

Fixed Point Constructions in Tilings and Cellular Automata

Ilkka Törmä  

Department of Mathematics and Statistics, University of Turku, Finland

Abstract

The fixed point construction is a method for designing tile sets and cellular automata with highly nontrivial dynamical and computational properties. It produces an infinite hierarchy of systems where each layer simulates the next one. The simulations are implemented entirely by computations of Turing machines embedded in the tilings or spacetime diagrams. We present an overview of the construction and list its applications in the literature.

2012 ACM Subject Classification Theory of computation → Models of computation

Keywords and phrases Tilings, Wang tiles, cellular automata, multidimensional symbolic dynamics, self-simulation

Digital Object Identifier 10.4230/OASICS.AUTOMATA.2021.4

Category Invited Talk

Related Version *Full Version:* <https://arxiv.org/abs/2105.00443>

1 Introduction

In this short survey we present an overview of the *fixed point construction* of Wang tile sets and cellular automata, also known as *programmatic self-simulation*. It is a method of defining an infinite sequence of tile sets $(T_k)_{k \geq 0}$ with the property that each T_k simulates T_{k+1} , in the sense that valid tilings over T_k have the form of infinite regular $N_k \times N_k$ grids for some constant $N_k > 1$ and each grid cell behaves like a tile of T_{k+1} . The simplest variant has $T_k = T_{k+1}$ for all k , so that T_0 simulates itself. The defining property of the technique is that the simulations are implemented almost entirely “in software” by computations of embedded universal Turing machines. This provides considerable flexibility, since modifying the next tile set T_{k+1} is as easy as modifying the program of the machine that runs on T_k . On the flip side, the flexibility comes with the price of nonmodularity, as nontrivial constructions are hard to reuse without presenting them in detail. The tile sets produced by the fixed point method are also inevitably massive, and it is usually impractical to determine them exactly. Of course, this is true for most constructions in symbolic dynamics that involve simulating nontrivial computation, but the fixed point construction stands out by requiring it even for results that do not have a computational flavor, such as the existence of an aperiodic tile set.

The fixed point construction has its roots in Kurdyumov’s informal article [24] on probabilistic cellular automata. Kurdyumov’s ideas were implemented rigorously by Gács in his disproof of the positive rates conjecture [13, 14], which uses programmatic self-simulation as the basis of an extremely intricate construction. Durand, Romashchenko and Shen isolated this part of the construction and applied it to Wang tile sets in a series of papers [8, 9, 10, 11] culminating in [12]. A series of further applications by them and other authors has followed. Interestingly, many of these share a common theme: a result on tiling systems, sofic shifts or cellular automata was proved in all dimensions except one or two using a geometric construction, such as Robinson tiles [28] or Mozes’s realization of substitutive shifts [27], and the remaining low-dimensional cases were settled by a fixed point construction.



© Ilkka Törmä;

licensed under Creative Commons License CC-BY 4.0

27th IFIP WG 1.5 International Workshop on Cellular Automata and Discrete Complex Systems (AUTOMATA 2021).

Editors: Alonso Castillo-Ramirez, Pierre Guillon, and Kévin Perrot; Article No. 4; pp. 4:1–4:13

OpenAccess Series in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2 Definitions and notation

Let A be a finite alphabet and $d \geq 1$. A *pattern* is a function $P : D \rightarrow A$ with domain $D = D(P) \subseteq \mathbb{Z}^d$. Patterns of domain \mathbb{Z}^d are called *configurations*, and they form the d -dimensional full shift $A^{\mathbb{Z}^d}$. Define the *shift map* $\tau : \mathbb{Z}^d \times A^{\mathbb{Z}^d} \rightarrow A^{\mathbb{Z}^d}$ by $(\tau^{\vec{n}}x)_{\vec{v}} = x_{\vec{v}+\vec{n}}$. A set of finite patterns F defines a *subshift* as the subset $X_F = \{x \in A^{\mathbb{Z}^d} \mid \forall P \in F, \vec{v} \in \mathbb{Z}^d : (\tau^{\vec{v}}x)|_{D(P)} \neq P\}$ where no pattern from F occurs at any position. If F is finite, X_F is a *shift of finite type* (SFT), and if F is computably enumerable, X_F is an *effective subshift*. A pattern is *locally valid* for F if no element of F occurs in it as a sub-pattern. The set of domain- D patterns occurring in a subshift X is denoted $\mathcal{L}_X(D) = \{x|_D \mid x \in X\}$. If a subshift X does not properly contain another nonempty subshift, it is *minimal*. If a subshift X is a union of minimal subshifts, it is *quasiperiodic*.¹

A particular class of 2-dimensional SFTs is given by sets of *Wang tiles*, which are squares with colored edges. Formally, a Wang tile set is a subset of C^4 for a finite set C of colors, with the four components representing the colors of the east, north, west and south edges of the square. We forbid exactly those 1×2 and 2×1 patterns where the adjacent edges of the two tiles have different colors. A d -dimensional SFT X is *aperiodic* if for all $x \in X$ and $\vec{p} \in \mathbb{Z}^d$ there exists $\vec{v} \in \mathbb{Z}^d$ with $x_{\vec{v}+\vec{p}} \neq x_{\vec{v}}$. A set of Wang tiles is called aperiodic if it defines a nonempty aperiodic SFT.

A function $\phi : X \rightarrow Y$ between subshifts $X \subseteq A^{\mathbb{Z}^d}, Y \subseteq B^{\mathbb{Z}^d}$ is a *block map* if there is a finite *neighborhood* $N \subset \mathbb{Z}^d$ and *local function* $\Phi : \mathcal{L}_X(N) \rightarrow B$ such that $\phi(x)_{\vec{v}} = \Phi((\tau^{\vec{v}}x)|_N)$ for all $x \in X$ and $\vec{v} \in \mathbb{Z}^d$. If X is an SFT and ϕ is surjective, then Y is a *sofic shift* and X is an *SFT cover* of Y . A *cellular automaton* (CA) is a block map from a full shift to itself. A *spacetime diagram* of a CA $\phi : A^{\mathbb{Z}^d} \rightarrow A^{\mathbb{Z}^d}$ is a configuration $x \in A^{\mathbb{Z}^{d+1}}$ with $x|_{\mathbb{Z}^d \times \{i+1\}} = \phi(x|_{\mathbb{Z}^d \times \{i\}})$ for each $i \in \mathbb{Z}$. In a spacetime diagram, the d -dimensional slices obtained by fixing the last coordinate form a trajectory of ϕ .

A standard reference for one-dimensional subshifts, with a short appendix on the multidimensional setting, is [25]. See [23] for a survey on the theory of CA. Wang tiles were first defined in [32], and aperiodic sets of Wang tiles were first constructed in [2].

3 The fixed point construction

3.1 Tile sets

In this section we present an outline of the fixed point construction in the context of Wang tiles following [12]. We omit much of the detail.

► **Definition 1.** *Let T and S be two sets of Wang tiles. A simulation of S by T with zoom factor N is an injective function $\alpha : S \rightarrow T^{N \times N}$ such that:*

- *For any $s_1, s_2 \in S$, the horizontal concatenation $s_1 s_2$ is locally valid over T if and only if $\alpha(s_1)\alpha(s_2)$ is locally valid over S , and similarly for their vertical concatenation.*
- *For every valid tiling $x \in T^{\mathbb{Z}^2}$, there is a unique vector $\vec{v} \in [0, N-1]^2$ such that $(\tau^{\vec{v}+(iN, jN)}x)|_{[0, N-1]^2} \in \alpha(S)$ for all $(i, j) \in \mathbb{Z}^2$.*

Suppose first that we are given the tile set S and wish to define, for each large enough zoom factor N , a tile set $T = T(N)$ and a simulation $\alpha : S \rightarrow T^{N \times N}$. We will implement T in a way that makes the choice $S = T$ possible with relatively minor modifications. Assume

¹ Note that an arbitrary union of minimal subshifts is generally not a subshift.

the colors of S -tiles are binary strings of some length k , that is, elements of $\{0, 1\}^k$. First, each tile T has an *address* $(i, j) \in [0, N - 1]^2$. Local rules ensure that its east and north neighbors have addresses $(i + 1, j)$ and $(i, j + 1)$ modulo N , respectively. Concretely, the set of colors of T is $[0, N - 1] \times C$ for some auxiliary set C , and each tile with address (i, j) has the form $((j + 1, c_e), (i + 1, c_n), (j, c_w), (i, c_s))$ for some $c_e, c_n, c_w, c_s \in C$. Then in a valid tiling $x \in T^{\mathbb{Z}^2}$, the tiles are partitioned into a grid of $N \times N$ blocks called *macrotiles* whose southwest corners are the tiles with address $(0, 0)$. The address of each tile is its relative position within the macrotile containing it.

Each tile of a macrotile $t \in T^{N \times N}$ has a specific role that depends on its address. A fixed rectangular subset $R = [a, b] \times [c, d] \subset [0, N - 1]^2$ of addresses forms the *computation zone*. It stores a simulated computation of a Turing machine M , which recognizes the set S in the sense that it halts on input $w \in \{0, 1\}^{4k}$ if and only if w encodes a tile of S . A reasonable choice for the parameters is $a = c = \lfloor N/3 \rfloor$ and $b = d = \lfloor 2N/3 \rfloor$, so that the dimensions of R increase linearly in N . The simulation of M is implemented in some standard way, such as the one presented in [28]. The bottom row of R initializes the computation, and each row above it simulates one computation step. If M does not halt before reaching the top row of R , a tiling error is produced. As long as N (and thus R) is large enough, the simulated machine has enough time and space to verify any input that encodes a tile of S .

Fix an interval $I \subset [0, N - 1]$ of length k , the number of bits in a color of S . Each tile with address in $B = (\{0, N - 1\} \times I) \cup (I \times \{0, N - 1\})$ stores an additional bit, called a *border bit*. Their purpose is to encode the four colors of the simulated tile of S . The border bits are constrained to be equal to those of the neighboring macrotiles. For each address in B , we fix a contiguous path of addresses to the bottom row of the computation cone, making sure that paths originating from distinct addresses are pairwise disjoint. Using local rules, we force all tiles with addresses on a given path to store the same bit. In this way the border bits are routed to the computation zone, and the resulting binary word of length $4k$ forms the input to the simulated machine M . If M accepts this word, then it represents a tile $s \in S$, and we set $\alpha(s) = t$. With this construction the set of locally valid macrotiles equals $\alpha(S)$, and the adjacency rules of macrotiles are identical to those of S . Thus T simulates S with zoom factor N .

We would now like to choose $S = T$, so that $T = T(N)$ simulates itself. There are a few complications that we need to handle. First, the colors of T must be binary words of constant length. We encode one component of the address in $\lceil \log_2 N \rceil$ bits and the remaining data in another $m = \lceil \log_2 |C| \rceil$ bits. Second, the state set and transition rules of the Turing machine M , which are part of the tile set T , depend on S . We replace M by a fixed universal Turing machine M_U that takes as input a binary word $w \in \{0, 1\}^*$ and, on a separate track of the tape, a program $p \in \{0, 1\}^*$, and computes $p(w)$. The machine M_U should also be efficient in the sense that for any Turing machine M' , there is a program p such that $M_U(p, w) = M'(w)$ for all w and the number of computation steps in $M_U(p, w)$ is polynomial in $|p|$, $|w|$, and the number of computation steps in $M'(w)$. Now the set C of auxiliary colors of $T(N)$ (and thus the number m) is fixed, and to simulate $T(N)$ itself, it suffices to find a suitable program $p_T = p_{T(N)}$. We enforce the simulation by requiring that each tile with address $(a + i, b)$ stores the bit $(p_T)_i$ on the program track of the simulated tape of M_U , where (a, b) is the southeast corner of the computation zone. We call this *the program condition*, and it is part of the definition of T .

Since we introduced the program condition into T , we must enforce it in the simulated version as well. Namely, suppose for a moment that the machine M_U always has enough time and space to finish its simulated computation in a macrotile. At this point of the construction

we then have a tile set T that simulates “tile set T without the program condition,” which is of course not equal to T . We can enforce the program condition in the simulated tiling as well by modifying the program p_T as follows. If the machine M_U simulated inside a macrotile $t \in T^{N \times N}$ is given inputs that correspond to a tile with address $(a + i, b)$ on the bottom row of the computation zone, then it reads the program bit $d \in \{0, 1\}$ of any tile of t with address $(a + i, b')$ for $b' \in [0, N - 1]$ (which are all equal), and verifies that the simulated program bit of t equals d . The program condition of T ensures that the initial tape of the machine M_U contains the program p_T , so we are guaranteed that $d = (p_T)_i$. After this modification, T simulates itself (given the assumption on M_U).

Finally, consider the time and space requirements of M_U . The number of bits needed to store a color of $T(N)$ is $\ell = \lceil \log_2 N \rceil + m = O(\log N)$, since m is fixed. The program p_T can store the number N in a variable, so its length is likewise $O(\log N)$. It should be clear that for each $w \in \{0, 1\}^{4\ell}$ the computation of $p_T(w)$ runs in time and space $O(\text{poly}(\log N))$ as long as we implement T and p_T in a reasonable way. For large enough N the computation zone can accommodate the computation, and then $T = T(N)$ is a self-simulating tile set. Each valid tiling over T is divided into macrotiles of level 1, which form macrotiles of level 2, which form macrotiles of level 3, and so on. In particular, the tile set is aperiodic. In the standard terminology, a level- k macrotile is a *child* of the level- $(k + 1)$ macrotile that contains it.

The simulation function $\alpha : T \rightarrow T^{N \times N}$ can be seen as a two-dimensional *substitution*. We can iterate it to obtain substitutions $\alpha^k : T \rightarrow T^{N^k \times N^k}$ for $k \geq 0$, and define the *substitutive subshift* $X_\alpha \subset T^{\mathbb{Z}^2}$ by forbidding all finite patterns that do not occur in any $\alpha^k(t)$ for $k \geq 0$ and $t \in T$. The set of valid tilings over T contains X_α , and T can quite easily be modified to guarantee that these subshifts are equal. In [7], Durand and Romashchenko presented a variant of the construction in which α is *primitive*, meaning that for some $k \geq 0$, every $t \in T$ occurs in $\alpha^k(s)$ for every $s \in T$. In this case X_α is a minimal subshift. We will discuss their results more thoroughly in Section 4.2.

3.2 Cellular automata

The main idea of the fixed point construction for one-dimensional cellular automata is the same as for tile sets, and in fact the “good” spacetime diagrams of a fixed point CA have a similar substitutive structure to those of fixed point tilings, except that the width and height of the macrotiles are distinct. The usual terminology is somewhat different, as is the implementation of information transfer between macrotiles. As before, we first construct a CA $\phi : A^{\mathbb{Z}} \rightarrow A^{\mathbb{Z}}$ that simulates a given CA $\psi : B^{\mathbb{Z}} \rightarrow B^{\mathbb{Z}}$. The notion of simulation has long been used informally in CA literature, but for the sake of being exact, let us use a version of *injective bulking* as studied in [5].

► **Definition 2.** A cellular automaton $\phi : A^{\mathbb{Z}} \rightarrow A^{\mathbb{Z}}$ simulates another CA $\psi : B^{\mathbb{Z}} \rightarrow B^{\mathbb{Z}}$ if there exist integers $Q, U \geq 1$ and $s \in [0, Q - 1]$ and an injection $\alpha : B \rightarrow A^Q$ such that $\phi^U(\tau^s(\alpha(x))) = \alpha(\psi(x))$ for all $x \in B^{\mathbb{Z}}$. Here $\alpha(x) \in A^{\mathbb{Z}}$ is defined by concatenating the blocks $\alpha(x_i)$ for $i \in \mathbb{Z}$.

We assume that ψ has neighborhood $N = \{-1, 0, 1\}$ and its state set consists of binary words of constant length k , and construct ϕ so that it also has neighborhood N . The simulation parameters $Q < U$ are called the *colony size* and *work period*, and we choose $s = 0$. The work period U is typically vastly larger than Q , but it is possible to implement the CA so that Q/U is arbitrarily close to 1. Every state of ϕ consists of an element of $[0, Q - 1] \times [0, U - 1]$

called its *address* and *age*² plus some additional data. Say that a configuration $x \in A^{\mathbb{Z}}$ is *valid at* $i \in \mathbb{Z}$ if the states $x_{i-1}x_i x_{i+1}$ have addresses $j, j+1, j+2 \pmod{Q}$ for some j and their ages are equal. We only define the prototypical CA ϕ on configurations that are valid everywhere. In this case, the address of a cell never changes and its age increases by 1 modulo U on every time step.

A *colony* is a sequence of Q adjacent cells with addresses $0, 1, 2, \dots, Q-1$ and equal ages. Each cell of a colony stores a *simulation bit*, and the k leftmost bits form the simulated state of the colony. The “life cycle” of a colony, usually called a *work period* when it cannot be confused with U , begins at age 0 and ends at age $U-1$. At the first step of the work period when each cell has age 0, the simulation bits of the cells are copied onto two separate tracks called the *left and right mailboxes*. If the age of a cell is between 1 and Q , it copies the contents of these tracks from its left and right neighbors. Otherwise they remain unchanged. In this way, during the first $Q+1$ steps of its life cycle every colony transmits its simulation bits to its nearest neighbors. The details of this process vary between implementations. Here we have followed [31] and in particular assumed that ϕ is synchronous and deterministic. If it is asynchronous or subject to local errors, a different scheme is needed to minimize loss of transmitted data.

Each cell of a colony also stores a *program bit*, and these bits together form a program $p \in \{0, 1\}^*$. At age $Q+1$, the leftmost cell of each colony initializes the computation of the efficient universal Turing machine M_U . The simulated head of M_U is usually called the *agent*. The agent scans the program bits, simulation bits and mailboxes of the colony, computes a new simulated state $w \in \{0, 1\}^k$ according to the program p , and writes w onto the simulation bits of the k leftmost cells. We assume that the agent never steps outside the colony and halts before the work period ends at age $U-1$. Then the age of the colony resets to 0 and its life cycle begins anew.

As in the previous section, the CA ϕ simulates ψ as long as Q and U are large enough to satisfy the time and space requirements of M_U . A suitable choice of the parameters allows self-simulation, meaning that we can choose $\phi = \psi$. Note that a correct simulation hierarchy requires a program p_ϕ implementing ϕ to be stored in every colony of the initial configuration, on every simulation level. On the first level this can be enforced by restricting the state set of ϕ analogously to the program condition of the tile set construction, and then M_U can check whether the next simulation level uses an identical program. It depends entirely on the application how the CA behaves if the programs do not match, or if the simulation structure is invalid in some other way: it can try to correct the error, produce a special error state, or have only a partially defined local rule.

3.3 Variable zoom factor and communication between simulation levels

The construction is flexible enough that instead of having the tile set or the CA simulate itself, we can modify the simulated system depending on the simulation level, resulting in a sequence of tile sets $(T_k)_{k \geq 0}$ or automata $(\phi_k)_{k \geq 0}$. The motivation is that the tile set or CA has some desirable property $P(N)$ that is limited by the zoom factor N , and by increasing the zoom factor on consecutive simulation levels it may be possible to obtain $P(N)$ for all $N \in \mathbb{N}$. A common use case is that $P(N)$ has a computational flavor, such as tiling the plane only if a fixed Turing machine halts within N steps.

² Notice the difference in terminology to the tile set construction, where *address* referred to a two-dimensional vector.

We discuss the implementation only in the case of tile sets, as the same ideas apply to the CA construction. Our goal is then to define a sequence $(T_k, N_k)_{k \geq 0}$ of tile sets and numbers such that each tile set T_k simulates T_{k+1} with zoom factor N_k , and we would like to have as much freedom as possible in choosing the latter. For this, we modify the construction of Section 3.1 as follows. The program p_T has an additional integer parameter: $p_T(k, w)$ determines if w encodes a tile of T_k . The program condition is extended so that in addition to the program p_T , each tile of T_0 stores the number 0. For each $k \geq 0$, the tile set T_k implements a variant of the program condition where each tile of T_{k+1} stores p_T and the number $k + 1$. This ensures that all macrotiles have access to their simulation level.

The constraints on the zoom factors are slightly different than in the uniform case. First, both N_{k+1} and T_{k+1} should be computable from k in time and space $O(N_k)$. Second, $N_k = \Omega(\log N_{k+1} + \log k)$ since the binary representation of any address in $[0, N_{k+1} - 1]^2$ and the simulation level k should fit in any level- k macrotile. To get a sense of the class of allowed sequences, $N_k = \lceil \log k \rceil$ for large enough k grows too slowly, while $N_{k+1} = 2^{N_k}$ grows too fast. The choices $N_k = \Theta(k^a)$ for $a \in \mathbb{Q}_{>0}$, $N_k = \Theta(2^k)$ and $N_k = \Theta(2^{2^k})$ are all acceptable.

In most applications, each macrotile stores some amount of additional information not listed above, and there are additional relations between the data stored by macrotiles of consecutive levels. In some sense, a macrotile can “communicate” with its children. The technical details vary, but the main idea is that each level- k macrotile contains a special field in its state set which the simulated machine M_U of its parent is allowed to access. The parent may thus read an arbitrary “message” from a subset of macrotiles on the bottom row of the computation zone. The children can synchronize this message among themselves using local rules, so that it is visible to all of them or some convenient subset. In the CA context, the agent of a colony can read and modify any data that was stored in the children during the last work period, so inter-level communication is slightly easier to implement.

4 History and applications

4.1 Fault-tolerant cellular automata

The fixed point construction was first introduced in the context of *probabilistic cellular automata* (PCA). We give some basic definitions; see [26] for a more thorough review on the subject. A d -dimensional PCA ϕ on $A^{\mathbb{Z}^d}$ consists of a neighborhood $N \subset \mathbb{Z}^d$ and a local rule $\Phi : A^N \rightarrow \mathbb{P}(A)$, where $\mathbb{P}(A)$ is the set of probability distributions over A . The local rule is applied to every coordinate of a configuration $x \in A^{\mathbb{Z}^d}$ as in the case of deterministic CA, but the result is the probability distribution $\phi(x) \in \mathbb{P}(A^{\mathbb{Z}^d})$ where the value of each cell $\phi(x)_{\vec{v}}$ is drawn independently from $\Phi((\tau^{\vec{v}}x)|_N)$. In this way ϕ lifts to a function $\phi : \mathbb{P}(A^{\mathbb{Z}^d}) \rightarrow \mathbb{P}(A^{\mathbb{Z}^d})$ on distributions, denoted $\mu \mapsto \mu\phi$. Every PCA has at least one fixed point, called an *invariant measure*. If ϕ has a unique invariant measure μ and for every other measure ν the sequence $(\nu\phi^n)_{n \in \mathbb{N}}$ converges weakly to μ , then ϕ is *ergodic*. Intuitively, an ergodic PCA gradually forgets all details about its initial state. We say ϕ has *positive rates* if the probability of $\Phi(P) = a$ is positive for all $P \in A^N$ and $a \in A$. In other words, every local transformation occurs under ϕ with positive (but possibly very small) probability. The *positive rates conjecture* states that if a one-dimensional PCA has positive rates, then it must be ergodic.

In 1986 (and with an expanded paper in 2001), Gács disproved the conjecture by an intricate construction of a PCA that behaves like a deterministic CA except that at each time step, each cell has an extremely small but positive probability of “making an error” and ending up in an arbitrary state [13, 14]. The design is based on a short paper of Kurdyumov from 1978 that likewise claims to refute the conjecture, but lacks detail [24].

► **Theorem 3** (Gács). *There exists a nonergodic 1D PCA with positive rates.*

The construction uses self-simulation to recognize and correct clusters of errors that occur at various scales. Non-ergodicity is achieved by defining the state set as a product $\{0, 1\} \times A$, initializing each cell in some state $x_i \in \{b\} \times A$ for the same bit b , and “remembering” the bit in the sense that the probability of seeing the opposite bit $1 - b$ in any given cell at any given time is low.³ If the error rate is small enough, an argument from percolation theory shows that errors will almost surely occur in semi-isolated “bursts” that fit in finite (but unbounded) space-time rectangles. At the lowest simulation level, each cell contains information about its own simulation values (address, age. . .) and the bit to remember, as well as those of its nearest neighbors, and isolated individual errors can be fixed in one time step by a majority vote between three cells. Larger bursts of errors are detected and repaired by higher-level colonies. One of the main difficulties is that long sequences of errors can produce a segment of states that resembles a valid simulation structure. Its incompatibility with the global hierarchy is only visible on its endpoints, where it cannot be determined locally which side belongs to the erroneous segment. See Gray’s condensed version of the construction for more (but not nearly all) technical details [16]. Gács and Çapuni have applied the same ideas to fault-tolerant Turing machines [4].

4.2 Realizations of sofic shifts

In [20], Hochman proved that d -dimensional effectively closed subshifts can be realized as the *projective sybdynamics* of $(d + 2)$ -dimensional sofic shifts, that is, sets of d -dimensional hyperplanes occurring in them. The construction uses the older result of Mozes that two-dimensional substitutive subshifts are sofic [27] to divide each configuration into regions of increasing size containing simulations of Turing machines. The dimension was reduced from $d + 2$ to $d + 1$ independently by Aubrun and Sablik in [1] with Robinson tiles [28], and by Durand, Romashchenko and Shen in [12] using fixed point tilings. It has since been used as a “black box” in numerous other constructions. Incidentally, the realization result of Mozes was also reproved with a fixed point construction in [12].

► **Theorem 4** (Aubrun & Sablik; Durand & Romashchenko & Shen). *Let $X \subseteq A^{\mathbb{Z}^d}$ be an effectively closed subshift. Then the higher-dimensional subshift $Y = \{y \in A^{\mathbb{Z}^{d+1}} \mid \exists x \in X : \forall \vec{w} \in \mathbb{Z}^{d+1} : y_{\vec{w}} = x_{\pi_d(\vec{w})}\}$ is sofic, where $\pi_d : \mathbb{Z}^{d+1} \rightarrow \mathbb{Z}^d$ is defined by $\pi_d(i_1, \dots, i_{d+1}) = (i_1, \dots, i_d)$.*

The article [12] is a culmination of a series of shorter papers by the same authors. It presents the prototypical fixed point construction, collects their previous results into one place, and extends some of them. In their proof of Theorem 4 (in the case $d = 1$), there is a separate layer of A -tiles in which each vertical stripe is tiled with copies of a single A -symbol. The simulations have zoom factors $N_k = Q^{c^k}$ for some large constant Q and $c > 2$. A macrotile of level k with address (x, y) , where $y < \prod_{i=0}^k N_i$, is tasked with retrieving and storing the symbols of $f(k)$ consecutive columns of the A -layer situated y steps away from its left border, where $f : \mathbb{N} \rightarrow \mathbb{N}$ is a very slowly growing function. The definition of N_k ensures that $N_{k+1} \ll \prod_{i=0}^k N_i$, so that in a level- $(k + 1)$ macrotile t , every length- $f(k)$ word of the A -layer occurring within it (or right at the border between it and its neighbor) is stored in at least one child. Using a series of inter-level messages, t is in turn able to retrieve

³ In fact, the construction can remember an infinite sequence of bits.

its designated length- $f(k+1)$ word. It also computes a prefix of the sequence of forbidden patterns that defines X and verifies that none of them occur in the word it retrieved. As a direct application, the authors reprove the main result of [6]: there exists a tile set where the Kolmogorov complexity of every $N \times N$ pattern that occurs in a tiling is $\Omega(N)$, which is asymptotically optimal for an SFT. Namely, we can define $X \subseteq \{0, 1\}^{\mathbb{Z}^2}$ by forbidding all words w whose Kolmogorov complexity is below $a|w| - b$ for suitable constants $a, b > 0$ and apply Theorem 4.

Some dynamical properties of the subshift X in Theorem 4 can be lifted to the SFT cover Z of Y . One of the main results of Durand and Romashchenko in [7] states that if X is minimal or quasiperiodic, then the same property can be imposed on Z . They modify the construction so that every macrotile contains a zone of “diversification slots” where all valid 2×2 patterns of tiles are manually forced to appear. In the minimal case, there is the additional complication that if the positions of the macrotile grids and the layer of A -tiles in the SFT cover can be independent, the result is generally not minimal. This is overcome by choosing a canonical configuration from each and implementing the minimal shift generated by their combination. In the same article, the authors characterize the Turing degree spectra of quasiperiodic SFTs, which are exactly the upper closures of effectively closed subsets of $\{0, 1\}^{\mathbb{N}}$ (see [7, Theorem 4] for details). In particular, there exist nonempty quasiperiodic SFTs with only noncomputable configurations.

In [33], Westrick proves two sofic realization results that utilize the fixed point construction. They concern two-dimensional binary subshifts where every connected component of 1-cells is a (possibly degenerate) square.

- **Theorem 5 (Westrick).** *Let $X_{\text{sq}} \subseteq \{0, 1\}^{\mathbb{Z}^2}$ be the 2D subshift where every connected component of 1-cells is either a finite square, a quarter-plane, a half-plane, or the entire \mathbb{Z}^2 .*
- (a) *Every effectively closed subshift $X \subseteq X_{\text{sq}}$ that forbids all pairs of distinct squares of identical size (and possibly other patterns) is sofic.*
 - (b) *For $S \subseteq \mathbb{N}$, let $X_S \subseteq X_{\text{sq}}$ be the subshift that forbids the $n \times n$ squares for all $n \in \mathbb{N} \setminus S$. If S is computably enumerable, then X_S is sofic.*

We only sketch the construction proving the first result, as the other is quite similar. The subshift X_{sq} is easily seen to be sofic, so we focus on the additional constraints of X . Westrick uses the variable zoom factors $N_k = 2^{2^k}$. In addition to the simulation structure, each macrotile t stores the following information: first, the size and relative position of each finite square of 1-cells that has at least one side completely inside t or one of its neighbors; second, the orientation and relative position of each corner and horizontal or vertical boundary of 1-cells in t or its neighbors that if not part of an already stored finite square; third, auxiliary bits that help enforce the consistency of this information between t and its parent.

If the sizes of all squares are distinct, then each level- k macrotile can contain at most $O(L_k^{2/3})$ squares, where $L_k = \prod_{i=0}^k N_i$ is its side length, and their sizes and positions can be encoded in $O(L_k^{2/3} \log L_k)$ bits. This data is small enough to fit in the macrotile, but transporting it from its children to the computation zone is nontrivial, since it is too large to be simply broadcast to each child. Westrick uses a graph theoretical argument to show that one can route the information in a nondeterministic way. Additionally, the universal Turing machine M_U needs to be replaced by a multi-tape machine to speed up the computation of the relative offsets of the squares. As in Theorem 4, the number of additional time steps used for computing the forbidden patterns of X is increased at each simulation level. The patterns formed by small squares are checked first, so any forbidden pattern will be eventually discovered and leads to a tiling error.

4.3 Robust tilings

In addition to Theorem 4 and related realization results, the paper [12] presents tile sets that are *robust with respect to random errors*, meaning that a configuration that is locally valid on a large subset of \mathbb{Z}^2 is guaranteed to be equal to a single valid tiling in a large⁴ subset of \mathbb{Z}^2 . In fact, the main result is the construction of a tile set that combines robustness with the aforementioned property that every tiling is algorithmically complex.

► **Definition 6.** Let T be a set of Wang tiles and $E \subseteq \mathbb{Z}^2$. A (T, E) -tiling is a configuration $x \in T^{\mathbb{Z}^2}$ such that for all neighboring coordinates $\vec{v}, \vec{w} \in \mathbb{Z}^2 \setminus E$, the colors at the common border of $x_{\vec{v}}$ and $x_{\vec{w}}$ match. The Bernoulli distribution on subsets of \mathbb{Z}^2 with $\mathbb{P}(\vec{v} \in E) = \epsilon$ for each \vec{v} is denoted B_ϵ .

► **Theorem 7** (Durand & Romashchenko & Shen). *There exists a tile set T that tiles the plane and $a, b > 0$ with the following properties. In every valid $x \in T^{\mathbb{Z}^2}$, every $n \times n$ square has Kolmogorov complexity at least $an - b$. For small enough $\epsilon > 0$, for B_ϵ -almost every $E \subseteq \mathbb{Z}^2$, for each (T, E) -tiling x , the Kolmogorov complexity of $x_{[-n, n]^2}$ is $\Omega(n)$, and there is a valid tiling $y \in T^{\mathbb{Z}^2}$ with $\limsup_n |\{\vec{v} \in [-n, n]^2 \mid x_{\vec{v}} \neq y_{\vec{v}}\}| / (2n + 1)^2 \leq 1/10$.*

The construction of robust self-simulating tile sets turns out to be much simpler than the error-correcting CA discussed in Section 4.1. Similarly to that construction, if ϵ is small enough, then almost all error sets E can be decomposed into isolated finite subsets that can be handled separately. It is then enough to modify the prototypical fixed point construction so that every locally valid pattern shaped like the 5×5 annulus $[-2, 2]^2 \setminus [-1, 1]^2$ can be extended into a valid 5×5 rectangle in a unique way. Namely, this property is inherited by macrotiles of all simulation levels, which can then correct arbitrarily large (but finite) incorrect patterns. The additional condition on the Kolmogorov complexity of (T, E) -tilings requires a little more machinery, since the uniformity of the vertical columns that store the bits of the high-complexity shift X can be broken by the error set E . This is remedied by duplicating the information in several columns of macrotiles and periodically checking it against an error-correcting code.

4.4 Unique ergodicity

The article [31] by Törmä is an application of the fixed point construction to the dynamics of cellular automata. It concerns a weakening of the notion of *nilpotency*.

► **Definition 8.** A CA $\phi : A^{\mathbb{Z}^d} \rightarrow A^{\mathbb{Z}^d}$ with a quiescent state $0 \in A$ is nilpotent if there exists $n \in \mathbb{N}$ such that $\phi^n(A^{\mathbb{Z}^d}) = \{0^{\mathbb{Z}^d}\}$. It is asymptotically nilpotent if for each $x \in A^{\mathbb{Z}^d}$ we have $\phi^n(x) \rightarrow 0^{\mathbb{Z}^d}$ in the product topology as $n \rightarrow \infty$. It is uniquely ergodic if it has a unique invariant measure.

A nilpotent CA sends every initial configuration into the same uniform “sink” in a bounded number of steps. Asymptotic nilpotency means that for every initial configuration, each cell will be in a nonzero state on only a finite (but not necessarily uniform) number of time steps. It was shown in [17] that asymptotically nilpotent one-dimensional CA are in fact nilpotent, and the result was generalized in [29, 30]. As for unique ergodicity, note that since 0 is quiescent, the unique invariant measure cannot be any other than the Dirac

⁴ Large in a slightly different sense: the first subset comes from a measure-1 set with respect to a biased Bernoulli distribution, the second has asymptotic density close to 1.

point measure concentrated on $0^{\mathbb{Z}^d}$. Then by classical results of ergodic theory, ϕ is uniquely ergodic if and only if for every initial configuration, the asymptotic proportion of time steps that any given cell spends in a nonzero state equals zero. The main result of [31] states that even one-dimensional uniquely ergodic CA are not necessarily nilpotent.

► **Theorem 9** (Törmä). *There exists a uniquely ergodic CA $\phi : A^{\mathbb{Z}} \rightarrow A^{\mathbb{Z}}$ that has a quiescent state and is not nilpotent.*

This CA is defined by a fixed point construction with variable zoom factors. The alphabet of the CA ϕ_k on each simulation level k contains a designated “blank” state B_k , with $B_0 = 0$. Each cell of a level- k colony will spend a proportion p_k of its work period simulating the state B_k , with $p_k \rightarrow 1$ as $k \rightarrow \infty$. If a colony and both of its neighbors simulate the state B_k , it is temporarily replaced by a segment of 0-cells. Any local inconsistency in the simulation structure causes all nearby colonies to assume the blank state. These properties ensure that every locally correct simulation structure is asymptotically infinitely sparse, and locally incorrect structures are quickly destroyed. On the other hand, since a carefully chosen initial condition gives rise to an infinite nested simulation, the CA is not nilpotent.

4.5 Realization of topological entropy

The *topological entropy* of a topological dynamical system measures its complexity and unpredictability. In the case of subshifts and cellular automata, it can be characterized as a limit of counting patterns of increasing sizes that occur in the valid configurations and spacetime diagrams.

► **Definition 10.** *The topological entropy of a subshift $X \subseteq A^{\mathbb{Z}^d}$ is the limit $h(X) = \lim_{n \rightarrow \infty} n^{-d} \log |\{x_{[0, n-1]^d} \mid x \in X\}|$. The topological entropy of a CA ϕ on $A^{\mathbb{Z}^d}$ is $h(\phi) = \lim_{r \rightarrow \infty} \lim_{n \rightarrow \infty} n^{-1} \log |\{(\phi^t(x)_{[0, r-1]^d})_{t=0}^{n-1} \mid x \in A^{\mathbb{Z}^d}\}|$.*

By classical results in symbolic dynamics (see Sections 4 and 11 of [25]), the entropy of a one-dimensional SFT is efficiently computable and its possible values are exactly the nonnegative integer multiples of logarithms of Perron numbers. The entropies of two- and higher-dimensional SFTs were characterized by Hochman in [22] as the nonnegative right-computable numbers, that is, limits of computable decreasing sequences of rational numbers. Independently in the preprint [15] using a variant of Robinson tiles and in [7] using fixed point tile sets, it was shown that they can be realized with transitive SFTs.

► **Theorem 11** (Gangloff & Sablik; Durand & Romashchenko). *The topological entropies of transitive 2D SFTs are exactly the nonnegative right-computable numbers.*

The topological entropies of three- and higher-dimensional CA were characterized in [20]. Guillon and Zinoviadis handled the remaining dimensions, 1 and 2, in [19]. Their proof uses the fixed point construction.

► **Theorem 12** (Hochman; Guillon & Zinoviadis). *The topological entropies of one-dimensional CA are exactly the nonnegative right-computable numbers, and those of two- and higher-dimensional CA are exactly the limits of increasing computable sequences of nonnegative right-computable numbers (including ∞).*

The high-level idea in each construction is the same. In the SFT case, given a number $\alpha \geq 0$ in the appropriate class, construct an SFT that factors onto a subshift $X \subseteq \{0, 1\}^{\mathbb{Z}^d}$ where the maximal asymptotic density of 1-symbols in a configuration equals α , and then

allow any subset of 1s to be independently replaced by 2s. In the case of cellular automata, a d -dimensional self-simulating CA with variable zoom factor is used to analyze a separate binary layer of the configuration, which is not modified by the CA except when erased by the error state. The CA produces a spreading error state if the local density of 1-symbols is too high, which guarantees that such patterns are transient and thus do not contribute to the entropy. This density is converted into entropy by replacing an arbitrary subset of 1s by 2s and composing the CA with a shift.

4.6 Expansive directions

Subshifts are *expansive* dynamical systems: a configuration $x \in X \subseteq A^{\mathbb{Z}^d}$ can be reconstructed from any family of good enough approximations of the shifts $\tau^{\vec{v}}x$ for $\vec{v} \in \mathbb{Z}^d$. Boyle and Lind studied a stronger variant called *directional expansivity*, where a certain subset of the shifts suffices to determine x , in [3]. For convenience, we define it only in the two-dimensional case.

► **Definition 13.** Let $\vec{0} \in L \subset \mathbb{R}^2$ be a line through the origin with direction (slope) $\ell \in \dot{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$. We say ℓ is expansive for a subshift $X \subset A^{\mathbb{Z}^2}$, if there exists $r \geq 0$ such that the function $x \mapsto x|_D$ is injective on X , where $D = (L + [-r, r]^2) \cap \mathbb{Z}^2$. Denote the nonexpansive directions by $N(X) \subseteq \dot{\mathbb{R}}$.

For any infinite two-dimensional subshift X , the set $N(X)$ is nonempty and closed. Boyle, Lind and Hochman proved in [3, 21] that every nonempty closed subset $L \subseteq \dot{\mathbb{R}}$ occurs as $N(X)$ for some subshift X . The subshifts constructed by Boyle and Lind consist of two layers, each of which contains regularly spaced discrete approximation of lines, and it realizes every L except singleton irrational directions. Hochman handled the remaining case with a subshift resembling the set of spacetime diagrams of a fixed point CA with variable zoom factor, but without the simulated Turing machines. Since X is not required to be of finite type, the hierarchical structure can be enforced. Building on Hochman's proof, Guillon and Zinoviadis have characterized the sets of nonexpansive directions of SFTs in an unpublished manuscript [18], which also contains several realization results of SFTs with a unique nonexpansive direction. The results are reported in Zinoviadis's PhD dissertation [34]. In the next result, a subset $A \subseteq \dot{\mathbb{R}}$ is *effectively closed* if there exists a computable sequence $(I_n)_{n \in \mathbb{N}}$ of open intervals (which may have the form $(a, \infty) \cup (-\infty, b)$) with rational endpoints such that $\dot{\mathbb{R}} \setminus A = \bigcup_{n \in \mathbb{N}} I_n$.

► **Theorem 14** (Guillon & Zinoviadis). *The sets $N(X)$ for two-dimensional SFTs X are exactly the effectively closed subsets of $\dot{\mathbb{R}}$.*

The main technical contribution of the dissertation is a fixed point construction for reversible partial cellular automata (RPCA), which are defined by a local rule that is a partial function. Note that the fixed point CA presented in Section 3.2 is not reversible. The set of spacetime diagrams of a one-dimensional RPCA is a two-dimensional SFT for which all directions $-1 < \ell < 1$ are expansive. For an RPCA obtained by the fixed point construction with variable zoom factor, if $\prod_{k=0}^{\infty} Q_k/U_k = 0$, then every direction except ∞ (the vertical line) is expansive. A shift map by $D_k \in \mathbb{Z}$ cells is then applied to each level k of the simulation hierarchy, which changes the direction of the unique nonexpansive line. These numbers and the simulation parameters Q_k, U_k are allowed to vary within a suitable set of sequences computed by the simulated Turing machines, which provides enough control on the nonexpansive directions to realize any effectively closed set.

References

- 1 Nathalie Aubrun and Mathieu Sablik. Simulation of effective subshifts by two-dimensional subshifts of finite type. *Acta Applicandae Mathematicae*, 126:35–63, 2013. doi:10.1007/s10440-013-9808-5.
- 2 Robert Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 66, 1966. 72 pages.
- 3 Mike Boyle and Douglas Lind. Expansive subdynamics. *Transactions of the American Mathematical Society*, 349(1):55–102, 1997. doi:10.1090/S0002-9947-97-01634-6.
- 4 Ilir Çapuni and Péter Gács. A Turing machine resisting isolated bursts of faults. *Chicago Journal of Theoretical Computer Science*, 2013(3), 2013. doi:10.4086/cjtc.2013.003.
- 5 M. Delorme, J. Mazoyer, N. Ollinger, and G. Theyssier. Bulking I: an abstract theory of bulking. *Theoretical Computer Science*, 412(30):3866–3880, 2011. doi:10.1016/j.tcs.2011.02.023.
- 6 Bruno Durand, Leonid A. Levin, and Alexander Shen. Complex tilings. *The Journal of Symbolic Logic*, 73(2):593–613, 2008. doi:10.2178/jsl/1208359062.
- 7 Bruno Durand and Andrei Romashchenko. The expressiveness of quasiperiodic and minimal shifts of finite type. *Ergodic Theory and Dynamical Systems*, 41(4):1086–1138, 2021. doi:10.1017/etds.2019.112.
- 8 Bruno Durand, Andrei Romashchenko, and Alexander Shen. Fixed point and aperiodic tilings. In *Developments in language theory*, volume 5257 of *Lecture Notes in Computer Science*, pages 276–288. Springer, Berlin, 2008. doi:10.1007/978-3-540-85780-8_22.
- 9 Bruno Durand, Andrei Romashchenko, and Alexander Shen. Fixed point theorem and aperiodic tilings. *Bulletin of the European Association for Theoretical Computer Science. EATCS*, 97:126–136, 2009.
- 10 Bruno Durand, Andrei Romashchenko, and Alexander Shen. High complexity tilings with sparse errors. In *Automata, languages and programming. Part I*, volume 5555 of *Lecture Notes in Computer Science*, pages 403–414. Springer, Berlin, 2009. doi:10.1007/978-3-642-02927-1_34.
- 11 Bruno Durand, Andrei Romashchenko, and Alexander Shen. Effective closed subshifts in 1D can be implemented in 2D. In *Fields of logic and computation*, volume 6300 of *Lecture Notes in Computer Science*, pages 208–226. Springer, Berlin, 2010. doi:10.1007/978-3-642-15025-8_12.
- 12 Bruno Durand, Andrei Romashchenko, and Alexander Shen. Fixed-point tile sets and their applications. *Journal of Computer and System Sciences*, 78(3):731–764, 2012. doi:10.1016/j.jcss.2011.11.001.
- 13 Péter Gács. Reliable computation with cellular automata. *Journal of Computer and System Sciences*, 32(1):15–78, 1986. doi:10.1016/0022-0000(86)90002-4.
- 14 Péter Gács. Reliable cellular automata with self-organization. *Journal of Statistical Physics*, 103(1-2):45–267, 2001. doi:10.1023/A:1004823720305.
- 15 Silvère Gangloff and Mathieu Sablik. Quantified block gluing, aperiodicity and entropy of multidimensional SFT. *ArXiv e-prints*, 2017. arXiv:1706.01627.
- 16 Lawrence F. Gray. A reader’s guide to Gacs’s “positive rates” paper. *Journal of Statistical Physics*, 103(1-2):1–44, 2001. doi:10.1023/A:1004824203467.
- 17 Pierre Guillon and Gaétan Richard. Nilpotency and limit sets of cellular automata. In *Mathematical foundations of computer science 2008*, volume 5162 of *Lecture Notes in Comput. Sci.*, pages 375–386. Springer, Berlin, 2008. doi:10.1007/978-3-540-85238-4_30.
- 18 Pierre Guillon and Charalampos Zinoviadis. Unpublished manuscript. Work in progress.
- 19 Pierre Guillon and Charalampos Zinoviadis. Densities and entropies in cellular automata. In S. Barry Cooper, Anuj Dawar, and Benedict Löwe, editors, *How the world computes*, pages 253–263. Springer, 2013. doi:10.1007/978-3-642-30870-3_26.
- 20 Michael Hochman. On the dynamics and recursive properties of multidimensional symbolic systems. *Inventiones Mathematicae*, 176(1):131–167, 2009. doi:10.1007/s00222-008-0161-7.

- 21 Michael Hochman. Non-expansive directions for \mathbb{Z}^2 actions. *Ergodic Theory and Dynamical Systems*, 31(1):91–112, 2011. doi:10.1017/S0143385709001084.
- 22 Michael Hochman and Tom Meyerovitch. A characterization of the entropies of multidimensional shifts of finite type. *Annals of Mathematics. Second Series*, 171(3):2011–2038, 2010. doi:10.4007/annals.2010.171.2011.
- 23 Jarkko Kari. Theory of cellular automata: a survey. *Theoretical Computer Science*, 334(1-3):3–33, 2005. doi:10.1016/j.tcs.2004.11.021.
- 24 G. L. Kurdyumov. An example of a nonergodic one-dimensional homogeneous random medium with positive transition probabilities. *Doklady Akademii Nauk SSSR*, 238(6):1287–1290, 1978. URL: <http://mi.mathnet.ru/eng/dan41542>.
- 25 Douglas Lind and Brian Marcus. *An introduction to symbolic dynamics and coding*. Cambridge University Press, Cambridge, 1995. doi:10.1017/CB09780511626302.
- 26 Jean Mairesse and Irène Marcovici. Around probabilistic cellular automata. *Theoretical Computer Science*, 559:42–72, 2014. doi:10.1016/j.tcs.2014.09.009.
- 27 Shahar Mozes. Tilings, substitution systems and dynamical systems generated by them. *Journal d'Analyse Mathématique*, 53:139–186, 1989. doi:10.1007/BF02793412.
- 28 Raphael M. Robinson. Undecidability and nonperiodicity for tilings of the plane. *Inventiones Mathematicae*, 12:177–209, 1971. doi:10.1007/BF01418780.
- 29 Ville Salo. On nilpotency and asymptotic nilpotency of cellular automata. In Enrico Formenti, editor, Proceedings 18th international workshop on *Cellular Automata and Discrete Complex Systems* and 3rd international symposium *Journées Automates Cellulaires*, La Marana, Corsica, September 19-21, 2012, volume 90 of *Electronic Proceedings in Theoretical Computer Science*, pages 86–96. Open Publishing Association, 2012. doi:10.4204/EPTCS.90.7.
- 30 Ville Salo and Ilkka Törmä. Nilpotent endomorphisms of expansive group actions. *International Journal of Algebra and Computation*, 2021. Published online. doi:10.1142/S021819672150020X.
- 31 Ilkka Törmä. A uniquely ergodic cellular automaton. *Journal of Computer and System Sciences*, 81(2):415–442, 2015. doi:10.1016/j.jcss.2014.10.001.
- 32 Hao Wang. Proving theorems by pattern recognition II. *Bell System Technical Journal*, 40:1–42, 1961. doi:10.1002/j.1538-7305.1961.tb03975.x.
- 33 Linda Brown Westrick. Seas of squares with sizes from a Π_1^0 set. *Israel Journal of Mathematics*, 222(1):431–462, 2017. doi:10.1007/s11856-017-1596-6.
- 34 Charalampos Zinoviadis. *Hierarchy and Expansiveness in Two-Dimensional Subshifts of Finite Type*. PhD thesis, University of Turku, 2016. URL: <http://urn.fi/URN:ISBN:%20978-952-12-3337-1>.