Computational Characterization of Surface Entropies for $\mathbb{Z}^2$ Subshifts of Finite Type

Antonin Callard
Université Paris-Saclay, ENS Paris-Saclay, Département Informatique, 91190 Gif-sur-Yvette, France

Pascal Vanier
Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

Abstract

Subshifts of finite type (SFTs) are sets of colorings of the plane that avoid a finite family of forbidden patterns. In this article, we are interested in the behavior of the growth of the number of valid patterns in SFTs. While entropy h corresponds to growths that are squared exponential 2^{hn^2}, surface entropy (introduced in Pace's thesis in 2018) corresponds to the eventual linear term in exponential growths. We give here a characterization of the possible surface entropies of SFTs as the Π_3 real numbers of $[0, +\infty]$.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics; Theory of computation → Models of computation

Keywords and phrases surface entropy, arithmetical hierarchy of real numbers, 2D subshifts, symbolic dynamics

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.122

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Funding This research was partially funded by ANR JCJC 2019 19-CE48-0007-01.

Acknowledgements The authors would like to thank Ronnie Pavlov for answering their many questions about surface entropy when they started this work, and Benjamin Hellouin de Menibus for the relecturing. Finally, we warmly thank the anonymous reviewers for their many helpful remarks and improvements.

1 Introduction

For a finite alphabet of colors, a two dimensional subshift is the set of all colorings of the plane $\mathbb{Z}^2$ that respect some local constraints. These constraints are usually given as a family of forbidden patterns. The most studied class of subshift are *Subshifts of Finite Type* (SFTs), subshifts that can be defined by a finite family of forbidden patterns. A famous special case is defined by Wang tilesets (unit squares with colored edges that may be placed side by side only when the colors on the edges match) are a famous special: the set of all tilings by some Wang tileset is always an SFT.

Subshifts were introduced in order to discretize continuous dynamical systems and are themselves dynamical systems. While it has long been known that most problems concerning Wang tiles (and thus subshifts) are undecidable [2, 5, 6], the role of computability has since shifted from an obstacle to a major tool in the study of SFTs and other related classes of subshifts. An aperiodic subshift has for instance been constructed [4] based on Kleene's *fixed-point theorem* [11], a classical theorem of computability theory. Many conjugacy invariants have been characterized using computability or complexity classes. The first such characterization was for *topological entropy*, which measures the exponential growth of the number of valid colorings of finite patterns. For a subshift X, its (topological) entropy is defined by:

© Antonin Callard and Pascal Vanier;
licensed under Creative Commons License CC-BY 4.0
48th International Colloquium on Automata, Languages, and Programming (ICALP 2021).
Editors: Nikhil Bansal, Emanuela Merelli, and James Worrell; Article No. 122; pp. 122:1–122:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$h(X) = \lim \frac{\log N_X(n)}{n^2}$$

where $N_X(n)$ denotes the number of $n \times n$ patterns appearing in colorings of X. Having entropy h corresponds to having $N_X = 2^{hn^2+o(n^2)}$. It turns out that the possible entropies for $\mathbb{Z}^2$ are exactly the upper semi-computable [8] real numbers. Another invariant related to growth, called *entropy dimension*, was then characterized using the arithmetical hierarchy of real numbers [13]. Many other invariants have since been linked to computability: for instance, there is a relationship between periodic colorings and computational complexity classes [10], subactions [7, 1, 3] can be characterized through recursively enumerable forbidden patterns, and many more.

We focus here on the notion of *surface entropy*, a notion which was introduced in Dennis Pace's thesis [14] in order to quantify the linear term inside exponential growth functions. Indeed, topological entropy cannot distinguish between the two following behaviors of the complexity function:

$$N_X(n) \approx 2^{hn^2} \qquad \text{and} \qquad N_X(n) \approx 2^{hn^2+2sn}.$$

Surface entropy corresponds roughly to the s term in the second behavior and will be introduced more formally in Subsection 2.2. In his thesis, Pace realizes Π_1 and Σ_1 numbers of the arithmetical hierarchy of real numbers and conjectures that surface entropies are exactly the Π_3 numbers. This is exactly what we prove here:

▶ **Theorem 1.** *The class of surface entropies of $\mathbb{Z}^2$ SFTs is $[0, +\infty] \cap \Pi_3$.*

In fact, surface entropies still cover the whole class of Π_3 real numbers for sofic subshifts, which are the letter-by-letter projections of SFTs, and for effective subshifts, which are the subshifts that can be defined with a recursively enumerable family of forbidden patterns:

▶ **Theorem 2.** *The class of surface entropies of $\mathbb{Z}^2$ sofic subshifts, and of $\mathbb{Z}^2$ effective subshifts, is $[0, +\infty] \cap \Pi_3$.*

The paper is organized as follows. The next section recalls some background and useful definitions. Section 3 and Section 4 focus on the proof of Theorem 1, and provide in particular a construction which creates $\mathbb{Z}^2$ SFTs with arbitrary Π_3 surface entropies. Some open questions are then discussed in Section 5.

2 Preliminaries

2.1 Subshifts

This subsection introduces some standard definitions and facts about subshifts. The reader may consult [12] for more details.

Let Σ be a finite alphabet of colors. A $\mathbb{Z}^d$ *configuration* (in this paper, $d = 1$ or $d = 2$) is a coloring $x : \mathbb{Z}^d \mapsto \Sigma$, and the value of x at position z is noted x_z. A *(d-dimensional) pattern* is a coloring $p : D \mapsto \Sigma$, with $D \subseteq \mathbb{Z}^d$ a finite domain. For a configuration x, we say that a pattern p appears in x (noted $p \sqsubseteq x$) if there exists some position $t \in \mathbb{Z}^d$ such that for all $z \in D$, $p_z = x_{t+z}$. A *subshift* is a set of colorings/configurations defined by some family of forbidden patterns. Each family of forbidden patterns $\mathcal{F}$ defines a subshift, possibly empty:

$$X_{\mathcal{F}} = \{(x : \mathbb{Z}^d \mapsto \Sigma) : \forall p \in \mathcal{F}, p \not\sqsubseteq x\}.$$

A subshift is *effective* if it can be defined by a recursively enumerable family of forbidden patterns. A subshift is *of finite type* (SFT) if it can be defined by a finite family of forbidden patterns. A subshift Y on Σ is *sofic* if there exists an SFT X on an alphabet Σ' and a letter-by-letter projection $\pi : \Sigma' \mapsto \Sigma$ that sends X on Y.

Finally, for any $\mathbb{Z}$ subshift X_1, the $\mathbb{Z}^2$ *lift of* X_1 is the subshift Y_2 whose configurations are vertically repeated configurations of X_1:

$$Y_2 = \{(y : \mathbb{Z}^2 \mapsto \Sigma) : \exists x \in X_1, \forall i, j \in \mathbb{Z}, y_{(i,j)} = x_i\}.$$

2.2 Complexity function of $\mathbb{Z}^2$ subshifts

Given a $\mathbb{Z}^2$ subshift X, its *complexity function* $N_X(m, n)$ (for $m, n \in \mathbb{N}$) is the number of different patterns that appear in a rectangle of size $m \times n$ in the configurations of X:

$$N_X(m, n) = \left|\left\{p \in \Sigma^{m \times n} : \exists x \in X, p \sqsubseteq x\right\}\right|.$$

This complexity function can be used to define the (topological) entropy $h(X)$:

$$h(X) = \lim_{n \to +\infty} \frac{\log N_X(n, n)}{n^2}.$$

This led Dennis Pace to introduce in [14] the notion of *surface entropy*, which corresponds to the "linear term" of the complexity function. Here, we define *surface entropy with eccentricity* α as:

$$h_s(X, \alpha) = \limsup_{n \to +\infty} \frac{\log N_X(pn, qn) - pqn^2 h(X)}{(p + q)n} \qquad \text{with } \alpha = \frac{p}{q} \in \mathbb{Q}^+,$$

where p, q are relatively prime integers. This definition slightly differs from the one of [14]; this will be further discussed in Section 5.

Note that in the definition of the topological entropy, only square patterns are used. In fact, any rectangular patterns would generate the same value. Interestingly, this is no longer the case with surface entropy: the *eccentricity* (ratio of the patterns' widths to heights) affects the value of the calculations. This explains why h_s is a function of both a subshift X and a rational parameter $\alpha = p/q$. The study in [14] focuses on both the realizability of specific surface entropies, and the behavior of surface entropies as functions of their eccentricities.

While surface entropy is not a conjugacy invariant, it was proved in [14] that the finiteness of surface entropy is invariant under conjugacy. For more details or examples about surface entropies, one may refer to [14, Section 3.2].

Links with other growth-type invariants

Links with (topological) entropy. If X_1 is a $\mathbb{Z}$ subshift, and Y_2 its $\mathbb{Z}^2$ lift (ie. the subshift whose configuration are vertically repeated configurations of X_1), then $h_s(Y_2, \alpha) = \frac{\alpha}{1+\alpha} h(X_1)$.

Links with entropy dimension. In [13], a growth-type invariant called *entropy dimension* was introduced as $h_d(X) = \limsup_{n \to +\infty} (\log \log N_X(n)) / \log n$. It roughly represents the exponent α if $\log N_X(n) \simeq n^\alpha$. For any $\mathbb{Z}^2$ subshift X, $0 \leq h_d(X) \leq 2$.

Entropy dimension and surface entropies are linked as follows:

- If $h_d(X) = 2$, then $h_s(X)$ can be either finite or infinite (similarly to $h_d(X) = 1$);
- If $1 < h_d(X) < 2$, then $h_s(X) = +\infty$;
- If $h_d(X) = 1$, $h_s(X)$ can either be finite (if $\log N_X(n) = O(n)$, see [14, Example 3.2.14]) or infinite (for example, if $\log N_X(n) \simeq n \log n$, see [14, Example 3.2.4]);
- If $0 \leq h_d(X) < 1$, then $h_s(X) = 0$.

2.3 Arithmetical hierarchy of real numbers

In order to state our main result, this section recalls from [15] the arithmetical hierarchy of real numbers, which classifies elements of the real line according to their computational properties. Denote by $\Gamma_{\mathbb{Q}}$ the set of total computable functions $f : \mathbb{N}^k \mapsto \mathbb{Q}$ for any k. For $n \geq 1$, the classes of real numbers Σ_n, Π_n and Δ_n are defined as follows:

$$\Sigma_n = \{ \sup_{i_1} \inf_{i_2} \sup_{i_3} \dots f(i_1, \dots, i_n) \mid f \in \Gamma_{\mathbb{Q}} \}$$
$$\Pi_n = \{ \inf_{i_1} \sup_{i_2} \inf_{i_3} \dots f(i_1, \dots, i_n) \mid f \in \Gamma_{\mathbb{Q}} \}$$
$$\Delta_n = \Sigma_n \cap \Pi_n.$$

It is known that for any $n \geq 1$, the inclusions $\Pi_n \subset \Sigma_{n+1}$ and $\Sigma_n \subset \Pi_{n+1}$ are proper. One may refer to [15] for more details.

In this paper, we will be interested in the third level of the hierarchy, and one of its equivalent characterization proved in [15]:

$$x \in \Pi_3 \text{ iff there exists } f \in \Gamma_{\mathbb{Q}} \text{ such that } x = \limsup_i \inf_j f(i, j).$$

3 Arithmetical restrictions of surface entropies

In this section, we prove the first and easiest direction of Theorem 1:

▶ **Theorem 3.** *For any $\mathbb{Z}^2$ SFT X and $\alpha \in \mathbb{Q}^+$:*

$$h_s(X, \alpha) \in [0, +\infty] \cap \Pi_3$$

Proof. Let X be a $\mathbb{Z}^2$ an SFT. We prove that $N_X(m, n)$ is a Π_1 real number (it is not computable in X: indeed, $N_X(m, n) = 0$ if and only if $X = \emptyset$; the latter is well-known for being undecidable). A pattern is said to be *admissible* if it does not contain a pattern which is forbidden: any valid pattern of X is admissible, but the converse is false. For $j \geq m, n$, define $N_X^{(j)}(m, n)$ as the number of $m \times n$ patterns that appear at the center of admissible patterns of size $j \times j$ (ie. the squares of size $j \times j$ in which none of the forbidden patterns of X appear).

By compactness, if a pattern is not valid in X, there exists some j such that it does not appear in any admissible pattern bigger than $j \times j$: this proves $N_X(m, n) = \inf_{j \geq m,n} N_X^{(j)}(m, n)$. In particular, this implies that $\log N_X(m, n)$ is a Π_1 real number.

It follows that $h(X)$ is a Π_1 real number (as proved in [8]). As the difference of two Π_1 real numbers is a Δ_2 real number, this proves that for any $\alpha = p/q \in \mathbb{Q}^+$, and for every $n \in \mathbb{N}$, the following is a Δ_2 real number:

$$\frac{\log N_X(pn, qn) - pqn^2 h(X)}{(p + q)n}$$

which then leads to $h_s(X, \alpha) \in \Pi_3$. Finally, by sub-additivity $h_s(X, \alpha) \geq 0$ (see [14]). ◀

A very similar reasoning proves that this is still the case for sofic or effective subshifts:

▶ Remark 4. For any $\mathbb{Z}^2$ sofic or effective subshift X and $\alpha \in \mathbb{Q}^+$, $h_s(X, \alpha) \in [0, +\infty] \cap \Pi_3$.

Proof. As sofic subshifts are effective, we can focus on the latter. For X an effective subshift, we slightly alter the definition of $N_X^{(j)}(m,n)$: it will now count the number of patterns of size $m \times n$ which appear in the squares of size $j \times j$ in which *none of the first j enumerated forbidden patterns* of X appear. Then we still have $N_X(m,n) = \inf_{j \geq m,n} N_X^{(j)}(m,n)$, and this still leads to $\log N_X(m,n) \in \Pi_1$ and $h(X) \in \Pi_1$.

After this, the end of the proof is the same. ◀

4 Realization of Π_3 real numbers as surface entropies

In this section, we prove the other (and harder) direction of Theorem 1. Because [14] provides examples of SFTs with infinite surface entropy, we now prove:

▶ **Theorem 5.** *For any $x \in [0, +\infty) \cap \Pi_3$, there exists an SFT X with surface entropy:*

$$\forall \alpha \in \mathbb{Q}^+, \qquad h_s(X, \alpha) = \frac{\min(\alpha, 1)}{1 + \alpha} \cdot x.$$

Theorem 1 and 2 are consequences of the previous section together with this statement.

A naive construction could consist in lifting $\mathbb{Z}$ subshifts to obtain $\mathbb{Z}^2$ SFTs with specific surface entropies. However, as the topological entropy of a $\mathbb{Z}$ effective subshift is a Π_1 real number (by ideas similar to Section 3), lifts can only have Π_1 surface entropies (cf. the link between $\mathbb{Z}$ entropies and $\mathbb{Z}^2$ surface entropies in Section 2.2).

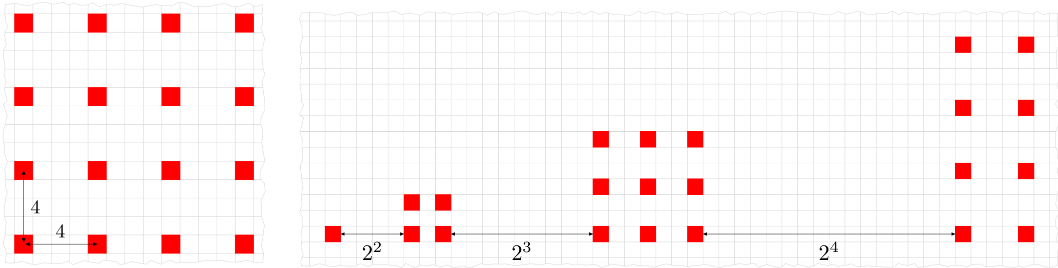
Any method to overcome this limitation would need some creative spatial distribution. Hence the *sparse squares* we develop below.

Proof. Let $x \in [0, +\infty) \cap \Pi_3$ be a Π_3 real number.

Let $e \in \mathbb{N}$ and $x' \in [0, 1)$ be such that $x = e + x'$. Because x' is also a Π_3 real number, there exists a computable function $f : \mathbb{N}^2 \mapsto \mathbb{Q}$ such that $x' = \limsup_k \inf_l f(k, l)$ (see the characterization in Subsection 2.3). We can assume that f only takes values in $[0, 1)$.

In the following subsections, we create an SFT X which verifies the property of Theorem 5. The proof is organized as follows:

1. Subsection 4.1 introduces our "sparse squares" construction, which aims at creating a set of colorings of the plane with controlled surface entropy. All the following sections focus on implementing this geometrical construction into an actual SFT X.
2. Subsection 4.2 recalls the Toeplitz sequences, which are sequences of uniform densities. They will be used in the sparse square layout to control the density of each square.
3. In Subsection 4.3, we create a $\mathbb{Z}$ effective subshift X_1. Effectiveness gives us a lot of room to control its patterns, and we will use X_1 as the foundation of X.
4. We use the *fixed-point* construction of [3] to create a $\mathbb{Z}^2$ SFT which simulates X_1. Subsection 4.4 provides an intermediary lemma about the entropy and the surface entropy of this construction: in our case, it proves that the intermediary construction has surface entropy zero.
5. In Subsection 4.5, we then create the desired SFT X, which arranges the sparse squares on the plane (with the help of the previous points).
6. Finally, in Subsection 4.6 and Subsection 4.7 we create and compute the surface entropy of X. This proves that X is a valid example for Theorem 5.



(a) 4^{th} sparse square. **(b)** (Portion of) the sparse square layout.

Figure 1 Presentation of the sparse square layout.

4.1 The sparse squares and the sparse square layout

To understand the idea behind this construction, consider the *full shift* over $\mathbb{Z}^2$ on the alphabet $\{0, 1\}$. Configurations are full grids of *free bits*, ie. bits that are allowed to vary freely in $\{0, 1\}$. It is not difficult to see that for the full shift, $\log N_{\text{full}}(n, n) = n^2$. In particular, its complexity function is quadratic in n, and its entropy is 1.

To realize specific surface entropies, we first need to figure out a way to contribute *linearly* to $\log N_X$, instead of quadratically. To do this, we create a sequence of *sparse squares*. A sparse square is, roughly, a finite piece taken from the full shift, but whose points are moved apart from one another: the square is *sparsified*.

More precisely, the *sparse square of index* k (see Figure 1a) is a set of positions that form a finite grid. Any position not in this set of points is blank. In the grid, there are k columns (the distance between two columns is also k), and in each column there are k points (the distance between two points in a column is also k).

The *sparse square layout* (see Figure 1b) makes the sparse squares sit next to one another on a single line, according to their indices. We set the distance between the square of index $k-1$ and the square of index k to 2^k.

The key feature of this geometrical layout lies in its linearity: the k^{th} square has edges of size $k(k-1)+1$: thus it has an area which is roughly $\left(k^2\right)^2$, while it also contains k^2 points. In addition, as the distance between two points increases as one considers squares of greater indices, compactness will only lead to degenerate configurations that contain at most one point. As such, they will not contribute significantly to the complexity function.

For now, the set of positions in the sparse square layout is not very interesting. In order to increase the complexity function, we will allow free bits to vary at each position inside a sparse square. Additionally, to create a surface entropy related to x, the density of free bits in each square will be related to x.

More precisely, define $x'_k = \inf_l f(k, l)$ (recall that $x = e + x'$, and $x' = \limsup_k \inf_l f(k, l)$). For each k, let t_k be a word of size k over the alphabet $\{\text{On}, \text{Off}\}$, whose density of On is x'_k. Then, define the subshift X' as (the closure of) the following configurations (see Figure 2):

- These configurations follow the sparse square layout.
- Each position not marked in the sparse square layout is blank (ie. marked with $\square$).
- The word t_k is written in each row of the k^{th} square.

On the positions marked by On, we then allow free bits to vary. On every position of the sparse square, we also allow free letters to vary in $\{1, \ldots, 2^e\}$. With this, the square of index k contributes to $\log N_{X'}$ with a term $k^2(x'_k + e)$. More precisely:

▶ **Lemma 6.** *The k^{th} sparse square contributes to $\log N_{X'}$ more than $k^2(e + x'_k) + O(k)$.*

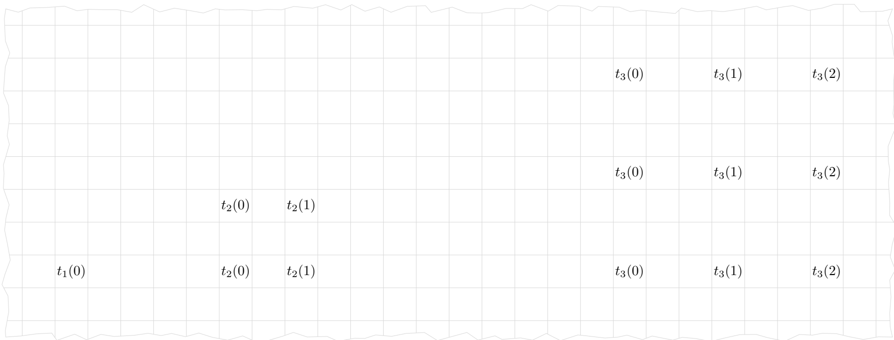


Figure 2 A sequence of words (t_k) written in (a finite piece of) the sparse square layout.

Proof. As this is a lower bound, consider the position of the k^{th} sparse square fixed, and only look at the contribution induced by its "free bits" and "free letters".

Free bits: (For now, we assume the "free letters" are fixed). A word t_k of density x'_k is written in this square, and the square contains $k|t_k|_{\text{ON}}$ free bits: indeed, there are k identical lines, and in each line there are $|t_k|_{\text{ON}}$ positions marked with ON. Additionally, one has $|t_k|_{\text{ON}} = kx'_k + O(1)$ (because the density is x'_k). This implies that all the free bits of the k^{th} sparse square (fixed at this position), when marked with t_k, contribute to the complexity function with a term $\exp_2(k|t_k|_{\text{ON}}) = \exp_2(k^2 x'_k + O(k))$.

Free letters: Recall that free letters vary in the alphabet $\{1, \dots, 2^e\}$ at each point in the k^{th} sparse squares. There are k^2 points in this square, so the previous contribution is multiplied by $\exp_2(k^2 e)$. This concludes the proof. ◀

This provides a lower bound for the complexity function: for a given size of pattern, we only count the biggest square that fits in it. If the squares do not interfere too much, one should expect the surface entropy to "converge" towards $\limsup_k x'_k + e = x' + e = x$.

The previous paragraphs were a draft of a geometrical construction. Below, we create an actual $\mathbb{Z}^2$ SFT X which implements this subshift (with some additional construction lines). Then, we formally prove that X has the desired surface entropy.

4.2 Effective $\mathbb{Z}$ upper-density Toeplitz subshifts

In order to create specific densities of letters in a subshift, we recall the useful Toeplitz sequences from [9]. Let $0 \leq y = \sum_{i=1}^{+\infty} y_i 2^{-i} \leq 1$ be a real number. A *Toeplitz sequence* associated to y is a bi-infinite sequence $b \in \{0,1\}^{\mathbb{Z}}$ such that:

$$\forall k \geq 1, \exists j_k \notin \left(\sum_{i=1}^{k-1} j_i + 2^i \mathbb{Z} \right), \; b_{j_k + 2^k \mathbb{Z}} = y_k$$

i.e. one bit in two is y_1; on the remaining bits, one bit in two is y_2; etc. . .

For any $y \in [0, 1)$, we define the *upper-density Toeplitz subshift* $\mathcal{T}(y)$ *associated to* y:

$$\mathcal{T}(y) = \{(b) \in \{0,1\}^{\mathbb{Z}} : \exists 0 \leq y' \leq y, (b) \text{ is a Toeplitz sequence associated to } y'\}.$$

In the following subsection, we will use subwords of Toeplitz sequences on the alphabet $\{\text{ON}, \text{OFF}\}$ (rather than $\{1, 0\}$). They have high regularity and tightly controlled densities. Indeed, assume that t_n is a factor of length n which appears in $\mathcal{T}(y)$. Then the number of letters ON in t_n is bounded by $0 \leq |t_n|_{\text{ON}} \leq ny + O(1)$.

We will use these subshits in the context of Π_1 real numbers. Indeed:

▶ **Lemma 7.** *If $y \in \Pi_1 \cap [0,1)$, then $\mathcal{T}(y)$ is an effective $\mathbb{Z}$ subshift.*

Proof. If $y \in \Pi_1$, there exists a computable total function $f : \mathbb{N} \mapsto \mathbb{Q}$ such that $y = \inf_n f(n)$. It is possible to recursively enumerate the different values of f, to computably forbid patterns of $\mathcal{T}(y)$ which do not respect the structure of Toeplitz sequences, and to computably forbid patterns of $\mathcal{T}(y)$ that respect the structure of Toeplitz sequences but whose density is too high. ◀

4.3 Building the base line

Consider the line on which all the sparse squares sit in the previous layout. We call it the *base line*. In this section, we create a $\mathbb{Z}$ effective subshift X_1 on the alphabet $\Sigma = \{\#, S, E, B, \textsc{On}, \textsc{Off}\}$ which implements the base line: its configurations will be all the possible values for this base line.

Base of the k^{th} sparse square. The base of the k^{th} sparse square is composed of k points, each separated by $k-1$ blanks (in X_1, these blanks will be marked by B). In each point, a letter On or Off is written, and the word t_k composed by these k letters is a Toeplitz subword of $\mathcal{T}(x'_k)$, where $x'_k = \inf_l f(k,l)$.

This naturally leads to defining the set of all possible bases for the k^{th} sparse square:

$$\mathcal{W}_k = \{t_k(1)\ B^{k-1}\ t_k(2)\ B^{k-1}\ \ldots\ B^{k-1}\ t_k(k) \mid t_k \text{ is a toeplitz subword in } \mathcal{T}(x'_k)\}.$$

Prefixes for the squares. In order to implement the sparse square layout, we will build the sparse squares on top of the base line. In order to build the squares properly, we set a specific prefix before the base of each square. These prefixes have no meaning in terms of surface entropy, and can be considered as construction lines.

In the layout, the positions of these prefixes were blank; in the implementation, they are marked with letters S, B and E, which will have different roles when building other construction lines in Subsection 4.5.

These prefixes are $s_1 = S$, and for $k \geq 2$, $s_k = SB^{k-2}E$.

The whole subshift. We now create a whole base line with these considerations: for any k, the base of the k^{th} sparse square is a word of $\mathcal{W}_k$; before the base of the k^{th} sparse square, we add the word s_k; and the other positions are marked with blanks $\#$.

For the rest of the paper, we denote X_1 as the closure of the following configurations:

$$X_1 = \text{cl}\{\#^\infty s_1 w_1\ \#^{2^2-2}\ s_2 w_2\ \ldots\ s_{k-1} w_{k-1}\ \#^{2^k-k}\ s_k w_k\ \cdots \mid \forall k, w_k \in \mathcal{W}_k\}$$

(for ease of understanding, we highlight the base of each sparse square in red).

The construction below relies on the fact that X_1 (ie. the set of all possible base lines) is a $\mathbb{Z}$ effective subshift. Indeed, it is possible to computably enumerate all the patterns in which at least two letters of the set $\{\textsc{On}, \textsc{Off}\}$ appear, and to forbid each of these patterns that do not respect the structure of the configurations above. Furthermore, by definition each $x'_k = \inf_l f(k,l)$ is a Π_1 real number, and its upper-density Toeplitz subshift $\mathcal{T}(x'_k)$ is effective. We conclude that X_1 is an effective $\mathbb{Z}$ subshift.

4.4 Entropy of the fixed-point realization of effective $\mathbb{Z}$ subshifts as subactions of $\mathbb{Z}^2$ SFTs

We now have a $\mathbb{Z}$ subshift X_1 which can be used as a foundation in order to implement the whole sparse square layout into an actual SFT. In this section, we recall a particular method (from [3]) which transforms $\mathbb{Z}$ effective subshifts into $\mathbb{Z}^2$ SFTs. Additionally, we compute how much this method impacts the surface entropy.

The construction of *fixpoint-based tile sets* was originally introduced in [4]. One particular application of this construction, explained in [3], is the following theorem: for any $\mathbb{Z}$ effective subshift X_1, the $\mathbb{Z}^2$ lift Y_2 of X_1 (ie. the subshift whose configurations are the configurations of X_1 repeated vertically) is sofic.

It was also proved in [1] with a different method. These constructions improve the original construction of [7], which realized $\mathbb{Z}$ effective subshifts as $\mathbb{Z}^3$ sofic subshifts. We use the construction of [3] in order to prove:

▶ **Lemma 8.** *Let X_1 be a $\mathbb{Z}$ effective subshift. There exists a $\mathbb{Z}^2$ SFT X_2 composed of two superimposed layers of tilings such that:*

1. *The projection of X_2 on its first layer is the $\mathbb{Z}^2$ lift Y_2 of X_1, whose configurations are the configurations of X_1 repeated vertically.*
2. *The second layer of X_2 is composed of tilings of a fixpoint based tile set.*
3. (New) *For any p, q relatively prime,*

$$h(X_2) = h(Y_2) = 0 \qquad \text{and} \qquad h_s(X_2, \alpha) = h_s(Y_2, \alpha) = \frac{\alpha}{1+\alpha} h(X_1).$$

Proof. Points 1 and 2 of this lemma come directly from the construction of Theorem 1 in [3]. We prove point 3, and use notations from [3]. The reader should feel free to skip this proof and go directly to the next section if they are more interested in the geometrical construction.

Let X_1 be some $\mathbb{Z}$ effective subshift, and X_2 (resp. Y_2) be the SFT (resp. the sofic subshift) given by the first two points of Lemma 8. Below, we compute the entropies and the surface entropies of X_2 and Y_2.

First, we prove that $\log N_{X_2}(pn, qn) = pnh(X_1) + o(n)$.

By definition of $\mathbb{Z}$ entropy, $\log N_{X_1}(n) = nh(X_1) + o(n)$. As any configuration of Y_2 is entirely determined by a single line, one has $\log N_{Y_2}(pn, qn) = \log N_{X_1}(pn) = pnh(X_1) + o(n)$. The complexity function of X_2 is at least the contribution of its first layer, which leads to $\log N_{X_2}(pn, qn) \geq \log N_{Y_2}(pn, qn) = pnh(X_1) + o(n)$.

On the other hand, one can find an upper bound of $\log N_{X_2}(pn, qn)$ by considering the contributions of the two layers independently. As the contribution of the first layer is the contribution of Y_2, we now focus on the contribution of all the tilings obtained from the fixpoint-based tile set. Here, we use notions and notations of [3].

In the basic construction of a self-simulating tile set, each macro-tile of level i (ie. of size N_i) is entirely determined by its four macro-colors, which fit in $O(\log N_i)$ bits. In the construction used in [3] to transform $\mathbb{Z}$ effective subshifts into $\mathbb{Z}^2$ SFTs, these macro-colors contain additional data: the level of the macro-tile ($\log N_i$ bits), one segment of l_i and three segments of l_{i+1} letters from configurations of X_1, and the position in the grand-father macro-tile ($\log N_{i+2}$ bits). By taking (as in [3]) $N_i = 2^{C2^i}$ with C being a constant, $L_i = \prod_{j=0}^{i-1} N_i$ and $l_i = \log \log L_i$, we obtain that these macro-colors still fit in $O(\log N_i)$ bits.

For any n big enough, there exists i verifying $L_i \leq pn \leq L_{i+1}$ and $qn \leq L_{i+2}$ (indeed, $\lim_{i \to +\infty} L_{i+2}/L_{i+1} = +\infty$). In this context, a pattern of size $pn \times qn$ can partially cover at most four macro-tiles of level $i+2$. These macro-tiles are entirely determined by their four macro-colors; each macro-tile of level $i+1$ entirely determines the macro-tiles of inferior levels that compose it; and by the previous paragraph each macro-color fits on $O(\log N_{i+2})$ bits (and the constant in the O does not depend on i): all these considerations imply that the number of patterns of size $pn \times qn$ on the second layer is at most polynomial in N_{i+2}.

Considering now the contribution of the two layers independently, one obtains that $\log N_{X_2}(pn, qn) \leq \log N_{Y_2} + \log poly(N_{i+2})$. We have just proved:

$$\log N_{X_2}(pn, qn) = pn\ h(X_1) + o(n) \qquad \text{and} \qquad \log N_{Y_2}(pn, qn) = pn\ h(X_1) + o(n).$$

This immediately leads to:

$$\begin{aligned} h(X_2) &= \lim_{n\to+\infty} \frac{nh(X_1) + o(n)}{n^2} = 0 = h(Y_2) \\ h_s(X_2, p/q) &= \limsup_{n\to+\infty} \frac{pnh(X_1) + o(n)}{(p+q)n} = \frac{p}{p+q} h(X_1) = h_s(Y_2, p/q). \end{aligned}$$ ◀

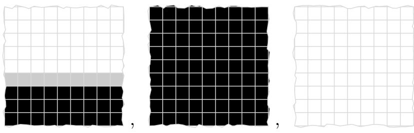
4.5 Building the sparse square layout

In this section, we use the $\mathbb{Z}$ subshift X_1 to build the whole sparse square layout.

First, apply Lemma 8 from the previous section: there exists a $\mathbb{Z}^2$ SFT X_2 with two superimposed layers, such that its first layer (**Base Layer 1**) contains all the vertical replications of configurations of X_1, and the second layer (**Computation Layer 2**) contains some embedded computations. We then create a $\mathbb{Z}^2$ SFT X_3 by superimposing a third layer to X_2, which we describe below.

This third layer (**Square Layer 3**) is itself a superimposition of several sub-layers:

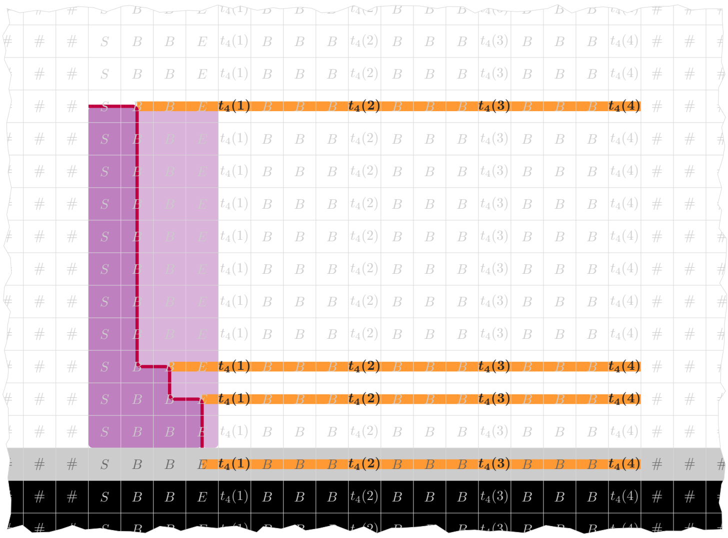
1. First, one must choose a line to be the base line on *Base Layer 1*. It is composed of the same line repeated vertically: we choose one. To do so, we add a [Layer 3a] with three colors (black, white and gray) whose only type of configurations are the following three:



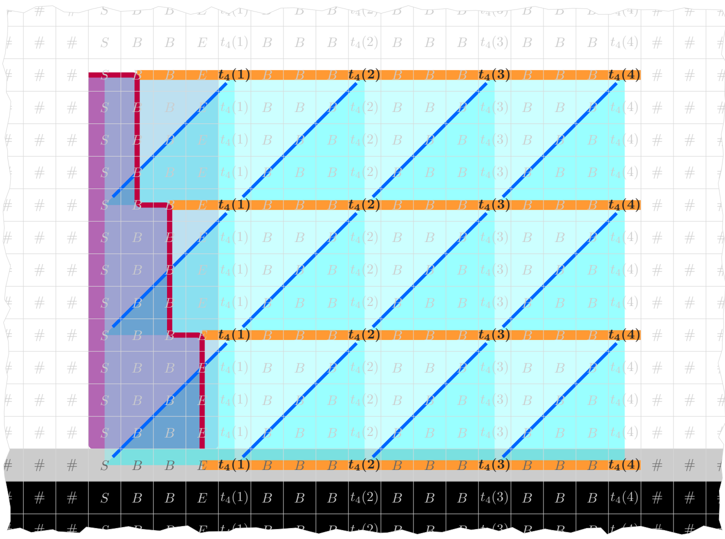
The base line will appear in gray (if it exists).

▶ **Important.** *The other markings of the Square Layer 3 (Layers 3b to 3d) will only be applied on white and gray areas. Additionally, they are not applied on areas marked by # on Base Layer 1.*

2. Add [Layer 3b] with purple construction lines (see Figure 3a). Any E marked in gray (on [Layer 3a]) starts a line at its top, and this line goes up.
Purple lines have the ability to "start" an orange line (on [Layer 3c]). Each time they start an orange line, they move to their left (it ensures that there are at most k orange lines in the k^{th} sparse square).
Purple lines can only end on an S.
Because compactness might lead to surprising results (like infinitely many purple lines behaving erratically), we add colored areas below and above these lines. In these areas of color, we forbid any other purple line to exist: this ensures that there are exactly one purple line per square.
3. Add [Layer 3c] with orange construction lines (see Figure 3a). *These lines will highlight the rows of the sparse squares.* They can only be started by the purple line at their left, and end just before a # on the right. Additionally, an E colored in gray (on [Layer 3a]) must start an orange line.



(a) Base Layer 1 and 3a–3c. Layer 3b ensures there are at most 4 orange lines on Layer 3c.



(b) Base Layer 1 and 3a–3d. Layer 3d enforces the positions of the orange lines.

Figure 3 Behavior of the construction for a pattern which contains the 4^{th} sparse square. These figures highlight the effect of each layer. (Positions of the sparse square are highlighted for the convenience of the reader).

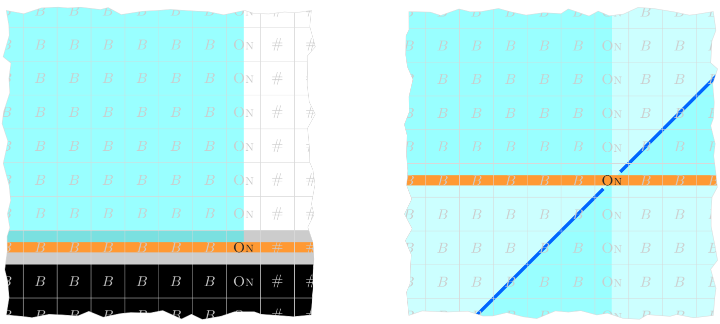


Figure 4 Two examples of degenerate configurations.

(In the figure, we highlight in bold the $t_k(i)$ marked by an orange line. These lines mark the future rows of the sparse squares, and these letters On, Off marked by an orange line will be at the exact non-blank positions of the sparse squares layout.)

4. Add [Layer 3d] with blue construction lines (see Figure 3b). These lines are diagonals. Each $t_k(i)$ colored in orange (on [Layer 3c]) and not in gray (on [Layer 3a]) should start a line that goes diagonally down and left. Additionally, these lines should end either on a letter S (colored in purple on [Layer 3b]), or on the next column marked with $t_k(i-1)$ at a position which is colored in orange (on [Layer 3c]).
   We also impose with colored areas that at each horizontal position, there can be only one blue line between two orange lines. These blue lines ensure the structure of the sparse squares, and constrain the behavior of degenerate configurations (see Figure 4 for examples of these degenerate configurations).

4.6 Contributing to the entropy

The subshift X_3 reflects the sparse square layout in the following way: by considering letters $t_k(i)$ marked in orange on [Layer 3c], we obtain a set of positions that respects the layout.

The final step of the construction consists in adding a fourth layer (**Entropy Layer 4**) to X_3 with *free letters* (ie. letters in the alphabet $1, \dots, 2^e$) and *free bits* (ie. bits in $\{0,1\}$) to obtain an SFT X with the right surface entropy. More precisely:

- *Free letters.* At each site in the sparse square layout (ie. for each position marked by On or Off on *Base Layer 1*, and which is marked in orange on [Layer 3c], we add a free letter which varies in $\{1, \dots, 2^e\}$.
- *Free bits.* At each activated position in the sparse square layout (ie. for each position marked by On on *Base Layer 1*, and which is marked in orange on [Layer 3c]), we add a free bit which varies in $\{0,1\}$.

Let X be the SFT composed of these four superimposed layers. We prove in Subsection 4.7:

▶ **Lemma 9.** *For any $\alpha = p/q \in \mathbb{Q}^+$ (for p and q relatively prime), and for any $n \in \mathbb{N}$, if k is the integer such that $k(k-1)+1 \leq \min(p,q)n < (k+1)k+1$, then the complexity function of the SFT X (defined in Section 4) behaves as follows:*

$$\min(p,q)n\,(e + x'_k) + o(n) \leq \log N_X(pn, qn) \leq \min(p,q)n\left(e + \sup_{\log n \leq i} x'_i\right) + o(n).$$

In particular, this first implies that $h(X) = 0$, then that:

$$h_s(X,\alpha) = \limsup_{n\to+\infty} \frac{\log N_X(pn,qn)}{(p+q)n} = \frac{\min(p,q)}{p+q}\left(e + \limsup_{n\to+\infty} x'_n\right) = \frac{\min(p,q)}{p+q} x$$

which concludes the proof of Theorem 5. ◀

4.7 Computation of the complexity function

This section solely proves Lemma 9. The reader not interested in the precise computation of the complexity function N_X should feel free to skip these pages.

Let $\alpha = p/q$ (for p and q relatively prime) be a positive rational number. In this section, we compute the complexity function $N_X(pn, qn)$ of the SFT X introduced in the previous section. We recall that $x = e + x'$ is a Π_3 real number with $x' \in \Pi_3 \cap [0,1)$, given by $x' = \limsup_i \inf_l f(i,l)$, and that we defined $x'_i = \inf_l f(i,l)$.

We now prove Lemma 9. In this statement, the value $k(k-1)+1$ corresponds to the length of the edges of the k^{th} sparse square.

Proof. As we already proved the lower bound (see Lemma 6), we focus on the upper bound. Here is the structure of the proof:

1. We first consider the contribution of "degenerate configurations" (ie. configurations that do not respect the structure of the sparse square layout: these are obtained when defining X_1 as a closure) to $\log N_X(pn, qn)$, and prove that they contribute only as $o(n)$ (subsection 4.7.1).
2. Then, we consider how many sparse squares can appear simultaneously in a pattern of size $pn \times qn$ (subsection 4.7.2), and:
   a. We provide an upper bound for the sparse square that appear simultaneously in a pattern of size $pn \times qn$ (subsection 4.7.3). As these squares have very low indices, they only contribute as $o(n)$.
   b. We provide an upper bound for the sparse squares that appear alone (subsection 4.7.4; they contribute with the significant term of the upper bound). To obtain it, we bound the number of positions that can appear simultaneously (in a pattern of size $pn \times qn$) of such a sparse square.

Before we begin, in the whole proof we denote by k the integer such that $k(k-1)+1 \leq \min(p,q)n < (k+1)k+1$. (As mentioned above, $k(k-1)+1$ is the length of the edges of the k^{th} sparse square). In the rest of the proof, we look for an upper bound of $\log N_X(pn, qn)$.

4.7.1 Contribution of degenerate configurations

We call *degenerate* the configurations that do not respect the structure of the sparse square layout. There are two possible sources for these configurations: some were obtained when X_1 was defined as a closure (first kind); and some are obtained if *Base Layer 1* respects the structure of a line in the sparse square layout, but [Layer 3a] does not choose a base line for the squares to sit on (second kind).

▷ Claim 10. The contribution of these configurations to $\log N_X(pn, qn)$ is $o(n)$.

Proof. First, we consider the case of a degenerate configurations of the first kind: *Base Layer 1* is not a base line. The number of such patterns only depends on the position of the gray line on [Layer 3a] (if it exists at all), of the position of the letter On or Off in the pattern, etc... There exists at most one varying bit in this pattern (there can be at most one letter On or Off colored in orange in such a degenerate configuration, because of the colored areas of the blue line on [Layer 3d]), which only multiplies the number of patterns by two.

All these patterns depend on finitely many parameters that range from 0 to $\max(p,q)n$. These configurations contribute polynomially in n to $N_X(pn, qn)$, so they contribute $O(\log n)$ to $\log N_X(pn, qn)$.

Consider then the case of degenerate configurations of the second kind: [Layer 3a] does not have a gray line (ie. it does not choose a line for the squares to sit on). If [Layer 3a] is a full black configuration, there are no markings at all on *Square Layer 3* or *Entropy Layer 4*, and the number of patterns depends only on *Base Layer 1*. If [Layer 3a] is a full white configuration, then purple lines on [Layer 3b] can only go up, and never go left: if they did, there would be an orange line on [Layer 3c], which is impossible because of the blue areas of color on [Layer 3d]. This implies that the number of patterns again only depends on *Base Layer 1* (and by Lemma 8, it contributes $o(n)$ to $\log N_X(pn, qn)$).

All in all, degenerate configurations contribute $o(n)$ to $\log N_X(pn, qn)$. ◁

In the rest of the proof, we assume that we consider non-degenerate configurations, ie. configurations that respect the sparse square layout.

We also assume that the gray line of [Layer 3a] is fixed at the bottom of the pattern of size $pn \times qn$ that we consider: this maximizes the number of free bits/free letters in the pattern. Furthermore, we will happily forget to count the different horizontal positions of the squares in the patterns.

Indeed, all these other patterns can be taken into account by translating the figure/varying some parameters which range between 0 and $\max(p, q)n$: these considerations only multiply the complexity function by a polynomial in n, or in other words only add a $o(n)$ to $\log N_X(pn, qn)$.

We also define the following set of words for any $l \in \mathbb{N}$ and $y \in [0, 1)$ (usually $\mathcal{T}(y)$ is defined on the alphabet $\{0, 1\}$ rather than $\{\textsc{On}, \textsc{Off}\}$; otherwise, there is no difference):

$$\mathcal{T}_l(y) = \{t_l \in \{\textsc{On}, \textsc{Off}\}^l : \ t_l \text{ is a subword of } \mathcal{T}(y)\}.$$

As we are looking at non-degenerate configurations, [Layers 3b–3d] (construction layers) are fixed by the line chosen by [Layer 3a]. This means that we can now focus the different Toeplitz words written on *Base Layer 1*, and on the contribution from the free bits and free letters that appear on *Entropy Layer 4*. To count these patterns, we mainly have to compute how many free bits/letters can fit in a pattern at the same time.

4.7.2 Which sparse squares can only appear alone in a pattern?

To find an upper bound of the complexity function, we ask the following question: how many sparse squares can fit in a pattern of size $pn \times qn$?

▷ Claim 11. For $n \geq p$, if at least two different sparse squares appear (maybe partially) in a pattern of size $pn \times qn$, then their indices are below $2 \log n$.

Proof. Assume that a range of squares from i to j, with $i < j$, appear (maybe partially) in a pattern of size $pn \times qn$. Then the horizontal space before the square of index j is entirely contained in the pattern, ie $2^j < pn$. Then for any $n \geq p$, one has $j \leq 2 \log n$. ◁

Reciprocally,

▷ Claim 12. If a sparse square can only appear alone in a pattern of size $pn \times qn$, then its index is greater than $\log n$.

Proof. Assume that a square of index j can "only" appear alone in a pattern of size $pn \times qn$. This means that the space before the sparse square, and the space after the sparse square, are bigger than the horizontal size of the pattern pn. In other words, $2^j \geq pn$, which becomes $j \geq \log n + \log p \geq \log n$. ◁

4.7.3 Contribution of simultaneously appearing sparse squares

▷ Claim 13. The sparse squares that can appear grouped with others contribute as $M_1 = o(n)$ to $\log N_X(pn, qn)$.

Proof. Assume that a range (between i and j, $i < j$) of sparse squares appear (partially) in a pattern of size $pn \times qn$. For n big enough, one has $j \leq 2 \log n$ by Claim 11. Additionally, because we are interested in an upper bound of $\log N_X(pn, qn)$, we can freely assume that all the free bits of the sparse squares of index i and j appear in this pattern.

If $C_{i,j}$ denotes the contribution to $N_X(pn,qn)$ of this slice of squares between i and j, then an upper bound on $C_{i,j}$ is (we count all the Toeplitz subwords written in the squares on *Base Layer 1*, and then their free bits and free letters on *Entropy Layer 4*):

$$\begin{aligned}
C_{i,j} &\leq \sum_{t_i \in \mathcal{T}_i(x'_i),\dots,t_j \in \mathcal{T}_j(x'_j)} \exp_2\left(\sum_{r=i}^{j} r|t_r|_{\textsc{On}} + e\sum_{t=i}^{j} r^2\right)\\
&\leq \sum_{t_i \in \mathcal{T}_i(x'_i),\dots,t_j \in \mathcal{T}_j(x'_j)} \exp_2\left(\sum_{r=1}^{j}(r|t_r|_{\textsc{On}} + er^2)\right)\\
&\leq \sum_{t_i \in \mathcal{T}_i(x'_i),\dots,t_j \in \mathcal{T}_j(x'_j)} \exp_2(j^3(1+e))\\
&\leq \sum_{t_1,\dots,t_j \in \{\textsc{On},\textsc{Off}\}^{1+\dots+j}} \exp_2((2\log n)^3(1+e))\\
&\leq \exp_2(j^2 + o(j^2)) \exp_2((2\log n)^3(1+e))\\
&\leq \exp_2(O((\log n)^3)).
\end{aligned}$$

As there are less than $(2\log n)^2$ different tuples of $i,j \leq 2\log n$, the contribution M_1 of these sparse squares to $\log N_X(pn,qn)$ verifies:

$$\begin{aligned}
M_1 \leq \log\left(\sum_{i<j\leq 2\log n} C_{i,j}\right) &\leq \log\left[(4(\log n)^2)\exp_2(O((\log n)^3))\right]\\
&\leq \log\exp_2(O((\log n)^3)))\\
&\leq o(n).
\end{aligned}$$

$\lhd$

4.7.4 Contribution of the other sparse squares

The other sparse squares can only appear alone (maybe partially) in a pattern of size $pn \times qn$. We distinguish two cases for them:

- The sparse squares of indices $i \leq k$. As they can fit entirely in a pattern of size $pn \times qn$ (recall that $k(k+1)-1 \leq \min(p,q)n \leq (k+1)k+1$), we assume they do (this maximizes the number of free bits that appear simultaneously).
- The sparse squares of indices $i > k$. They can only fit partially in a pattern of size $pn \times qn$, and we need to "count" the number of their free bits/letters that can appear simultaneously.

Contribution of the sparse squares of index $i \leq k$.

$\rhd$ Claim 14. The sparse squares that can only appear alone in a pattern of size $pn \times qn$, and of indices $i \leq k$, contribute to $\log N_X(pn,qn)$ with a term:

$$M_2 \leq \min(p,q)n\left(e + \max_{\log n \leq i \leq k} x'_i\right) + o(n).$$

Proof. As $\#\mathcal{T}_i(x'_i) \leq 2^i$, each of these squares contribute with a term (again, we count the words of $\mathcal{T}_i(x'_i)$ on *Base Layer 1*, and free letters and free bits on *Entropy Layer 4*):

$$\begin{aligned}
C_i &\leq (2^e)^{i^2} \times \sum_{t_i \in \mathcal{T}_i(x'_i)} \exp_2(i|t_i|_{\mathrm{ON}}) \\
&\leq \exp_2(ei^2) \times \exp_2(i) \exp_2(i^2 x'_i + O(i)) \\
&\leq \exp_2(i^2(e + x'_i) + O(i)).
\end{aligned}$$

By Claim 12, such a square must be of index $\geq \log n$. This implies that all these squares contribute to $\log N_X(pn, qn)$ with a term:

$$\begin{aligned}
M_2 &\leq \log\left(\sum_{i=\log n}^{k} C_i\right) \\
&\leq \log\left(\sum_{i=\log n}^{k} \exp_2(i^2(e + x'_i) + O(i))\right) \\
&\leq \log\left(k \exp_2\left(k^2\left(e + \max_{\log n \leq i \leq k} x'_i\right) + O(k)\right)\right) \\
&\leq \log \exp_2\left(k^2\left(e + \max_{\log n \leq i \leq k} x'_i\right) + O(k)\right) \\
&\leq k^2\left(e + \max_{\log n \leq i \leq k} x'_i\right) + O(k) \\
&\leq \min(p, q)n\left(e + \max_{\log n \leq i \leq k} x'_i\right) + o(n).
\end{aligned}$$ ◁

Contribution of the sparse squares of index $i \geq k$. Finally, we consider the sparse squares of indices $i \geq k$: these square only appear partially in a pattern of size $pn \times qn$. They contribute significantly to $\log N_X(pn, qn)$, as explained in the following claim:

▷ Claim 15. The sparse squares of indices $i \geq k$ contribute to $\log N_X(pn, qn)$ with a term:

$$M_3 \leq \min(p, q)n\left(e + \max_{k \leq i \leq \max(p,q)(k+1)^2} x'_i\right) + o(n).$$

Proof. This proof is organized as follows:

1. We provide an upper bound on the contribution C_i of the i^{th} sparse square to $N_X(pn, qn)$, for $i \geq k$, according to the number of free bits/free letters of these squares that can appear simultaneously in a pattern of size $pn \times qn$.
2. Then, we provide an upper bound on the contribution M_3 of all these sparse squares of indices $i \geq k$ to $\log N_X(pn, qn)$.
3. By studying the variations of two functions $h : \mathbb{N} \mapsto \mathbb{N}$ and $v : \mathbb{N} \mapsto \mathbb{N}$ we prove that for any $i \geq k$ the number of simultaneously appearing free bits/free letters of the i^{th} sparse square is $h(i)v(i) \leq \min(p, q)n + o(n)$. This will conclude the proof.

Contribution C_i of the sparse square of index i. First, we need to answer the following question: how many bits can appear simultaneously in a pattern of size $pn \times qn$? Recall that there are exactly i bits in the square of index i per row (and per column). If $h(i)$ denotes the number of horizontal bits that can appear simultaneously in a slice of width pn (and height 1), and $v(i)$ the number of vertical bits in a slice of height qn (and width 1), then:

$$h(i) = \min\left(i, \left\lfloor \frac{pn-1}{i} \right\rfloor + 1\right) \qquad v(i) = \min\left(i, \left\lfloor \frac{qn-1}{i} \right\rfloor + 1\right).$$

h and v are eventually decreasing, and $\lim_{i\to+\infty} h(i) = \lim_{i\to+\infty} v(i) = 1$. There exists an integer $J_k \le \max(p,q)(k+1)^2$ such that for any $i \ge J_k$, one has $v(i) = h(i) = 1$.

We now count free letters and free bits to compute an upper bound on the contribution C_i of a square of index $i \ge k$. As opposed to the previous cases, in which the squares appeared entirely in the pattern of size $pn \times qn$, here we can only see at once $h(i)$ different column: but thanks to the use of Toeplitz sequences, which have a very uniform distribution, the density of a Toeplitz subword of size $h(i)$ is still less than or equal to $h(i)x'_i + O(1)$:

$$\begin{aligned} C_i &\le \sum_{t_i \in \mathcal{T}(x'_i)h(i)} \exp_2\left[h(i)v(i)e + v(i)|t_i|_{\mathrm{ON}}\right] \\ &\le \exp_2\left[h(i)v(i)(e + x'_i) + O(v(i))\right]. \end{aligned}$$

Additionally, as $v(i) \le \frac{qn-1}{i} + 1$, $i \ge k$ and $k = \Theta(\sqrt{n})$, one has $O(v(i)) = O(\sqrt{n})$.

Contribution M_3 of the sparse square of index $i \ge k$. This implies that the contribution M_3 of all these squares of indices $i \ge k$ is:

$$\begin{aligned} M_3 &\le \log\left(\sum_{i=k}^{J_k} \exp_2[h(i)v(i)(e + x'_i) + o(h(i)v(i))]\right) \\ &\le \log\left(J_k \exp_2\left(\left[\max_{k\le i\le J_k} h(i)v(i) + o(h(i)v(i))\right] \times \left[e + \max_{k\le i\le J_k} x'_i\right]\right)\right) \\ &\le \left[\max_{k\le i\le J_k} h(i)v(i) + o(h(i)v(i))\right] \times \left[e + \max_{k\le i\le J_k} x'_i\right] + o(n). \end{aligned}$$

Study of the product $h(i)v(i)$ for $k \ge i \ge J_k$. We now have to study the product $h(i)v(i)$ for $k \le i \le J_k$. Below, we will prove that $\forall k \le i \le J_k, h(i)v(i) \le \min(p,q)n + O(\sqrt{n})$.

Without any loss of generality, assume $q \ge p$. We prove that:

$$h(i)v(i) \le pn + O(\sqrt{n}), \qquad \text{i.e.} \qquad M_3 \le pn\left(e + \max_{k\le i\le \max(p,q)(k+1)^2} x'_i\right) + o(n).$$

As $p = \min(p,q)$, one has $h(i) = \left\lfloor \frac{pn-1}{i} \right\rfloor + 1$ and $v(i) = \min\left(i, \left\lfloor \frac{qn-1}{i} \right\rfloor + 1\right)$. For the first values of i, h is an increasing function, and it then decreases for i large enough: below, we study these variations and conclude about the product $h(i)v(i)$.

- For any $i \le \lfloor\sqrt{qn-1}\rfloor$, one has $\left\lfloor \frac{qn-1}{i} \right\rfloor + 1 \ge i$ (which implies $v(i) = i$). Indeed,

$$\begin{aligned} \left\lfloor \frac{qn-1}{i} \right\rfloor + 1 &\ge \left\lfloor \sqrt{qn-1} \right\rfloor + 1 \\ &\ge i. \end{aligned}$$

- For any $k \le i \le \lfloor\sqrt{qn-1}\rfloor$, one has $h(i)v(i) \le pn + O(\sqrt{n})$. Indeed,

$$\begin{aligned} h(i)v(i) = h(i)i &\le \left(\left\lfloor \frac{pn-1}{i} \right\rfloor + 1\right) i \\ &\le i\left\lfloor \frac{pn-1}{i} \right\rfloor + i \\ &\le pn - 1 + i \\ &\le pn + O(\sqrt{n}). \end{aligned}$$

- For any $i \geq \lfloor\sqrt{qn-1}\rfloor + 1$, one has $i \geq \left\lfloor \frac{qn-1}{i} \right\rfloor + 1$ (which implies $v(i) = \left\lfloor \frac{qn-1}{i} \right\rfloor + 1$). Indeed,

$$\begin{aligned}\left\lfloor \frac{qn-1}{i} \right\rfloor + 1 &\leq \left\lfloor \frac{qn-1}{\sqrt{qn-1}} \right\rfloor + 1 \\ &\leq \left\lfloor \sqrt{qn-1} \right\rfloor + 1 \\ &\leq i.\end{aligned}$$

- For any $\lfloor\sqrt{qn-1}\rfloor + 1 \leq i \leq J_k$, one has $h(i)v(i) \leq pn + O(\sqrt{n})$. Indeed,

$$\begin{aligned}h(i)v(i) = h(i)\left(\left\lfloor \frac{qn-1}{i} \right\rfloor + 1\right) &= \left\lfloor \frac{pn-1}{i} + 1 \right\rfloor \left\lfloor \frac{qn-1}{i} + 1 \right\rfloor \\ &\leq \left\lfloor \frac{pn-1}{\sqrt{qn-1}} + 1 \right\rfloor \left\lfloor \frac{qn-1}{\sqrt{qn-1}} + 1 \right\rfloor \\ &\leq \left\lfloor \frac{pn-1}{\sqrt{qn-1}} + 1 \right\rfloor \left(\sqrt{qn-1} + 1\right) \\ &\leq pn + O(\sqrt{n}).\end{aligned}$$

With all of these computations, we conclude that

$$\max_{k \leq i \leq J_k} h(i)v(i) \leq pn + o(n).$$

In the case $p \geq q$, the computations are completely symmetric and one obtains:

$$\max_{k \leq i \leq J_k} h(i)v(i) \leq qn + O(\sqrt{n}).$$ ◁

4.7.5 End of the proof

We can now conclude the proof of Lemma 9 about the bounds of $\log N_X(pn, qn)$:

$$\min(p,q)n\left(e + x'_k\right) + o(n) \leq \log N_X(pn, qn) \leq \min(p,q)n\left(e + \sup_{\log n \leq i} x'_i\right) + o(n).$$

The lower bound comes from Lemma 6. Indeed, if $k(k-1)+1 \leq \min(p,q)n \leq k(k+1)+1$, then $k^2 = \min(p,q)n + o(n)$, which leads to the desired lower bound. In order to compute the upper bound, we can count the contributions of the different layers independently.

Base Layer 1 and *Computation Layer 2* (the 1D subshift composed of the base line repeated vertically, and the tilings obtained with self-simulating tile sets) do not contribute to the surface entropy by Lemma 8. Indeed, the 1D entropy of the base line is $h(X_1) = 0$. In other words, these two layers add $o(n)$ to $\log N_X(pn, qn)$.

Degenerate configurations, *Square Layer 3*, along with considerations on the different shifts of the configurations, also contribute as $o(n)$ to $\log N_X(pn, qn)$ (they indeed contribute polynomially in n to $N_X(pn, qn)$, see the remark at the end of "Contribution of degenerate configurations"). Finally, in the previous sections, we provided 3 quantities M_1, M_2, M_3 whose sum is greater than the contribution of *Entropy Layer 4* (and which take into account the different written words on non-degenerate configurations of *Base Layer 1*).

With these considerations, we conclude that:

$$\begin{aligned}\log N_X(pn, qn) &\leq o(n) + M_1 + M_2 + M_3 \\ &\leq o(n) + o(n) + \min(p,q)n\left(e + \max_{\log n \leq i \leq \max(p,q)(k+1)^2} x'_i\right) + o(n) \\ &\leq \min(p,q)n\left(e + \max_{\log n \leq i \leq \max(p,q)(k+1)^2} x'_i\right) + o(n).\end{aligned}$$ ◀

5 Conclusive remarks and open questions

Many questions remain open about the notion of surface entropy.

Computational behavior of the definition in [14]. The definition of *surface entropy* used in this paper differs from the original notion of surface entropy in Pace's thesis [14], which was: for any eccentricity $\alpha \in \mathbb{R}^+$,

$$h_s(X,\alpha) = \sup_{\substack{(x_n,y_n)\in(\mathbb{N}^2)^{\mathbb{N}}: \\ x_n,y_n\to+\infty,\ \frac{x_n}{y_n}\to\alpha}} \limsup_{n\to+\infty} \frac{\log N_X(x_n,y_n) - x_n y_n h(X)}{x_n+y_n}.$$

This definition was chosen in [14] because it provides a unified approach for rational (ie. $\alpha \in \mathbb{Q}^+$) and irrational eccentricities. However, we are currently unsure of how the supremum over all sequences impacts the computational characterization of surface entropies. Our construction still realizes any Π_3 surface entropy with the definition of [14], but surface entropies may not be Π_3 real numbers anymore. For all we know, they may not be at any level of the arithmetical hierarchy.

Equivalence between the two definitions. Furthermore, as we modified [14]'s definition of surface entropy, a natural question is whether our new definition coincides with it in the case of rational eccentricities. In other words, can the supremum over all sequences be removed when the eccentricity is a rational number. We are not sure of the answer at the time of writing.

Arbitrary topological entropy with an arbitrary surface entropy. Finally, in the main section of this paper, we created SFTs with zero topological entropy and any Π_3 surface entropy. It was proved in [8] that the class of entropies of $\mathbb{Z}^2$ SFTs is exactly the class of Π_1 real numbers. This led us to wonder whether we could create a family of $\mathbb{Z}^2$ SFTs with arbitrary Π_1 entropy and arbitrary Π_3 surface entropy.
As we do not know the surface entropies of the main constructions of Π_1 entropies, we could not answer this problem with the straightforward solution (ie. a Cartesian product of our construction with one for Π_1 entropies).

References

1. Nathalie Aubrun and Mathieu Sablik. Simulation of effective subshifts by two-dimensional subshifts of finite type. *Acta Applicandae Mathematicae*, 126(1):35–63, 2013. doi:10.1007/s10440-013-9808-5.
2. Robert Berger. *The Undecidability of the Domino Problem.* Number 66 in Memoirs of the American Mathematical Society. The American Mathematical Society, 1966.
3. Bruno Durand, Andrei Romashchenko, and Alexander Shen. Effective Closed Subshifts in 1D Can Be Implemented in 2D. In *Fields of Logic and Computation*, number 6300 in Lecture Notes in Computer Science, pages 208–226. Springer, 2010. doi:10.1007/978-3-642-15025-8_12.
4. Bruno Durand, Andrei Romashchenko, and Alexander Shen. Fixed-point tile sets and their applications. *Journal of Computer and System Sciences*, 78(3):731–764, May 2012. doi:10.1016/j.jcss.2011.11.001.
5. Yuri Gurevich and I Koryakov. Remarks on Berger's paper on the domino problem. *Siberian Math. Journal*, pages 319–320, 1972.

6 David Harel. Recurring Dominoes: Making the Highly Undecidable Highly Understandable. *Annals of Discrete Mathematics*, 24:51–72, 1985.

7 Michael Hochman. On the dynamics and recursive properties of multidimensional symbolic systems. *Inventiones Mathematicae*, 176(1):2009, April 2009.

8 Michael Hochman and Tom Meyerovitch. A characterization of the entropies of multidimensional shifts of finite type. *Annals of Mathematics*, 171(3):2011–2038, May 2010. doi:10.4007/annals.2010.171.2011.

9 Konrad Jacobs and Michael Keane. 0-1-sequences of Toeplitz type. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 13(2):123–131, 1969. doi:10.1007/BF00537017.

10 Emmanuel Jeandel and Pascal Vanier. Characterizations of periods of multidimensional shifts. *Ergodic Theory and Dynamical Systems*, 35(2):431–460, 2015. doi:10.1017/etds.2013.60.

11 S.C. Kleene. *Two Papers on the Predicate Calculus*, chapter Finite Axiomatizability of Theories in the Predicate Calculus Using Additional Predicate Symbols, pages 31–71. Number 10 in Memoirs of the American Mathematical Society. American Mathematical Society, 1952.

12 Douglas A. Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, New York, NY, USA, 1995.

13 Tom Meyerovitch. Growth-type invariants for $\mathbb{Z}^d$ subshifts of finite type and arithmetical classes of real numbers. *Inventiones Mathematicae*, 184(3), 2010. doi:10.1007/s00222-010-0296-1.

14 Dennis Pace. *Surface Entropy of Shifts of Finite Type*. PhD thesis, University of Denver, 2018.

15 Xizhong Zheng and Klaus Weihrauch. Arithmetical hierarchy of real numbers. In *Mathematical Foundations of Computer Science (MFCS)*, pages 23–33, 1999. doi:10.1007/3-540-48340-3_3.