectors $(0,m)$ and $(m,0)$ are periods). Let E [resp. O] be all m-periodic tile sets for all even m [resp. odd m].

Theorem 5. *The sets E and O are inseparable enumerable sets.*

Proof. It is easy to see that the property "to be an m-periodic tile set" is enumerable (both the existence of tiling and enforcing periods $(m,0)$ and $(0,m)$ are enumerable properties).

It remains to reduce some standard pair of inseparable sets (say, machines that terminate with output 0 and 1) to (E,O). It is easy to achieve using the technique explained. Assume that the numbers N_k increase being odd integers as long as the computation of a given machine does not terminate. When and if it terminates with output 0 [1], we require periodicity with odd [resp. even] period at the next level. □

Another application of a variable zoom factor is the proof of the following result obtained by Lafitte and Weiss (see [19]) using Turing machine simulation inside Berger–Robinson construction.

Theorem 6. *Let f be a total computable function whose arguments and values are tile sets. Then there exists a tile set τ that implements a tile set $f(\tau)$.*

Here we assume that some computable encoding for tile sets is fixed. Since there are no restrictions on the computation complexity of f, the choice of the encoding is not important.

Proof. Note that for identity function f this result provides a self-simulating tile set of Section 2.3. To prove it we may use the same kind of a fixed-point technique. However, there is a problem: the computation resources inside a tile are limited (by its size) while time needed to compute f can be large (and, moreover, depends on the tile size).

The solution is to postpone the simulation to large levels: if a tile set τ_0 simulates τ_1 that simulates τ_2 that simulates etc. up to τ_n, then τ_0 simulates τ_n, too. Therefore we may proceed as follows.

We use the construction explained above with a variable zoom factor. Additionally, at each level the computation starts with a preliminary step that may occupy up to (say) half of the available time. This step involves:

- interpreting a program that it is on the tape and unfolding a tile set that is implemented by this program on the ground level; this set should be then converted into a form used by f;
- applying f to this tile set;

- converting the output of f into a list of tiles written down in some straightforward encoding.

This part of the computation checks also that it does not use more that half of the available time and that the output is small enough compared to the tile size. If this time turns out to be insufficient or the output is too big, this part is dropped and we start a normal computation for variable zoom factor (as explained above). However, if the time is enough and result (list of tiles that corresponds to f's output) is small compared to the tile size, we check that macro-tile (of the current level) belongs to the tile set computed.

Since the program is the same at all level and the computation of f should be finite (though may be very long), at some (big enough) level the second possibility starts to play, and we get a tile set isomorphic to $f(\tau)$ where τ is the tile set on the ground level. □

Another application is the construction of tile sets with any given computable density. Assume that a tile set is given and, moreover, all tiles are divided into two classes, say, A-tiles and B-tiles. We are interested in a fraction of A-tiles in a tiling of an entire plane or its large region. If the tile set is flexible enough, this fraction can vary. However, for some tile sets this ratio tends to a limit value when the size of a tiled region increases. This phenomenon is captured in the following definition: we say that tile set τ divided into A- and B-tiles *has a limit density* α if for every $\varepsilon > 0$ there exists N such that for any $n > N$ the fraction of A-tiles in any tiling of the $n \times n$ square is between $\alpha - \varepsilon$ and $\alpha + \varepsilon$.

Theorem 7. (i) *If a tile set has a density α, then α is a computable real number in* $[0,1]$. **(ii)** *Any computable real number $\alpha \in [0,1]$ is a density of some tile set.*

Proof. The first part is a direct corollary of the definitions. For each n we can consider all tilings of the $n \times n$ square and look for the minimal and maximal fractions of A-tiles in them. Let us denote them by m_n and M_n. It is easy to see that the limit frequency (if exists) is in the interval $[m_n, M_n]$. Indeed, in a large square split into squares of size $n \times n$ the fraction of A-tiles is between m_n and M_n being at the same time arbitrarily close to α. Therefore, α is computable (to get its value with ε-precision, we increase n until the difference between M_n and m_n becomes smaller than ε).

It remains to prove (ii). Since α is computable, there exist two computable sequences of rational numbers l_i and r_i that converge to α in such a way that

$$[l_1, r_1] \supset [l_2, r_2] \supset [l_3, r_3] \supset \ldots$$

Our goal will be achieved if macro-tiles of the first level have density either l_1 or r_1, macro-macro-tiles have density either l_2 or r_2, and so on. Indeed, each large square can be split into macro-tiles (and the border that does not change the density much), so in any large square the fraction of A-tiles is (almost) in $[l_1, r_1]$. The same argument works for macro-macro-tiles, etc.

However, this plan cannot be implemented directly: the main difficulty is that the computation of l_i and r_i may require a lot of time while the computational abilities of macro-tiles of level i are limited (we use variable zoom factors, but they cannot grow too fast).

The solution is to postpone the switch from densities l_i and r_i to densities l_{i+1} and r_{i+1} to the higher level of the hierarchy where the computation has enough time to compute all these four

rational numbers and find out in which proportion l_i- and r_i-tiles should be mixed in l_{i+1}- and r_{i+1}-tiles. (This proportion is restricted by our construction: the denominator should be the number of i-level macro-tiles in $(i+1)$-level macro-tile, but this restriction can be always satisfied by a slight change in l_i and r_i which leaves α unchanged.) So, we allocate, say, the first half of the available time for controlled computation of all these values; if the computation does not finish in time, the densities for the next level are the same as for the current level. If the computation terminates in time, we use the result of the computation to have two types of the next level tiles: one with density l_{i+1} and one with density r_{i+1}. They are made by using prescribed amount of l_i- and r_i-tiles (since each tile knows its coordinates, it can find out whether it should be of the first or second type). This finishes the construction. □

6 Strongly aperiodic tile sets revisited

In Section 4 we constructed a tile set such that every tiling is α-aperiodic (for every $\alpha < 1/4$). Now we want to improve this result and construct a tile set such that every tiling is, say, 0.99-aperiodic (here 0.99 can be replaced by any constant less than 1). It is easy to see that this cannot be achieved by the same argument, with Thue–Morse substitutions, as well as with any substitutions in a two-letter alphabet; we need a large alphabet to make the constant close to 1.

May be it is possible to achieve this result with some other substitution rule just applying Theorem 3, but we do not know how to construct a substitution rule that gives 0.99-aperiodic configurations. Instead, we will modify the construction and use substitution rules with variable zoom factor (and different substitutions on each level).

Instead of one alphabet, A, we now consider a sequence of finite alphabets, A_0, A_1, A_2,…; the cardinality of A_i will grow as i grows. Then we consider a sequence of mappings:

$$s_1 \colon A_1 \to A_0^{N_0 \times N_0}, \quad s_2 \colon A_2 \to A_1^{N_1 \times N_1}, \quad s_3 \colon A_3 \to A_2^{N_2 \times N_2}, \ldots$$

where N_0, N_1, N_2,… are some positive integers (zoom factors); they will also increase as i increases.

Then we can compose this mappings. For example, a letter z in A_2 can be first replaced by a $N_1 \times N_1$ square $s_2(z)$ filled by A_1-letters. Then each of these letters can be replaced by a $N_0 \times N_0$-square filled by A_0-letters according to s_1 and we get a $N_0N_1 \times N_0N_1$-square filled by A_0-letters; we denote this square by $s_1(s_2(z))$ (slightly abusing the notation).

All this (the sequence of A_i, N_i, s_i) is called a *substitution family*. Such a family defines a class of A_0-configurations compatible with it (in the same way as in Section 3). Our plan is to construct a substitution family such that:

- every configuration compatible with this family is 0.99-aperiodic;
- there exists a tile set and projection of it to A_0 such that only compatible configurations (and all compatible configurations) are projections of tilings.

In other words, we use the same argument as before (proving Theorem 4) but use a substitution family instead of one substitution rule. This substitution family will have special properties:

A. Symbols used in different locations are different. This means that A_i-letters that appear in a given position of the squares $s_{i+1}(z)$ for $z \in A_{i+1}$, never appear in any other places of these squares; the sets A_i is split into $N_i \times N_i$ disjoint subset used for different positions in $N_i \times N_i$ square.

B. Different letters are mapped to squares that are far away in Hamming distance. This means that if $z, w \in A_{i+1}$ are different, the images $s_{i+1}(z)$ and $s_{i+1}(w)$ are far away in the Hamming distance: the fraction of positions in $N_i \times N_i$ squares where $s_{i+_1}(z)$ and $s_{i+1}(w)$ have equal equal letters, does not exceed ε_i.

Here ε_i is a sequence of positive reals such that $\sum_{i\geq 0} \varepsilon_i < 0.01$.

This implies that composite images of different letters are also far apart. For example, the fraction of positions in $N_0N_1 \times N_0N_1$ square where $s_1(s(2(z))$ and $s_1(s_2(w))$ coincide does not exceed $\varepsilon_0 + \varepsilon_1 < 0.01$. (Indeed, in $s_2(z)$ and $s_2(w)$ we have at most ε_1-fraction of matching letters; these letters generate ε_1-fraction of matching A_0-letters on the ground level; all other, non-matching, pairs add ε_0-fraction. In fact, we get a stronger bound $1-(1-\varepsilon_0)(1-\varepsilon_1)$.)

In the same way, if we take two different letters in A_i and then go down to the ground level and obtain two squares of size $N_0N_1\ldots N_{i-1} \times N_0N_1\ldots N_{i-1}$ filled by A_0-letters, the fraction of coincidences is at most $\varepsilon_0 + \ldots + \varepsilon_{i-1} < 0.1$.

This property of the substitution family implies the desired property:

Lemma 2. *Any A_0-configuration U compatible with such a tiling family is 0.99-aperiodic.*

Proof. Consider a shift vector T. If T is not a multiple of N_0 (one of the coordinates is not a multiple of N_0), then property A guarantees that original configuration and its T-shift differ everywhere. Now assume that T is a multiple of N_0. Then T induces a T/N_0-shift of an A_1-configuration U_1 that is a s_1-pre-image of U. If T is not a multiple of N_0N_1, then T/N_0 is not a multiple of N_1 and for the same reason this T/N_0-shift changes all the letters in U_1. And different letter in A_1 are mapped to $N_0 \times N_0$ squares that coincide in (at most) ε_0-fraction of positions.

If T is a multiple of N_0N_1 but not $N_0N_1N_2$, we get a $T/(N_0N_1)$ shift of A_2-configuration U_2 that changes all its letters, and different letters give squares that are $1-(\varepsilon_0+\varepsilon_1)$ apart. The same argument works for the higher levels. □

Now we have to construct a substitution family that has properties A and B and can be enforced by a tile set. The requirement of large Hamming distance is standard for coding theory, and the classical tool is the Reed–Solomon code.

First, let A_i be equal to $B_i \times \{0,1,\ldots,N_i\} \times \{0,1,\ldots,N_i\}$; let us agree that we use letters $\langle b,i,j\rangle$ only in (i,j)-position of the square. This ensures the requirement A.

Then we construct a code that encodes each A_{i+1}-letter w by a a string of length N_i^2 made of B_i-letters (arranged in a square); adding the coordinates, we get s_{i+1}-image of w. We use a sequence of codes:

$$
\begin{aligned}
s_1 &: A_1 = B_1 \times N_1 \times N_1 \to B_0^{N_0\times N_0}, \quad \varepsilon_0 \text{ coincidences between } s_1(a_i),\ s_1(a_j)\ (i\neq j)\\
s_1 &: A_2 = B_2 \times N_2 \times N_2 \to B_1^{N_1\times N_1}, \quad \varepsilon_1 \text{ coincidences between } s_2(a_i),\ s_2(a_j)\ (i\neq j)\\
&\ldots
\end{aligned}
$$

To satisfy requirement B, we need to have a code with distance $(1-\varepsilon_i)N_i^2$. The standard construction uses polynomials of small degree over some finite field. The size of the field should be (at least) the length of the codeword, i.e., N_i^2. Let us decide that N_i is a power of 2 and the size of the field is exactly N_i^2. (We can use also $\mathbb{Z}/p\mathbb{Z}$ for prime p of an appropriate size.) To achieve the required code distance, we have to use polynomials of degree less than $\varepsilon_i N_i^2$. Using (for simplicity) only coefficients 0 and 1, we get $2^{\varepsilon_i N_i^2}$ polynomials of this type, it is enough if

$$|A_{i+1}| \leq 2^{\varepsilon_i N_i^2}.$$

Recalling that $A_{i+1} = B_{i+1} \times N_{i+1} \times N_{i+1}$ and that we agreed that B_{i+1} is a field of size N_{i+1}^2, we get the inequality

$$N_{i+1}^4 \leq 2^{\varepsilon_i N_i^2}, \quad \text{or} \quad 4\log N_{i+1} \leq \varepsilon_i N_i^2.$$

Now let $N_i = 2^{i+c}$ for some constant c; we see that for large enough c this inequality is satisfied for ε_i with sum less than 0.01 (or any other constant), since the left-hand side is linear in i while the right-hand side is exponential.

Now it remains to implement all this using tiling rules. As we have discussed, the zoom factor $N_i = 2^{i+c}$ is OK for the construction. This factor leaves enough space to keep two substitution letters (for the tile itself and its father tile), since these letters require linear size (in i). Moreover, we have enough time (exponential time) to perform the computations in the finite fields needed to construct the error correction code mappings, so the construction used to prove Theorem 3 still works.

Remark. We can also get an 0.99-aperiodic tile set as a corollary of the result of next section; indeed, we construct there a tile set such that any tiling embeds a horizontal sequence with high complexity substrings, and such a sequence cannot match itself well after a shift (in fact, we need to replace a binary alphabet by a larger finite alphabet in this argument). Then we can superimpose this with a 90°-rotated construction; then any non-zero translation will shift either vertical or horizontal sequence and therefore change most of the positions. Note that in this way we can also get a tile set that is 0.99-far from every periodic pattern (a slightly different approach to define strong aperiodicity).

However, we prefer to present a more explicit (and simpler) construction in this section that does not refer to (rather complicated) arguments in Section 7.

7 Tile set that has only complex tilings

In this section we provide a new proof of the following result from [5]:

Theorem 8. *There exists a tile set τ and constants $c_1 > 0$ and c_2 such that τ-tilings exist and in every τ-tiling T every $N \times N$-square has Kolmogorov complexity at least $c_1 N - c_2$.*

We refer to [5] for the discussion of this result (why it is optimal, why the exact value of c_1 does not matter etc.) and other related results.

7.1 A biinfinite bit sequence

Proof. We start the proof in the same way as in [5]: we assume that each tile keeps a bit that propagates (unchanged) in the vertical direction. Then any tiling contains a biinfinite sequence of bits ω_i (where $i \in \mathbb{Z}$). Any $N \times N$ square contains a N-bit substring of this string, so if (for large enough N) every N-bit substring of ω has complexity at least $c_1 N$ for some fixed c_1, we are done.

We say that a sequence ω has *Levin's property* if every N-bit substring x of ω has complexity $\Omega(N)$. Such a biinfinite sequence indeed exists (see [5]; another proof can be obtained by using Lovasz local lemma, see [27]). So our goal is to formulate tilings rules in such a way that a correct tiling "ensures" that the biinfinite sequence embedded in it indeed has this property.

The set of all "forbidden" binary strings, i.e., strings x such that $K(x) < c_1|x| - c_2$ (here $K(x)$ stands for Kolmogorov complexity of x and $|x|$ stands for the length of x) is enumerable: there is a program that generates all forbidden substrings. It would be nice to embed into the tiling a computation that runs this program and compares its output strings with the substrings of ω; such a computation may blow up (create a tiling error) if a forbidden substring is found.

However, this is not easy. There are several difficulties.

- First of all, our self-similar tiling contains only finite computations; the duration depends on the zoom factor and may increase as the level increases (bigger macro-tiles keep longer computations), but at any level the computations are finite.
- The computation at some level deals with bits encoded in the cells of that level, i.e., with macro-tile states. So the computation cannot achieve the bits of the sequence (that are "deep in the subconscious") directly and some mechanism to dig them out is needed.

Let us explain how to overcome these difficulties.

7.2 Bits delegation

Macro-tile of level k is a square whose side is $L_k = N_0 \cdot N_1 \cdot \ldots \cdot N_{k-1}$, so there are L_k bits of the sequence that intersect this macro-tile. Let us delegate each of these bits to one of the macro-tiles it intersects. Note that macro-tile of the next level is made of $N_k \times N_k$ macro-tiles of level k. We assume that N_k is much bigger than L_k (more about choice of N_k later); this guarantees that there is enough macro-tiles of level k (in the next level macro-tile) to serve all bits that intersect them. Let us decide that ith macro-tile of level k (from bottom to top) in a $(k+1)$-level macro-tile serves (consciously knows, so to say) ith bit (from the left) in its zone. (In this way we have several macro-tiles of level k in each macro-tile of level $k+1$ that are "responsible" for the same bit, but this does not create any problems.)

So each bit (each vertical line) has a representative on every level — a macro-tile that consciously knows this bit. However, we need some mechanisms that guarantee that this information is indeed true (consistent on different levels). On the bottom level it is easy, since the bits are available on the same level.

To guarantee the consistency we use the same trick as in Section 3: at each level we keep the information not only for this level but also for its father and made necessary consistency checks. Namely, each macro-tile knows (has on its computation tape):

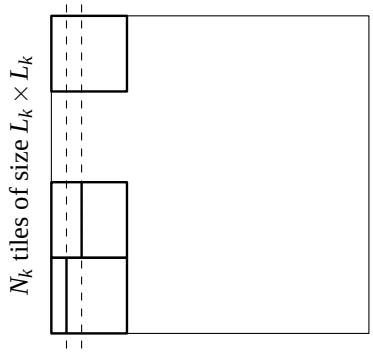

Figure 5: Bit delegation

- the bit delegated to this macro-tile;
- the coordinates of this macro-tile in its father macro-tile (that are already used in the fixed-point construction); the y-coordinate determines the position of the bit delegated to this macro-tile (relative to the left boundary of the macro-tile).
- the bit delegated to the father of this macro-tile;
- the coordinates of the father macro-tile in the grandfather macro-tile

This information is subject to consistency checks:

- the information about the father macro-tile should coincide with the same information in neighbor tiles (unless they have a different father, i.e., one of the coordinates is zero).
- if it happens that the bit delegated to the father macro-tile is the same bit as delegated for the tile, these bits should match;
- it can happen that the macro-tile occupies a place in its father macro-tile where some bits of its coordinates (inside grandfather macro-tile) or the bit delegated to the father are kept; then this partial information on the father level should be checked against the information about father coordinates and bit.

These tests guarantee that the information about father is the same in all brothers, and some of these brothers (that are located on the father tape) can check it against actual father information; at the same time some other brother (that has the same delegated bit as the father) checks the consistency of the delegated bits information.

Note that this scheme requires that not only $\log N_k$ but also $\log N_{k+1}$ is much less than N_{k-1}. This requirement, together with the inequality $L_k = N_0 \cdot N_1 \cdot \ldots N_{k-1} \leq N_k$ (discussed earlier) is satisfied if $N_k = Q^{c^k}$ where Q is a large enough constant (this is needed also to make macro-tiles of the first level large enough) and $c > 2$ (so $1 + c + c^2 + \ldots + c^{k-1} < c^k$).

Later, in Section 13, the choice of c has to be reconsidered: we need $c < 3$ to achieve error correction.

7.3 Bit blocks checked

We explained how macro-tile of any level can have a true information about one bit (delegated to it). However, we need to check not bits, but substrings (and create a tiling error if a forbidden string appears). Note that it is OK to test only very short substrings compared to the macro-tile size (N_k): if this test is done on all levels, this short substring becomes long enough to detect any violation. (Also note the short forbidden substrings can appear very late in the generation process, so we need computation in arbitrary high levels for this reason, too.)

So we need to provide more information to tiles. It can be done in the following way. Let us assume that a tile contains not one bit but a group of bits that starts at the delegated bit and has length depending on the level k (and growing very slowly with k, say, $\log\log\log k$ is slow enough). If this group goes out of the region occupied by a tile, we truncate it.

Similarly, a macro-tile should have this information for the father macro-tile (even if the bits are outside its own region), this information should be the same for brothers and needs to be checked against the delegated bits on the macro-tile level and pieces of information on the father level.

Then the computation in the computation zone can start the generating process and checking the forbidden strings that appear against all the substrings of the group of bits available to this computation. This process is time- and space-bounded, but this does not matter since every string is considered on a high enough level.

7.4 Last correction

The argument explained above still needs some correction. We claim that every forbidden string will be detected at some level where it is short enough compared to the level parameters. However, there could be strings that never become a part of one macro-tile. Imagine that there is some vertical line that is a boundary between macro-tiles of all levels (so we have bigger and bigger tiles on both sides, and this line is still the boundary between them). Then a substring that crosses this line will be never checked and therefore we cannot guarantee that it is not forbidden.

There are several ways to get around this problem. One can decide that each macro-tile contains information not only about blocks inside its father macro-tile but in a wider regions (say, three times wider including uncle macro-tiles); this information should be checked for consistency between cousins, too.

But there is a simpler solution. Note that even if a string on the boundary is never checked, its parts (on both sides of the boundary) are, so their complexity is proportional to their length. And one of the parts has length at least half of the original length, so we still have a complexity bound, just the constant is twice smaller.

This finishes the proof of Theorem 8. □

8 Subshifts

The analysis of the proof in the previous section shows that it can be divided into two parts. We defined *forbidden* strings as bit strings that are sufficiently long and have complexity at most $\alpha \cdot$

(length). We started by showing that biinfinite strings without forbidden factors (substrings) exist. Then we constructed a tile set that contains such a biinfinite string in any tiling.

The second part can be separated from the first one, and in this way we get new proofs for some results of S. Simpson [28] and M. Hochman [16] about effectively closed subshifts.

Fix some alphabet A. Let F be a set of A-strings. Consider a set S_F of all biinfinite A-sequences that have no factors (substrings) in F. This is a *closed* 1*-dimensional subshift* over A, i.e., a closed shift-invariant subset of the space of all biinfinite A-sequences. If the set F is (computably) enumerable, S_F is called an *effectively closed* 1*-dimensional subshift* over A. If F is finite, S_F is called a *subshift of finite type*.

In one dimension (non-empty) subshifts of finite type always contain periodic sequences. Berger's theorem says that for two-dimensional subshifts it is not the case. More precisely, let F be a set of two-dimensional patterns (squares filled with A-letters). Then we can consider a set S_F of all A-configurations (= mappings $\mathbb{Z}^2 \to A$) that do not contain any pattern from F. This is a closed shift-invariant set of A-configurations (= 2-dimensional closed subshift over A). If F is (computably) enumerable, S_F is called *a* 2*-dimensional effectively closed subshift* over A. If F is finite, S_F is called *a* 2*-dimensional subshift of finite type*.

Let X and Y be two alphabets and let $r\colon X \to Y$ be a mapping. Then every X-configuration can be mapped to a Y-configuration by applying r to every letter. It is easy to see that an image of a closed subshift is a closed subshift (compactness argument). An effective version of this compactness argument shows that an image of an effectively closed subshift is again an effectively closed subshift.

The following theorem shows that for 2-dimensional subshifts of finite type it is not the case (an image of a finite type subshift is not necessarily of finite type).

Theorem 9. *Let A be some alphabet and let S be a* 1*-dimensional effectively closed subshift over A. Then there exists an alphabet B, a mapping $r\colon B \to A$, and a* 2*-dimensional subshift S' of finite type over B such that r-images of configurations in S' are* (*exactly*) *elements of S extended vertically* (*vertically aligned cells contain the same A-letter*).

Proof. The proof uses the same argument as in Theorem 8. Each cell now contains an A-letter that propagates vertically. Computational zones in macro-tiles generate (in available space and time) elements of the enumerable set of forbidden A-substrings and compare them with A-substrings that are made available to them. It remains to note that tiling requirements (matching colors) are local, i.e., they define a finite type 2-dimensional subshift.

Note that now the remark of Section 7.4 becomes crucial, since otherwise the image of S'-configuration may be a concatenation of two sequences (a left-infinite one and a right-infinite one); each sequence does not contain forbidden patterns but they may appear near the meeting point. (This makes a fixed-point construction essential in the proof: the argument from [5] does not work here.) □

A similar argument shows that every 2-dimensional effectively closed subshift can be represented as an image of a 3-dimensional subshift of finite type (after a natural extension along the third dimension), any 3-dimensional effectively closed subshift is an image of a 4-dimensional subshift of finite type, etc.

This result is an improvement of a similar one proved by M. Hochman (Theorem 1.4 in [16], where the dimension increases by 2), thus providing a solution of Problem 9.1 in this paper. Note also that it implies the result of S. Simpson [28] where 1-dimensional sequences are embedded into 2-dimensional tilings but in some weaker sense (defined in terms of Medvedev degrees).

One can ask whether a dimension reduction is essential here. For example, is it true that every 2-dimensional effectively closed subshift is an image of some 2-dimensional subshift of finite type? The answer for this question (and related questions in higher dimensions) is negative. This follows from an upper bound in [5] saying that every tile set has a tiling where $n \times n$ squares have complexity $O(n)$ (this result immediately translates for subshifts of finite type) and a result from [27] that shows that some non-empty effectively closed 2-dimensional subshift has $n \times n$ squares of complexity $\Omega(n^2)$. Therefore the latter cannot be an image of the first one (complexity can only decrease when we apply an alphabet mapping).

9 Random errors

9.1 Motivation and discussion

The result of Section 10 states that an isolated hole in a tiling can be patched, if the tile set is constructed in a special way. Moreover, it implies that many holes of bounded size can be patched simultaneously if the distance between the holes is large enough compared to their size (since the corrected neighborhoods of holes are disjoint). However, this is a rather special case of holes set, and we are interested in more general results: we would like to prove that for a "robust" tile set any tiling with "sparse enough" errors or holes can be patched (by changing a small fraction of tiles).

Note that it does not matter much whether we speak about errors (places where two neighbor tiles do not match) or holes (places without tiles). Indeed, we can convert a tiling error into a hole (by deleting one of the two non-matching tiles) and convert a hole into a small number(at most 4) errors by placing an arbitrary tile there. (Holes look more naturally if we start with a set of holes and then try to tile the rest; on the other hand, if we imagine some process similar to crystallization when a tiling tries to become correct by some trial-and-error procedure, it is more natural to consider tiling errors. Since it does not make serious difference from the mathematical point of view, we use both metaphors.)

We use a hierarchical approach to hole patching that goes back to P. Gacs who used it in a much more complicated situation [11]. This means that first we try to patch small holes that are not too close to each other (by changing small neighborhoods around them). This (if we are lucky enough) makes larger (and still unpatched) holes more isolated since there are less small holes around. Some of these larger holes (that are not too large and not too close to each other) can be patched again. Then the same procedure can be repeated again for the next level. Of course, we need some conditions (that guarantee that holes are not too dense) to make this procedure successful. These conditions are described later in full details, but the important question is: How do we ensure that these conditions are reasonable (i.e., general enough)? Our answer is: we prove that if holes are generated at random (each position becomes a hole independently of other positions with small enough probability ε) then the generated set satisfies these conditions with probability 1.

From the physics viewpoint, this argument sounds rather weak: if we imagine some crystallization process, errors in different positions are not independent at all. However, this approach could be a first approximation until a more adequate one is found.

Note that patching holes in a tiling could be considered as a generalization of the percolation theory. Indeed, let us consider a simple tile set made of two tiles: one has all black sides and the other has all white sides. Then the tiling conditions reduce to the following simple condition: each connected component of the complement to the holes set is either completely black or completely white. We want to make small corrections in the tiling that patch the holes (and therefore make the entire plane black or white). This means that initially either we have small black "islands" in a white ocean or vice versa, which is exactly what percolation theory says (it guarantees that if holes are generated at random independently with small probability, the rest consists of one large connected component and many small islands.)

This example shows also that simple conditions like small density (in Besicovitch sense) of the holes set are not enough: a regular grid of thin lines can have small density but still splits the plane into non-connected squares; if half of these squares are black and the others are white, no small correction can patch the holes.

One can define an appropriate notion of a sparse set in the framework of algorithmic randomness (Martin-Löf definition of randomness) considering individual random sets (with respect to Bernoulli distribution B_ε) and their subsets as "sparse". Then we can prove that any "sparse" set satisfies the conditions that are needed to make the iterative patching procedure work. This algorithmic notion of "sparseness" is discussed in [3]. However, in the current paper we do not assume that reader is familiar with algorithmic randomness and restrict ourselves to the classical probability theory.

So our statements become quite lengthy and use probabilistic quantifiers "for almost all" (=with probability 1). The order of quantifiers (existential, universal and probabilistic) is important here. For example, the statement "a tile set τ is robust" means that *there exists* some $\varepsilon > 0$ such that *for almost all* H (with probability 1 with respect to the distribution where each point independently belongs to H with probability ε) the following is true: *for every* (τ, H)-tiling U *there exists* a τ-tiling U' (of the entire plane) that is "close" to U. Here by (τ, H)-tiling we mean a tiling of $\mathbb{Z}^2 \setminus H$ (where existing pairs of neighbor tiles match).

9.2 Islands of errors

In this section we develop the notion of sparsity based on the iterative grouping of errors (or holes) and prove its properties.

Let $E \subset \mathbb{Z}^2$ be a set of points; points in E are called *dirty*; other points are *clean*. Let $\beta \geq \alpha > 0$ be integers. A non-empty set $X \subset E$ is an (α, β)-*island* in E if:

(1) the diameter of X does not exceed α;

(2) in the β-neighborhood of X there is no other point from E.

(Diameter of a set is a maximal distance between its elements; the distance d is defined as l_1, i.e., the maximum of distances along both coordinates; β-neighborhood of X is a set of all points y such that $d(y,x) \leq \beta$ for some $x \in X$.)

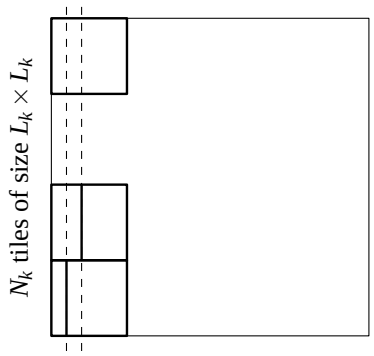

Figure 6: Explanation tree; vertical lines connect different names for the same points.

It is easy to see that two (different) islands are disjoint (and the distance between their points is greater than β).

Let (α_1,β_1), (α_2,β_2),... be a sequence of pairs of integers and $\alpha_i \le \beta_i$ for all i. Consider the following iterative "cleaning" procedure. At the first step we find all (α_1,β_1)-islands (*rank* 1 *islands*) and remove all their elements from E (thus getting a smaller set E_1). Then we find all (α_2,β_2)-islands in E_1 (*rank* 2 *islands*); removing them, we get $E_2 \subset E_1$, etc. Cleaning process is *successful* if every dirty point is removed at some stage.

At the ith step we also keep track of the β_i-neighborhoods of islands deleted during this step. A point $x \in \mathbb{Z}^2$ is *affected* during a step i if x belongs to one of these neighborhoods.

The set E is called *sparse* (for a given sequence α_i,β_i) if the cleaning process is successful, and, moreover, every point $x \in \mathbb{Z}^2$ is affected at finitely many steps only (i.e., x is far from islands of sufficiently large ranks).

The values of α_i and β_i should be chosen in such a way that for sufficiently small $\varepsilon > 0$ a B_ε-random set is sparse with probability 1. (As we have said, this justifies that our notion of sparsity is not unreasonably restrictive.) The sufficient conditions are provided by the following statement:

Lemma 3. *Assume that*

$$8\sum_{k<n}\beta_k < \alpha_n \le \beta_n \quad \textit{for every } n \textit{ and} \quad \sum_i \frac{\log \beta_i}{2^i} < \infty.$$

Then for all sufficiently small $\varepsilon > 0$ a B_ε-random set is sparse with probability 1.

Proof of Lemma 3. Let us estimate the probability of the event "x is not cleaned after n steps" for a given point x (this probability does not depend on x). If $x \in E_n$, then x belongs to E_{n-1} and is not cleaned during the nth step (when (α_n,β_n)-islands in E_{n-1} are removed). Then $x \in E_{n-1}$ and, moreover, there exists some other point $x_1 \in E_{n-1}$ such that $d(x,x_1)$ is greater than $\alpha_n/2$ but not greater than $\beta_n + \alpha_n/2 < 2\beta_n$. Indeed, if there were no such x_1 in E_{n-1}, then the $\alpha_n/2$-neighborhood of x in E_{n-1} is an (α_n,β_n)-island in E_{n-1} and x would be removed.

Each of the points x_1 and x (that we denote also x_0 to make the notation uniform) belongs to E_{n-1} because it belongs to E_{n-2} together with some other point (at the distance greater than $\alpha_{n-1}/2$ but not exceeding $2\beta_{n-1}$). In this way we get a tree (Figure 6) that "explains" why x belongs to E_n.

The distance between x_0 and x_1 in this tree is at least $\alpha_n/2$ while the diameter of the subtrees starting at x_0 and x_1 does not exceed $\sum_{i<n} 2\beta_i$. Therefore, the Lemma's assumption guarantees that these subtrees cannot intersect and, moreover, that all the leaves of the tree are different. Note that all 2^n leaves of the tree belong to $E = E_0$. As every point appears in E independently from other points, such an "explanation tree" is valid with probability ε^{2^n}. It remains to estimate the number of possible explanation trees for a given point x.

To specify x_1 we need to specify horizontal and vertical distance between x_0 and x_1. Both distances do not exceed $2\beta_n$, therefore we need about $2\log(4\beta_n)$ bits to specify them (including the sign bits). Then we need to specify the distances between x_{00} and x_{01} as well as distances between x_{10} and x_{11}; this requires at most $4\log(4\beta_{n-1})$ bits. To specify the entire tree we therefore need

$$2\log(4\beta_n) + 4\log(4\beta_{n-1}) + 8\log(4\beta_{n-2}) + \ldots + 2^n\log(4\beta_1)$$

bits, and that is (reversing the sum and taking out the factor 2^n) equal to $2^n(\log(4\beta_1) + \log(4\beta_2)/2 + \ldots)$. Since the series $\sum \log\beta_n/2^n$ converges by assumption, the total number of explanation trees for a given point (and given n) does not exceed $2^{O(2^n)}$, so the probability for a given point x to be in E_n for a B_ε-random E does not exceed $\varepsilon^{2^n}2^{O(2^n)}$, which tends to 0 (even super-exponentially fast) as $n \to \infty$, assuming that ε is small enough.

We conclude that the event "x is not cleaned" (for a given point x) has zero probability; the countable additivity guarantees that with probability 1 all points in $\mathbb{Z}^2$ are cleaned.

It remains to show that every point with probability 1 is affected by finitely many steps only. Indeed, if x is affected by step n, then some point in its β_n-neighborhood belongs to E_n, and the probability of this event is at most

$$O(\beta_n^2)\varepsilon^{2^n}2^{O(2^n)} = 2^{2\log\beta_n + O(2^n) - \log(1/\varepsilon)2^n};$$

the convergence conditions guarantees that $\log\beta_n = o(2^n)$, so the first term is negligible compared to others, the probability series converges (for small enough ε) and the Borel–Cantelli lemma gives the desired result. □

Our next step: by definition a sparse set is split into a union of islands of different ranks; now we prove that these islands together occupy only a small part of the plane. To make this statement formal, we use the notion of Besicovich size (density) of a set $E \subset \mathbb{Z}^2$. Let us recall the definition. Fix some point O of the plane and consider squares of increasing size centered at O. For each square consider the fraction of points in this square that belong to E. The lim sup of these frequencies is called *Besicovitch density* of E. (Note that the choice of the center point O does not matter, since for any two points O_1 and O_2 large squares of the same size centered at O_1 and O_2 share most of their points.)

By definition the distance between two rank k islands is at least β_k. Therefore the $\beta_k/2$-neighborhoods of these islands are disjoint. Each of the islands contains at most α_k^2 points (it can be placed in a rectangle that has sides at most α_k). Each neighborhood has at least β_k^2 points (since it contains a $\beta_k \times \beta_k$-square centered at any point of the island). Therefore the union of all rank k islands has Besicovitch density at most $(\alpha_k/\beta_k)^2$. Indeed, for a large square the islands near its border can be ignored, and all other islands are surrounded by disjoint neighborhoods where their density is bounded by $(\alpha_k/\beta_k)^2$.

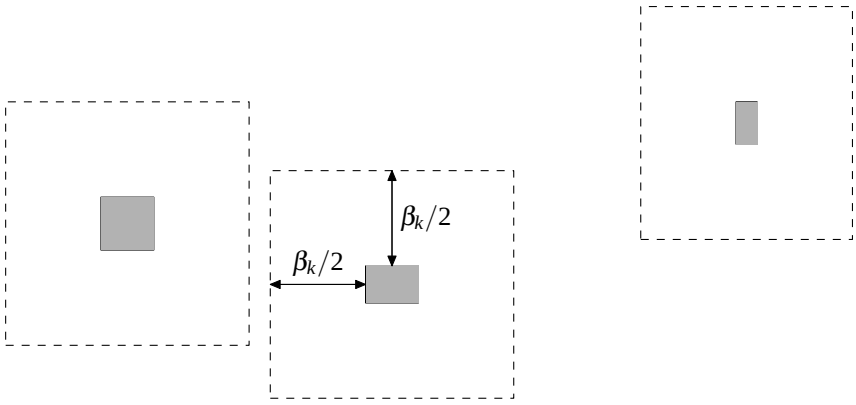

Figure 7: Rank k islands form a set of a small density. (In this picture each island is shown as a rectangle, which is not always the case.)

One would like to conclude that the overall density of all islands (of all ranks) does not exceed $\sum_k(\alpha_k/\beta_k)^2$. However, the Besicovitch density is in general not countably semi-additive (for example, the union of finite sets having density 0 may have density 1). But in our case the second condition of the definition of a sparse set (each point is covered by only finitely many neighborhoods of islands) helps.

Lemma 4. *Let E be a sparse set for a given family of α_k and β_k. Then Besicovitch density of E is $O(\sum(\alpha_k/\beta_k)^2)$.*

Proof of Lemma 4. Let O be a center point used in the definition of Besicovitch density. By definition of sparsity, this point is not covered by β_k-neighborhoods of rank k islands if k is greater than some K. Now we split the set E into two parts: one ($E_{\le}$) is formed by islands of rank at most K and other ($E_{>}$) is formed by all islands of bigger ranks. As we have just seen, in a large square the share of $E_{\le}$ is bounded by $\sum_{k\le K}(\alpha_k/\beta_k)^2$ up to negligible (as the size goes to infinity) boundary effects (we consider separately each $k \le K$ and then sum over all $k \le K$). The similar bound is valid for rank k islands with $k > K$, though the argument is different and a constant factor appears. Indeed, such an island I has β_k-neighborhood that does not contain the center point O. Therefore, any square S centered at O that intersects the island, contains also a significant part of its $\beta_k/2$-neighborhood N: the intersection of N and S contains at least $(\beta_k/2)^2$ elements.
Therefore, the share of $E_{>}$ in S is bounded by $4\sum_{k>K}(\alpha_k/\beta_k)^2$. $\square$

Remark. It is easy to choose α_k and β_k satisfying the conditions of Lemma 3 and having arbitrarily small $\sum(\alpha_k/\beta_k)^2$ (take geometric sequences that grow fast enough). Therefore we get the following well known result as a corollary of Lemmas 3 and 4: for every $\alpha > 0$ there exists $\varepsilon > 0$ such that with probability 1 a B_ε-random set has Besicovitch density less than α.

In fact we will need a slightly more complicated version of Lemma 4. We are interested not only in the Besicovitch density of a sparse set E but also in the Besicovitch density of a larger set:

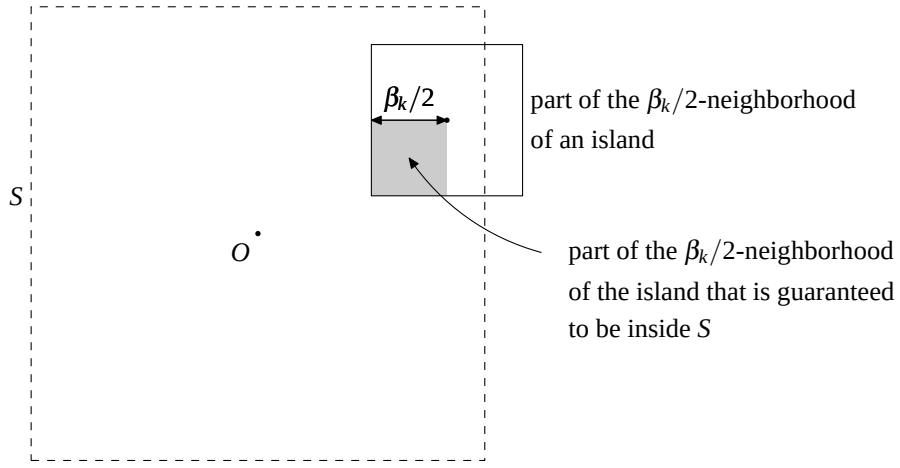

Figure 8: Together with a point in a rank k island, a square S contains at least $(\beta_k/2)^2$ points of its $(\beta_k/2)$-neighborhood.

the union of γ_k-neighborhoods of rank k islands in E. Here γ_k are some numbers (in most applications $\gamma_k = c\alpha_k$ for some constant c). The same argument gives the bound $4\sum((\alpha_k + 2\gamma_k)/\beta_k)^2$. Assuming that $\gamma_k \geq \alpha_k$, we can rewrite this bound as $O(\sum(\gamma_k/\beta_k)^2)$. So we arrive at the following statement:

Lemma 5. *Let E be a sparse set of a given family of α_k and β_k and let $\gamma_k \geq \alpha_k$ be some integers. Then the union of γ_k-neighborhoods of level k islands* (*over all k and all islands*) *has Besicovitch density $O(\sum(\gamma_k/\beta_k)^2)$.*

9.3 Islands as a tool in percolation theory

Let us show how some basic results of percolation theory can be proved using the island technique.

Theorem 10. *For some α_k and β_k satisfying Lemma 3 the complement of any sparse set E contains exactly one infinite connected component C; the complement of C has Besicovitch density $O(\alpha_k/\beta_k)^2$.*

Proof. Let $\gamma_k = 2\alpha_k$. (The choice of α_k and β_k will be discussed later.) For every k and for every rank k island fix a point in this island and consider the γ_k-neighborhood of this point. It is a square containing the entire island plus an additional security zone of width α_k and contained in the γ_k-neighborhood of the island.

It is enough to prove the following three statements:

- *The union U of all these squares* (*for all ranks*) *contains the set E and has Besicovitch size $O(\sum(\alpha_k/\beta_k)^2)$.*
- *The complement of U is connected.*

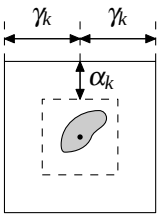

Figure 9: A point in a rank k island, its γ_k-neighborhood and the security zone of width α_k.

- *There are no other infinite connected component in the complement of E.*

The first statement is a direct corollary of Lemma 4 above.

To prove the second statement consider two points x and y that lie outside U. We need to prove that x and y can be connected by a path that is entirely outside U. Let us connect x and y by some path (say, one of the shortest paths) and then push this path out of U. Consider squares of maximal rank that intersect this path. For each of them consider the first moment when the path gets into the square and the last moment when the path goes out, and connect these two points by a path outside the square:

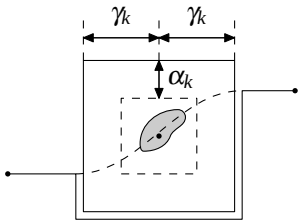

Figure 10: Pushing a path out of the square.

Let us assume that $\beta_k > 2\gamma_k$; then the new path is α_k-separated from rank k islands. Note also that the shift (the distance between the original path and the shifted one) does not exceed $3\gamma_k$.

Then we can do the same for islands of rank $k-1$ (pushing the path out of surrounding squares). Note that since the shift is bounded by $3\gamma_{k-1}$, we will not bump into islands of rank k assuming that $3\gamma_{k-1}$ is less than the width of the security zone, α_k.

Repeating this process for decreasing k, we finally get a path that connects x and y and goes entirely outside U. For this we need only that the total shift on the smaller levels, the sum $3\sum_{i<k}\gamma_i$ is less than α_k. (This is easy to achieve if α_k, β_k and γ_k are suitable geometric sequences.)

It remains to show that every infinite connected set intersects the complement of U. To show this, let us take a big circle centered at the origin and then push it out of U as described above. Since the center is outside β_k-neighborhoods of islands for large enough k, we may assume that the size of islands that intersect this circle are small compared with its radius (say, less than 1% of it; this can be guaranteed if the geometric sequences α_k, β_k and γ_k grow fast enough). Then after the change the circle will still encircle a large neighborhood of the origin, so any infinite connected component should cross such a circle. □

9.4 Bi-islands of errors

In the proof of our main result (Section 13) we need a more delicate version of the definition of islands. In fact we need such a definition that some counterpart of Lemma 3 could be applied even if the sequence $\log \beta_n$ grows much faster than 2^n (e.g., for $\beta_n = c^{(2.5)^n}$). In this section we define bi-islands (that generalize the notion of islands from Section 9.2) and prove bi-islands versions of Lemma 3, Lemma 4, and Lemma 5. The reader can safely skip this section for now and return here before reading Section 13.

Let $E \subset \mathbb{Z}^2$ be a set of points. As in Section 9.2, we call points in E *dirty*, and the other points are *clean*. Let $\beta \geq \alpha > 0$ be integers. A non-empty set $X \subset E$ is an (α, β)-*bi-island* in E if X can be covered by the union of some sets X_0, X_1 such that:

(1) the diameters of X_0 and X_1 do not exceed α;

(2) in the β-neighborhood of $X_0 \cup X_1$ there are no points from $E \setminus (X_0 \cup X_1)$.

(3) the distance between X_0 and X_1 does not exceed β.

(See Fig. 11.) In particular, an (α, β)-island is a special case of an (α, β)-bi-island (let X_1 be

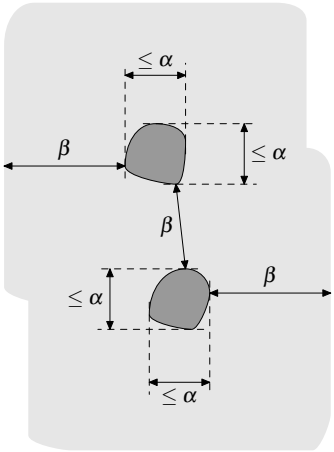

Figure 11: Bi-island is a union of two "islands" that are close to each other.

empty).

Note that one may split the same bi-island into X_0 and X_1 in different ways.

Obviously, every two different bi-islands are disjoint. Moreover, the distance between them is greater than β. The diameter of a bi-island is at most $(2\alpha + \beta)$.

Let (α_1, β_1), (α_2, β_2),... be a sequence of pairs of integers and $\alpha_i \leq \beta_i$ for all i. We define an iterative cleaning procedure for bi-islands. At the first step we find all (α_1, β_1)-bi-islands and remove all their elements from E (getting a smaller set E_1). Then we find in E_1 all (α_2, β_2)-bi-islands; removing them, we get $E_2 \subset E_1$, etc. Cleaning process is *successful* if every dirty point is removed at some stage.

Similarly to the case of islands, we say that a point $x \in \mathbb{Z}^2$ is *affected* during step i if x belongs to the β_i-neighborhood of one of bi-islands of rank i.

The set E is called *bi-sparse* (for a given sequence α_i, β_i) if the cleaning process defined above is successful, and, moreover, every point $x \in \mathbb{Z}^2$ is affected at finitely many steps only (that means that x is far from bi-islands of sufficiently large ranks).

We choose the values of α_i and β_i in such a way that for sufficiently small $\varepsilon > 0$ a B_ε-random set is bi-sparse with probability 1. The main achievement here is that the convergence condition is now weaker:

Lemma 6. *Assume that*

$$12 \sum_{k<n} \beta_k < \alpha_n \le \beta_n \text{ for every } n, \text{ and } \quad \sum_i \frac{\log \beta_i}{3^i} < \infty.$$

Then for all sufficiently small $\varepsilon > 0$ a B_ε-random set is bi-sparse with probability 1.

Proof of Lemma 6 is very similar to the proof of Lemma 3. At first we estimate the probability of the event "x is not cleaned after n steps" for a given point x. If $x \in E_n$, then x belongs to E_{n-1} and is not cleaned during the nth step (when (α_n, β_n)-bi-islands in E_{n-1} are removed). Then $x \in E_{n-1}$. Moreover, we show that there exist *two other* points $x_1, x_2 \in E_{n-1}$ such that the three distances $d(x,x_1)$, $d(x,x_2)$, $d(x_1,x_2)$ are all greater than $\alpha_n/2$ but not greater than $2\beta_n + 2(\alpha_n/2) < 3\beta_n$.

Let X_0 be the $\alpha_n/2$-neighborhood of x in E. If X_0 were an island, it would be removed. Since it does not happen, there is a point x_1 outside X_0 but in the β_n-neighborhood of X_0.

Let X_1 be the $\alpha_n/2$-neighborhood of x_1 in E. Again X_0 and X_1 do not form a bi-island. Each of them has diameter at most α_n and the distance between them is at most β_n. So the only reason why they are not a bi-island is that there exists a point $x_2 \in E$ outside $X_0 \cup X_1$ but in the β_n-neighborhood of it. The points x_1, and x_2 have the required properties.

To make the notation uniform, we denote x by x_0. Each of the points x_0, x_1, x_2 belongs to E_{n-1}. This means that each of them belongs to E_{n-2} together with a pair of other points (at the distance greater than $\alpha_{n-1}/2$ but not exceeding $3\beta_{n-1}$). In this way we get a 3-ary tree that "explains" why x belongs to E_n.

The distance between every two points among x_0, x_1, and x_2 in this tree is at least $\alpha_n/2$ while the diameters of the subtrees starting at x_0, x_1, and x_2 do not exceed $\sum_{i<n} 3\beta_i$. Thus, the Lemma's assumption guarantees that these subtrees cannot intersect and that all the leaves of the tree are different. The number of leaves in this 3-ary tree is 3^n, and they all belong to $E = E_0$. Every point appears in E independently from other points; hence, one such an "explanation tree" is valid with probability ε^{3^n}. It remains to count the number of all explanation trees for a given point x.

To specify x_1 and x_2 we need to specify horizontal and vertical distance between x_0 and x_1, x_2. These distances do not exceed $3\beta_n$, therefore we need about $4\log(6\beta_n)$ bits to specify them (including the sign bits). Then we need to specify the distances between x_{00} and x_{01}, x_{02} as well as the distances between x_{10} and x_{11}, x_{12}, and between x_{20} and x_{21}, x_{22}. This requires at most $12\log(6\beta_{n-1})$ bits. To specify the entire tree we therefore need

$$4\log(6\beta_n) + 12\log(6\beta_{n-1}) + 36\log(6\beta_{n-2}) + \ldots + 4 \cdot 3^{n-1}\log(6\beta_1),$$

which is equal to $4 \cdot 3^{n-1}(\log(6\beta_1) + \log(6\beta_2)/3 + \ldots)$. The series $\sum \log \beta_n / 3^n$ converges by assumption; so, the total number of explanation trees for a given point (and given n) does not exceed

$2^{O(3^n)}$. Hence, the probability for a given point x to be in E_n for a B_ε-random E does not exceed $\varepsilon^{3^n} 2^{O(3^n)}$, which tends to 0 as $n \to \infty$ (assuming that ε is small enough).

We conclude that the event "x is not cleaned" (for a given point x) has zero probability; hence, with probability 1 *all* points in $\mathbb{Z}^2$ are cleaned.

It remains to show that every point with probability 1 is affected by finitely many steps only. Indeed, if x is affected by step n, then some point in its β_n-neighborhood belongs to E_n, and the probability of this event is at most

$$O(\beta_n^2)\varepsilon^{3^n} 2^{O(3^n)} = 2^{2\log\beta_n + O(3^n) - \log(1/\varepsilon)3^n}.$$

From the convergence conditions we have $\log\beta_n = o(3^n)$, so the first term is negligible compared to others. The probability series converges (for small enough ε) and the Borel–Cantelli lemma gives the result. □

By definition, a bi-sparse set is split into a union of bi-islands of different ranks. Such bi-islands occupy only a small part of the plane:

Lemma 7. *Let E be a bi-sparse set for a given family of α_k and β_k. Then Besicovitch density of E is $O(\sum(\alpha_k/\beta_k)^2)$.*

Proof of Lemma 7 repeats the proofs of Lemma 4. □

Recalling Lemma 5, we may consider a sequence of numbers γ_k such that $\gamma_k \geq \alpha_k$. Then the Besicovitch density of the union of γ_k-neighborhoods of rank k bi-islands (for all k and for all islands) is bounded by $O(\sum(\gamma_k/\beta_k)^2)$.

However, this statement is not enough for us. In Section 13 we will need a kind of "closure" of γ_k-neighborhood of a bi-island:

Definition. *Let S be an n-level bi-island. We say that $(x,y) \in \mathbb{Z}^2$ belongs to the* extended γ_k*-neighborhood of S if this bi-island can be represented as $S = S_0 \cup S_1$ (diameters of S_0 and S_1 are not greater than α_n), and there exist points $(x,y'),(x,y'') \in \mathbb{Z}^2$ such that* $\operatorname{dist}(S_0,(x,y')) \leq \gamma_k$, $\operatorname{dist}(S_1,(x,y'')) \leq \gamma_k$*, and y is between y' and y'', see Fig. 12.*

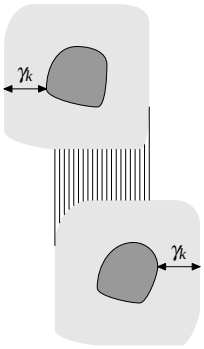

Figure 12: An extended neighborhood of a bi-island consists of the neighborhoods of its two parts and a zone between them.

The meaning of the last definition is quite simple: we take not only the points that are close to S but also those points that are placed somehow between the neighborhoods of S_0 and S_1.

Lemma 8. *Let E be a bi-sparse set for a given family of α_k and β_k satisfying the conditions of Lemma 6. Let γ_k be a sequence of numbers such that $\alpha_k < \gamma_k < \beta_k/8$, and the series $\sum(\gamma_i/\beta_i)$ converges. Then*

(1) *The Besicovitch density of the union of extended γ_k-neighborhoods of rank k bi-islands in E is bounded by $O(\sum(\gamma_k/\beta_k))$.*

(2) *For every large enough centered square of size $\Delta\times\Delta$, on each vertical section of this square (of size $1\times\Delta$) there exists a point not covered by the union of extended γ_k-neighborhoods of all k-level bi-islands (for all $k=1,2,\ldots$) in E.*

Proof of statement (1): Arguments are similar to the proof of Lemma 5. An extended γ_n-neigborhood of an n-level island can be covered by a rectangle of width $O(\gamma_n)$ and height $O(\beta_n+\gamma_n)$; so its area is $O(\gamma_n\beta_n)$ (since $\gamma_n\le\beta_n$)). The distance between any two bi-islands of rank n is at least β_n. Hence, the fraction of *extended* γ_n-neigborhoods of islands is $O(\sum\gamma_k/\beta_k)$ (we get it instead of the bound $O(\sum(\gamma_k/\beta_k)^2)$, which holds for simple γ_n-neigborhoods).

Proof of statement (2): Let O be the center point. Consider a $\Delta\times\Delta$-square S with the center O (so the distance between O and any other point in the square is at most $\Delta/2$). Denote by n the maximal integer such that $\beta_n<\Delta/2$. If $k>n$ and extended the γ_k-neighborhood of some k-level bi-island intersects the square, then the β_k-neighborhood of the same bi-island covers O (recall that $\gamma_k<\beta_k$). Since by definition of a bi-sparse set the point O is affected by only finitely many islands, we may assume that for a large enough Δ, all bi-islands whose extended γ_k-neighborhoods intersect S, have rank at most n.

Let us fix any vertical line in S. We show that there is a point on this line not covered by the extended γ_k-neighborhood of any rank k bi-island (for any k). Choose a bi-island of maximal rank that touches this line (i.e., its extended γ_k-neighborhood intersects it). Let it be a bi-island B_0 of rank k_0.

Let us now try to find a non-covered point going up and starting from this bi-island. Its extended neighborhood covers some interval I_0 on our line. If the point p_0 right above I_0 is not covered, we are done. But it can be covered by the extended γ_{k_1}-neighborhood of some bi-island of rank k_1. Note that $k_1<k_0$ since rank k_0 bi-islands cannot be close (β_{k_0}-zone between them).

It may happen that p_0 is covered by several extended neighborhoods for different ranks. We take the maximal among them and get a bi-island B_1 of rank k_1 whose extended neighborhood covers p_0. It touches an interval I_1 on the line that contains p_0. Take a point p_1 right above I_1, etc.

This process either gives us a point that is not covered or reaches the border of the $\Delta\times\Delta$-square. In the latter case we can start again from B_0 and go down. If we again reach the square border, then the entire vertical section of the square is covered by intervals I_0, I_1, etc. (in both directions). The length of I_k does not exceed $\beta_k+2\alpha_k+2\gamma_k$, so the total length of all intervals is bounded by

$$(2\alpha_{k_0}+\beta_{k_0}+2\gamma_{k_0})+2\sum_{i<k_0}(2\alpha_i+\beta_i+2\gamma_i)<\beta_{k_0}+4\gamma_{k_0}+2\sum_{i<k_0}(5\beta_i)<$$
$$<\beta_{k_0}+4\gamma_{k_0}+\alpha_{k_0}\le 2\beta_{k_0}\le 2\beta_n$$

(here we used the fact that β_i satisfy the conditions of Lemma 6). Since $\beta_n < \Delta/2$, extended neighborhoods of all B_i cannot cover the entire vertical section of S. $\square$

Lemmas 6–8 will be used in Section 13. (The arguments of Sections 10–12 do not refer to bi-islands.) These lemmas will be used for α_n, β_n such that $\log \alpha_n \sim q^n$ for $q > 2$, $\beta_n \sim \alpha_{n+1}$, and $\gamma_n = O(\alpha_n)$ or $\gamma_n = O(\alpha_n^2)$. For these parameters we cannot apply Lemmas 3 and 4 because $\log \beta_n$ grows faster than 2^n.

10 Robust tile sets

Now we are ready to construct an aperiodic tile set where isolated defects can be healed.

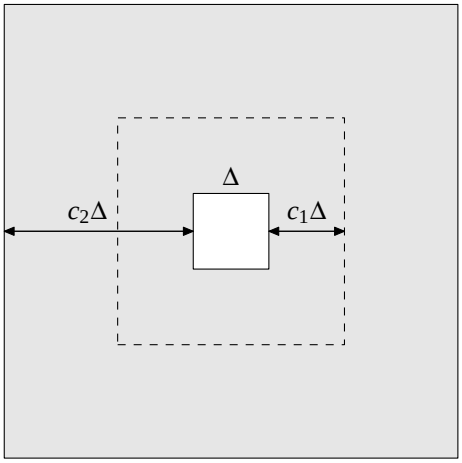

Figure 13: Patching holes

Let $c_1 < c_2$ be positive integers. We say that a tile set τ is (c_1, c_2)-*robust* if the following holds: For every Δ and for every τ-tiling U of the $c_2 n$-neighborhood of a square $\Delta \times \Delta$ excluding the square itself there exists a tiling V of the entire $c_2\Delta$-neighborhood of the square (including the square itself) that coincides with U outside of the $c_1 n$-neighborhood of the square (see Fig. 13).

Theorem 11. *There exists a self-similar tile set that is (c_1, c_2)-robust for some c_1 and c_2.*

Proof. For every tile set μ it is easy to construct a "robustified" version μ' of μ, i.e., a tile set μ' and a mapping $\delta\colon \mu' \to \mu$ such that: (a) δ-images of μ'-tilings are exactly μ-tilings; (b) μ' is "5-robust": every μ'-tiling of a 5×5 square minus 3×3 hole (see Fig. 14) can be uniquely extended to the tiling of the entire 5×5 square. (One can replace 5 by 4 in our argument using more careful estimates.)

Indeed, it is enough to keep in one μ'-tile the information about, say, 5×5 square in μ-tiling and use the colors on the borders to ensure that this information is consistent in neighbor tiles.

This robustification can be easily combined with the fixed-point construction. In this way we can get a 5-robust self-similar tile set τ if the zoom factor N (which is considered to be fixed in this

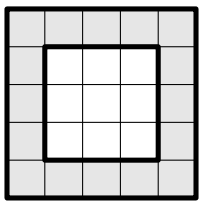

Figure 14: Filling a 3×3-hole

argument) is large enough. Let us show that this set is also (c_1, c_2)-robust for some c_1 and c_2 (that depend on N, but N is fixed.)

Indeed, assume that a tiling of a large enough neighborhood around an $\Delta \times \Delta$ hole is given. Denote by k the minimal integer such that $N^k \geq \Delta$ (so the k-level macro-tiles are greater than the hole under consideration). Note that the size of k-level macro-tiles is linear in Δ since $N^k \leq N \cdot \Delta$.

In the tiling around the hole, an $N \times N$ block structure is correct except for the N-neighborhood of the central $n \times n$ hole. Indeed, the colors encode coordinates, so in every connected tiled region coordinates are consistent. For similar reasons $N^2 \times N^2$-structure is correct except for the $N + N^2$-neighborhood, etc. So for the chosen k we get a k-level structure that is correct except for (at most) $9 = 3 \times 3$ squares of level k, and such a hole can be filled (due to 5-robustness) with $N^k \times N^k$ squares, and these squares can be then detailized back. (Note that this detalization is unique because of the isomorphism between tiles and macro-tiles.)

To start this procedure (and fill the hole), we need a correct tiling only in the $O(N^k)$-neighborhood of the hole (technically, we need to have a correct tiling in the $(3N^k)$-neighborhood of the hole; as $3N^k \leq 3N\Delta$, we let $c_2 = 3N$). The correction procedure involves changes in another $O(N^k)$-neighborhood of the hole (technically, changes touch $(2N^k)$ of the hole; $2N^k \leq 2N\Delta$, so we let $c_1 = 2N$). $\square$

11 Robust tile sets with variable zoom factors

The construction from the previous Section works only for self-similar tilings with a fixed zoom factor. It is enough for simple applications, as we see below in Section 12. However, in the proof of our main result in Section 13 we need variable zoom factor. So here we develop some technique suitable for this case. Reading of this Section can be skipped now but should be read before Section 13.

Now we explain how to get "robust" fixed-point tilings with a variable zoom factors N_1, N_2,... As well as in the case of a fixed zoom factor, the idea is that k-level macro-tiles are "responsible" for healing holes of size comparable with this macro-tiles.

Let $\Delta_0 \leq \Delta_1 \leq \Delta_2 \leq \ldots$ be a sequence of integers. Let $c_1 < c_2$ be positive integers. We say that a tile set τ is (c_1, c_2)-*robust against holes of size* $\Delta_0, \Delta_1, \ldots$, if the following holds: For every n and for every τ-tiling U of $c_2\Delta_n$-neighborhood of a square $\Delta_n \times \Delta_n$ excluding the square itself there exists a tiling V of the entire $c_2\Delta_n$-neighborhood of the square (including the square itself) that coincides with U outside of the $c_1\Delta_n$-neighborhood of the square. The difference with the

definition from Section 10 is that we take quantized values $\Delta \in \{\Delta_0, \Delta_1, \ldots\}$ instead of arbitrary Δ .

Proposition 2. *Assume a sequence of zoom factors N_k grows not too fast and not too slow* (*it is enough to assume that $N_k \geq C \log k$ and $C \log N_{k+1} < N_k$ for a large enough C, cf. discussions in Section 5*). *Then there exists a self-similar tile set with zoom factors N_k* (*k-level macro-tiles should be of size $L_k = N_0 \cdot \ldots \cdot N_{k-1}$*) *that is (c_1, c_2)-robust* (*for some c_1 and c_2*) *against holes of size $L_0, L_1, \ldots$*

Proof. First, we apply the fixed-point construction from Section 5 and get a tile set which is "self-similar" with variable zoom factors $N_1, N_2, \ldots$ Denote by μ_k the family of k-level macro-tiles corresponding to this tile set.

Further we make a "robustified" version of this tile set. It should be also a self-similar tile set with the same zoom factors $N_1, N_2, \ldots$ Denote by μ'_k the family of k-level macro-tiles for the new tiling. We need that there exists a mapping $\delta\colon \mu' \to \mu$ such that: (a) δ-images of μ'-tilings are exactly μ-tilings; (b) μ' is "5-robust": every μ'-tiling of a 5×5 square minus 3×3 hole (see Fig. 14) can be uniquely extended to the tiling of the entire 5×5 square.

To get such a robustification, it is enough to keep in every μ'-macro-tile the information about 5×5 square in μ-tiling and use the colors on the borders to ensure that this information is coherent in neighbor macro-tiles.

As usual, this robustification can be combined with the fixed-point construction. We get a 5-robust macro-tiles for all levels of our construction. "Self-similarity" guarantees that the same property holds for macro-tiles of all ranks, which implies the required property of generalized robusteness.

Indeed, assume that a tiling of a large enough neighborhood around a $\Delta \times \Delta$ hole is given, and $\Delta \leq L_n$ for some n. In the tiling around the hole, an $(L_1 \times L_1)$ block structure is correct except only for the L_1-neighborhood of the hole. For similar reasons $(L_2 \times L_2)$-structure is correct except for the $(L_1 + L_2)$-neighborhood, etc. So for n we get an n-level structure that is correct except for (at most) $9 = 3 \times 3$ squares of size $L_n \times L_n$, and such a hole can be filled (due to 5-robustness) with n-level macro-tiles. Note that detalization of a high-level macro-tile is unique after we know its "conscious" memory (reconstructed from the neighbors). [For the maximal complexity tile set (Section 7) it is not the case, and the absence of this property will become a problem in Section 13 where we robustify it. To solve this problem, we will need to use error correcting codes.]

To implement the patching procedure (and fill the hole) we need to have a correct tiling in the $O(L_n)$-neighborhood of the hole. The correction procedure involves changes in another $O(L_n)$-neighborhood of the hole. □

We can robustify tiling not only against holes, but against *pairs of holes*. To this end we slightly modify our definition of robusteness. Let $\Delta_0 \leq \Delta_1 \leq \Delta_2 \leq \ldots$ be an increasing sequence of integers, and $c_1 < c_2$ be positive integers. We say that a tile set τ is (c_1, c_2)*-robust against pairs of holes of size* $\Delta_0, \Delta_1, \ldots$, if the following holds: Let us have two sets $H_1, H_2 \subset \mathbb{Z}^2$, each of them of diameter at most Δ_n (for some $n > 0$). For every τ-tiling U of $c_2\Delta_n$-neighborhood of the union $(H_1 \cup H_2)$ excluding H_1 and H_2 themselves there exists a tiling V of the entire $c_2\Delta_n$-neighborhood of $(H_1 \cup H_2)$ (including H_1 and H_2 themselves) that coincides with U outside of the $c_1\Delta_n$-neighborhood of $(H_1 \cup H_2)$.

A robustification against *pairs* of holes can be done in the same way as the robustification against a single isolated hole above. Indeed, if these two holes are far apart from each other, we can "correct" them independently; if they are rather closed to each other, we correct them as one hole of (roughly) doubled size. So we can employ the same robustification technique as above; we need only to take a large enough "radius of multiplication" D (and use D-robusteness instead of 5-robusteness). So we get the following generalization of Proposition 2:

Proposition 3. *Assume a sequence of zoom factors N_k grows not too fast and not too slow* (*e.g., $N_k \geq C \log k$ and $C \log N_{k+1} < N_k$ for a large enough C*). *Then there exists a self-similar tile set with zoom factors N_k (k-level macro-tiles should be of size $L_k = N_0 \cdot \ldots \cdot N_{k-1}$) that is (c_1, c_2)-robust* (*for some c_1 and c_2*) *against pairs of holes of size $L_0, L_1, \ldots$*

Of course, similar propositions can be also proven for triples, quadruples and any other sets of holes of bounded cardinality. But we restrict ourselves to pairs only, because it is enough for the required applications in Section 13.

12 Strongly aperiodic robust tile set

Now we are ready to apply islands technique to construct a robust strongly aperiodic tile set.

Definition. *For a subset $E \subset \mathbb{Z}^2$ and a tile set τ we call by (τ, E)-tiling any mapping*

$$T \ : \ (\mathbb{Z}^2 \setminus E) \to \tau$$

such that for every two neighbor cells $x, y \in \mathbb{Z}^2 \setminus E$, the tiles $T(x)$ and and $T(y)$ satisfy the local restriction rules of τ. We may say that T is a τ-tiling of the plane with errors at points of E.

Theorem 12. *There exists a tile set τ with the following properties:* (1) *τ-tilings of $\mathbb{Z}^2$ exist;* (2) *for all sufficiently small ε for almost every* (*with respect to B_ε*) *subset $E \subset \mathbb{Z}^2$ every (τ, E)-tiling is at least $1/10$ Besicovitch-apart from every periodic mapping $\mathbb{Z}^2 \to \tau$.*

Remark 1. Since the tiling contains holes, we need to specify how we treat the holes when defining Besicovitch distance. We do *not* count points in E as points where two mappings differ; this makes our statement stronger.

Remark 2. The constant $1/10$ is not optimal and can be replaced by any other constant $\alpha < 1$.

Proof. Consider a tile set τ such that (a) all τ-tilings are α-aperiodic for every $\alpha < 1/4$; (b) τ is (c_1, c_2)-robust for some c_1 and c_2. Such a tile set can be easily constructed by combining the arguments used for Theorem 11 (p. 34) and Theorem 4 (p. 12).

Our plan is to choose some α_k and β_k such that:

- the conditions of Lemma 3 (p. 25) are satisfied (and therefore a random error set is sparse with respect to these α_k and β_k;
- for every sparse set $E \subset \mathbb{Z}^2$ every (τ, E)-tiling can be iteratively corrected (by changing it in the neighborhoods of islands of all ranks) into a τ-tiling of the entire plane;

- the Besicovitch distance between the tilings before and after correction is small.

Then we conclude that the original (τ, E)-tiling is strongly aperiodic since the corrected tiling is strongly aperiodic and close to the original one.

To implement this plan, we use the following lemma that describes the error correction process.

Lemma 9. *Assume that a tile set τ is (c_1, c_2)-robust, $\beta_k > 4c_2\alpha_k$ for every k and a set $E \subset \mathbb{Z}^2$ is sparse (with respect to α_i, β_i). Then every (τ, E)-tiling can be transformed into a τ-tiling of the entire plane by changing it in the union of $2c_1\alpha_k$-neighborhoods of rank k islands (for all islands of all ranks).*

Proof. Note that $\beta_k/2$-neighborhoods of rank k islands are disjoint and large enough to perform the error correction of rank k islands, since $\beta_k > 4c_2\alpha_k$. The definition of a sparse set guarantees also that every point is changed only finitely many times (so the limit tiling is well defined) and that the limit tiling has no errors. □

The Besicovitch size of the changed part of a tiling can be estimated by using Lemma 4: here $\gamma_k = 2c_1\alpha_k$ is proportional to α_k, so the Besicovitch distance between the original and corrected tilings (in Lemma 9) does not exceed $O(\sum_k(\alpha_k/\beta_k)^2)$. (Note that the constant in O-notation depends on c_1.)

It remains to chose α_k and β_k. We have to satisfy all the inequalities in Lemmas 3, Lemma 4 and Lemma 9 at the same time. To satisfy Lemma 4 and Lemma 9, we may let $\beta_k = ck\alpha_k$ for large enough c. To satisfy Lemma 3, we may let $\alpha_{k+1} = 8(\beta_1 + \ldots + \beta_k) + 1$. Then α_k and β_k grow faster that any geometric sequence (like $k!$ multiplied by some exponent in k), but still $\log\beta_i$ is bounded by a polynomial in i and the series in Lemma 3 converges.

With these parameters (taking c large enough) we may guarantee that Besicovitch distance between the original (τ, E)-tiling and the corrected τ-tiling does not exceed, say $1/100$. Since the corrected tiling is $1/5$-aperiodic and $1/10 + 2\cdot(1/100) < 1/5$, we get the desired result. □

13 Robust tile set that enforces complex tilings

In this section we prove the main result of the paper. We construct a tile set that guarantees large Kolmogorov complexity of every tiling, and which is robust with respect to random errors.

Theorem 13. *There exists a tile set τ and constants $c_1, c_2 > 0$ with the following properties:*

(1) *a τ-tiling of $\mathbb{Z}^2$ exists;*

(2) *for every τ-tiling T of the plane, every $N \times N$-square of T has Kolmogorov complexity at least $c_1 N - C_2$;*

(3) *for all sufficiently small ε for almost every (with respect to the Bernoulli distribution B_ε) subset $E \subset \mathbb{Z}^2$ every (τ, E)-tiling is at most $1/10$ Besicovitch-apart from some τ-tiling of the entire plane $\mathbb{Z}^2$;*

(4) *for all sufficiently small ε for almost every B_ε-random subset $E \subset \mathbb{Z}^2$, for every (τ, E)-tiling T Kolmogorov complexity of centered squares of T of size $N \times N$ is $\Omega(N)$.*

The rest of the section is devoted to the proof of this theorem. It combines virtually all technique developed in this paper: self-similar tile sets with variable zoom factors, embedding a sequence sequence with Levin's property (i.e., with linear Kolmogorov complexity of all factors) into tilings, bi-sparse sets, incremental error correcting and robustness against doubled holes.

13.1 The main difficulties and ways to get them round

We want to combine the construction from Section 7 with error correcting methods based on the idea of "islands" of faulty points. There are two main difficulties in this plan: fast growing zoom factors and gaps in vertical columns. Let us discuss these two items in some detail.

The first problem is that our construction of tiling with high Kolmogorov complexity from Section 7 requires *variable zoom factors*. What is even worse, zoom factors N_i must increase very fast (their logarithms grow faster than 2^i). Hence, we cannot apply directly the technique of islands from Section 9.2 since it works only when $\sum \frac{\log \beta_i}{2^i} < \infty$ (here β_i is the parameter from the definition of islands; in our construction it must be comparable with the size of i-level macro-tiles). To overcome this obstacle, we replace islands by bi-islands (the technique developed in Section 9.4). We deal with bi-islands mostly in the same way as we did with islands. We have seen in Section 9.4 that with probability 1 a Bernoulli-random set of faults is bi-sparse, so we can incrementally correct bi-islands of errors.

The second problem is that now it is not enough to know the "conscious" memory of a macro-tile to detalize it consistently with its neighbors. The missing information is the bits on the vertical columns (that carry bits of a high-complexity sequence ω). But random errors make gaps in vertical columns, so now the columns are split into parts that can (a priori) carry different bits. We organize some additional information flows between macro-tiles (of all ranks) to guarantee that each vertical column of tiles carries in most places the same bit value.

13.2 General scheme

Here we explain general ideas of our proof. First of all, we use macro-tiles with variable zoom factors $N_k = Q^{\lfloor 2.5^k \rfloor}$ for a large enough integer $Q > 0$. This means that every k-level macro-tile is an $(N_{k-1} \times N_{k-1})$-array of $(k-1)$-level macro-tiles. So the size (the number of columns = the number of rows) of a k-level macro-tile is $L_k = N_0 \cdot \ldots \cdot N_{k-1}$, and $L_k < N_k$. (Here 2.5 can be replaced by any constant between 2 and 3.)

To get tilings with high Kolmogorov complexity, we re-use the construction from Section 7 with the zoom factors defined above. Let us remind the idea of that construction (proof of Theorem 8). In the ith column of tiles (in a correct tiling) all tiles keep some bit ω_i, and we want that in the corresponding biinfinite sequence ω every N-bits substring of ω has Kolmogorov complexity $\Omega(N)$. Technically, we fix some constants $\alpha \in (0,1)$ and c and guarantee the following local property:

> *for every k-level macro-tile M ($k = 1,2,\ldots$), and for every substring x of ω that is contained in M's zone of responsibility (of length L_k) the inequality $K(x) \geq \alpha|x| - c$ holds.* (∗)

This property implies that the entire sequence ω has the required property: every N-bits substring of ω has Kolmogorov complexity $\geq \alpha' N - c'$ (for some other constants $\alpha' \in (0, \alpha)$ and c'). Indeed, every substring of ω is either contained in a large enough macro-tile or consists of two parts with this property (as he discussed in Section 7.4).

To enforce property $(*)$ we organize some computation on macro-tiles of all levels. The crucial point of the construction is propagation of bits ω_i to the computational zones of macro-tiles of high levels. [In fact, our construction in Section 7 guaranteed that only for two halves of the sequence, and we needed a special trick (Section 7.4) to guarantee this property for the entire sequence. For simplicity we do not consider the robustification of this enhanced version of a complex tiling; it can be done but is not needed for the proof of Theorem 13.]

To run this propagation, we delegate the bits of ω to macro-tiles of different levels. Every macro-tile M of level k is "responsible" for the L_k bits ω_i that correspond to the columns intersecting this macro-tile. One of these bits is "delegated" to macro-tile M itself. For convenience we assume that the ordinal number of the delegated bit (an integer from the range $1 \ldots L_k$) corresponds to the vertical position of M in the enclosing macro-tile of level $(k+1)$. More precisely, the ordinal number (i.e., its position in the zone of "responsibility") of the delegated bit for M is calculated as the ordinate of M in the enclosing macro-tile of level $(k+1)$ (this ordinate is an integer from the range $1 \ldots N_k$) modulo L_k (recall that $N_k \geq L_k$).

The bit delegated to macro-tile M must be kept in M's "consciousness", i.e., it is available to the computation running on M's computational zone. Besides the delegated bit, in the consciousness of M there should be also a few other bits from M's zone of "responsibility" (see Section 7). The bits from M's zone of "responsibility" that are not kept explicitly in this macro-tile consciousness, are kept in M implicitly. Indeed, each bit from the zone of "responsibility" of a k-level macro-tile M is delegated to some $(k-1)$-level macro-tile inside of M. Moreover, for each of these bits we can easily calculate the position of the $(k-1)$-level macro-tile to which this bit is delegated. We say that M keeps all these bits in "subconsciousness" (in fact, in a "conscious" memory of $(k-1)$-level blocks of M).

In this Section we tolerate random errors in a tiling, and vertical columns can be broken by faults. So we need to make additional efforts to enforce that the copies of ω_i consciously kept by different macro-tiles are coherent (at least for the macro-tiles that are not seriously damaged by local errors). We do it by means of small enough checksums, which guarantee that neighbor macro-tiles have coherent conscious and subconscious memory (unless they are damaged by error islands of very high rank).

To deal with random errors we use the technique of bi-islands (see Section 9.4). Our arguments will work if diameters of k-level bi-islands are comparable with the size of k-level macro-tiles. Technically we set $\alpha_k = 13L_{k-1}$ and $\beta_k = L_k$. Lemmas 6, 7 can be applied for these values of parameters. (Also in the sequel we will apply Lemma 8 with $\gamma_k = O(\alpha_k)$.)

13.3 The new construction of the tile set

We take the construction from Section 7 as a starting point and superimpose some new structures on k-level macro-tiles. We introduce these supplementary structures in several steps.

First step (introducing checksums): Every k-level macro-tile M (in a correct tiling) consists of an $N_{k-1} \times N_{k-1}$-array of $(k-1)$-level macro-tiles; each of these $(k-1)$-level macro-tiles keeps the bit delegated to it (these bits together determine the subconscious memory of M). Take in this 2D-array one horizontal row of a $(k-1)$-level macro-tiles (such a row consists of N_{k-1} macro-tiles). Denote the corresponding sequence of delegated bits by $\eta_1, \ldots, \eta_{N_{k-1}}$. Now we introduce some checksums for this sequence.

Let $D > 0$ be a constant (to be fixed later). The checksums should be enough to reconstruct all bits $\eta_1, \ldots, \eta_{N_{k-1}}$ *if at most D of these bits are corrupt* (it is enough for us to be able to reconstruct the corrupt bits if their positions are known). Also we want the checksums to be easily computable. The standard solution is the Reed–Solomon error correcting code.

Let us explain this solution in more detail. We take a finite field $\mathbb{F}_k$ of large enough size (the size of $\mathbb{F}_k$ must be approximately twice greater than N_{k-1}). Now we calculate a polynomial of degree less than N_{k-1} that takes values $\eta_1, \ldots, \eta_{N_{k-1}}$ at some N_{k-1} points of the field. Then take as checksums the values of this polynomial at some other $2D$ points from $\mathbb{F}_k$ (all $(N_{k-1} + 2D)$ points are fixed in advance).

Two different polynomials of degree less than N_{k-1} can coincide in at most $(N_{k-1} - 1)$ points. Hence, if at most D bits from the sequence $\eta_1, \ldots, \eta_{N_{k-1}}$ are corrupt, we can reconstruct them given the checksums defined above.

These checksums contain $O(\log N_{k-1})$ bits of information. Now we should discuss how to compute them.

Second step (calculating checksums): For each row in a k-level macro-tile, we can calculate its checksums (values of the corresponding polynomial) in the conscious memory of $k-1$-level macro-tile that form this row. This is done (in a standard way) as follows.

Let $\eta_1, \ldots, \eta_{N_{k-1}}$ be the values of a polynomial $p(x)$ (of degree less than N_{k-1}) at points $x_1, \ldots, x_{N_{k-1}}$. Assume we want to reconstruct all coefficients of this polynomial. We can do it by the following iterative procedure. For $i = 1, \ldots, N_{k-1}$ we calculate polynomials $p_i(x)$ and $q_i(x)$ (of degree $\leq (i-1)$ and i respectively) such that

$$p_i(x_j) = \eta_j \text{ for } j = 1, \ldots, i$$

and

$$q_i(x) = (x - x_1) \ldots (x - x_i)$$

It is easy to see that p_{i+1} and q_{i+1} can be calculated from polynomials p_i, q_i, and values x_{i+1} and η_{i+1}.

If we do not need to know the resulting polynomial $p = p_{N_{k-1}}(x)$ but want to get only the value $p(a)$ for some particular point a, then we can do all these calculations modulo $(x - a)$. Thus, to obtain the value of $p(x)$ at $2D$ different points, we run in parallel $2D$ copies of this process. At each step of the calculation we need to keep in memory only $O(1)$ elements of $\mathbb{F}_k$, which is $O(\log N_{k-1})$ bits of temporary data (the multiplicative constant in this $O(\cdot)$-notation depends on the value of D).

These calculations can be easily simulated by a tiling. To organize this simulation we include into conscious memory of $(k-1)$-level macro-tiles additional $O(\log N_{k-1})$ bits of data. This fits well our fixed-point construction since zoom factors N_k grow fast, and we have enough room in the

computational zone. We may assume that the resulting values of checksums are kept in conscious memory of the rightmost $(k-1)$-level macro-tile in each row.

Third step (consistency of checksums between macro-tiles): So far, every k-level macro-tile contains $O(N_{k-1}\log N_{k-1})$ bits of checksums, $O(\log N_{k-1})$ bits for every row. We want these checksums to be the same for every two vertical neighbor macro-tiles. It is inconvenient to keep the checksums for all rows only in the rightmost column (since it would create too much traffic in this column if we try to transmit the checksums to the neighbor macro-tiles). So we propagate the checksums of the ith row in a k-level macro-tile M ($i = 1,\ldots,N_{k-1}$) along the entire ith row and along the entire ith column of M. In other words, these checksums must be "consciously" known to all $(k-1)$-level macro-tiles in the ith row and in the ith column of M. On Fig. 15 we show the area of propagation of checksums for two rows (the ith and the jth rows).

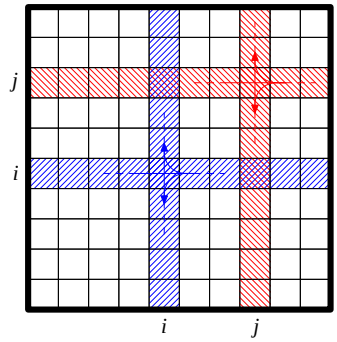

Figure 15: Propagation of checksums inside of a macro-tiles

On the border of two neighbor k-level macro-tiles (one above another) we check that in each column $i = 1,\ldots,N_{k-1}$ all the corresponding checksums calculated in both macro-tiles coincide. This check is redundant if there is no errors in the tiling: the checksums are calculated from delegated bits (which come from the sequence of bits ω encoded into tiles of the basic level), so the corresponding values must be equal for vertically aligned k-level macro-tiles. However this redundancy is useful to resist random errors, as we show in the sequel.

Fourth step (robustification): The explained above features organized in every k-level macro-tile (conscious and subconscious memory, calculating and propagating checksums, and all calculations simulated in the computational zone of a macro-tile) are implemented by means of bits kept in "conscious" memory of $(k-1)$-level macro-tiles. Now we made this construction C-robust in the following sense: each $(k-1)$-level macro-tile M keeps in consciousness not only "its own" data but also the bits previously assigned to $(k-1)$-level macro-tiles from its $(C\cdot L_{k-1})$-neighborhood (i.e., the $(2C+1)\times(2C+1)$ array of $(k-1)$-level macro-tiles centered at M). So, the content of the conscious memory of each macro-tile is multiplied by some constant factor $O(C^2)$.

We choose the constant C so that any k-level bi-island (that consist of two parts of size α_k) and even the $\gamma_k = O(\alpha_k)$-neighborhood of any k-level bi-island (we specify γ_k below) can involve only a small part of the (CL_{k-1})-neighborhood of any $(k-1)$-level macro-tile. (Note that we speak here about neighborhoods, not extended neighborhoods.)

This robustification allows us to talk about conscious memory of of a k-level macro-tile and its

building $(k-1)$-level blocks even if this macro-tile is damaged by one k-level bi-island (assuming there is no other errors).

The last remark (the number of bits in the conscious memory): The construction explained above requires that we put into the computational zones of $(k-1)$-level macro-tiles additional $\mathrm{poly}(\log N_{k-1})$ bits of data (the most substantial part is the data used for calculating the check-sums). This fits well our fixed-point construction because $\mathrm{poly}(\log N_{k-1})$ is much less than N_{k-2}, so we have enough room to keep and process all these data.

The tile set τ is defined. Since there exist ω with Levin's property, it follows that τ-tiling exist, and every $N \times N$-square of such a tiling has Kolmogorov complexity $\Omega(N)$. Further we prove that this τ satisifies also the statement (3) of Theorem 13.

13.4 Error correcting procedure

Denote by τ the tile set described in Section 13.3. Let $\varepsilon > 0$ be small enough. Lemma 11 says that B_ε-random set with probability 1 is bi-sparse. Now we assume that $E \subset \mathbb{Z}^2$ is a bi-sparse set (for the chosen values of α_i and β_i), and T is a τ-tiling of $\mathbb{Z}^2 \setminus E$. Further we explain how to correct errors and convert T into a tiling T' of the entire plane (T' should be close to T).

We follow the usual strategy. The set E is bi-sparse, i.e., it can be represented as a union of isolated bi-islands of different ranks. We correct them one by one, starting from bi-islands of low rank. We need only to explain how to correct bi-island S of rank k assuming that it is well isolated. i.e., in the β_k-neighborhood of this bi-island there are no other (still non-corrected) errors.

Let us recall that a k-level bi-island S is a union of two "clusters" S_0, S_1; diameters of both S_0 and S_1 are at most $\alpha_k = O(L_{k-1})$. Hence the clusters S_0 and S_1 touch only $O(1)$ macro-tiles of level $(k-1)$. The distance between S_0 and S_1 is at most β_k, and the β_k-neighborhood of S is free of other bi-islands of rank k and higher (so we can assume that the β_k-neighborhood of S is already cleaned of errors). Our correction procedure around S will involve only points in extended the γ_k-neighborhood of S, where $\gamma_k = 2\alpha_k$. Since $\beta_k + 2\gamma_k < 2L_k$, the correction procedure can involve points of only $O(1)$ macro-tiles of level k (four if it happens near the corner of a macro-tile).

Let M be one of macro-tiles intersecting the extended γ_k-neighborhood of k-level bi-island S. Basically, we need to reconstruct all $(k-1)$-level macro-tiles in M destroyed by S. First we will reconstruct the conscious memory of all $(k-1)$-level macro-tiles in M. This is enough to get all bits of ω from the zone of "responsibility" of M. Then we will reconstruct subconscious memory of all the blocks that make M, and reconstruct all n-level macro-tiles inside M for all $n < k$ (in a consistent way).

Thus, we start with reconstructing the conscious memory of all $(k-1)$-level macro-tiles M' enclosed in M. First of all we remind that conscious memory (the content of the computational zone) of every $(k-1)$-level macro-tile M' consists of several groups of bits:

[A] the binary representation of the number $(k-1)$ and coordinates of M' in the enclosing macro-tile M (these coordinates are integers from the range $1 \ldots N_{k-1}$);

[B] the bit (from the sequence ω) delegated to M';

[C] the bit (from ω) delegated to the enclosing macro-tile M (and propagated among all its $(k-1)$-level blocks, see our construction in Section 7);

[D] bits used to calculate and communicate the checksums for the corresponding row in the enclosing macro-tile M;

[E] the bits used to simulate a Turing machine on the computational zone of the enclosing k-level macro-tile M;

[F] a short segment of bits from M's zone of responsibility, and some computation that verifies that this segment does not violate property $(*)$ (i.e., does not contain factors of low Kolmogorov complexity).

Bits of field [A] in a small isolated group of $(k-1)$-level macro-tiles are trivially reconstructed from the surrounding macro-tiles of the same level. Fields [B,C,D,E] can be reconstructed because of robustification on the level of $(k-1)$-level macro-tiles (we organized the robustification on the level of $(k-1)$-level macro-tiles in such a way that we are able to reconstruct these fields for any $C \times C$ group of missing or corrupt $(k-1)$-level macro-tiles). So far the correcting procedure goes absolutely in the same way as in Section 11.

We postpone the question of reconstructing fields [F] of the conscious memory since we have a more urgent problem: to reconstruct a $(k-1)$-level blocks inside M it is not enough to know it conscious memory; we need to know also the bits of ω corresponding to the columns that cross M. Where can we obtain these bits? We can read them in the conscious memory of blocks in M; we can also look for these bits in a neighbor k-level tile (recall that S touches only $O(1)$ k-level macro-tiles and there is a "healthy" zone of k-level macro-tiles around them). However, the problem remains since we are not guaranteed that ω-bit above M, below M, and inside M are consistent. (This is the reason to use checksums.)

Denote by M_u and M_d the k-level macro-tiles just above and below S that are free of errors, see Fig. 16 (in this picture bi-island S touches only one k-level macro-tile; in case when S touches several k-level macro-tiles, almost the same arguments work, so all our explanations refer to Fig. 16). It is enough to prove that the values of bits ω reconstructed for M are equal to the corresponding values in M_u (and in M_d).

The macro-tiles M_u and M_d are error-free; therefore, they contain some sequences of bits on vertical lines (a priori different in M_u and M_d) and information on all the levels is consistent with the bit sequences (including checksums).

Note that bit sequences for M_u and M_d coincide at most places (outside the grey zone that is $O(1)$ blocks wide and consists of blocks involved in the correction and their vertical stripes). Indeed, the vertical transmission of these bits is not obstructed by errors.

Note also the after the correction the conscious information for all $(k-1)$-level blocks is consistently reconstructed; this implies that checksums are transmitted through M and are the same for M_u and M_d.

Therefore, the error correction property guarantees that bits in M_u and M_d are the same and we use these bits (with already existing conscious information) to reconstruct all the blocks in M, and get a consistent tiling in M and around.

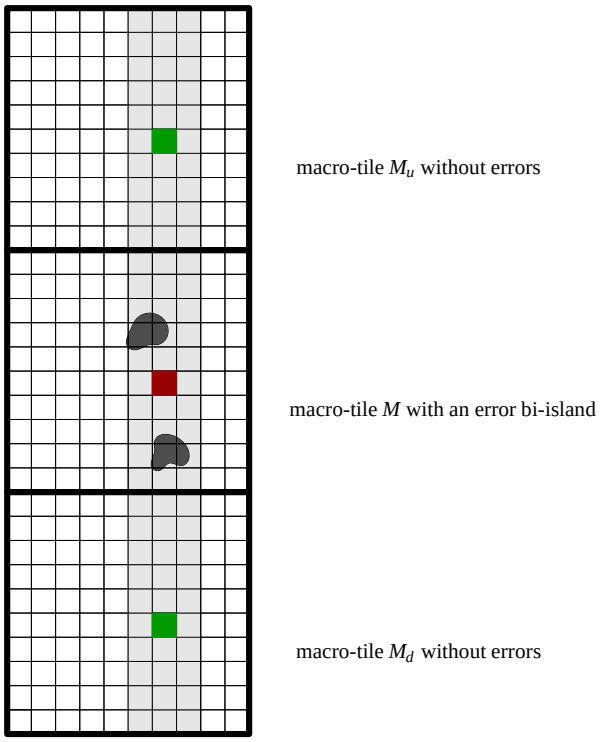

Figure 16: Bi-island of errors in a macro-tile

Finally, we have to look which blocks could be changed during this correction process. Blocks outside grey zone already had correct bits and conscious memory, so they are not changed. Moreover, not all the grey zone needs to be changed, only the part between two clusters (and their small neighborhoods) can be affected. This is the extended neighborhood of S as we have defined it. (This argument is the motivation of the definition.)

It remains to explain how to reconstruct fields [F] in the damaged $(k-1)$-level macro-tiles. It may look trivial: we already know all conscious and subconscious memory of the involved macro-tiles. So we know the values of the bits that should be tested in [F]. The testing procedure is deterministic and fixed (we enumerate strings with small Kolmogorov complexity and check that these strings are not factors of the tested interval). The only problem is to show that this procedure never crashes. In other words, we want to prove that reconstructed bits (inside M) indeed satisfy property $(*)$. Unfortunately, this is false: this finite string of bits can contain simple substrings (e.g., it may happen that we are between two error islands of high rank, and the values of bits on the vertical lines are not the "global" ones). However, by a miracle the global process of error correction converges to a correct tiling of $\mathbb{Z}^2$. We explain this in the next Section 13.5.

13.5 Levin's property for ω embedded into a (τ, E)-tiling

As we have seen, while correcting a bi-island of errors we can get stuck: a macro-tile that fills the gap does not exist, because the computation performed in this macro-tile would find that the bits of ω inside its zone of responsibility do not have the property $(*)$.

Nevertheless, we still want to apply the correction procedure described in Section 13.4 despite of this obstacle. Let us explain very informally how the correction process could work. If it happens that we cannot correct some bi-island of rank k completely, we do some "partial correcting". This means that for the involved k-level macro-tiles we (a) reconstruct their conscious memory, and (b) guarantee that bits of ω are correctly transmitted from top to bottom (i.e., the sequences of bits written on the top and bottom sides of a "partially corrected" macro-tile are equal). At the same time we tolerate several inconsistencies between subordinate macro-tiles of levels $m < k$ inside of these "partially corrected" k-level macro-tiles. Then we continue the error correction for bi-islands of higher ranks. A "partially corrected" macro-tile of level k can participate (in the same way as explained in Section 13.4) in the correction procedure for bi-islands of ranks $n > k$ as it were a correct macro-tile.

It can be shown (and this fact is nontrivial) that in the limit all "partially correct" macro-tiles disappear: every point (and every finite region) of the plane is changed finitely many times during the correction process; the limit tiling is defined and is a correct (not only "partially correct") tiling. Unfortunately, the implementation of this idea is technically complicated. To make the proof easier, we use a different and simpler (though slightly artificial) argument.

This argument constructs a tile set satisfying the statement of the theorem in two steps. First, we introduce a new tile set τ_1; then we select some part τ_0 of τ_1 and prove the theorem for τ_0. Every τ_1-tiling of the plane (without errors) carries a bi-infinite sequence ω of bits propagated along the vertical columns. The difference with τ is that now *every* ω can be embedded into some tiling. In τ_1-tilings, the procedure of testing the property $(*)$ is allowed to fail without destroying a tiling. Even if a simulation of the Turing machine in the computational zone of some k-level macro-tile M discovers that property $(*)$ is violated for some bits from M's zone of responsibility, the computational zone of M still can be correctly tiled, but a special "alarm" is raised.

More precisely, in τ_1-tiling every macro-tile (consciously) knows an additional "alarm bit" (saying whether the alarm is on or off for this macro-tile). Alarm bits on different levels are related: all subordinates of a macro-tile where alarm is on, should also have alarm on; subordinates of a macro-tile where alarm is off, may have any value of their alarm bits. This can be implemented in the same way as the substitution rules (see Section 3); the only difference is that now the rule is not deterministic, but this is not important for the construction (each tile consciously knows its alarm bit and its father's alarm bit). In particular, on the ground level the value of the alarm bit divides all tiles into two groups. The tiles where it is off form the set $\tau_0 \subset \tau_1$. All the arguments of the previous sections remain valid for τ_0; the advantage is that now we have an extension τ_1 of τ_0 which tolerates all sequences of bits (the violations of the property $(*)$ do not destroy the tiling, they just raise the alarm bit in the corresponding macro-tile and all its subordinate tiles).

The correction procedure from Section 13.4 guarantees that for a bi-sparse set E, every τ_0-tiling T of $(\mathbb{Z}^2 \setminus E)$ can be converted into a τ_1-tiling T' of the entire plane such that T and T' coincide everywhere except for the extended γ_k-neighborhoods of k-level bi-islands from E. On each step

of the error correction we resort to "alarm" tiles or "alarm" macro-tiles in case we cannot correctly reconstruct field [F] of some macro-tile damaged by errors from E. Our goal is to show that the resulting τ_1-tiling is in fact a τ_0-tiling. In other words, though we need to use "alarm" macro-tiles at some steps, in the limit tiling T' they are not used anywhere (every introduced "alarm" macro-tile is removed during one of the subsequent steps). Technically, it is enough to prove that the sequence ω embedded into the resulting T' satisfies the property $(*)$.

Informally, the argument goes as follows. Why the check of the sequence ω that appear in T' cannot fail? Assume that it fails on some level k. But in original tiling we already had k-level macro-tiles that were not touched when correcting higher (than k) levels, so they (a) carry the same bits as in the limit tiling T'; (b) check these bits for correctness. We need also to ensure that all possible (for level k) checks are performed in this way (in the original tiling). This is because in the corresponding vertical stripe there are sufficiently long vertical groups of level k macro-tiles not affected by the correction.

We start a more formal argument with a definition. We say that a k-level macro-tile in T' is *healthy* if it is outside extended γ_n-neighborhoods of all islands of ranks $n > k$ (from E). Recalling the argument explained in Section 13.4, we see that in a healthy macro-tile M we never raise the alarm bit. Indeed, a healthy k-level macro-tile cannot be involved into correction of bi-islands of rank greater than k. And even if M is damaged by a bi-island of rank k, the computational zone of M is reconstructed due to robustification, so there is no risk that the computation of a Turing machine that is embedded into M's computation zone terminates with an alarm.

Let ω be the biinfinite sequence of bits corresponding to the tiling T'. By the way of contradiction, assume that $x = \omega_i \dots \omega_{i+m-1}$ is a substring (of length m) of ω where $(*)$ is violated (Kolmogorov complexity of x is much less than m). Let k be a large enough number so that k-level macro-tiles have enough space on their computational zones to detect this violation.

Consider the (infinite) vertical stripe of k-level macro-tiles in T' that keep bits of x in their zones of responsibility. By construction, macro-tiles of this stripe test all short enough subsequences of bits from their (common) zone of responsibility. There are L_k bits of ω in this zone. Different macro-tiles of this stripe test different substrings of this zone (all having the same length but different starting points). The macro-tiles that test x appear periodically in this stripe with period L_k (measured in macro-tiles). It remains to show that one of these tests was already present in the original tiling (and therefore does not raise an alarm).

We know that the error correction procedure involves only γ_n-neighborhoods of n-level bi-islands of errors (for all n), where $\gamma_n = 2\alpha_n = O(L_{n-1})$. Denote $\tilde{\gamma}_n = \gamma_n + L_{n-1}^2$. The value $\tilde{\gamma}_n$ is larger than γ_n but still much less than β_n. Hence we can apply Lemma 8 (part 2) to our usual α_n and β_n, and $\tilde{\gamma}_n$ instead of γ_n. This Lemma implies that in any vertical line there exists a point z which is not covered by the extended $\tilde{\gamma}_n$-neighborhood of any n-level bi-island (whatever n is). Fix such a point z on some vertical line inside the stripe.

The choice of z guarantees that k-level macro-tile that contains z is healthy, because the distance between z and extended γ_n-neighborhoods of n-level bi-islands (for $n > k$) is at least the gap between γ_n and $\tilde{\gamma}_n$, and $\tilde{\gamma}_n - \gamma_n = L_{n-1}^2 \gg L_k$ (recall that $n - 1 \geq k$). For the same reason not only this macro-tile, but also L_k macro-tiles around it (in fact, even more) are healthy. One of them tests the substring x, therefore x cannot have low complexity.

Thus, we have proven that ω satisifies $(*)$, and the obtained τ_1-tiling T' is in fact a correct τ_0-tiling (without alarm macro-tiles). The difference between T and T' is covered by extended γ_k-neighborhoods of k-level bi-islands. Now Theorem 13 (part 3) follows from Lemma 8 (part 1) applied to the usual $\alpha_k, \beta_k, \gamma_k$.

It remains to prove part (4) of Theorem 13. Let E be a bi-sparse set, T be a tiling of $\mathbb{Z}^2 \setminus E$, and T' be a correct tiling of $\mathbb{Z}^2$ such that T and T' differ only in extended γ_k-neighborhoods of k-level bi-islands from E (for $k = 1, 2, \ldots$). The existence of T' is already proven.

Fix a point O. Since E is bi-sparse, O is covered by β_k-neighborhoods of only finitely many bi-islands. Hence, for large enough Δ, the $\Delta \times \Delta$-square Q_Δ centered at O intersects extended γ_k-neighborhoods of k-level bi-islands only if $\beta_k < \Delta$. (If the extended γ_k-neighborhood of some bi-island intersects Q_Δ and $\beta_k \geq \Delta$, then $\beta_k - \gamma_k > \Delta/2$ and O is covered by β_k-neighborhood of this bi-island.) Therefore, to reconstruct T' in Q_Δ it is enough to correct all the bi-islands of bounded levels ($\beta_k < \Delta$), and all the information needed to perform this correction in Q_Δ is determined by the restriction of T to the centered $O(\Delta) \times O(\Delta)$-square. Therefore, the Kolmogorov complexity of the latter is $\Omega(\Delta)$ (as the complexity of the restriction of T' to Q_Δ is), and we are done. $\square$

14 Acknowledgements

The results included in this paper were discussed with many colleagues, including Peter Gács, Leonid Levin, our LIF collaborators Emmanuel Jeandel, Gregory Lafitte, Nicolas Ollinger, and Michael Weiss. We are grateful to all of them and to the participants of seminars (Kolmogorov seminar at Moscow State University, IITP seminar, FRAC) and conferences (DLT 2008, ICALP 2009) where some of these results were presented.

References

[1] C. Allauzen, B. Durand. Appendix A: Tiling Problems. In: E. Börger, E. Grädel, Y. Gurevich, *The Classical Decision Problems*, Springer-Verlag, 1996.

[2] R. Berger, The Undecidability of the Domino Problem. *Mem. Amer. Math. Soc.*, **66**, 1966.

[3] L. Bienvenu, A. Romashchenko, A. Shen. Sparse sets. *Journées Automates Cellulaires 2008* (*Uzès*), p. 18–28. MCCME Publishers, 2008. Available online: http://hal.archives-ouvertes.fr/docs/00/27/40/10/PDF/18-28.pdf

[4] K. Culik, An Aperiodic Set of 13 Wang Tiles, *Discrete Math.*, **160**, 245–251, 1996.

[5] B. Durand, L. Levin, A. Shen, Complex Tilings. *J. Symbolic Logic*, **73** (2), 593–613, 2008 (See also Proc. *33rd Ann. ACM Symp. Theory Computing*, 732–739, 2001, and `www.arxiv.org/cs.CC/0107008` for an earlier version.)

[6] B. Durand, L. Levin, A. Shen, Local Rules and Global Order, or Aperiodic Tilings, *Math. Intelligencer*, **27**(1), 64–68, 2004.

[7] B. Durand, A. Romashchenko, On Stability of Computations by Cellular Automata, In Proc. *European Conf. Compl. Syst.*, Paris, 2005.

[8] B. Durand, A. Romashchenko, A. Shen, Fixed Point and Aperiodic Tilings, *Developments in Language Theory, 12th International Conference, DLT 2008, Kyoto, Japan, September 16–19, 2008. Proceedings*, Springer, Lecture Notes in Computer Science, volume 5257, 2008. ISBN: 978-3-540-85779-2. P. 276–288.

[9] B. Durand, A. Romashchenko, A. Shen, High Complexity Tilings with Sparse Errors, *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5–12, 2009, Proceedings, Part I.* (Lecture NOtes in Computer Science, 5555.) Springer-Verlag, 2009, ISBN: 978-3-642-02926-4. P. 403–414.

[10] P. Gács, Reliable Cellular Automata with Self-Organization, In Proc. *38th Ann. Symp. Found. Comput. Sci.*, 90–97, 1997.

[11] P. Gács, Reliable Cellular Automata with Self-Organization, *J. Stat. Phys.*, **103** (1/2), 45–267, 2001.

[12] L. Gray, A Reader's Guide to Gács' Positive Rates Paper. *J. Stat. Phys.*, **103** (1/2), 1–44, 2001.

[13] B. Grünbaum, G.C. Shephard, *Tilings and Patterns*, W.H. Freeman and Company, New York, 1987.

[14] Yu. Gurevich, I. Koryakov, Remarks of Berger's paper on the domino problem, *Siberian Math. Journal*, **13**, 319–321, 1972.

[15] W. Hanf, Nonrecursive tilings of the plane, I, *Journal of Symbolic Logic*, **39**:283–285, 1974.

[16] M. Hochman, On the dynamic and recursive properties of multidimensional symbolic systems. *Inventiones mathematicae*, **176**, 131–167 (2009).

[17] J. Kari, A Small Aperiodic Set of Wang tiles, *Discrete Math.*, **160**, 259–264, 1996.

[18] H. Rogers, *The Theory of Recursive Functions and Effective Computability*, Cambridge, MIT Press, 1987.

[19] Lafitte, Weiss, The paper where the formal statement about fixed point is proved

[20] L. Levin, Aperiodic Tilings: Breaking Translational Symmetry, *Computer J.*, **48**(6), 642–645, 2005. On-line: `http://www.arxiv.org/cs.DM/0409024`

[21] S. Mozes, Tilings, Substitution Systems and Dynamical Systems Generated by Them, *J. Analyse Math.*, **53**, 139–186, 1989.

[22] D. Myers. Nonrecursive tilings of the plane, II, *Journal of Symbolic Logic*, **39**:286–294, 1974.

[23] J. von Neumann, *Theory of Self-reproducing Automata*, Edited by A. Burks, University of Illinois Press, 1966.

[24] N. Ollinger, *Two-by-two Substitution Systems and the Undecidability of the Domino Problem*, In Proc. *Computability in Europe*, LNCS **5028**, 476–485, 2008.

[25] Yu. Pritykin, J. Ulyashkina, Aperiodicity Measure for Infinite Sequences, Computer Science — Theory and Applications. Fourth International Computer Science Symposium in Russia, CSR 2009, Novosibirsk, Russia, August 18–23, 2009. (Lecture Notes in Computer Science, v. 5675) Springer, 2009. ISBN: 978-3-642-03350-6. P. 274–285.

[26] R. Robinson, Undecidability and Nonperiodicity for Tilings of the Plane. *Inventiones Mathematicae*, **12**, 177–209, 1971.

[27] An. Rumyantsev, M. Ushakov, Forbidden Substrings, Kolmogorov Complexity and Almost Periodic Sequences, *STACS 2006 Proceedings*, Lecture Notes in Computer Science, Vol. 3884, Springer, 2006.

[28] S.G. Simpson, Medvedev degrees of 2-dimensional subshifts of finite type, MPIM2007-67 preprint, 2007.

[29] M. Zaks, A.S. Pikovsky, J. Kurths, On the Correlation Dimension of the Spectral Measure for the Thue–Morse Sequence, *J. Stat. Phys.*, **88**(5/6), 1387–1392, 1997.