



HAL
open science

Resource-Bounded Kolmogorov Complexity Provides an Obstacle to Soficness of Multidimensional Shifts

Julien Destombes, Andrei Romashchenko

► **To cite this version:**

Julien Destombes, Andrei Romashchenko. Resource-Bounded Kolmogorov Complexity Provides an Obstacle to Soficness of Multidimensional Shifts. *Journal of Computer and System Sciences*, 2022, 128, pp.107-134. 10.1016/j.jcss.2022.04.002 . lirmm-03646717

HAL Id: lirmm-03646717

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-03646717v1>

Submitted on 11 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Resource-Bounded Kolmogorov Complexity Provides an Obstacle to Soficness of Multidimensional Shifts

Julien Destombes

Andrei Romashchenko

May 24, 2022

Abstract

We suggest necessary conditions for soficness of multidimensional shifts formulated in terms of resource-bounded Kolmogorov complexity. Using this technique we provide examples of effective and non-sofic shifts on \mathbb{Z}^2 with very low block complexity: the number of globally admissible patterns of size $n \times n$ grows only as a polynomial in n . We also show that more conventional proofs of non-soficness for multi-dimensional effective shifts, including the techniques of Pavlov [Proc. AMS, 141(3):987-996, 2013] and Kass and Madden [Proc. AMS, 141(11):3803-3816, 2013], can be expressed in terms of Kolmogorov complexity with unbounded computational resources.

1 Introduction

Symbolic dynamics originally appeared in mathematics as a branch of the theory of dynamical systems that studies smooth or topological dynamical systems by discretizing the underlying space. Since the late 1930s, symbolic dynamics became an independent field of research, see [10, 11]. A classical *dynamical system* is a space (of states) \mathcal{S} with a function F acting on this space; this function represents the “evolution rule,” i.e., the time dependence of a configuration in the space. The central notion of the theory of dynamical systems is a trajectory — a sequence of configurations obtained by iterating the evolution rules,

$$x, F(x), F(F(x)), \dots, F^{(n)}(x), \dots$$

In symbolic dynamics the space of states reduces to a finite set (an *alphabet*). The trajectories are represented by infinite (or bi-infinite) sequences of letters over this alphabet, and the “evolution rule” is the shift operator acting on these sequences. Symbolic dynamics focuses on the *shift spaces* — the sets of bi-infinite sequences of letters (over a finite alphabet) that are defined by a shift-invariant constraint on the factors of finite length. More precisely, a *shift space* (also called a *shift*) over an alphabet Σ is a non-empty subset of bi-infinite sequences over Σ that is translation invariant and closed in the standard topology of the Cantor space. Every shift can be defined in terms of *forbidden finite patterns*: we fix a set of (finite) words \mathcal{F} and say that a configuration (a bi-infinite sequence) belongs to the corresponding shift $\mathcal{S}_{\mathcal{F}}$ if and only if it does not contain any factor from \mathcal{F} . We bear in mind that different sets of forbidden patterns can induce the same shift.

Obviously, the properties of shifts heavily depend on the corresponding set of forbidden patterns. The following three large classes of shifts play an important role in symbolic dynamics and computability theory:

- *shifts of finite type* (SFT), which can be defined by a finite set of forbidden finite patterns;
- *sofic* shifts (introduced in [20]), where the set of forbidden finite patterns is a regular language;
- *effective* (or *effectively closed* shifts), which can be defined by a computable¹ set of forbidden finite patterns.

¹We can require that the set of forbidden finite patterns is *decidable* or *recursively enumerable*. Though formally speaking these requirements are different, they result in one and the same class of shifts. That is, if some shift is defined by a recursively enumerable set of forbidden finite patterns, then the same shift can be also defined by a decidable set of forbidden patterns.

The class of effective shifts is often understood as a class of “explicitly constructible” shifts, so it is of considerable interest beyond the scope of computability theory. These three classes are distinct:

$$[\text{the SFTs}] \subsetneq [\text{the sofic shifts}] \subsetneq [\text{the effective shifts}].$$

The sofic shifts can be equivalently defined as the coordinate-wise projections of configurations from an SFT:

Definition 1. A shift \mathcal{S} over an alphabet Σ is *sofic* if there is an SFT \mathcal{S}' over an alphabet Σ' and a mapping $\pi : \Sigma' \rightarrow \Sigma$, such that \mathcal{S} consists of the coordinate-wise projections

$$(\dots \pi(y_{-1})\pi(y_0)\pi(y_1)\pi(y_2)\dots)$$

of all configurations $(\dots y_{-1}y_0y_1y_2\dots)$ from \mathcal{S}' .

There is a simple characterization of soficness. Let us say that two words w_1, w_2 are *equivalent* in a shift \mathcal{S} , if they have the same *follower sets*, i.e., if exactly the same half-infinite configurations occur in \mathcal{S} immediately to the right of w_1 and to the right of w_2 . A shift is sofic if and only if the finite patterns in this shift are subdivided in a finite number of equivalence classes (see [9, Theorem 3.2.10]). Loosely speaking, this criterion guarantees that when we read a configuration from the left to the right and verify that it belongs to a sofic shift, we need to keep in mind only finite information.

The SFTs and even the sofic shifts are rather restrictive classes of shifts with several very special properties. Not surprisingly, many important examples of effective shifts are not sofic. Non-soficness of a shift is usually proved with some version of the pumping lemma from automata theory.

Multidimensional shifts. The formalism of shifts can be naturally extended to the grids \mathbb{Z}^d for $d > 1$. A shift on \mathbb{Z}^d (over a finite alphabet Σ) is defined as a set of d -dimensional configurations $f : \mathbb{Z}^d \rightarrow \Sigma$ that are (i) translation-invariant (under translations in all directions) and (ii) closed in Cantor’s topology. Similar to the one-dimensional case, the shifts can be defined in terms of forbidden finite patterns.

The definitions of the *effective shifts* (the set of forbidden patterns is computable) and of the *SFTs* (the set of forbidden patterns is finite) apply to the multidimensional shift spaces directly, without any revision. The *sofic shifts* on \mathbb{Z}^d are defined as in Definition 1 above (as the coordinate-wise projections of SFTs). We refer the reader to [4] for an extensive discussion of the relations between SFT, sofic, and effective shifts on \mathbb{Z}^d .

For multidimensional shifts spaces, the classes of the effective shifts, the sofic shifts, and the SFTs remain distinct, though the difference between these classes is more elusive than in the one-dimensional case. In this paper we discuss the tools that help to reveal the reasons why one or another effective multidimensional shift is *not* sofic.

The class of sofic shifts in dimension $d \geq 2$ is surprisingly wide. Besides many simple and natural examples, there are shifts whose soficness follow from rather subtle considerations. For instance, S. Mozes showed that the shifts generated by (a natural class of) non deterministic multidimensional substitutions systems are sofic [12]. L. B. Westrick proved that the two-dimensional shift on the alphabet $\{0, 1\}$ whose configurations consist of squares of 1s of pairwise different sizes on a background of 0s, is sofic; moreover, any effectively closed subshift of this shift is also sofic [21].

On the other hand, there are several examples of effective multidimensional shifts that are known to be non-sofic. In what follows we briefly discuss two of them.

Example 1 (the mirror shift). One of the standard examples of a non-sofic shift is the shift of mirror-symmetric configurations on \mathbb{Z}^2 . Let Σ be the alphabet with three letters (e.g., *black*, *white*, and *red*), and the configurations of the shift are all black-and-white configurations (without any red letter) and the configurations with an infinite horizontal line of red letters and symmetric black-and-white half-planes above and below this line, see Fig. 1. We denote this shift by $\mathcal{S}_{\text{mirror}}$.

It is easy to see that this shift is effective (the forbidden patterns are those where the red letters are not aligned, and those where the areas above and below the horizontal red line are not symmetric). At the same time, this shift is not sofic. The intuitive explanation of this fact is as follows. Let us focus on a pair of symmetric patterns of size $n \times n$ in black and white, above and below the horizontal red line (see the blue squares in Fig. 1). To make sure that the configuration belongs to the shift, we must “compare” these two patterns with each other. To this end, we need to transmit the information about a pattern of size n^2 through its border line (of length $O(n)$). However, in a sofic shift, the “information flow” across a

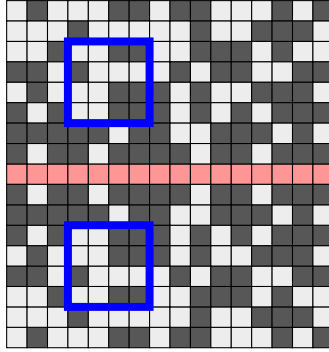


Figure 1: A configuration with mirror symmetry with respect to the horizontal red line. The blue squares select two symmetric black-and-white patterns.

contour of length $O(n)$ is bounded by $O(n)$ bits, and this contradiction implies non-soficness. For a more formal argument see, e.g. [1] and [5], or a similar example [6, Example 2.4]. We revisit this example in Section 4.1 and use it as a simple illustration of our technique.

Example 2 (the high complexity shift). Let \mathcal{S} be the set of all binary configurations on \mathbb{Z}^2 where for each $n \times n$ pattern P its Kolmogorov complexity is quadratic, $C(P) = \Omega(n^2)$. Technically, this means that no globally admissible pattern can be produced by a program of size below cn^2 , for some factor $c > 0$ (see the formal definition of Kolmogorov complexity below).

This shift is not sofic. This follows from two facts (proven in [2]):

- (i) For some $c < 1$, the shift defined above is not empty.
- (ii) In every non-empty sofic shift on \mathbb{Z}^2 , there is a configuration where the Kolmogorov complexity of each $n \times n$ pattern is bounded by $O(n)$.

At the same time, this shift is obviously effective: we can algorithmically enumerate the patterns whose Kolmogorov complexity is below the specified threshold. In Section 3 we study a similar construction based on a less conventional version of Kolmogorov complexity.

Note that the non-sofic shifts in the two examples above have positive entropy (the number of globally admissible patterns of size $n \times n$ grows as $2^{\Omega(n^2)}$). This is not surprising: the proofs of non-soficness of these shifts use the intuition about the information flows (*super-linear amount of information cannot flow through a linear contour*). This type of argument can be adapted for some shifts where the number of globally admissible patterns of size $n \times n$ grows slower than $2^{\Omega(n^2)}$ but still faster than $2^{O(n)}$ (see, e.g. [18, Proposition 15]). As it was noticed in [21], “*all examples known to the author of effectively closed shifts which are not sofic were obtained by in some sense allowing elements to pack too much important information into a small area.*”

This type of argument was formalized as rather general sufficient conditions of non-soficness in [15] and [6]. The theorems by Kass and Madden ([6, Theorem 3.2.10]) and Pavlov ([15, Theorem 1.1]) apply only to the two-dimensional shifts where the number of globally admissible $n \times n$ patterns is greater than $2^{O(n)}$. However, there is no reason to think that this condition is necessary for non-soficness (see, e.g. the discussion in [5, Section 1.2.2]). It is instructive to observe that non-effective non-sofic shifts can have very low block complexity, see [6, 14].

In this paper we propose a new approach to the proof of non-soficness. Our technique uses algorithmic descriptive complexity (Kolmogorov complexity). Though the definition of the sofic shifts does not involve explicitly any resource bounded computations, we show that a suitable tool to deal with soficness is the time-bounded version of Kolmogorov complexity. We argue that a shift on \mathbb{Z}^2 cannot be sofic if the essential information contained in an $n \times n$ pattern cannot be compressed to the size $O(n)$ in *bounded time*.

The ideas behind our argument are similar to those used in [2] and [6], and there is no surprise that Kolmogorov complexity helps to formalize the intuition of “information flows.” The main new element of our approach is the idea of information compression with *bounded computational resources*.

This technique applies to several shifts with very low block complexity: we cannot “communicate” the essential information about a pattern across its contour not because this information is too large, but since we do not have enough time and space to compress it. In particular, we provide examples of non-sofic effective shifts with only polynomial block complexity (and thus zero entropy).

Remark 1. A more straightforward approach to the measure of the “*information flow through the border line of a pattern*” might use the notion of *extender sets* introduced in [6]. Extender sets are a natural generalization of follower sets to multi-dimensional shifts. Let \mathcal{S} be a shift and P be a globally admissible finite pattern in this shift. The extender set of P in \mathcal{S} is the set of all (co-finite) patterns Q completing P to a valid configuration of \mathcal{S} (the support of Q should be the complement of the support of P).

Let \mathcal{S} be a shift on \mathbb{Z}^2 ; denote by N_k the number of different extender sets for the globally admissible patterns with a support of size $k \times k$. (Several patterns can share one and the same extender set, so the number of extender sets might be much less than the number of all globally admissible patterns of this size). It seems natural to interpret $\log N_k$ as the “information flow” crossing the border line of a $k \times k$ pattern and imposing the consistency between the pattern and its complement. As the length of the contour around a $k \times k$ pattern is $O(k)$, the naive intuition suggests that “a super-linear flow of information cannot cross a contour of linear length.” This is indeed true for shifts of finite type: for every SFT on \mathbb{Z}^2 , the number N_k is not bigger than $2^{O(k)}$. However, in case of sofic shifts, this approach fails. For a sofic shift on \mathbb{Z}^2 , the value of $\log N_k$ can grow much faster than the length of the border line of the pattern ([6] attributes this observation to unpublished works of C. Hoffman, A. Quas, and R. Pavlov). In fact, for a sofic shift on \mathbb{Z}^2 , the value of $\log N_k$ can grow even as $\Omega(k^2)$ (we mention such an example in Section 4.2). Therefore, we cannot use the asymptotic of $\log N_k$ to prove non-soficness of a multi-dimensional shift. This is why we need a subtler refinement of the intuition of “information flows” in sofic shifts.

Though the *cardinality* of the family of extender sets is not enough to distinguish between sofic and non-sofic shifts, some finer properties of extender sets can be helpful. Kass and Madden proposed in [6] an approach to the proof of non-soficness based on the structure of these sets. Technically, they focus on the length of union-increasing chains of extender sets, which is a more appropriate parameter than the total number of these sets. We discuss this method in Section 4.3 and show that it can be interpreted as a special case of our approach.

The rest of the paper is organized as follows. After recalling the necessary definitions from symbolic dynamics and the theory of Kolmogorov complexity in Section 2, we prove in Section 3 our main technical results and construct a non-sofic effective shift with very low block complexity. In Section 4 we extend our technique to more general settings; we discuss several examples and show that every proof of non-soficness using the argument from [6] (the method of union-increasing chains of extender sets) can be reformulated in terms of our technique with the resource-unbounded Kolmogorov complexity. For the sake of self-containedness, in Appendix we prove Theorem 3 (implicitly proven in [2]) and Lemma 1 (formulated and proven in different terms in [6]). We also prove in Appendix Theorem 4, which is a stronger version of Theorem 4' (though the simpler Theorem 4' proven in the main text of the paper is strong enough for all applications where we use it to construct effective non-sofic shifts).

2 Preliminaries

Shift spaces In the introduction we have defined the notion of *shift spaces* (also called *shifts*). More specifically, we study *shifts of finite type*, *sofic shifts*, and *effective shifts*. In this paper we focus on two-dimensional shifts, though all arguments can be extended to the shifts on \mathbb{Z}^d for all $d \geq 2$.

A (finite) *pattern* on \mathbb{Z}^2 over a finite alphabet Σ is a mapping from a (finite) subset of \mathbb{Z}^2 to Σ ; the domain of this mapping is the *support* of the pattern. Sometimes a pattern P with a support \mathcal{A} is called a *coloring* of \mathcal{A} (the “colors” are letter from Σ).

For a shift \mathcal{S} , we say that a pattern P is *globally admissible*, if P is a restriction of a configuration from \mathcal{S} onto some finite support. For a shift of finite type determined by a set of forbidden patterns \mathcal{F} , we say that a pattern is *locally admissible* if it contains no forbidden patterns from \mathcal{F} . Similarly, two patterns with disjoint domains are *locally compatible* if the union of these patterns is locally admissible.

The *block complexity* of a shift is a function that gives for each integer $n > 0$ the number of globally admissible patterns of size $n \times n$ (patterns with support $\{1, \dots, n\}^2$) in this shift.

If a sofic shift \mathcal{S} is a coordinate-wise projection of configurations from $\hat{\mathcal{S}}$, we say that $\hat{\mathcal{S}}$ is a *covering* of \mathcal{S} . Every sofic shift has a covering SFT that can be defined by a set of forbidden patterns F , where every pattern in F is a pair of neighboring letters (see, e.g. [9]).

Computability and Kolmogorov complexity In this section we recall the main definitions of the theory of Kolmogorov complexity. In the way that is usual in computability theory (see, e.g., [13, 16]), we identify *algorithms* with Turing machines. A partial function $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ is called *computable*, if there is an algorithm (a Turing machine) \mathcal{M} that computes this function in the following sense: given as the input a string x from the domain of f , machine \mathcal{M} prints the value $f(x)$ and stops; given as the input a string x such that $f(x)$ is not defined, machine \mathcal{M} never stops. It is known that the class of computable functions does not change if we vary minor details of the definition of a Turing machine (we may admit machine with only one tape or with any number of tapes, the tapes can be bi-infinite or only infinite to the right, and so on). It is generally agreed that this definition is an adequate formalization of the intuitive idea of computability. In this paper we discuss several simple algorithms; we give only an informal description of these algorithms and do not construct explicitly the corresponding Turing machines.

Let U be a (partial) computable function. The complexity of x with respect to the description method U is defined as

$$C_U(x) := \min\{ |p| : U(p) = x \},$$

where $|p|$ stands for the length of the binary string p . If there is no p such that $U(p) = x$, we assume that $C_U(x) = \infty$. Here U is understood as a programming language; p is a program that prints x ; the complexity of x is the length of (one of) the shortest programs p that generate x .

The obvious problem with this definition is its dependence on U . The theory of Kolmogorov complexity becomes possible due to the following invariance theorem.

Theorem 1 (Kolmogorov, [7]). *There exists a computable function U such that for any other computable function V there is a constant d such that*

$$C_U(x) \leq C_V(x) + d$$

for all x .

Notation 1. This U is called *an optimal description method*. We fix an optimal U and in what follows omit the subscript in $C_U(x)$ and use the notation $C(x)$. The value $C(x)$ is called the (plain) Kolmogorov complexity of x .

In a similar way, we define Kolmogorov complexity in terms of programs with bounded resources (the time of computation). Let U be a Turing machine; we define the Kolmogorov complexity $C_U^t(x)$ as the length of the shortest p such that $U(p)$ produces x in at most t steps. There exists an *optimal description method* U in the following sense: for every Turing machine V

$$C_U^{\text{poly}(t)}(x) \leq C_V^t(x) + O(1).$$

For multi-tape Turing machines a slightly stronger statement can be proven:

Theorem 2 (see [8]; the proof uses the simulation technique from [3]). *There exists an optimal description method (multi-tape Turing machine) U_m in the following sense: for every multi-tape Turing machine V (with any number of tapes) there exists a constant d such that*

$$C_{U_m}^{ct \log t}(x) \leq C_V^t(x) + d$$

for all strings x .

Notation 2. We fix such a machine U_m , and in the sequel use for the resource-bounded version of Kolmogorov complexity the notation $C^t(x)$ instead of $C_{U_m}^t(x)$. Without loss of generality we may assume that $C(x) \leq C^t(x)$ for all x and for all t .

It is easy to show that there exists a constant d such that for all bit strings $x = (x_1 \dots x_n) \in \{0, 1\}^n$ we have $C(x) \leq n + d$ (roughly speaking, this is true because there always exists a trivial algorithm of type “ $print(x_1 \dots x_n)$ ” where the bits x_i are listed explicitly; such an algorithm can be expressed in any natural programming language, including Turing machines). At the same time, for every n there exists a string $x \in \{0, 1\}^n$ such that $C(x) \geq n$, because the total number of descriptions p that are shorter than n bits, is less than 2^n . (Such a string x is usually called *incompressible*.) More generally, in every set of 2^n strings there is at least one string x such that $C(x) \geq n$.

We fix a computable enumeration of finite patterns (over a finite alphabet) that assigns a binary string (a *code*) to each pattern in dimension two. In the sequel we take the liberty of talking about Kolmogorov complexity of finite patterns in dimension two (assuming the Kolmogorov complexity of the *codes* of these patterns). Similarly, we can talk about Kolmogorov complexity of natural numbers (assuming the Kolmogorov complexity of the binary expansion of these numbers).

In computability theory it is common to study *relativized computations*, i.e., computations with an oracle. Let \mathcal{O} be an infinite binary sequence. A Turing machine *with oracle* \mathcal{O} is a Turing machine with an extra read-only tape where the bits of \mathcal{O} are written; the machine can read bits of the oracle when necessary and then use them in the computation on the usual read/write working tape. Though the access to an oracle can extend the power of a Turing machine, many fundamental properties of Turing machines remain valid with any oracle (in particular, the arguments based on diagonalization apply to Turing machines with an oracle). We refer the reader to the textbooks [13, 16] for a detailed discussion of relativized computations.

Notation 3. Having fixed an infinite sequence \mathcal{O} , we can define the relativized version of Kolmogorov complexity for machines with oracle \mathcal{O} (in exactly the same way as above but with Turing machines having the access to the chosen oracle). We use the notation $C^{\mathcal{O}}(x)$ for Kolmogorov complexity with oracle \mathcal{O} ; similarly, $C^{t, \mathcal{O}}(x)$ denotes Kolmogorov complexity with oracle \mathcal{O} and time bound t .

A systematic introduction to the theory of Kolmogorov can be found in [8, 19]. In what follows we use only rather simple properties of Kolmogorov complexity. Throughout the paper, the claims of the form $C^t(x) = m$ can be understood as an informal statement *there is a program of length m that prints x in t steps*; we avoid statements where subtle details of the formal definition of Kolmogorov complexity become significant.

3 High resource-bounded Kolmogorov complexity is compatible with low block complexity

The following theorem was proven implicitly in [2]:

Theorem 3. *In every non-empty sofic shift \mathcal{S} on \mathbb{Z}^2 there exists a configuration x such that for all $n \times n$ -patterns P in x , we have*

$$C^{T(n)}(P) = O(n)$$

for a time threshold $T(n) = 2^{O(n^2)}$.

In [2] a weaker version of this theorem is stated: it is claimed only that the *plain* complexity of $n \times n$ patterns is $O(n)$. However, the argument from [2] implies a bound for a *resource-bounded* version of Kolmogorov complexity. For the sake of self-containedness, in what follows we provide a proof of this theorem in A.

Theorem 3 guarantees that a shift is non-sofic, if in every configuration we can find a sequence of growing size patterns with super-linear Kolmogorov complexity. In the next theorem we prove the existence of such effective shifts.

Theorem 4. *For every $\epsilon > 0$ and for every computable $T(n)$ there exists an effective shift on \mathbb{Z}^2 such that for every n and for every globally admissible pattern P of size $n \times n$, we have that*

(i) $C(P) = O(\log n)$, and

(ii) $C^{T(n)}(P) = \Omega(n^{2-\epsilon})$.

We defer the proof of Theorem 4 to B. In what follows we prove a slightly weaker version of this theorem, which is nevertheless strong enough for our main applications:

Theorem 4'. *For every computable $T(n)$ there exists an effective shift on \mathbb{Z}^2 such that*

(i) *for every n and for every globally admissible pattern P of size $n \times n$, we have $C(P) = O(\log n)$, and*

(ii) *for infinitely many n and for every globally admissible pattern P of size $n \times n$, we have that $C^{T(n)}(P) = \Omega(n^{1.5})$.*

From Theorem 3 and Theorem 4' we deduce the following corollary:

Corollary 1. *There exists an effective non-sofic shift on \mathbb{Z}^2 with block complexity $\text{poly}(n)$, i.e., with $\leq \text{poly}(n)$ globally admissible blocks of size $n \times n$.*

Proof. We take the shift from Theorem 4' assuming that the threshold $T(n)$ is much greater than $2^{\Omega(n^2)}$ (e.g., we can let $T(n) = 2^{n^3}$). On the one hand, property (ii) of Theorem 4' and Theorem 3 guarantee that this shift is not sofic. On the other hand, property (i) of Theorem 4' implies that the number of globally admissible blocks of size $n \times n$ is not greater than $2^{O(\log n)}$. \square

Remark 2. Our proof of Theorem 4' implies a stronger bound than property (i). In fact, instead of the bound $C(P) = O(\log n)$ we can prove for some (large enough) computable function $\hat{T}(n)$ that for every globally admissible $n \times n$ pattern P in this shift we have

$$C^{\hat{T}(n)}(P) = O(\log n).$$

Now we introduce some notation.

Notation 4. For any function $T(n)$ and for any number $\lambda > 0$, let $\mathcal{S}_{\lambda, T}$ denote the following shift over the alphabet $\{0, 1\}$: a configuration x belongs to $\mathcal{S}_{\lambda, T}$ if and only if $C^{T(n)}(P) \leq \lambda \log n$ for every $n \times n$ pattern P in x .

Obviously, for an integer λ and a computable $T(n)$, the shift introduced in Notation 4 is effective.

Remark 2 can be rephrased as follows: there are an integer number $\hat{\lambda}$ and a computable function $\hat{T}(n)$ such that for the corresponding shift $\mathcal{S}_{\hat{\lambda}, \hat{T}}$ there exists a sequence of globally admissible patterns P_n of size $n \times n$ ($n = 1, 2, \dots$) such that

$$C^{2^{n^3}}(P_n) = \Omega(n^{1.5});$$

the same time, for every globally admissible pattern of size $n \times n$ we have $C(P) \leq C^{\hat{T}(n)}(P) = O(\log n)$.

Notation 5. Let \mathcal{S}_0 denote the shift defined above.

The shift constructed in Theorem 4' is a proper subshift of \mathcal{S}_0 . Indeed, besides all configurations of the shift from Theorem 4' (those configuration must have only patterns with high resource-bounded Kolmogorov complexity), \mathcal{S}_0 admits also patterns with a very low time bounded complexity (e.g., the configuration with all 0s and the configuration with all 1s). In the next section we use \mathcal{S}_0 to construct some other examples of effective non-sofic shifts.

(Proof of Theorem 4'). In this proof we provide a construction of a required shift assuming that we have fixed in advance a universal Turing machine in the definition of Kolmogorov complexity. The choice of a universal Turing machine is not unique (and quite arbitrary), and the final choice of parameters in our construction depends on this arbitrariness.

Let us fix a sequence (n_i) where n_0 is a large enough integer number², and

$$n_{i+1} := (n_0 \cdot \dots \cdot n_i)^c \quad \text{for } i = 0, 1, 2, \dots, \quad (1)$$

where $c \geq 3$ is a constant. We set $N_i := n_0 \cdot \dots \cdot n_i$. In what follows we construct for each i a pair of *standard* binary patterns Q_i^0 and Q_i^1 of size $N_i \times N_i$ such that

²For every instance of an optimal Turing machine U in the definition of Kolmogorov complexity one can compute the corresponding minimum value of n_0 that makes the construction work.

- the plain Kolmogorov complexities of the standard patterns $C(Q_i^0)$ and $C(Q_i^1)$ are not greater than $O(\log N_i)$, and
- the resource-bounded Kolmogorov complexities $C^{T(N_i)}(Q_i^0)$ and $C^{T(N_i)}(Q_i^1)$ are not less than $\Omega(N_i^{1.5})$.

The construction is hierarchical: both Q_i^0 and Q_i^1 are defined as $n_i \times n_i$ matrices composed of patterns Q_{i-1}^0 and Q_{i-1}^1 ; for each i the blocks Q_i^0 and Q_i^1 are bitwise inversions of each other.

When the standard patterns Q_i^0 and Q_i^1 are constructed for all i , we define the shift as the closure of these patterns: we say that a finite pattern is globally admissible if and only if it appears in some standard pattern Q_i^j or at least in a 2×2 -block composed of Q_i^0 and Q_i^1 (for some i).

Remark 3. Due to the hierarchical structure of the standard patterns, we can guarantee that every globally admissible pattern P of size $N_i \times N_i$ appears in a 2×2 -block composed of Q_i^0 and Q_i^1 (no need to try the blocks Q_s^j for $s > i$).

Since the construction of Q_i^j is explicit, the resulting shift is effective. Properties (i) and (ii) of the theorem will follow from the properties of the standard patterns.

In what follows we explain an inductive construction of Q_i^0 and Q_i^1 . Let Q_0^0 and Q_1^0 be the squares composed of only 0s and only 1s respectively. Further, for every i we take the lexicographically first binary matrix R_i of size $n_i \times n_i$ such that

$$C^{t_i}(R_i) \geq n_i^2 \quad (2)$$

(the time bound t_i is fixed in the sequel). We claim that such a matrix exists. Indeed, there exists a matrix of size $n_i \times n_i$ that is incompressible in the sense of the plain Kolmogorov complexity. The resource-bounded Kolmogorov complexity of a matrix can be only greater than the plain complexity. Therefore, there exists at least one matrix satisfying (2). If t_i is a computable function of i , then given i we can find R_i algorithmically.

Now we substitute in R_i instead of each zero and one entry the copies of Q_{i-1}^0 and Q_{i-1}^1 respectively, e.g.,

$$R_i = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \implies Q_i^0 := \begin{pmatrix} Q_{i-1}^0 & Q_{i-1}^0 & Q_{i-1}^0 & Q_{i-1}^0 & Q_{i-1}^1 \\ Q_{i-1}^0 & Q_{i-1}^1 & Q_{i-1}^0 & Q_{i-1}^0 & Q_{i-1}^1 \\ Q_{i-1}^1 & Q_{i-1}^1 & Q_{i-1}^1 & Q_{i-1}^1 & Q_{i-1}^0 \\ Q_{i-1}^0 & Q_{i-1}^1 & Q_{i-1}^1 & Q_{i-1}^0 & Q_{i-1}^0 \\ Q_{i-1}^0 & Q_{i-1}^1 & Q_{i-1}^0 & Q_{i-1}^1 & Q_{i-1}^0 \end{pmatrix}$$

The resulting matrix (of size $N_i \times N_i$) is denoted Q_i^0 . Matrix Q_i^1 is defined as the bitwise inversion of Q_i^0 .

Claim 1. Assuming that $t'_i \ll t_i$ (in what follows we discuss the choice of t'_i in more detail) we have

$$C^{t'_i}(Q_i^0) = \Omega(N_i^{1.5}) \text{ and } C^{t'_i}(Q_i^1) = \Omega(N_i^{1.5}).$$

Proof of Claim 1. Given Q_i^j (for $j = 0, 1$) we can retrieve the matrix R_i (this retrieval can be implemented in polynomial time). Therefore, for every time bound t

$$C^{t+\text{poly}(N_i)}(R_i) \leq C^t(Q_i^j) + O(1).$$

Therefore, if $t_i > t'_i + \text{poly}(N_i)$ then

$$n_i^2 \leq C^{t_i}(R_i) \leq C^{t'_i}(Q_i^j).$$

It remains to observe that our choice of parameters in (1) with $c \geq 3$ implies $n_i^{1/2} \geq (n_0 \cdot \dots \cdot n_{i-1})^{3/2}$, and therefore

$$n_i^2 \geq (n_0 \cdot \dots \cdot n_i)^{1.5} = (N_i)^{1.5}.$$

Thus, we obtain $C^{t'_i}(Q_i^j) \geq (N_i)^{1.5} - O(1)$, and the claim is proven. \square

Remark 4. By choosing a larger constant c in (1), we can achieve a lower bound $C^{t'_i}(Q_i^j) = \Omega(n^{2-\epsilon})$ for any $\epsilon > 0$.

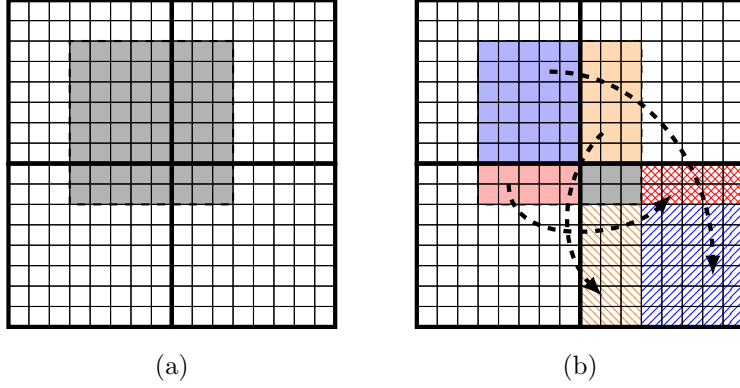


Figure 2: A pattern of size $N_k \times N_k$ (shown in gray in fig. (a)) covered by a quadruple of standard blocks of the same size contains enough information to reconstruct a standard pattern (fig. (b)).

Claim 2. For every globally admissible pattern P of size $N_i \times N_i$ (and not only for the standard patterns, as it was in Claim 1) its time-bounded Kolmogorov complexity $C^{T(N_i)}(P)$ is $\Omega(n^{1.5})$ (assuming that $T(N_i) \ll t'_i$).

Proof of Claim 2. If a pattern P of size $N_i \times N_i$ is globally admissible then it is covered by a quadruple of standard patterns of rank i , see Remark 3 above. Then P can be divided into four rectangles which are “corners” of standard patterns of rank i , see Fig. 2 (a). Since the standard blocks Q_i^0 and Q_i^1 are the inversions of each other, these four “corners” (with a bitwise inversion if necessary) form together the entire standard pattern, as shown in Fig. 2 (b). Therefore, we can reconstruct Q_i^j from P given (a) the position of P with respect to the grid of standard blocks (this involves $O(\log N_i)$ bits) and (b) the four bits identifying the standard blocks covering P (we need to know which of them is a copy of Q_i^0 and which one is a copy of Q_i^1).

The retrieval of Q_i^j from P requires only $\text{poly}(N_i)$ steps of computation (in addition to the time we need to produce P). Now the claim follows from the bound for the resource-bounded Kolmogorov complexity of the standard patterns Q_i^0 and Q_i^1 . \square

Claim 3. For every $k \times k$ -pattern in Q_i^0 or Q_i^1 , its plain Kolmogorov complexity is at most $O(\log k)$.

Proof of Claim 3. First of all, we observe that the standard patterns Q_i^0 or Q_i^1 can be computed given i . Therefore, $C(Q_i^0) = O(\log i)$ and $C(Q_i^1) = O(\log i)$.

Every globally admissible $k \times k$ -pattern is covered by at most four standard patterns Q_i^0 or Q_i^1 with

$$N_{i-1} < k \leq N_i,$$

see Remark 3. Therefore, to obtain a globally admissible pattern P of size $k \times k$ we need to produce a quadruple of standard patterns of size $N_i \times N_i$ and then to specify the position of P with respect to the grid of standard blocks. This description consists of only $O(\log N_i)$ bits, and we conclude that $C(P) = O(\log k)$. \square

It remains to fix the time bounds t_i . Given a computable threshold $T(N_i)$, we choose a suitable $t'_i \gg T(N_i)$ and then a suitable $t_i \gg t'_i$. The theorem follows from Claim 2 and Claim 3. \square

Remark 5. For all large enough i , the incompressible pattern R_i constructed in the proof of Theorem 4' contains copies of all 2^4 binary patterns of size 2×2 . Therefore, we can guarantee that every standard block Q_i^j contains all globally admissible patterns of size $N_{i-1} \times N_{i-1}$.

There exists a non-empty effective shift on \mathbb{Z}^2 where the Kolmogorov complexity of all $n \times n$ patterns is $\Omega(n^2)$ (see [2] and [17]). So a natural question arises: can we improve Theorem 4' and strengthen condition (ii) to $C^{T(n)}(P) = \Omega(n^2)$? The answer is negative: we cannot achieve the resource bounded complexity $\Omega(n^2)$, even with a much weaker version of property (i) for the plain complexity:

Proposition 1. *For all large enough time bounds $T(n)$, there is no shift on \mathbb{Z}^2 such that*

- (i) *for every globally admissible pattern P of size $n \times n$, we have that $C(P) = o(n^2)$, and*
- (ii) *for infinitely many n and for every globally admissible pattern P of size $n \times n$, we have that $C^{T(n)}(P) = \Omega(n^2)$.*

Proof. Assume for the sake of contradiction that such a shift exists. Let us fix a real number $\varepsilon > 0$. Condition (i) means that there exists an integer $k = k(\varepsilon)$ such that the number of globally admissible $k \times k$ patterns in this shift is $L_k \leq 2^{\varepsilon k^2}$.

Let us estimate Kolmogorov complexity of globally admissible patterns of size $(Nk) \times (Nk)$. For any N , every globally admissible pattern P of size $(Nk) \times (Nk)$ can be specified by

- the list of all globally admissible patterns of size $k \times k$, which requires $L_k \cdot k^2$ bits (with $O(L_k \cdot k^2)$ bits we can even provide a self-delimiting description of this list), and
- by an array of size $N \times N$ with indices of $k \times k$ blocks that constitute P (which requires $N^2 \cdot \log L_k$ bits).

We can combine these two parts of the description in one program. Clearly, P can be reconstructed from such a description in polynomial time. Thus, it follows that for some polynomial $\text{poly}_1(n)$

$$C^{\text{poly}_1(Nk)}(P) \leq O(L_k \cdot k^2) + N^2 \cdot \varepsilon k^2. \quad (3)$$

Now we estimate Kolmogorov complexity of a globally admissible pattern P of size $n \times n$ in case when n is not divisible by k . Let N be the integer number such that

$$(N-1)k < n \leq Nk.$$

Pattern P can be embedded in a bigger globally admissible pattern P' of size $(Nk) \times (Nk)$. To describe P , we need to describe P' and then specify the position of P with respect to the borders of P' . The gaps between P and the borders of P' cannot be bigger than k . Therefore, to obtain an upper bound for Kolmogorov complexity of P , we only need to add $O(\log k)$ bits to (3). It follows that for some polynomial $\text{poly}_2(n)$

$$\begin{aligned} C^{\text{poly}_2(n)}(P) &\leq O(L_k \cdot k^2) + N^2 \cdot \varepsilon k^2 + O(\log k) \\ &\leq \varepsilon(n+k)^2 + O(L_k \cdot k^2) \\ &\leq \varepsilon n^2 + 2\varepsilon nk + \varepsilon k^2 + O(L_k \cdot k^2). \end{aligned} \quad (4)$$

For a fixed k and large enough n , the first term in (4) dominates the remaining terms, and we get

$$C^{\text{poly}_2(n)}(P) \leq 2\varepsilon n^2.$$

Let us recall that we can prove this bound for every $\varepsilon > 0$ and for all large enough n . This contradicts condition (ii) of the proposition. \square

4 Epitomes

The technique from Section 3 does not apply to the shifts that contain very simple configurations (with low resource-bounded Kolmogorov complexity of all patterns). In particular, it does not apply to Example 1 from the introduction. In this section we propose a different technique (also based on Kolmogorov complexity) that helps to handle such shifts. The intuitive idea behind this technique is as follows: we try to capture the “essential” information in each pattern (discarding irrelevant data) and then measure the plain or resource-bounded Kolmogorov complexity of an “epitome” extracted from this essential information. We show that in a sofic shift the complexity of properly defined “epitomes of the essential information” must be bounded by the length of the contour around the pattern. Thus, if this condition is violated, then the shift is not sofic.

Let us fix some notation. We denote by B_n the set $\{0, \dots, n-1\}^2 \subset \mathbb{Z}^2$ and by F_n its complement, $F_n := \mathbb{Z}^2 \setminus B_n$. We say that two patterns with disjoint supports are *compatible* (for a shift \mathcal{S}) if the union

of these patterns is globally admissible in \mathcal{S} . In particular, a finite pattern P with support B_n and an infinite pattern R with support F_n are compatible, if the union of these patterns is a valid configuration of the shift.

In Section 4.1 we start with a simple and more restrictive definition of *plain epitomes*. Then, in Section 4.2 we proceed with a more general definition of *ordered epitomes* and discuss several examples of proofs of non-soficness based on this technique. In Section 4.3 we compare the technique of epitomes with the technique of *union-increasing chains of extender sets* proposed by Kass and Madden in [6]; we show that the method of Kass and Madden is equivalent to a special case of the method of ordered epitomes.

4.1 Plain epitomes

In this section we propose a general approach to the proof of non-soficness that captures the intuition behind the standard proof from Example 1. We begin with a definition of “plain” epitome functions.

Definition 2. Let \mathcal{P} be the set of all patterns on \mathbb{Z}^2 over some finite alphabet. Let

$$\mathcal{E} : \mathcal{P} \mapsto \{0, 1\}^*$$

be a partial function mapping patterns to binary strings. We say that \mathcal{E} is an *epitome function* for the shift \mathcal{S} if for every pattern P with support B that is globally admissible in \mathcal{S} (in brief, P is a *valid pattern*) such that $\mathcal{E}(P)$ is defined, there exists an *epitome witness* pattern R on $\mathbb{Z}^2 \setminus B$ (the complement of B) for which

- (i) R is compatible with P , i.e., the union of P and R is a valid configuration in \mathcal{S} , and
- (ii) for every pattern P' with support B that is compatible with R and for which $\mathcal{E}(P')$ is defined, $\mathcal{E}(P) = \mathcal{E}(P')$. In other words, the epitome witness R uniquely determines the value of \mathcal{E} for the valid patterns on the complement of its support.

Given an epitome function \mathcal{E} and a pattern P , we refer to $\mathcal{E}(P)$ as the epitome of P , or as the \mathcal{E} -epitome of P if the epitome function is not clear from the context.

In general, an epitome function can be defined for patterns of any shape. However, in this paper we only use epitome functions \mathcal{E} defined on patterns with support B_n (a square if size $n \times n$) for different $n > 0$. In certain situations it can be helpful to focus on the restriction of an epitome function onto the patterns of a fixed size. Given an epitome function \mathcal{E} , we write \mathcal{E}_n for the restriction of \mathcal{E} onto the patterns with support B_n . We refer to the sequence \mathcal{E}_n ($n = 1, 2, \dots$) as a *family of restricted epitome functions for \mathcal{S}* . If an epitome function is defined only for patterns with supports B_n (which is always the case in this paper), a family of restricted epitome functions \mathcal{E}_n for all $n > 0$ and an epitome function \mathcal{E} are two equivalent languages to describe one and the same object.

We say that \mathcal{E} is *computable* if there is an algorithm that computes \mathcal{E} . As is customary in computability theory, we assume that an algorithm computing a partial function returns no result for inputs outside its domain. If, moreover, there is an algorithm computing the value $\mathcal{E}(P)$ in time $2^{O(n^2)}$ for each input pattern P of size $n \times n$ ($n = 1, 2, \dots$) from the domain of \mathcal{E} , then we say that this epitome function is *exp-time computable*.

Similarly, we say that an epitome function is *computable with oracle \mathcal{O}* if the domain of \mathcal{E} is decidable with oracle \mathcal{O} and there is an algorithm that computes \mathcal{E} given the access to oracle \mathcal{O} . We can also talk about epitome functions that are *exp-time computable with an oracle*.

Remark 6. If an epitome function is exp-time computable, then the domain of \mathcal{E} can be decided in exponential time: it is enough to run the algorithm computing \mathcal{E} and abort it if the computation does not converge in due time.

Remark 7. Formally speaking, \mathcal{E} -epitomes are binary strings — objects for which we have defined Kolmogorov complexity. In applications, it can be convenient to assign the values of an epitome function to integer numbers, to vectors of integer numbers, or to multi-dimensional finite patterns over a finite alphabet. All these constructive objects can be represented by finite binary strings (with a naturally chosen encoding); so in some cases we abuse notation and talk about epitomes that are integer numbers (or vectors of integer numbers, or finite patterns, and so on) and their Kolmogorov complexities.

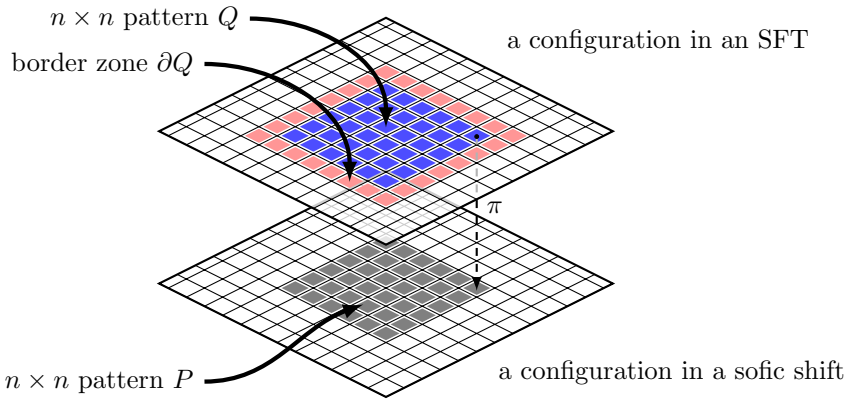


Figure 3: Projection of an $n \times n$ pattern from an SFT onto a sofic shift.

Proposition 2. (a) For every sofic shift with a computable epitome function \mathcal{E} , for every globally admissible pattern P of size $n \times n$ such that $\mathcal{E}(P)$ is defined, we have $C(\mathcal{E}(P)) = O(n)$.

(b) For every sofic shift with an exp-time computable epitome function \mathcal{E} , for every globally admissible pattern P of size $n \times n$ such that $\mathcal{E}(P)$ is defined, we have $C^{T(n)}(\mathcal{E}(P)) = O(n)$ for a time threshold $T(n) = 2^{O(n^2)}$.

(c) The statements (a) and (b) relativize. That is, for every sofic shift with an epitome function \mathcal{E} that is computable with oracle \mathcal{O} (or exp-time computable with oracle \mathcal{O}), for every globally admissible pattern P of size $n \times n$ such that $\mathcal{E}(P)$ is defined, we have $C^{\mathcal{O}}(\mathcal{E}(P)) = O(n)$ (or, respectively, $C^{T(n), \mathcal{O}}(\mathcal{E}(P)) = O(n)$ for a time threshold $T(n) = 2^{O(n^2)}$).

Proof. (a) Assume that \mathcal{S} is a sofic shift with a covering SFT $\hat{\mathcal{S}}$ (i.e., \mathcal{S} is a coordinate-wise π -projection of $\hat{\mathcal{S}}$). Let P be a pattern with support B_n in \mathcal{S} and R be the pattern on the complement of B_n that enforces the value of the \mathcal{E} -epitome of P (as specified in Definition 2). Denote by Y a configuration in $\hat{\mathcal{S}}$ whose π -projection gives the union of P and R . Let Q be a pattern of size $n \times n$ in Y such that P is a coordinate-wise projection of Q . Denote by ∂Q the “border zone,” i.e., the nearest neighborhood of Q (the border zone of Q is not included in Q ; if Q is an $n \times n$ square, then border zone ∂Q around Q consists of $4(n+1)$ letters, see Fig. 3).

We assume without loss of generality that the local constraints in $\hat{\mathcal{S}}$ involve only pairs of neighboring nodes in \mathbb{Z}^2 . Then, every locally admissible pattern Q' of size $n \times n$ that is locally compatible with border zone ∂Q , must be compatible with the rest of configuration Y . Therefore, the π -projections of these Q' are compatible with R . We apply the definition of an epitome function: if \mathcal{E} is defined on the π -projection of such Q' , then the epitome of Q' must be equal to the epitome of P .

It follows that $\mathcal{E}(P)$ can be computed given only the coloring of ∂Q : we use the brute-force search to find all patterns Q' that are locally compatible with ∂Q , apply to them projection π , and run in parallel the computation of their epitomes. As soon as we find at least one Q' compatible with ∂Q such that the epitome function is defined on its π -projection, we are done: though the projection $\pi(Q')$ can be different from P , the epitome of $\pi(Q')$ must coincide with the epitome of P . Since the size of ∂Q is linear in n , we conclude that $C(\mathcal{E}(P)) = O(n)$.

To prove claim (b), we observe that the brute force search over all patterns of size $n \times n$ runs in time $2^{O(n^2)}$, so we get $C^{2^{O(n^2)}}(\mathcal{E}(P)) = O(n)$. It is easy to verify that the proof relativizes (each step of the argument works with any fixed oracle \mathcal{O}), so we get claim (c). \square

Proposition 2 gives a necessary condition for soficness. To prove that a shift is not sofic, we need to provide a computable (or an exp-time computable) epitome function with high Kolmogorov complexities (or, respectively, resource-bounded Kolmogorov complexities). In what follows we discuss simple examples of application of this technique. We start with the standard example of mirror symmetric shift.

Example 1 revisited. Let $\mathcal{S}_{\text{mirror}}$ be the shift from Example 1 in the introduction (the mirror-symmetric configurations). For this example we can define an epitome function \mathcal{E} as follows:

- if an $n \times n$ pattern P contains only black and white letters, then $\mathcal{E}(P)$ is a binary string of length n^2 that identifies P uniquely (roughly speaking, \mathcal{E} does not compress the patterns in black and white);
- \mathcal{E} is undefined for all patterns containing at least one red letter.

The defined \mathcal{E} satisfies the definition of a computable epitome function, since the part of a configuration above the red line determines uniquely all black-and-white patterns below this line. More specifically, for every $n \times n$ pattern P that contains only black and white letters, let R be the infinite pattern defined as follows:

- the domain of R is the complement of the $n \times n$ domain of P ,
- the horizontal line in R that is one row above P must consist of only red letters,
- the positions that are symmetric (with respect to the red line) to the positions with black letters in P must be black,
- all other letters in R must be white,

as shown in Fig. 4. It is clear that R is compatible with P and not compatible with other $n \times n$ patterns, as required in the definition of an epitome function.

Since there are 2^{n^2} patterns of size $n \times n$ with black and white letters, for some patterns of size $n \times n$ we have $C(\mathcal{E}(P)) \geq n^2$. Therefore, we can apply Proposition 2 (a) and conclude that the shift is not sofic.

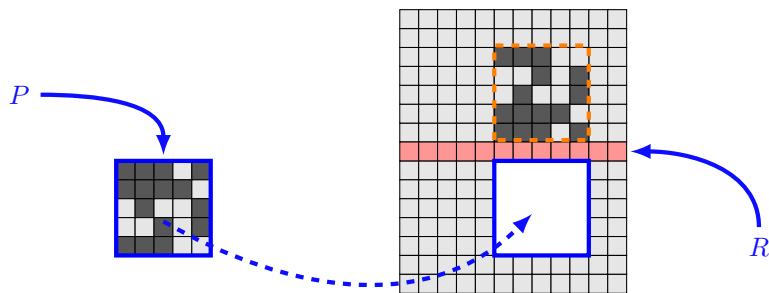


Figure 4: An $n \times n$ pattern P with black and white letters and the corresponding pattern R with the complementary domain. The orange (dashed) frame contains the pattern that is mirror symmetric to P .

Example 3 (Mirror shift with low plain Kolmogorov complexity). Let us consider a subshift of $\mathcal{S}_{\text{mirror}}$ from Example 1: we still admit only symmetric configurations, but we now allow only those $n \times n$ patterns P in black and white that are globally admissible for the shift \mathcal{S}_0 introduced in Notation 5, p. 7. A non-degenerate configuration of the new shift looks as follows: there is an infinite horizontal line composed of red letters, and two symmetric half-planes (filled with black and white letters) above and below this line, with $n \times n$ patterns P such that

$$C^{\hat{T}(n)}(P) \leq \hat{\lambda} \log n$$

(the choice of $\hat{\lambda}$ and $\hat{T}(n)$ is explained before Notation 5). The new shift is effective, and the number of globally admissible patterns is at most $2^{O(\log n)} = \text{poly}(n)$. We know also that *some* globally admissible $n \times n$ patterns in this shift satisfy

$$C^{2^{n^3}}(P) = \Omega(n^{1.5}). \quad (5)$$

We cannot apply Theorem 3 directly and conclude that the new shift is non-sofic. The obstacle is that this shift admits some patterns with very low time-bounded complexity; for example, the shift admits the configuration with an infinite horizontal line in red and only white letters above and below this line. Proposition 2 (a) also does not apply since the plain Kolmogorov complexity of all admissible patterns (and, therefore, of the epitomes of these patterns) is logarithmic. However, in this case we can use exp-time computable epitome functions.

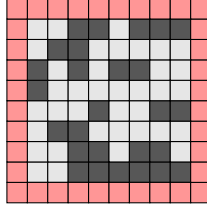


Figure 5: An example of a special pattern of size $n \times n$ from Example 4: the border line (included in the pattern) consists of red letters, and the remaining $(n - 2) \times (n - 2)$ letters can be only black or white.

The functions \mathcal{E} defined for Example 1 (see p. 12) provide for this shift an *exp-time computable* epitome function. For some (though not for all) $n \times n$ patterns P we have (5), so it follows from Proposition 2 (b) that the shift is not sofic.

Remark 8. In the definition of epitome functions we permit that \mathcal{E} can be defined for non-admissible patterns. In the next example we show that this possibility can be useful.

Example 4. In this example we use again the alphabet of three letters (*red*, *black*, and *white*). Before we define a shift on this alphabet, we need to introduce some terms. We say that a pattern P of size $n \times n$ is *special* if it contains only red letters on the border line, and only black and white letters in the internal points, as shown in Fig. 5. It is convenient to require that a special pattern contains at least one black letter, so there are $2^{(n-2)^2} - 1$ special patterns of size $n \times n$. For every n we fix a one-to-one correspondence between the special patterns of this size and the binary strings of length smaller than $(n - 2)^2$. We denote by x_P the binary string corresponding to pattern P .

Let U be the computable function from the definition of Kolmogorov complexity (see Theorem 1). Fix an algorithm computing U . It is known that function U is partial, so the algorithm computing U diverges on some inputs. For every integer number $m > 0$ we denote by \hat{p}_m the string of length smaller than m such that the computation of $U(\hat{p}_m)$ converges but takes longer than the computation of $U(p)$ for any other p with $|p| < m$. In case of a tie (if there are several p requiring the same number of steps to complete the computation) we chose \hat{p}_m that is lexicographically biggest.

Now we define a shift on \mathbb{Z}^2 by the following rule: for every n , a special pattern P of size $n \times n$ is globally admissible if $U(x_P)$ is not defined or if $x_P = \hat{p}_{(n-2)^2}$ (i.e., if the pattern corresponds to the longest converging computation of U among all inputs of length less than $(n - 2)^2$). There is no other constraints: a configuration belongs to the shift if and only if it does not contain any special pattern forbidden by this rule.

This shift is effective: the set of inputs p where the computation of $U(p)$ converges is recursively enumerable and, therefore, so is the set of forbidden special patterns. Now we prove that this shift is not sofic. To this end, we define an epitome function \mathcal{E} as follows:

- for every special pattern P of size n we let $\mathcal{E}(P) = P$ if $U(x_P)$ is defined;
- \mathcal{E} is not defined for other patterns.

This mapping \mathcal{E} is obviously computable. Observe that \mathcal{E} is defined for many inadmissible patterns and for one and only one globally admissible pattern of size $n \times n$ (namely, for the pattern representing the longest-running computation of U). In this situation, the definition of an epitome function is trivially satisfied.

Given $\hat{p}_{(n-2)^2}$, we can find the maximal number of steps required for a computation of $U(p)$ to converge for $|p| < (n - 2)^2$. This information permits to find all converging computations of $U(p)$ for $|p| < (n - 2)^2$, to reveal all strings x with Kolmogorov complexity strictly smaller than $(n - 2)^2$, and then to take the first incompressible string of length $(n - 2)^2$. In other words, given $\hat{p}_{(n-2)^2}$ we can find an object with Kolmogorov complexity at least $(n - 2)^2$. Hence, Kolmogorov complexity of $\hat{p}_{(n-2)^2}$ itself is close to $(n - 2)^2$. Thus, for the only $n \times n$ pattern P that is globally admissible and where $\mathcal{E}(P)$ is defined, we have

$$C(\mathcal{E}(P)) = \Omega(n^2).$$

It follows from Proposition 2 (a) that the shift is not sofic.

Let us note in conclusion that we used in this example epitome functions defined on many non-admissible patterns, and that the domain of \mathcal{E} is recursively enumerable but not decidable. These features seem to be essential for this argument.

4.2 Ordered epitomes

The argument based on Definition 2 does not apply to [6, Example 2.5] and similar shifts. To handle this class of non-sofic shifts, we introduce a slightly more general definition of epitome functions:

Definition 3. Let \mathcal{P} be the set of all patterns on \mathbb{Z}^2 over some finite alphabet, \preceq be a partial order on $\{0, 1\}^*$, and

$$\mathcal{E} : \mathcal{P} \mapsto \{0, 1\}^*$$

be a *partial* function. We say that \mathcal{E} (together with the assigned partial order \preceq) is an *o-epitome function* (a function computing *ordered* epitomes) for a shift \mathcal{S} , if for every globally admissible pattern P with support B such that $\mathcal{E}(P)$ is defined, there exists a pattern R on $F = \mathbb{Z}^2 \setminus B$ such that

- (i) R is compatible with P , i.e., the union of P and R provides a valid configuration in \mathcal{S} , and
- (ii) for every pattern P' on support B compatible with R , if $\mathcal{E}(P')$ is defined then

$$\mathcal{E}(P') \preceq \mathcal{E}(P)$$

(i.e., this configuration R on the complement of B determines the maximum of the \mathcal{E} -epitomes over all valid P').

To denote an o-epitome function (a mapping considered together with a partial order) we write (\mathcal{E}, \preceq) . We say that values of \mathcal{E} are *ordered epitomes* or *o-epitomes*.

Similarly to the case of plain epitomes, in general, ordered epitomes can be defined for patterns of any shape. However, in this paper we use only o-epitome functions defined on patterns with support B_n (square of size $n \times n$) for $n > 0$. It can be helpful to consider the restriction of an o-epitome function onto the patterns of a fixed size. Given an o-epitome function (\mathcal{E}, \preceq) , we write \mathcal{E}_n and \preceq_n for the restrictions of \mathcal{E} and \preceq respectively onto the patterns with support B_n . We refer to the sequence $(\mathcal{E}_n, \preceq_n)$ ($n = 1, 2, \dots$) as a *family of restricted o-epitome functions*³.

We say that an o-epitome function (\mathcal{E}, \preceq) is *computable* if there are algorithms that compute the mapping \mathcal{E} and the relations \preceq . If, moreover, \mathcal{E} and \preceq are computable in time $2^{O(n^2)}$ (for patterns of size $n \times n$), we say that this o-epitome function is *exp-time computable*. As usual, an algorithm computing a partial function must not give any result for inputs outside its domain.

Similarly, we say that an o-epitome function is *computable with oracle* \mathcal{O} if there are algorithms that can compute the mappings \mathcal{E} and the relations \preceq given the access to oracle \mathcal{O} . We can also talk about o-epitome functions that are *exp-time computable* with an oracle.

Plain epitome functions from Definition 2 can be viewed as a special case of Definition 3. If \mathcal{E} is a computable (or exp-time computable) epitome in the sense of Definition 2 and if \preceq is an arbitrary effectively computable (exp-time computable) order on the \mathcal{E} -epitomes, then (\mathcal{E}, \preceq) is a computable (or, respectively, an exp-time computable) o-epitome function in the sense of Definition 3. In Definition 2 the neighborhood R enforces the exact value of $\mathcal{E}(P')$ over all P' compatible with R , while in Definition 3 R enforces only the maximum of $\mathcal{E}(P')$.

Remark 7 applies to the o-epitome functions: we may abuse notation and talk about ordered epitomes that are integer numbers (or vectors of integer numbers, or finite patterns, and so on) and their Kolmogorov complexities.

Proposition 3. (a) For every sofic shift with a computable o-epitome function (\mathcal{E}, \preceq) , for every globally admissible pattern P of size $n \times n$ such that $\mathcal{E}(P)$ is defined, we have $C(\mathcal{E}(P)) = O(n)$.

(b) For every sofic shift with an exp-time computable o-epitome function (\mathcal{E}, \preceq) , for every globally admissible pattern P of size $n \times n$ such that $\mathcal{E}(P)$ is defined, we have $C^{T(n)}(\mathcal{E}(P)) = O(n)$ for a time threshold $T(n) = 2^{O(n^2)}$.

³If an o-epitome function (\mathcal{E}, \preceq) is defined only for patterns with supports B_n , then it is determined uniquely by the corresponding family of restricted o-epitome functions $(\mathcal{E}_n, \preceq_n)$.

(c) The statements (a) and (b) relativize. That is, for every sofic shift with an o-epitome function (\mathcal{E}, \preceq) that is computable with oracle \mathcal{O} (or exp-time computable with oracle \mathcal{O}), for every globally admissible pattern P of size $n \times n$ such that $\mathcal{E}(P)$ is defined, we have $C^{\mathcal{O}}(\mathcal{E}(P)) = O(n)$ (or, respectively, $C^{T(n), \mathcal{O}}(\mathcal{E}(P)) = O(n)$ for a time threshold $T(n) = 2^{O(n^2)}$).

Proof. We reuse the notation from the proof of Proposition 2: \mathcal{S} denotes a sofic subshift, $\hat{\mathcal{S}}$ denotes an SFT covering \mathcal{S} , and π is a letter-to-letter projection mapping $\hat{\mathcal{S}}$ onto \mathcal{S} . We may assume that the local constraints in $\hat{\mathcal{S}}$ involve only pairs of neighboring nodes in \mathbb{Z}^2 . For an $n \times n$ pattern P in \mathcal{S} , we denote by Q the $n \times n$ pattern in $\hat{\mathcal{S}}$ projected on P , and ∂Q denotes the *border zone* (nearest neighborhood) around Q consisting of $4(n+1)$ letters, as shown in Fig. 3. We call such patterns ∂Q *n-border zones*.

We start with a proof of (b), which is very similar to the proof of Proposition 2. In the previous proof, it was enough to find *at least one* pattern Q' compatible with the given border zone ∂Q such that the epitome of $\pi(Q')$ is defined; then computing the epitome of $\pi(Q')$ we obtain the epitome of P . Now we should find *all* patterns Q' compatible with ∂Q , apply to each of them the projection π , compute their epitomes (for those patterns where \mathcal{E} is defined), and then take the maximum of the obtained results. The fact that \mathcal{E} are partial functions provides no obstacle. Indeed, we assumed that the o-epitome function is computable in exponential time; so we can safely abort the computations that do not converge in due time. It only remains to observe that the exhaustive search over all patterns of size $n \times n$ requires to examine $2^{O(n^2)}$ possibilities, which can be performed in exponential time.

We cannot prove (a) in the same way because the computation of epitomes can take arbitrary long time, and we cannot find algorithmically the finalized list of $\mathcal{E}(P')$ for all P' compatible with ∂Q . In this case we need a more involved argument.

Claim 1. *The number of n-border zones ∂Q that are globally admissible in $\hat{\mathcal{S}}$ is not greater than $2^{O(n)}$.*

Proof. This claim is pretty trivial: the domain of such a border zone consists of $4(n+1)$ nodes, and the number of all patterns of this size (admissible or not) is at most $|\Sigma|^{4(n+1)}$, where Σ is the alphabet of $\hat{\mathcal{S}}$. \square

Claim 2. *The number of values $\mathcal{E}(P)$ for globally admissible in \mathcal{S} patterns P is not bigger than the number of all n-border zones ∂Q that are globally admissible in $\hat{\mathcal{S}}$.*

Proof. By the definition of an o-epitome function, for every value $\mathcal{E}(P)$ there is a pattern R on F_n that determines implicitly $\mathcal{E}(P)$: the maximum of $\mathcal{E}(P')$ for all P' compatible with R is equal to $\mathcal{E}(P)$. Though R is infinite, the “interaction” between P and R goes through a contour of linear length. As we observed in the proof of Proposition 2, we can choose an n -border zone ∂Q (globally admissible in $\hat{\mathcal{S}}$) with the following property: if we take all $n \times n$ patterns Q locally compatible with ∂Q in $\hat{\mathcal{S}}$, and then take π -projections of these patterns, we obtain the required pattern P and possibly several other (globally admissible in \mathcal{S}) patterns P' such that $\mathcal{E}(P')$, if defined, is inferior or equal to $\mathcal{E}(P)$ in the sense of \preceq . Therefore, the chosen n -border zone ∂Q determines uniquely the value of $\mathcal{E}(P)$.

Observe that ∂Q determines $\mathcal{E}(P)$ in an abstract sense; we cannot compute $\mathcal{E}(P)$ from ∂Q algorithmically. Indeed, since \mathcal{E} is a partial function (and the algorithm computing \mathcal{E} does not stop on inputs where the function is not defined), we cannot find algorithmically the maximum value of $\mathcal{E}(P')$ for all P' obtained from ∂Q . However, we can claim that the number of values $\mathcal{E}(P)$ for globally admissible (in shift \mathcal{S}) patterns P is not greater than the number of globally admissible (in shift $\hat{\mathcal{S}}$) ∂Q . \square

The set of n -border zones that are globally admissible in any SFT (actually, in any effective shift) is co-enumerable (the complement is recursively enumerable): we can examine progressively all patterns of bigger and bigger size, detect those that are forbidden, and reveal one by one the border zones that are inadmissible (because all their extensions contain forbidden patterns). Using this procedure we can eventually reveal each non-admissible pattern; but *a priori* we do not know when the last inadmissible n -border zone is detected.

Denote by N_n the number of n -border zones that are globally admissible in $\hat{\mathcal{S}}$. Given this number (its binary expansion) we can determine the moment when the described algorithmic process finalizes the list of globally admissible n -border zones (i.e., when all inadmissible n -border zones are revealed). Then, given the list of all non-eliminated (globally admissible) n -border zones, we can find all $n \times n$ patterns in $\hat{\mathcal{S}}$ that are locally compatible with at least one globally admissible n -border zone (so these patterns

are also globally admissible). Then we apply to each of these patterns the mapping π and obtain the $n \times n$ patterns that are globally admissible in \mathcal{S} . Finally, we apply the algorithm computing \mathcal{E} to each obtained pattern and run these computations in parallel. Every value of $\mathcal{E}(P)$ for globally admissible P will be eventually found by this procedure.

Thus, to describe one specific value of \mathcal{E} , we need to describe the enumeration procedure explained above and to specify the ordinal number of the sought value of \mathcal{E} in this enumeration (in the order of appearance). To run the enumeration process, we need to know the numbers n and N_n . Due to Claim 1, the binary expansion of N_n consists of only $O(\log N_n) = O(n)$ bits. The number of different values of \mathcal{E} for globally admissible patterns is bounded by N_n (Claim 2); so the ordinal number of an element in this enumeration can be specified by $\log N_n = O(n)$ bits (again, due to Claim 1). Thus, every value of \mathcal{E} can be described by an algorithm of size $O(n)$, which is the claim (a) of the theorem.

The arguments used in the proofs of (a) and (b) obviously relativizes, and we get statement (c) for computations with an oracle. \square

Proposition 3 provided us with another necessary condition for soficness. To prove that a shift is not sofic, it is enough to construct a computable (or exp-time computable) \mathcal{o} -epitome function with super-linear resource-bounded Kolmogorov complexities.

Remark 9. When we provide an algorithm with an access to an oracle, its computational power can increase dramatically. Thus, when we say that $C^{\mathcal{O}}(w) = O(n)$ with some (unspecified) oracle \mathcal{O} , we do not say much about string w . However, for every oracle we still can say that there are at most $2^{O(n)}$ objects of complexity $O(n)$. So when we claim that $C^{\mathcal{O}}(\mathcal{E}(P)) = O(n)$ for all P , we say essentially that there are $2^{O(n)}$ values of epitomes \mathcal{E} . In other words, the bounds on Kolmogorov complexity with an oracle (and without a time bound) is only a different language for the usual counting argument. On the other hand, the meaning of Kolmogorov complexity becomes much subtler when we constrain the computational resources of algorithms in use; the claims $C^{T(n)}(w) = O(n)$ or $C^{T(n), \mathcal{O}}(w) = O(n)$ cannot be reduced to a simple counting.

Example 5 (semi-mirror shift). In this example we define a shift space extending the one defined in Example 1. Let Σ be the alphabet with three letters (e.g., *black*, *white*, and *red*); the admissible configurations of the shift are all black-and-white configurations (without any red letter) and the configurations with an infinite horizontal line of red letters and black-and-white half-planes above and below this line that are *semi-symmetric* in the following sense: if the letter in the i th column at the distance k below the horizontal red line is black, then the letter in the same i th column at the distance k above the red line must be also black. In other words, we may take symmetric half-planes above and below the red line, and then convert some of the black letters in the lower half-plane into white ones, see Fig. 6. We denote this shift by $\mathcal{S}_{\text{semi-mirror}}$.

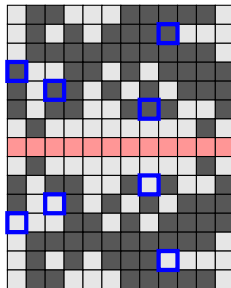


Figure 6: A configuration of the shift $\mathcal{S}_{\text{semi-mirror}}$ (we show in blue the positions where the upper and lower half-planes mismatch).

It is easy to see that this shift is effective: the forbidden patterns are those where the red letters are not horizontally aligned, and those where the areas above and below the horizontal red line do not satisfy the property of semi-symmetry. Below we prove that this shift is not sofic. To this end we use the technique of ordered epitomes.

We define the epitome function \mathcal{E} trivially for the patterns of size $n \times n$ that contain only black and white letters. For these patterns, we let $\mathcal{E}(P)$ be P (or its binary encoding). For patterns with at least

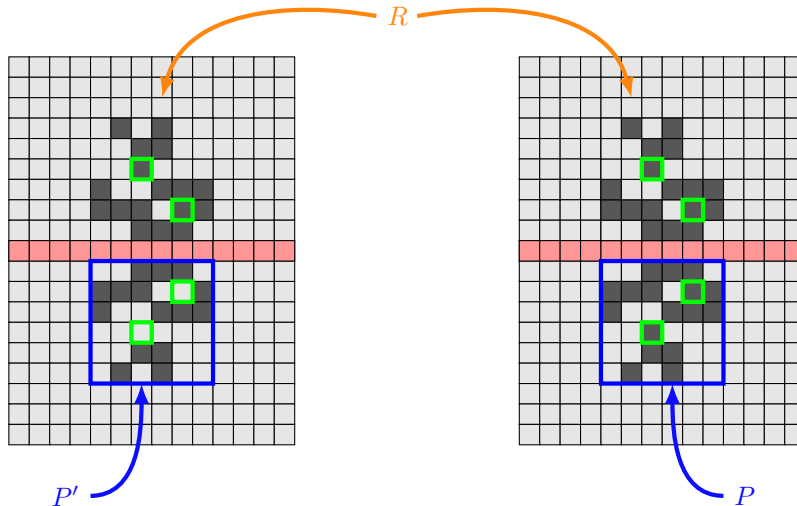


Figure 8: On the right: a black-and-white pattern P (in the blue frame) joined with the corresponding pattern R with the complementary domain (outside of the blue frame). On the left: another black-and-white pattern P' (such that $P' \preceq P$) joined with the same R . We show in green the positions that distinguish P and P' , and the corresponding positions in R (symmetric with respect to the red line).

one red letter, \mathcal{E} is not defined. With this definition, for each $n > 0$ there are 2^{n^2} possible values of $\mathcal{E}(P)$ for $n \times n$ patterns.

We define the partial order \preceq as follows: $\mathcal{E}(P_1) \preceq \mathcal{E}(P_2)$ if P_1 and P_2 are patterns of the same size $n \times n$, and for every position with a black letter in P_1 the corresponding position in P_2 also contains a black letter, see Fig. 7.

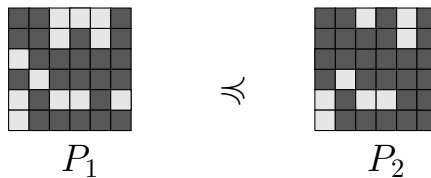


Figure 7: A pair of patterns P_1 and P_2 : the relation $\mathcal{E}(P_1) \preceq \mathcal{E}(P_2)$ means that the set of black position in P_1 is included in the set of black positions in P_2 .

Let us verify that (\mathcal{E}, \preceq) satisfies the definition of an o-epitome function. For every $n \times n$ pattern P with black and white letters we define the corresponding pattern R in exactly the same way as in Example 1 on p. 12, see Fig. 4. It is clear that when we join P with the corresponding R , we obtain an admissible configuration of the shift. Further, if we join the same R with some other pattern P' (as in Fig. 8), we obtain an admissible configuration if and only if P' together with R form a semi-symmetric configuration, i.e., a letter in P' can be black *only if* the letter at the symmetric position in R is black. Hence, R is compatible with P' if and only if $\mathcal{E}(P') \preceq \mathcal{E}(P)$, which is the property required in the definition of an o-epitome function.

We have observed that for every n we have 2^{n^2} possible values of \mathcal{E} . From the counting argument it follows that for every n there exists at least one pattern P of size $n \times n$ such that $C(\mathcal{E}(P)) \geq n^2 \gg n$. Thus, the property $C(\mathcal{E}(P)) = O(n)$ cannot be true for this shift. From Proposition 3 (a) it follows that the shift is not sofic.

A non-degenerate configuration of shift $\mathcal{S}_{\text{semi-mirror}}$ defined in Example 5 consists of two “semi-symmetric” half-planes separated by an infinite horizontal line of red letters. The property of semi-symmetry means that the lower half-plane can be obtained from the upper half-plane one by the composition of the usual mirror symmetry and repainting of some black letters into white ones. In that definition we do not restrict the number of black letters that are converted in white ones: the two half-planes can be

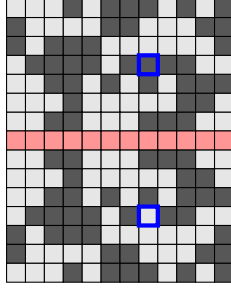


Figure 9: A configuration of the shift $\mathcal{S}'_{\text{semi-mirror}}$ (we show in blue the unique pair of symmetric positions where the upper and lower half-planes mismatch).

exactly symmetric or, conversely, the lower half-plane can consist of only white letters. It is instructive to investigate subshifts of $\mathcal{S}_{\text{semi-mirror}}$ where the mismatch between the upper and lower half-planes is under control. In the following two examples we briefly discuss two extremal cases: a subshift in $\mathcal{S}_{\text{semi-mirror}}$ where the symmetry can be broken for at most one pair of letters, and another subshift where the symmetry is broken for all black letters except (possibly) one.

Example 5' (semi-mirror shift with very small discrepancy between half-planes). Let us define a new shift $\mathcal{S}'_{\text{semi-mirror}}$ (such that $\mathcal{S}'_{\text{semi-mirror}} \subset \mathcal{S}_{\text{semi-mirror}}$) by adding to the definition of $\mathcal{S}_{\text{semi-mirror}}$ another condition saying that the symmetry between the two half-planes (above and below the line of red letters) is violated *for at most one pair of letter*, as shown in Fig 9. It is easy to verify that this shift is effective. Exactly the same argument as in Example 5 implies that $\mathcal{S}'_{\text{semi-mirror}}$ is also non-sofic.

Example 5'' (semi-mirror shift with very large discrepancy between half-planes). Let us define one more shift $\mathcal{S}''_{\text{semi-mirror}}$ (again $\mathcal{S}''_{\text{semi-mirror}} \subset \mathcal{S}_{\text{semi-mirror}}$) by the condition that the half-plane under the horizontal line of red letters contains *at most one black letter*, see Fig. 10. This shift is still effective. With the new restriction, the argument from Example 5 does not apply. Indeed, as we have at most one black letter in the lower half-plane, Kolmogorov complexity of the epitomes $\mathcal{E}(P)$ defined Example 5 sinks to $O(\log n)$, and we cannot get a contradiction with Proposition 3 (a). The fact that our proof of non-soficness fails is not misleading: it is not hard to verify that $\mathcal{S}''_{\text{semi-mirror}}$ is actually sofic. We discuss this example below in Section 4.3 in the context of extender sets.

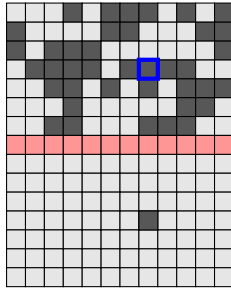


Figure 10: A configuration of the shift $\mathcal{S}''_{\text{semi-mirror}}$. We show in blue the position in the upper half-plane that is symmetric to the unique black letter in the lower half-plane.

In the next example we discuss a shift with very low block complexity, so its non-soficness can be proven only with help of time-bounded Kolmogorov complexity of ordered epitomes.

Example 6 (semi-mirror shift with low block complexity). In this example we use again subshift \mathcal{S}_0 introduced in Notation 5, p. 7. We define a new shift $\mathcal{S}^0_{\text{semi-mirror}}$ (such that $\mathcal{S}^0_{\text{semi-mirror}} \subset \mathcal{S}'_{\text{semi-mirror}}$) by adding to the definition of $\mathcal{S}'_{\text{semi-mirror}}$ the following requirement: every black-and-white pattern below the horizontal line of red letters must be globally admissible in \mathcal{S}_0 .

It is easy to verify that $\mathcal{S}^0_{\text{semi-mirror}}$ is also effective. By the construction, for every globally admissible pattern P of size $n \times n$ that can appear below the horizontal line of red letters, we have $C(P) = O(\log n)$. The same property holds for the patterns that appear above the horizontal line of red letters, since the mirror symmetry and adding one black letter increases Kolmogorov complexity by at most $O(\log n)$. The

same is true for globally admissible patterns involving red letters, since such a pattern can be described by its parts above and below the horizontal red line, and both these parts have Kolmogorov complexity $O(\log n)$. Thus, for *every* globally admissible pattern P of size $n \times n$ we have $C(P) = O(\log n)$. Hence, there are at most $2^{O(\log n)} = \text{poly}(n)$ globally admissible patterns, i.e., the shift has only polynomial block complexity.

The technique of ordered epitomes with the plain Kolmogorov complexity does not apply to this shift since for every globally admissible pattern P the value of $C(P)$ is very low, so $C(\mathcal{E}(P))$ is also sub-linear, and we cannot obtain a contradiction with Proposition 3 (a). However, we can prove non-soficness of this shift with help of time bounded Kolmogorov complexity. Indeed, the shift is defined in such a way that for some $n \times n$ patterns that appear below the horizontal line of red we have $C^{2^{n^3}}(\mathcal{E}(P)) = \Omega(n^{1.5})$. So we can repeat the argument from Example 5 with the same o-epitome function \mathcal{E} (it is exp-time computable) but this time using Kolmogorov complexity with a time bound $T = 2^{\omega(n^2)}$. By applying Proposition 3 (b), we conclude that the shift is non-sofic.

In the next example we discuss an interesting instance of a shift proposed by Kass and Madden in [6, Example 2.5]. We reformulate the proof of non-soficness given in [6] in the language of Kolmogorov complexity, in terms of ordered epitomes. In this case, the definition of an epitome function is less straightforward than in the previous examples.

Example 7 (the shift with no hidden red-black squares). Let Σ be the alphabet with three letters (e.g., *black*, *white*, and *red*), and the forbidden patterns be all squares (of all sizes) where the top side consists of red letters, and the bottom one consists of black letters (*hidden red-black squares*), as shown in Fig. 11a.

Proposition 4 ([6]). *The shift on \mathbb{Z}^2 defined by the set of forbidden patterns specified above is not sofic.*

In [6] this proposition was proven with the technique of *union-increasing chains of extender sets*. In what follows we propose essentially the same argument, but explain it in terms of ordered epitomes.

Proof of Proposition 4. We define for this shift an o-epitome function. First of all, we define a class of *simple patterns*: the simple patterns are all square patterns that (i) consist of only black and white letters (with no red letters), where (ii) every row starts with a few successive black letters followed by a sequence of white letters, as show in Fig. 11b. Every simple pattern of size $n \times n$ can be specified by its *profile* — a tuple of integers (k_1, \dots, k_n) , where k_i is the number of black letters in the i -th row of the pattern. (Thus, a simple pattern with the profile (k_1, \dots, k_n) is an $n \times n$ square where each i -th row starts with k_i black letters followed by $(n - k_i)$ white letters.)

Let epitome \mathcal{E} assign to each simple pattern its profile, and be undefined for all other patterns. For example, for the pattern P show in Fig. 11b we have $\mathcal{E}(P) = (4, 3, 8, 5, 4, 2, 4, 6)$.

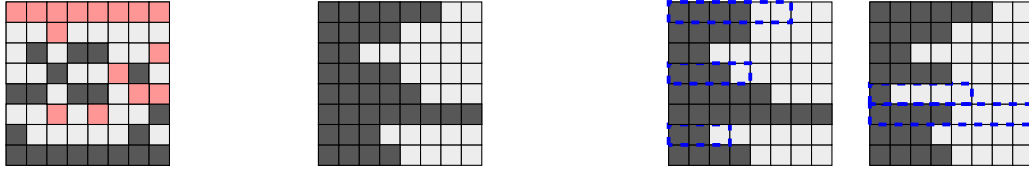
We introduce the natural order \preceq on the profiles of simple patterns of the same size $n \times n$; we say that the profile of P_1 is *not greater* than the profile of P_2 , if the first profile is coordinate-wise not greater than the second one. For example, the profiles of the two patterns shown in Fig. 11c are not greater than the profile of the pattern in Fig. 11b (and incomparable with each other).

The introduced \mathcal{E} and \preceq are obviously computable, even in polynomial time. Some work is required to show that (\mathcal{E}, \preceq) satisfies Definition 3:

Lemma 1. *The defined above (\mathcal{E}, \preceq) provides an exp-time computable o-epitome function for the shift under consideration.*

This lemma is proven implicitly in [6]. We sketch this proof in C.

It remains to observe that for every n there are $(n+1)^n$ simple patterns of size $n \times n$ (in each row of a simple pattern the frontier between black and white areas varies between 0 and n). Therefore, for some simple patterns P of size $n \times n$ the Kolmogorov complexity of their profile is greater than $n \log(n+1)$, i.e., even the plain Kolmogorov complexity $C(\mathcal{E}(P))$ is super-linear. We apply Proposition 3 (a) and conclude that the shift is not sofic. \square



(a) Forbidden pattern: a square with a red top and a black bottom.

(b) A pattern for which the epitome \mathcal{E} is defined: each row starts with a few black letters on the left followed by white letters on the right.

(c) A pair of incomparable patterns.

Figure 11

4.3 The technique of ordered epitomes and union-increasing chains of extender sets

In Example 7 we translated in the language of epitomes a proof proposed by S. Kass and K. Madden in [6]. In this section we show that this example is a special case of the general situation: the technique of Kass and Madden is equivalent to a special case of the technique of ordered epitomes. In this special case we require that the epitome functions have decidable domains, and we use the plain Kolmogorov complexity (i.e., with no bounds for computational resources). We begin with two definitions from [6].

Definition 4. Let \mathcal{S} in \mathbb{Z}^2 be a shift and let P be a pattern with support B_n . The extender set of P , denoted $E_{\mathcal{S}}(P)$, consists of all patterns Q on F_n such that the union of P and Q gives a configuration in \mathcal{S} . The family of extender sets for all globally admissible patterns P with support B_n is denoted

$$Ext_{\mathcal{S}}^n := \{E_{\mathcal{S}}(P) \mid P \text{ is a globally admissible pattern on } B_n\}.$$

In some sense, the cardinality of $Ext_{\mathcal{S}}^n$ corresponds to the “information flow” between an $n \times n$ pattern and the rest of the configuration on \mathbb{Z}^2 . Intuitively, this “information flow” crosses the borderline around an $n \times n$ pattern, while the length of this borderline is only $O(n)$. This intuition can be made more precise for SFTs: for every shift of finite type \mathcal{S} on \mathbb{Z}^2 the size of $Ext_{\mathcal{S}}^n$ cannot be bigger than $2^{O(n)}$. However, this property is not necessary true for multi-dimensional sofic shifts. In fact, it is known that for some sofic shifts on \mathbb{Z}^2 the size of $Ext_{\mathcal{S}}^n$ can grow faster than $2^{O(n)}$ (see Remark 1). For instance, for the sofic shift in Example 5'' the size of $Ext_{\mathcal{S}}^n$ grows as $2^{\Omega(n^2)}$.

Thus, to show that a shift is not sofic, it is not enough to study the *size* of the family of extender sets, we need to reveal subtler properties of these objects. More specifically, we are going to use the structure of the class of extender sets imposed by the relation of inclusion. Technically, we use union-increasing chains of extender sets.

Definition 5. Let S_1, \dots, S_m be a finite sequence of non-empty sets. This sequence is called *union-increasing* chain if the partial unions of the S_i strictly increase, i.e., if for each $1 \leq i \leq m$

$$S_i \not\subseteq \bigcup_{j=1}^{i-1} S_j.$$

Now we can formulate the theorem of Kass and Madden.

Theorem 5 ([6]). *Let \mathcal{S} in \mathbb{Z}^2 be a shift space. Suppose given any $M > 0$, there exists an $n > 0$, an $m > M^n$, and globally admissible patterns P_1, \dots, P_m with support B_n for which*

$$Ext_{\mathcal{S}}(P_1), \dots, Ext_{\mathcal{S}}(P_m)$$

is a union-increasing chain. Then \mathcal{S} is non-sofic.

Loosely speaking, this theorem claims that the “essential information” that must cross the borderline of an $n \times n$ pattern (in a sofic shift) corresponds not just to the number of extender sets but to the length of union-increasing chains of extender sets. This theorem can be reformulated as follows.

Corollary 2. *Let \mathcal{S} in \mathbb{Z}^2 be a shift space. For every n we choose a sequence of globally admissible patterns $P_1^n, \dots, P_{m(n)}^n$ with support B_n for which the sequence of extender sets*

$$Ext_{\mathcal{S}}(P_1^n), \dots, Ext_{\mathcal{S}}(P_{m(n)}^n)$$

is a union-increasing chain. If $\log m(n) = \omega(n)$, then \mathcal{S} is non-sofic.

This approach is pretty general. Kass and Madden have shown in particular that Theorem 5 allows to express any argument based on the technique of Pavlov ([15, Theorem 1.1]).

We show that every proof of non-soficness based on Theorem 5 can be rephrased in terms of o-epitome functions from Definition 3 with decidable domains and with the plain (resource-unbounded) Kolmogorov complexity, and vice-versa. We prove this equivalence in two steps: first we show that an argument in the language of union-increasing chains of extender sets can be translated in the language of o-epitome functions (Proposition 5); then we prove the opposite implication and show that an argument explained in the language of o-epitome functions can be reformulated in terms of union-increasing chains of extender sets (Proposition 6 and Proposition 7).

Next proposition is formulated and proven in terms of families of restricted o-epitomes (let us remind that $(\mathcal{E}_n, \preceq_n)$ is the restriction of (\mathcal{E}, \preceq) onto patterns of size $n \times n$).

Proposition 5. *Let \mathcal{S} in \mathbb{Z}^2 be a shift space. Assume that for every n there is a sequence of globally admissible patterns $P_1, \dots, P_{m(n)}$ with support B_n such that the set of extender sets*

$$Ext_{\mathcal{S}}(P_1), \dots, Ext_{\mathcal{S}}(P_{m(n)})$$

is union-increasing. Then for every n there exists an onto mapping

$$\mathcal{E}_n : [\text{patterns with domain } B_n] \rightarrow \{1, \dots, m\}$$

defined on a subset of globally admissible patterns in \mathcal{S} with support B_n and a partial order \preceq_n on $\{1, \dots, m(n)\}$ such that $(\mathcal{E}_n, \preceq_n)$ satisfies the definition of a family of restricted o-epitome functions for \mathcal{S} .

Proof. The construction of a suitable family of epitome functions is straightforward: we let $\mathcal{E}_n(P_i) = i$ for $i = 1, \dots, m(n)$, and assume that for all other patterns the function \mathcal{E}_n is undefined. We introduce on the values of \mathcal{E}_n the linear order that is inverse to the natural order on the integer numbers,

$$m(n) \preceq_n \dots \preceq_n 2 \preceq_n 1.$$

Let us verify that the obtained $(\mathcal{E}_n, \preceq_n)$ satisfies the definition of a family of o-epitome functions. Since the sequence of extender sets $Ext_{\mathcal{S}}(P_i)$ is union-increasing, for every i we have

$$Ext_{\mathcal{S}}(P_i) \not\subset \bigcup_{j=1}^{i-1} Ext_{\mathcal{S}}(P_j).$$

This means that there exists a pattern R with support F_n that is compatible with P_i (i.e., belongs to $Ext_{\mathcal{S}}(P_i)$) but not with P_1, \dots, P_{i-1} (i.e., does not belong to $\bigcup_{j=1}^{i-1} Ext_{\mathcal{S}}(P_j)$). Thus, for every $n \times n$ pattern P' compatible R , if $\mathcal{E}_n(P')$ is defined then $\mathcal{E}_n(P') \preceq_n i$, which is exactly the definition of restricted o-epitome function. \square

Assume that for every positive integer number n we have a sequence of globally admissible patterns $P_1^n, \dots, P_{m(n)}^n$ with support B_n such that the corresponding sequence of extender sets

$$Ext_{\mathcal{S}}(P_1^n), \dots, Ext_{\mathcal{S}}(P_{m(n)}^n)$$

is a union-increasing chain. By applying Proposition 5 we obtain a family of restricted o-epitome functions \mathcal{E}_n that have $m(n)$ values for patterns of size $n \times n$. In general, this epitome function can be non computable. However, due to a trivial reason it is computable with the oracle \mathcal{O} that contains a description of the sequences $P_1^n, \dots, P_{m(n)}^n$.

If the number $\log m(n)$ grows faster than a linear function (i.e., $m(n) = 2^{\omega(n)}$, which is necessary to apply Corollary 2), then from the counting argument it follows that Kolmogorov complexity of $\mathcal{E}_n(P_i^n)$ (even with oracle \mathcal{O}) is not bounded by a linear function. More precisely, for every $\lambda > 0$ and for a large enough n there is a pattern P_i^n such that $C^{\mathcal{O}}(\mathcal{E}_n(P_i^n)) > \lambda n$. So we can apply Proposition 3(c) and conclude that \mathcal{S} is non-sofic.

Thus, if we can prove that a shift \mathcal{S} is non-sofic with the technique of Kass and Madden (Corollary 2), then due to Proposition 5 we can apply Proposition 3 (c) (with Kolmogorov complexity relativized conditional on the chosen above oracle \mathcal{O}) and prove non-soficness of \mathcal{S} with the technique of epitomes.

In the next two propositions we show that the reverse translation (from the language of ordered epitomes to the language of chains of extender sets) is also possible as long as we use o-epitome functions with decidable domains and resource-unbounded Kolmogorov complexity.

Proposition 6. *Let \mathcal{S} be a shift space, $(\mathcal{E}_n, \preceq_n)$ be a family of restricted o-epitome functions for \mathcal{S} , and $m = m(n)$ be the number of images of \mathcal{E}_n computed for globally admissible patterns. Then there exists a family of globally admissible patterns P_1, \dots, P_m such that the chain of extender sets*

$$\text{Ext}_{\mathcal{S}}(P_1), \dots, \text{Ext}_{\mathcal{S}}(P_m)$$

is union-increasing.

Proof. Denote by e_1, \dots, e_m the images of \mathcal{E}_n . Without loss of generality we may assume that for every i

$$e_j \not\preceq_n e_i \text{ for all } j < i \tag{6}$$

(in every finite partially ordered set we can arrange the indices of e_j so that (6) holds true: e_1 should be one of the maximal elements, e_2 must be maximal among the remaining elements, and so on). Then, we fix globally admissible patterns P_i with support B_n such that $\mathcal{E}_n(P_i) = e_i$. By the definition of an epitome function, for each i there exists a pattern R_i with support F_n that is compatible with P_i but not compatible with P_1, \dots, P_{i-1} . Therefore, $\text{Ext}_{\mathcal{S}}(P_i)$ is not contained in the union $\bigcup_{j=1}^{i-1} \text{Ext}_{\mathcal{S}}(P_j)$. Thus, the sequence of extender sets $\text{Ext}_{\mathcal{S}}(P_i)$ is union-increasing, and we are done. \square

Proposition 7. *Let (\mathcal{E}, \preceq) be a computable o-epitome function for an effective shift \mathcal{S} .*

(a) Assume that for an admissible pattern P of size $n \times n$

$$C(\mathcal{E}(P)) = m, \tag{7}$$

and $m \gg \log n$. Then there exist $2^{\Omega(m)}$ different values of \mathcal{E} corresponding to patterns of size $n \times n$ (globally admissible or non-admissible).

(b) Assume the domains of \mathcal{E} are decidable (there is an algorithm that can decide for every n and every pattern P of size $n \times n$ whether $\mathcal{E}(P)$ is defined or not) and that (7) with $m \gg \log n$ holds for at least one globally admissible pattern P of size $n \times n$. Then there exist $2^{\Omega(m)}$ values of \mathcal{E} corresponding to globally admissible patterns of this size.

(c) Both propositions (a) and (b) relativize: they remain true for Kolmogorov complexity relativized with any oracle \mathcal{O} .

Proof. (a) For a given n , we can apply the algorithm computing \mathcal{E} to each pattern of size $n \times n$ and run these computations in parallel. As some of these computations converge, we can write down the revealed values of \mathcal{E} one by one. (If the domain of \mathcal{E} is undecidable, we never know whether the last terminating computation is over, and some of these computations last infinitely long time without any result.) Let N_n be the number of all values that eventually appear. Each value of \mathcal{E} can be specified by its index in this list (in the order of appearance) together with the binary expansions of n . The index (position in the list) can be specified by $\lceil \log N_n \rceil$ bits. Therefore, for every $n \times n$ pattern P we have

$$C(\mathcal{E}(P)) = O(\log N_n) + O(\log n),$$

and the statement (a) follows.

(b) Let N_n be the number of values of $\mathcal{E}(P)$ for all patterns P of size $n \times n$, and let N'_n be the number of values of $\mathcal{E}(P)$ for globally admissible patterns P of this size. Denote $\ell = \lceil \log N'_n \rceil$.

If $N'_n > N_n/2$, the statement (b) follows from (a). Otherwise, we compute $\mathcal{E}(P)$ for all patterns of size $n \times n$ (this is possible since the domain of an epitome function is decidable) and start enumerating non-admissible patterns of the shift. As non-admissible patterns are progressively revealed, we eliminate one by one the values of $\mathcal{E}(P)$ that do not correspond to any non-discarded (and potentially globally admissible) pattern P . We stop this process when there remain exactly 2^ℓ non-eliminated values of epitomes (that can *a priori* represent globally admissible patterns). Now the epitome of every globally admissible P can be specified by the numbers n and ℓ and by the ordinal number of the value $\mathcal{E}(P)$ in the list of 2^ℓ non-eliminated candidates. Such an ordinal number can be represented by a string of ℓ_n bits. Thus,

$$C(\mathcal{E}(P)) = O(\ell) + O(\log n),$$

and we are done.

(c) It is easy to see that the argument remains valid for algorithms accessing an oracle. \square

Assume that we can prove non-soficness of a shift \mathcal{S} with the technique of ordered epitomes from Proposition 3 using Kolmogorov complexity without time bounds. This is possible if we have for \mathcal{S} an o-epitome function (\mathcal{E}, \preceq) that is computable (possibly with some oracle \mathcal{O}) and such that $C(\mathcal{E}(P))$ is not bounded by a linear function. Assume now that the union of domains of \mathcal{E} is a decidable set. Then it follows from Proposition 7 (b) that the number of values of \mathcal{E} for globally admissible patterns grows as $2^{\omega(n)}$. Now we apply Proposition 6 and obtain a list of globally admissible $n \times n$ patterns $P_1^n, \dots, P_{m(n)}^n$ with $m(n) = 2^{\omega(n)}$ such that the chain of extender sets $Ext_{\mathcal{S}}(P_1^n), \dots, Ext_{\mathcal{S}}(P_{m(n)}^n)$ is union-increasing. Hence, we can apply Corollary 2 and prove non-soficness of \mathcal{S} with the argument of Kass and Madden.

Proposition 5 and Proposition 6 show that the technique of union-increasing chains of extender sets is closely connected with the version of the technique of ordered epitomes with a decidable domain and the plain Kolmogorov complexity. On the other hand, it seems that an argument using plain epitome functions with undecidable domains (see Example 4) cannot be translated directly in the language of union-increasing chains of extender sets. We also recall that the method of epitomes is getting stronger when we use resource-bounded version of Kolmogorov complexity. In particular, it helps to handle shifts with sub-exponential block complexity, which is unachievable with Theorem 5 and Corollary 2.

5 Conclusion

To the best of our knowledge, all previously known proofs of non-soficness for multi-dimension effective shifts (including the one proposed in [15]) can be reformulated in terms of union-increasing chains of extender sets, as proposed in [6]. Therefore, these proofs can be rephrased in terms of ordered epitomes with the plain Kolmogorov complexity, as we explained in Section 4.3. The technique of epitomes becomes more powerful if we use resource-bounded Kolmogorov complexity; in particular, it allows to handle shifts with sub-exponential block complexity. So it seems that Kolmogorov complexity helps to capture substantial properties that are *necessary* for soficness. It would be interesting to adapt the technique of resource-bounded Kolmogorov complexity to reveal properties that are *sufficient* for a shift to be sofic.

Open Problem. Find sufficient conditions of soficness that can be formulated in terms of resource-bounded Kolmogorov complexity.

Acknowledgments We thank Bruno Durand, Alexander Shen, and Ilkka Törmä for fruitful discussions. We are grateful to Pierre Guillon and Emmanuel Jeandel for motivating comments. We also thank the anonymous referees of STACS 2019 and JCSS for many insightful comments.

In the source of Fig. 3 we use a Tikz code based on examples by Marco Miani on texample.net and Martin Wegmann on en.wikipedia.org/wiki/PGF/TikZ.

References

- [1] N. Aubrun, S. Barbieri, and E. Jeandel. About the domino problem for subshifts on groups. Sequences, groups, and number theory. *Birkhäuser, Cham*, pages 331–389, 2018.

- [2] B. Durand, L. Levin, and A. Shen. Complex tilings. *The Journal of Symbolic Logic*, 73:2:593–613, 2008.
- [3] F. C. Hennie and R. E. Stearns. Two-tape simulation of multitape Turing machines. *Journal of the ACM*, 13(4):533–546, 1966.
- [4] M. Hochman. On the dynamics and recursive properties of multidimensional symbolic systems. *Inventiones mathematicae*, 176(1):131, 2009.
- [5] E. Jeandel. Propriétés structurelles et calculatoires des pavages. *Habilitation thesis, Université Montpellier 2*, 2011.
- [6] S. Kass and K. Madden. A sufficient condition for non-soficness of higher-dimensional subshifts. *Proceedings of the American Mathematical Society*, 141:11:3803–3816, 2013.
- [7] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of information transmission*, 1:1:1–7, 1965.
- [8] M. Li and P. Vitányi. An introduction to Kolmogorov complexity and its applications. *3rd ed. Springer, New York*, 2008.
- [9] D. Lind and B. Marcus. An introduction to symbolic dynamics and coding. *Cambridge University Press*, 1995.
- [10] M. Morse and G. A. Hedlund. Symbolic dynamics. *Amer. J. Math.*, 60:815–866, 1938.
- [11] M. Morse and G. A. Hedlund. Symbolic dynamics II: Sturmian trajectories. *Amer. J. Math.*, 62:1–42, 1940.
- [12] S. Mozes. Tilings, substitution systems and dynamical systems generated by them. *Journal d’analyse mathématique (Jerusalem)*, 53:139–186, 1989.
- [13] André Nies. *Computability and randomness*, volume 51. OUP Oxford, 2009.
- [14] N. Ormes and R. Pavlov. Extender sets and multidimensional subshifts. *Ergodic Theory and Dynamical Systems*, 36:3:908–923, 2016.
- [15] R. Pavlov. A class of nonsofic multidimensional shift spaces. *Proceedings of the American Mathematical Society*, 141:3:987–996, 2013.
- [16] Hartley Rogers Jr. *Theory of recursive functions and effective computability*. MIT Press, 1987.
- [17] A. Rumyantsev and M. Ushakov. Forbidden substrings, Kolmogorov complexity and almost periodic sequences. *In Proc. Annual Symposium on Theoretical Aspects of Computer Science*, pages 396–407, 2006.
- [18] V. Salo. Subshifts with sparse projective subdynamics. *arXiv preprint*, arXiv:1605.09623, 2016.
- [19] Alexander Shen, Vladimir A. Uspensky, and Nikolay Vereshchagin. *Kolmogorov complexity and algorithmic randomness*, volume 220. American Mathematical Soc., 2017.
- [20] B. Weiss. Subshifts of finite type and sofic systems. *Monatsh. Math.*, 77:462–474, 1973.
- [21] L. B. Westrick. Seas of squares with sizes from a Π_1^0 set. *Israel Journal of Mathematics*, 22:1, 2017.

A Proof of Theorem 3

It is enough to prove the theorem for shifts of finite type (since a configuration in a sofic shift is a coordinate-wise projection of a configuration from a shift of finite type).

We fix an SFT (which can be defined by a set of forbidden patterns F where every pattern in F is a pair of neighboring letters) and show that for every k there exists a locally admissible $(2^k + 1) \times (2^k + 1)$ -pattern P_k with the required property: every $n \times n$ square pattern inside P_k has Kolmogorov complexity $O(n)$.

To this end we choose an arbitrary coloring of the border line of a square of size $(2^k + 1) \times (2^k + 1)$ so that this coloring can be extended in at least one way to a locally admissible coloring of the entire square, see Fig. 12 (a). Since the set of all colorings of the square is finite, we can find algorithmically (by a naive brute-force search) the lexicographically first coloring of the horizontal and vertical centerlines of the square that are compatible with the fixed coloring of the border line (i.e., the coloring of the border line together with coloring of the centerlines can be extended to a locally admissible coloring of the whole square), see Fig. 12 (b). Observe that the centerlines split the square into four squares of size $(2^{k-1} + 1) \times (2^{k-1} + 1)$.

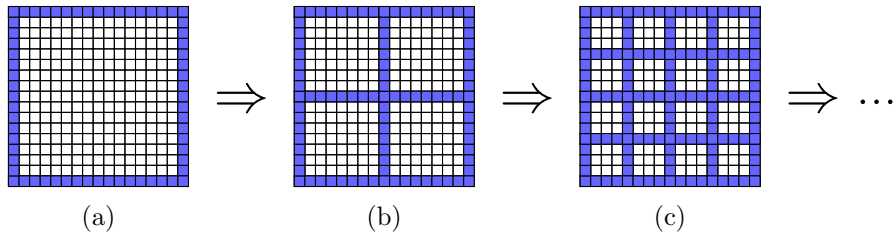


Figure 12: Three steps of the recursive coloring procedures for a pattern of size $(2^n + 1) \times (2^n + 1)$.

Then, we repeat the same procedure recursively: in each of the four squares of size $(2^{k-1} + 1) \times (2^{k-1} + 1)$ we find the lexicographically first coloring of their horizontal and vertical centerlines that are compatible with the coloring of the border around each of these squares (see Fig. 12 (c)), and so on. On the last step we end up with the lexicographically first valid coloring of isolated patterns of size 1×1 (that must be coherent with the chosen above coloring of the neighborhood). This concludes the construction of P_k .

In the procedure explained above, the initial square of size $(2^k + 1) \times (2^k + 1)$ is split into four squares of size $(2^{k-1} + 1) \times (2^{k-1} + 1)$, then into sixteen squares of size $(2^{k-2} + 1) \times (2^{k-2} + 1)$, etc. This recursive procedure finds the coloring of each of these squares in “standard” positions given (as a kind of boundary condition) the coloring of the border around this square.

Thus, to find the assigned coloring of a square of size $(2^m + 1) \times (2^m + 1)$ in a “standard” position, the recursive procedure needs to know only the coloring of the border around this square (which requires $O(2^m)$ letters and therefore $O(2^m)$ bits of information); it is not hard to see that the recursive call runs in time

$$2^{O((2^m+1) \times (2^m+1))} + 4 \times 2^{O((2^{m-1}+1) \times (2^{m-1}+1))} + 16 \times 2^{O((2^{m-2}+1) \times (2^{m-2}+1))} + \dots = 2^{O(2^m)}$$

(this is a recursive computation of depth m , with a brute-force search and four recursive calls on each level of the hierarchy).

An arbitrary square of size $n \times n$ (possibly in a non-standard position) is covered by at most four “standard” squares (of size at most twice bigger than n), see Fig. 13.

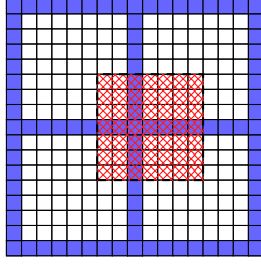


Figure 13: A pattern of size 7×7 (hatched in red) covered by a quadruple of “standard” blocks of size 9×9 .

Thus, to identify an $n \times n$ -pattern inside P_k , it is enough to describe the quadruple of standard squares covering this pattern and, in addition, the position (the coordinates of the corners) of this pattern with respect to the covering standard squares. In turn, each standard square is, by construction, determined by its border line. Hence, every $n \times n$ square in P_k can be specified with only $O(n)$ bits of information; moreover, it can be recovered from this description in time $2^{O(n^2)}$ (at first we reconstruct the four standard squares from their border lines, and then cut out the pattern with the given coordinates of the corners).

To conclude the proof of the theorem, we observe that P_k are defined for arbitrarily large k , and the compactness argument gives a valid coloring of the entire \mathbb{Z}^2 with the required property.

B Proof of Theorem 4

The high level scheme of the proof of Theorem 4 is similar to the proof of Theorem 4'. At first we construct “standard” building blocks of growing size so that all patterns in these blocks have a large time-bounded Kolmogorov complexity. Then we define a shift as the closure of these standard patterns.

Stage 1: building standard blocks. We fix parameters n_i , $i = 0, 1, 2, \dots$

$$n_0 \gg 1, \quad n_{i+1} = (n_0 \cdot \dots \cdot n_i)^c \quad (8)$$

(the constant c to be specified later) and

$$N_i := n_0 \cdot \dots \cdot n_i.$$

In what follows we define inductively *standard blocks* Q_i^j of size $N_i \times N_i$ for $i = 0, 1, 2, \dots$. The major difference between this proof and the proof of Theorem 4' is that for each i we define a set of $\ell_i = n_{i+1}^2$ standard blocks of level i

$$Q_i^1, \dots, Q_i^{\ell_i}$$

(instead of only two of standard patterns used in the previous proof). The construction is hierarchical: each standard block of rank i is defined as an $n_i \times n_i$ array of standard blocks of level $(i - 1)$. In this inductive construction, we maintain for each i the following property:

Main Property: For every $i > 0$, for the list of standard blocks $Q_i^1, \dots, Q_i^{\ell_i}$, we have that

$$C^{t_i}(Q_i^1, \dots, Q_i^{\ell_i}) \geq \ell_i \log(\ell_{i-1}!) = (n_{i+1})^2 \cdot \log((n_i^2)!) \quad (9)$$

(the time threshold t_i to be specified later).

Our construction is explicit, and the standard blocks Q_i^j are computable (given i and j) though the time required for this computation can be pretty large.

Remark 10. Let us explain the intuition behind this construction. We inductively construct standard blocks that have the following two properties. On the one hand, all large enough patterns in each standard $N_i \times N_i$ blocks should have a high time-bounded Kolmogorov complexity due to the “local structure”: each of the standard blocks of rank $(i - 1)$ (of size $N_{i-1} \times N_{i-1}$) is complex, and these blocks are independent of each other. On the other hand, the whole standard block of size $N_i \times N_i$ should have a high time bounded Kolmogorov complexity (with even a bigger time bound) due to the information

embedded in its “global structure” (how the blocks of level $(i - 1)$ are arranged inside of a block of level i). To carry out the inductive argument, we construct for each i a large family of standard blocks of size $N_i \times N_i$, and these blocks must be in some sense “independent.”

Base of induction: We defined blocks $Q_0^1, \dots, Q_0^{\ell_0}$ as arbitrary distinct binary matrices of size $n_0 \times n_0$ (we can do it assuming that $2^{n_0^2} > \ell_0 = n_1^2$).

Inductive step: Given a set of standard patterns $Q_i^1, \dots, Q_i^{\ell_i}$ of size $N_i \times N_i$, we construct the standard patterns of the next level, $Q_{i+1}^1, \dots, Q_{i+1}^{\ell_{i+1}}$ of size $N_{i+1} \times N_{i+1}$. Each new pattern Q_{i+1}^j is defined as an $n_{i+1} \times n_{i+1}$ array composed of blocks $Q_i^1, \dots, Q_i^{\ell_i}$.

Since $\ell_i = n_{i+1}^2$, we may require that each block Q_i^k is used exactly once in every Q_{i+1}^j . In other words, every standard block Q_{i+1}^j can be represented by a permutation

$$\pi_j : \{1, \dots, \ell_i\} \rightarrow \{1, \dots, \ell_i\}$$

that arranges the set of standard patterns of the previous level.

There are $(\ell_i!)$ possibilities to choose each permutations π_j and $(\ell_i!)^{\ell_{i+1}}$ possibilities to choose all permutations $\pi_1, \dots, \pi_{\ell_{i+1}}$. From a trivial counting argument it follows that there exists a tuple of permutations $\pi_1, \dots, \pi_{\ell_{i+1}}$ such that

$$C(\pi_1, \dots, \pi_{\ell_{i+1}}) \geq \ell_{i+1} \log(\ell_i!)$$

Therefore, for every computable threshold h_i , we can algorithmically find a tuple of permutations such that

$$C^{h_{i+1}}(\pi_1, \dots, \pi_{\ell_{i+1}}) \geq \ell_{i+1} \log(\ell_i!).$$

Since each π_j is uniquely determined by the corresponding pattern Q_{i+1}^j , we obtain

$$C^{h_{i+1} - \text{gap}_{i+1}}(Q_{i+1}^1, \dots, Q_{i+1}^{\ell_{i+1}}) \geq \ell_{i+1} \log(\ell_i!),$$

where gap_{i+1} is the time required to extract the permutations π_j from the corresponding patterns Q_{i+1}^j .

Thus, if the threshold h_{i+1} is much bigger than t_i , we conclude that

$$C^{t_i}(Q_{i+1}^1, \dots, Q_{i+1}^{\ell_{i+1}}) \geq \ell_{i+1} \log(\ell_i!),$$

i.e., the *Main Property* holds true for the level $i + 1$.

Lemma 2. *For every $i > 0$ and every tuple of k (pairwise different) standard blocks $Q_i^{j_1}, \dots, Q_i^{j_k}$, we have that*

$$C^{t'_i}(Q_i^{j_1}, \dots, Q_i^{j_k}) \geq \frac{1}{2}k \log(\ell_{i-1}!)$$

(the threshold t'_i is specified in what follows).

Remark 11. The factor of $\frac{1}{2}$ in this lemma could be changed to any constant less than 1.

Proof of lemma. Assume for the sake of contradiction that for some blocks $Q_i^{j_1}, \dots, Q_i^{j_k}$ we have

$$C^{t'_i}(Q_i^{j_1}, \dots, Q_i^{j_k}) < \frac{1}{2}k \log(\ell_{i-1}!)$$

Then we can provide the following description of the entire list of standard blocks of level i :

- a description of size $\frac{1}{2}k \log(\ell_{i-1}!)$ for these particular k blocks,
- $k \log \ell_i$ bits to specify the indices j_1, \dots, j_k of these particular k block,
- a straightforward description for the other standard blocks of level i , which requires $(\ell_i - k) \log(\ell_{i-1}!)$ bits.

We need to add an overhead of size $O(\log k + \log i)$ to join these parts in one description. It remains to observe that the summarized length of these data is less than $\ell_{i+1} \log(\ell_i!)$ (the right-hand side of (9)).

Let us estimate the time required to produce the list of all standard blocks of level i using this description. First of all, we compute the list of all standard blocks of lower rank Q_{i-1}^j (we have to assume that the time required to compute these blocks is much less than t_i). Then, in time t'_i we obtain the k “peculiar” blocks with a short description. With $\text{poly}(N_i)$ steps we generate the remaining $(\ell_i - k)$ standard blocks, and in $\text{poly}(N_i)$ more steps we merge all blocks in one list. Assuming that t_i is large enough, we get a contradiction with the *Main Property* of standard blocks. \square

Lemma 3. *For every standard block of rank i , the plain Kolmogorov complexity of Q_i^j is very low:*

$$C(Q_i^j) = O(\log N_i).$$

Proof. There are ℓ_i standard blocks of level i , and the list of all standard blocks can be computed given i . Therefore,

$$C(Q_i^j) = O(\log \ell_i).$$

For the chosen parameters we have $\log \ell_i = \log(n_{i+1}^2) = O(\log N_i)$. \square

Observe that the construction of standard blocks is explicit, and we can choose a computable function $\hat{T}(n)$ so that all standard blocks are computed given i in time $\hat{T}(n)$. This observation implies the statement of Remark 2.

Stage 2: complexity of patterns inside standard blocks. Let P be an array of $m \times m$ (pairwise distinct) standard blocks of level i . Observe that the size of P (measured in individual letters) is $k \times k$, where $k = mN_i$.

Due to Lemma 2, we have

$$C^{t'_i}(P) \geq \frac{1}{2} m^2 \log(\ell_{i-1}!) = \Omega(m^2 \ell_{i-1}) = \Omega((mn_i)^2).$$

In other words, the time-bounded Kolmogorov complexity of this pattern of size $k \times k$ is bigger than

$$\Omega((mn_i)^2) = \Omega(k^2 / (n_0 \cdot \dots \cdot n_{i-1})^2).$$

Stage 3: the closure of standard blocks. We define a shift as the closure of the standard blocks: a finite pattern is globally admissible if and only if it appears in a 2×2 array composed of standard blocks (of some level i). Due to the hierarchical structure of standard blocks, we can conclude that an $N_i \times N_i$ pattern is *not* globally admissible if it never appears in 2×2 arrays composed of standard blocks of rank i .

Let P be a globally admissible pattern of size $k \times k$ in the shift defined above. Let $2N_i \leq k < 2N_{i+1}$ for some $i > 0$. Then P must be covered by a 2×2 array of standard blocks of level $i + 1$. Therefore, a constant fraction of P can be represented as an array of blocks of level i (inside a standard block of level $(i + 1)$). We can apply the bound from Stage 2 and conclude that

$$C^{T_i}(P) = \Omega(k^2 / (n_0 \cdot \dots \cdot n_{i-1})^2) \tag{10}$$

(assuming that the gap between t'_i and T_i is large enough).

On the other hand, the plain Kolmogorov complexity of every globally admissible block is very low. Indeed, to describe a globally admissible block P of size k , we need to specify four standard blocks covering P and the position of P with respect to the grid of standard blocks. Due to Lemma 3, we obtain

$$C(P) = O(\log k).$$

Stage 4: the choice of parameters. We chose constants c in (8) so that for $k = \Omega(N_i)$ equality (10) rewrites to

$$C^{T_i}(P) = \Omega(k^{2-\epsilon}).$$

It remains to comment the choice of time bounds. The threshold T_i is given in the theorem. Given T_i , we choose a suitable threshold t'_i (the gap between t'_i and T_i must be large enough, so that the argument

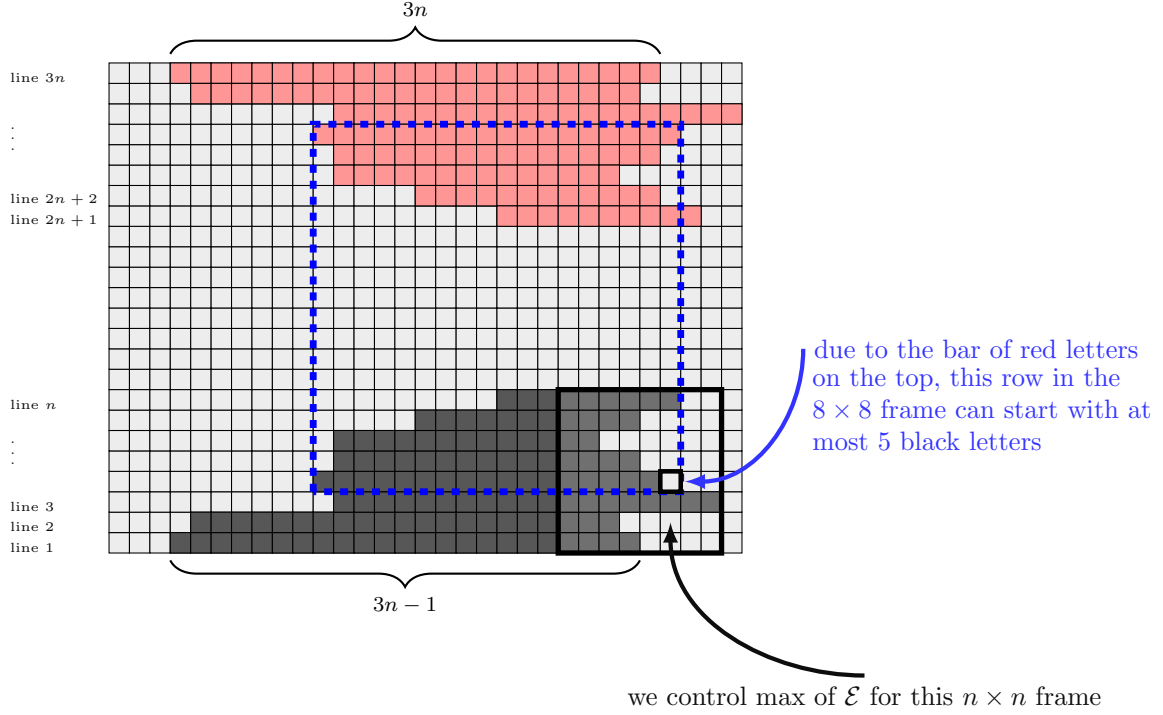


Figure 14: An $n \times n$ pattern P with a neighborhood that enforces the desired maximum of \mathcal{E} .

on Stage 3 works). Then, given t'_i we choose a suitable value t_i (the gap between t_i and t'_i must be large enough, so that the proof of Lemma 2 is valid). The choice of t_i determines the value of h_i on Stage 1. The definition of these sequences is inductive: to define t_i we need to know t_{i-1} and the time required to produce the standard blocks of level $(i-1)$ (see the proof of Lemma 2).

As T_i is a computable function of i , we can choose computable sequences t''_i , t'_i , t_i , and h_i . This observation concludes the proof.

C Proof of Lemma 1

In this section we sketch the proof of Lemma 1. For every *simple pattern* P of size $n \times n$ we should construct a configuration R on the complement of B_n , so that

- (i) P and R are compatible,
- (ii) for every other simple pattern P' compatible with R we have $\mathcal{E}(P') \preceq \mathcal{E}(P)$.

To build the required R , we follow the construction from [6].

By definition, each row of P consists of a contiguous sequence of black letters followed by a contiguous sequence of white letters, as shown in Fig. 11b. The pattern R will consist of a finite number of black and red letters (the other letters will be white).

Black letters in R . To construct R , we extend each stripes of black letters in P to the left, so that in the first line we get a contiguous sequence of $(3n-1)$ black letters (including those black letters that belong to P), in the second line a contiguous sequence of $(3n-3)$ black letters, in the third line a contiguous sequence of $(3n-5)$ black letters, etc. In the n -th line we obtain a contiguous sequence of $(n+1)$ black letter, see Fig. 14.

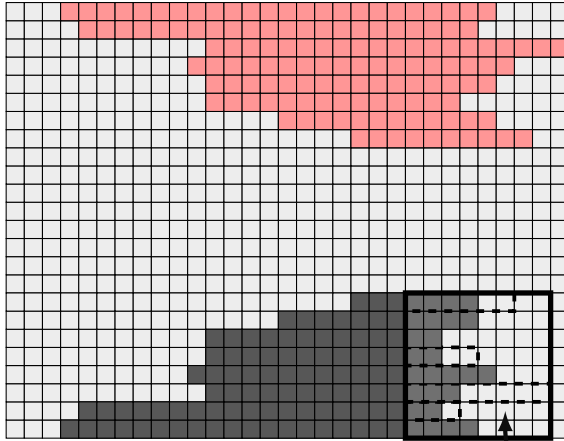
Red letters in R . Similarly, we put in R stripes of red letters: $3n$ contiguous red letters in line $3n$, $(3n-2)$ contiguous red letters in line $3n-1$, \dots , $(n+2)$ contiguous red letters in line $(2n+1)$. We place these stripes of red letters so that for each $i = 1, \dots, n$ the leftmost red letter in the line $(3n-i+1)$ is vertically aligned with the leftmost black letter in the line i , as shown in Fig. 14.

All other letters outside B_n are made white.

Claim 1. The constructed R is compatible with P .

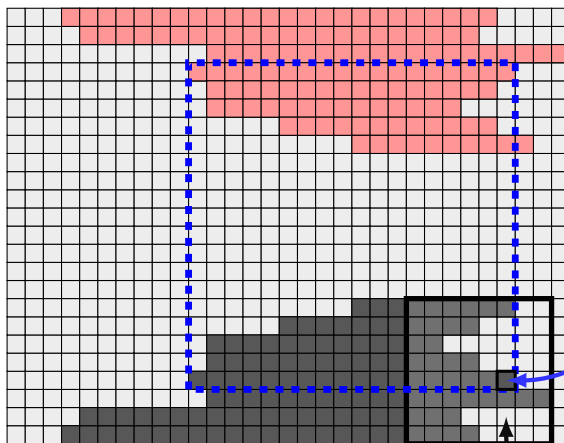
Proof of Claim 1: This fact is easy to verify: we have chosen the lengths of black and red stripes so that they cannot form a forbidden pattern (as in Fig. 11a), regardless the horizontal placement of each stripes. Indeed, on the one hand, the black letters of the i -th line cannot interfere with the red stripes in lines $3n, 3n - 1, \dots, 3n - i$, since *this black stripe* is too short to form a forbidden pattern together with any of these red stripes; on the other hand, the black letters of the i -th line cannot interfere with the red stripes in lines $3n - i - 1, 3n - i - 2, \dots, 2n + 1$, since *those red stripes* are too short. \square

Claim 2. The constructed R is compatible only with simple patterns P' such that $\mathcal{E}(P') \preceq \mathcal{E}(P)$.



this $n \times n$ pattern P' is compatible with the neighborhood

Figure 15: A pattern P' with $\mathcal{E}(P') \preceq \mathcal{E}(P)$ matches the neighborhood.



by adding one supplementary black letter we get a forbidden pattern

this $n \times n$ pattern P'' is incompatible with the neighborhood

Figure 16: A pattern P'' with $\mathcal{E}(P'') \not\preceq \mathcal{E}(P)$ does not match the neighborhood.

Proof of Claim 2: If R is compatible with an $n \times n$ pattern P' , the profile of P' is not determined uniquely. In fact, R can be compatible with simple patterns P' whose profiles are *strictly less* than the profile of P (in each row of P' the number of black letters must be not greater than the number of black letters in the corresponding row of P), see Fig. 15. On the other hand, if at least one row of P' contains more black letters than the same row in P , then P' and R are incompatible, i.e., the joint of P' and R contains a forbidden pattern, as shown in Fig. 16. \square

The lemma follows from Claim 1 and Claim 2. For a more detailed argument we refer the reader to [6].

Remark 12. In the construction discussed above, pattern R does not determine uniquely the epitomes of P' compatible with R (these epitomes can be different, though they all must be *not greater* than the epitome of the initial pattern P). This is why we cannot apply Proposition 2, and we have to use Proposition 3.