Hardness of conjugacy, embedding and factorization of multidimensional subshifts☆

Emmanuel Jeandel[a], Pascal Vanier[b]

[a] *LORIA*
Campus Scientifique - BP 239
54506 Vandoeuvre-les-Nancy
France
[b] *Laboratoire d'Algorithmique Complexité et Logique*
Université de Paris Est, LACL, UPEC, France

Abstract

Subshifts of finite type are sets of colorings of the plane defined by local constraints. They can be seen as a discretization of continuous dynamical systems. We investigate here the hardness of deciding factorization, conjugacy and embedding of subshifts in dimensions $d > 1$ for subshifts of finite type and sofic shifts and in dimensions $d \geq 1$ for effective shifts. In particular, we prove that the conjugacy, factorization and embedding problems are Σ_3^0-complete for sofic and effective subshifts and that they are Σ_1^0-complete for SFTs, except for factorization which is also Σ_3^0-complete.

Keywords: Subshifts, Computability, Factorization, Embedding, Conjugacy, Subshift of finite type, Arithmetical Hierarchy, Tilings, SFTs.

A d-dimensional subshift is the set of colorings of $\mathbb{Z}^d$ by a finite set of colors in which a family of forbidden patterns never appear. These are shift-invariant spaces, hence the name. If the family of forbidden patterns is finite, then it is a subshift of finite type (SFT). If the family of forbidden patterns is recursively enumerable, then the subshift is called effective. Another class of subshifts can be defined by the help of local maps, namely the class of sofic shifts: they are the letter by letter projections of SFTs.

One can also see SFTs as tilings of $\mathbb{Z}^d$, and in dimension 2 they are equivalent to the usual notion of tilings introduced by Wang [17]. Subshifts are a way to discretize continuous dynamical systems: if X is a compact space and $\phi : X \to X$ a continuous map, we can partition X in a finite number of parts $\mathcal{A} = \{1, \ldots, n\}$ and transform the orbit of a point $x \in X$ into a sequence $(x_n)_{n \in \mathbb{N}^*}$, where x_i denotes the part of X in which $\phi^i(x)$ lies.

Conjugacy is the right notion of isomorphism between subshifts, and plays a major role in their study: when two subshifts are conjugate they code each other and hence have the same dynamical properties. Conjugacy is an equivalence relation and allows

☆This work was sponsored by grants ANR-09-BLAN-0164, EQINOCS ANR 11 BS02 004 03, TARMAC ANR 12 BS02 007 01

Email addresses: `emmanuel.jeandel@loria.fr` (Emmanuel Jeandel), `pascal.vanier@lacl.fr` (Pascal Vanier)

to separate SFTs into equivalence classes. Deciding whether two SFTs are conjugate is called the classification problem. It is a long standing open problem in dimension one [5], although has been proved decidable in the particular case of one-sided SFTs on $\mathbb{N}$, see [18]. It has been known for a long time that in higher dimensions the problem is undecidable when given two SFTs, since it can be reduced to the emptiness problem which is Σ^0_1-complete [2]. However, we prove here a slightly stronger result: even by fixing the class in advance, it is still undecidable to decide whether some given SFT belongs to it:

Theorem 0.1. *For any fixed SFT X, given some SFT Y as an input, it is Σ^0_1-complete to decide whether X and Y are conjugate (resp. equal).*

As for the classes of sofic and effective shifts, the complexity is higher:

Theorem 0.2. *Given two sofic/effective shifts X, Y, it is Π^0_2-complete to decide whether X and Y are equal.*

Theorem 0.3. *Given two sofic/effective subshifts X, Y, it is Σ^0_3-complete to decide whether X and Y are conjugate.*

An interesting open question for higher dimension that would probably help solve the one dimensional problem would be *is conjugacy of subshifts decidable when provided an oracle answering whether or not a pattern is extensible ?*. A positive answer to this question would solve the one dimensional case, even if the SFTs are considered on $\mathbb{N}^2$ instead of $\mathbb{Z}^2$.

Factorization is the notion of surjective morphism adapted to SFTs: when X factors on Y, then Y is a recoding of X, possibly with information loss: the dynamic of Y is "simpler" than X's,*i.e.* it can be deduced from X's. The problem of knowing if some SFT is a factor of another one has also been much studied. In dimension one, it is only partly solved for the case when the entropies of the two SFTs X, Y verify $h(X) > h(Y)$, see [4]. Factor maps have also been studied with the hope of finding universal SFTs: SFTs that can factor on any other and thus contain the dynamics of all of them. However it has been shown that such SFTs do not exist, see [7, 3]. We prove here that it is harder to know if an SFT is a factor of another than to know if it is conjugate to it.

Theorem 0.4. *Given two SFTs/sofic/effective subshifts X, Y as input, it is Σ^0_3-complete to decide whether X factors onto Y.*

The last problem we will tackle is the embedding problem, that is to say: when can an SFT be injected into some other SFT? If an SFT X can be injected into another SFT Y, that means that there is an SFT $Z \subseteq Y$ such that X and Z are conjugate. In dimension 1, this problem is also partly solved when the two SFTs X, Y are irreducible and their entropies verify $h(X) > h(Y)$ [12]. We prove here that the problem is Σ^0_1-complete for SFTs and Σ^0_3-complete for effective and sofic subshifts:

Theorem 0.5. *Given two SFTs X, Y as inputs, it is Σ^0_1-complete to decide whether X embeds into Y.*

Theorem 0.6. *Given two sofic/effective subshifts X, Y as inputs, it is Σ^0_3-complete to decide whether X embeds into Y.*

The paper is organized as follows: first we give the necessary definitions and fix the notation is section 1, after what we give the proofs of the theorems about conjugacy and equality in Section 2, about factorization in Section 3 and about embedding in Section 4.

This article covers the results announced in [10] with the additions of the results on sofic and effective subshifts.

1. Preliminary definitions

1.1. SFTs and effective subshifts

We give here some standard definitions and facts about multidimensional subshifts, one may consult Lind [14] or Lind/Marcus [13] for more details.

Let $\mathcal{A}$ be a finite alphabet, its elements are called *symbols*, the d-dimensional full shift on $\mathcal{A}$ is the set $\mathcal{A}^{\mathbb{Z}^d}$ of all maps (colorings) from $\mathbb{Z}^d$ to the $\mathcal{A}$ (the colors). For $v \in \mathbb{Z}^d$, the shift functions $\sigma_v : \mathcal{A}^{\mathbb{Z}^d} \to \mathcal{A}^{\mathbb{Z}^d}$, are defined locally by $\sigma_v(c_x) = c_{x+v}$. The full shift equipped with the distance $d(x, y) = 2^{-\min\left\{\|v\| \middle| v \in \mathbb{Z}^d, x_v \neq y_v\right\}}$ is a compact metric space on which the shift functions act as homeomorphisms. An element of $\mathcal{A}^{\mathbb{Z}^d}$ is called a *configuration*.

Every closed shift-invariant (invariant by application of any σ_v) subset X of $\mathcal{A}^{\mathbb{Z}^d}$ is called a *subshift*, or *shift*. An element of a subshift is called a *point* of this subshift.

Alternatively, subshifts can be defined with the help of forbidden patterns. A *pattern* is a function $p : P \to \mathcal{A}$, where P, the *support*, is a finite subset of $\mathbb{Z}^d$. Let $\mathcal{F}$ be a collection of *forbidden* patterns, the subset X_F of $\mathcal{A}^{\mathbb{Z}^d}$ containing the configurations having nowhere a pattern of F. More formally, $X_{\mathcal{F}}$ is defined by

$$X_{\mathcal{F}} = \left\{ x \in \mathcal{A}^{\mathbb{Z}^d} \middle| \forall z \in \mathbb{Z}^d, \forall p \in F, x_{|z+P} \neq p \right\}.$$

In particular, a subshift is said to be a *subshift of finite type* (SFT) when the collection of forbidden patterns is finite. Usually, the patterns used are *blocks* or *r-blocks*, that is they are defined over a finite subset P of $\mathbb{Z}^d$ of the form $B_r = [\![-r, r]\!]^d$, r is called its *radius*. We may assume that all patterns of $\mathcal{F}$ are defined with blocks of the same radius r, and say the family $\mathcal{F}$ has radius r. We note r_X the radius of the SFT X, the smallest r for which there is a family $\mathcal{F}$ of radius r defining X. When the collection of forbidden patterns is recursively enumerable (*i.e.* Σ_1^0), the subshift is an *effective subshift*.

Given a subshift X, a pattern p is said to be *extensible* if there exists $x \in X$ in which p appears, p is also said to be extensible to x. We also say that a pattern p_1 is extensible to a pattern p_2 if p_1 appears in p_2. A block or pattern is said to be *admissible* if it does not contain any forbidden pattern. Note that every extensible pattern is admissible but that the converse is not necessarily true. As a matter of fact, for SFTs, it is undecidable (in Π_1^0 to be precise) in general to know whether a pattern is extensible while it is always decidable efficiently (polynomial time) to know if a pattern is admissible, further details about that will be introduced in Section 1.4.

As we said before, subshifts are compact spaces, this gives a link between admissibility and extensibility: if a pattern appears in an increasing sequence of admissible patterns, then it appears in a valid configuration and is thus extensible. More generally, if we have an increasing sequence of admissible patterns, then we can extract from it a sequence converging to some point of the subshift.

Note that instead of using the formalism of SFTs for the theorems we could have used the formalism of Wang tiles, in which numerous results have been proved. In particular the undecidability of knowing whether an SFT is empty. Since we will use a construction based on Wang tiles, we review their definitions.

Wang tiles are unit squares with colored edges which may not be flipped or rotated. A *tileset* T is a finite set of Wang tiles. A *coloring of the plane* is a mapping $c : \mathbb{Z}^2 \to T$ assigning a Wang tile to each point of the plane. If all adjacent tiles of a coloring of the plane have matching edges, it is called a tiling.

The set of tilings of a Wang tileset is a SFT on the alphabet formed by the tiles. Conversely, any SFT is isomorphic to a Wang tileset. From a recursivity point of view, one can say that SFTs and Wang tilesets are equivalent. In this paper, we will be using both terminologies indiscriminately.

1.2. Conjugacy, Embedding and Factorization

In the rest of the paper, we will use the notation $\mathcal{A}_X$ for the alphabet of the subshift X.

Let $X \subseteq \mathcal{A}_X^{\mathbb{Z}^2}$ and $Y \subseteq \mathcal{A}_Y^{\mathbb{Z}^2}$ be two subshifts, a function $F : X \to Y$ is a block code if there exists a finite set $V = \{v_1, \ldots, v_k\} \subset \mathbb{Z}^2$, the *window*, and a local map $f : \mathcal{A}_X^{|V|} \to \mathcal{A}_Y$, such that for any point $x \in X$ and $y = F(x)$, for all $z \in \mathbb{Z}^d$, $y_z = f(x_{z+v_1}, \ldots, x_{z+v_k})$. That is to say F is defined locally. Without loss of generality, we may suppose that the window is an r-block, r being then called the radius of F and $(2r+1)$ its diameter, we note r_F the radius of F.

A *factorization* or *factor map* is a surjective block code $F : X \to Y$. When the function is injective instead of being surjective, it is called an *embedding*, and we say that X embeds into Y.

By the Curtis/Lyndon/Hedlund Theorem [6], when a block map F is bijective then it is invertible and its inverse is also a block code. Subshifts X and Y for which there exist a bijective block map $F : X \to Y$ are said to be *conjugate*. In the rest of the paper, we will note with the same symbol the local and global functions, the context making clear which one is being used.

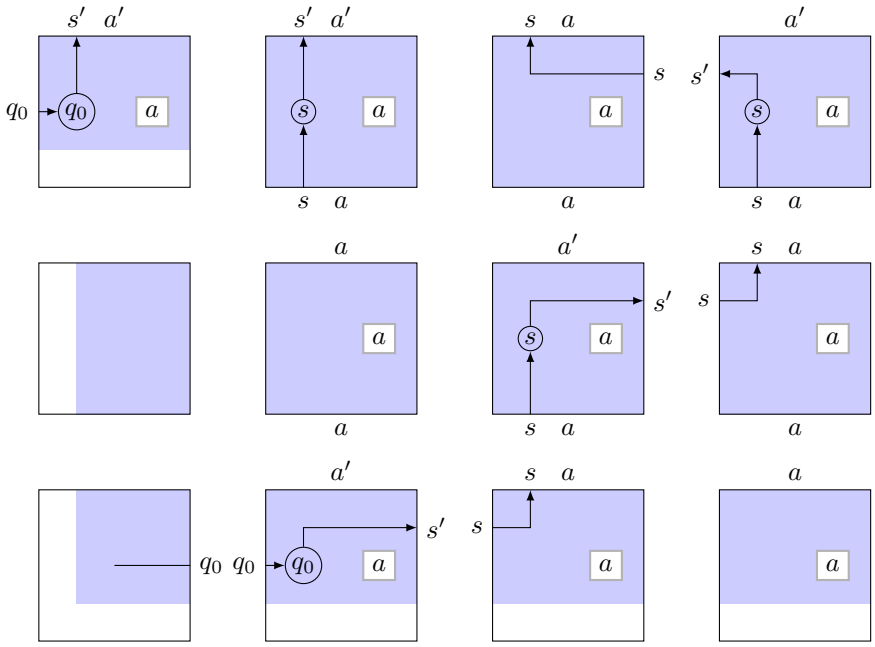
The entropy of a subshift X is defined as

$$h(X) = \lim_{n\to\infty} \frac{\log E_n(X)}{n^d}$$

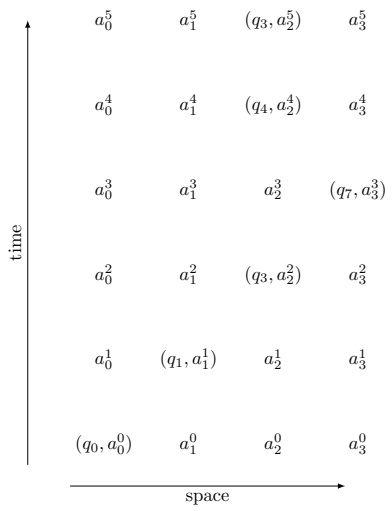
where $E_n(X)$ is the number of extensible patterns of X of support $[\![0, n-1]\!]^d$ where d is the dimension. For instance, the entropy of the full shift is $h(\mathcal{A}^{\mathbb{Z}^d}) = \log |\mathcal{A}|$. The entropy is a conjugacy invariant, that is to say, if X and Y are conjugate, then $h(X) = h(Y)$. It is in particular easy to see thanks to the entropy that the full shift on n symbols is not conjugate to the full shift with n' symbols when $n \neq n'$.

1.3. Sofic subshifts and their relation to effective subshifts

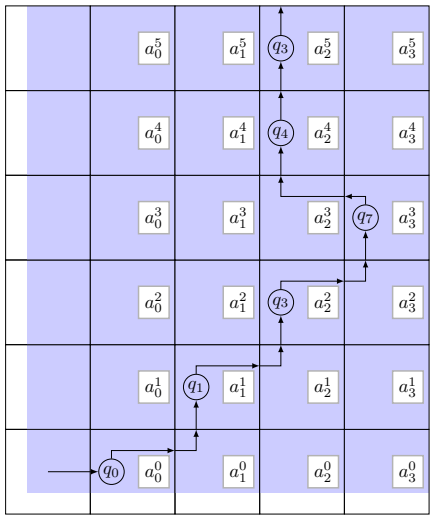
The class of subshifts that are images of SFTs by a factor map is the class of *sofic subshifts*. It is the smallest class of subshifts that is closed by factorization. There is a link between sofic subshifts of dimension d and effective subshifts of dimension $d+1$.



(a) Tiles allowing to encode computations of a Turing machine: the adjacency rules are given by the machine's transition table $\delta(s, a) = (s', a', d)$ with d determining to which side of the tile the arrow goes, s' the new state and a' the letter written on the tape. q_0 is the initial state of the Turing machine. Note that there is no tile for the halting state.



(b) Space-Time diagram of a Turing machine.



(c) Valid tiling by the tileset corresponding to the Turing machine. If the Turing machine takes no input, then the a_i^0 are all blanks.

Figure 1: How to encode Turing machine computations in tilings. Only the valid infinite runs of the Turing machine may tile the quarter plane without any errors when the starting tile is present, since we forbid halting states to appear in a tile.

Definition 1.1 (Lift). *Let X be a d-dimensional subshift, then the* lift X' *of X to dimension $d+1$ is the subshift that is formed of configurations $x \in X$ and that are identical on the next component.*

Particularly, the projection of the lift of X along its d first components is X.

Through lifting, one can link sofic and effective subshifts with the following theorem:

Theorem 1.1 (Hochman [8], Aubrun and Sablik [1]). *A subshift is effective if and only if its lift is sofic.*

This theorem will be mainly used to transpose constructions of effective subshifts to the sofic case, but in one more dimension.

1.4. Arithmetical Hierarchy and computability

We give now some background in computability theory and in particular about the arithmetical hierarchy. More details can be found in Rogers [16].

Given a Turing machine M, we will note $M(n)\downarrow$ when M halts on input n and $M(n)\uparrow$ when it does not.

In computability theory, the arithmetical hierarchy is a classification of sets according to their logical characterization. A set $A \subseteq \mathbb{N}$ is Σ^0_n if there exists a total computable predicate R such that $x \in A \Leftrightarrow \exists \overline{y_1}, \forall \overline{y_2}, \ldots, Q\overline{y_n} R(x, \overline{y_1}, \ldots, \overline{y_n})$, where Q is a $\forall$ or an $\exists$ depending on the parity of n. A set A is Π^0_n if there exists a total computable predicate R such that $x \in A \Leftrightarrow \forall \overline{y_1}, \exists \overline{y_2}, \ldots, Q\overline{y_n} R(x, \overline{y_1}, \ldots, \overline{y_n})$, where Q is a $\forall$ o r an $\exists$ depending on the parity of n. Equivalently, a set is Σ^0_n iff its complement is Π^0_n.

We say a set A is many-one reducible to a set B, $A \leq_m B$ if there exists a computable function f such that for any x, $f(x) \in A \Leftrightarrow x \in B$. Given an enumeration of Turing machines M_i with oracle X, the Turing *jump* X' of a set X is the set of integers i such that M_i halts on input i. We note $X^{(0)} = X$ and $X^{(n+1)} = (X^{(n)})'$. In particular $0'$ is the set of halting Turing machines.

A set A is Σ^0_n-hard (resp. Π^0_n) iff for any Σ^0_n (resp. Π^0_n) set B, $B \leq_m A$. Furthermore, a Σ^0_n-hard (resp. Π^0_n) is Σ^0_n-complete (resp. Π^0_n-complete) if it is in Σ^0_n. An example of Σ^0_n-complete problem is $0^{(n)}$. The sets in Σ^0_1 are also called recursively enumerable and the sets in Π^0_1 are called the co-recursively enumerable or effectively closed sets. In this article, we will mainly use two complete problem:

- TOTAL: this is the set of Turing machines which halt on all inputs, see example 35.3 of [11].
- COFIN: this is the set of Turing machines which halt on all inputs but a finite number. This problem is Σ^0_3-complete, see example 35.5 and lemma 36.1 of [11].

Another fact that will be used several times is that extensibility for the several classes of subshifts we consider is Π^0_1.

Lemma 1.2. *Extensibility for SFTs, sofic and effective subshifts is Π^0_1.*

Proof. We may restrict ourselves to effective subshifts since SFTs and sofic subshifts are effective. One may check if a pattern M is extensible step by step: at the k-th step one enumerates k forbidden pattern and checks whether there exist a pattern of

radius $r(M)+k$ containing M at its center containing none of the k forbidden patterns enumerated so far. If this algorithm halts, then M is not extensible. Otherwise we have an infinite sequence of increasing patterns M_k from which we can extract a converging subsequence whose limit is an extension of M containing no forbidden pattern. $\square$

It is quite clear that deciding whether a pattern is admissible for SFTs is decidable, while for effective shifts is Π_1^0 since one needs to enumerate all forbidden patterns to check. Admissibility for sofic shifts is also Π_1^0 since they are effective.

2. Conjugacy and equality

2.1. SFTs

We prove here the Σ_1^0-completeness of the conjugacy problem for SFTs in dimension $d \geq 2$, even if we fix an SFT in advance. We first prove the following lemma, which is the first step to show that conjugacy is Σ_1^0 and also proves that equality of SFTs is Σ_1^0.

Lemma 2.1. *Given F a local map, X and Y SFTs as inputs, deciding if $F(X) \subseteq Y$ is Σ_1^0.*

Proof. It is clear that $F(X) \subseteq Y$ if and only if $F(X)$ does not contain any configuration where a forbidden patterns of Y appears.

We now show that this is equivalent to the following Σ_1^0 statement: *there exists a radius $r > \max(r_F + r_Y, r_X)$ such that for any admissible r-block M of X, $F(M)$ does not contain any forbidden pattern in its center.*

We prove the result by contraposition, in both directions. Suppose there is a configuration $x \in X$ such that $F(x)$ contains a forbidden pattern. Then for any radius $r > \max(r_F + r_Y, r_X)$, there exists an extensible, and thus admissible, pattern M of size r such that $F(M)$ contains a forbidden pattern in its center.

Conversely, if for any radius $r > \max(r_F + r_Y, r_X)$, there exists an admissible pattern M of X of size r such that $F(M)$ contains a forbidden pattern in its center, then by compactness one can extract a converging subsequence from these forbidden patterns, its limit x is in X and $F(x)$ contains a forbidden pattern in its center. $\square$

In particular, if F is the identity we obtain:

Corollary 2.2. *Given two SFTs X, Y as an input, it is Σ_1^0 to decide whether $X = Y$.*

We may now prove one of the announced theorems:

Theorem 2.3. *Given two SFTs X, Y as an input, it is Σ_1^0 to decide whether X and Y are conjugate.*

Proof. To decide whether two SFTs X and Y are conjugate, we have to check whether there exists two local functions $F : \mathcal{A}_X^{B_{r_F}} \to \mathcal{A}_Y$ and $G : \mathcal{A}_Y^{B_{r_G}} \to \mathcal{A}_X$ such that the global functions associated verify $F_{|X} \circ G_{|Y} = id_{|Y}$ and $G_{|Y} \circ F_{|X} = id_{|X}$. These functions being local, we can guess them with a first order existential quantifier. We prove that X and Y are conjugate if and only if the following $\mathcal{A}_1^0$ statement is true :

There exist F, G and $k > \max(r_X + r_Y) + r_F + r_G$ such that $F(X) \subseteq Y$ and $G(Y) \subseteq X$ and :

- *for all k-block b, if b is admissible for X, then $G \circ F(b)_0 = b_0$*
- *for all k-block b, if b is admissible for Y, then $F \circ G(b)_0 = b_0$*

We only prove the statement for $G \circ F$ the other one being identical. The proof is by contraposition in both directions :

- Let $x \in X$ be a point such that $G \circ F(x) \neq x$, we may suppose that the difference is in 0 by shifting. For all k, there exists an extensible pattern b of size k such that $G \circ F(x)_0 \neq b_0$.
- Conversely, if there exists a sequence b_k of admissible k-blocks such that $G \circ F(b_k)_0 \neq (b_k)_0$, then by compactness we can extract a subsequence converging to some point $x \in X$ which by construction is different from its image by $G \circ F$ in 0.

As we have seen in Lemma 2.1 that checking whether $F(X) \subseteq Y$ is Σ^0_1, we have the desired result. □

Theorem 2.4. *For any fixed SFT X of dimension $d \geq 2$, the problem of deciding whether given Y an SFT of dimension $d \geq 2$ as input, Y is conjugate (resp. equal) to X is Σ^0_1-hard.*

Proof. We reduce the problem from $0'$, the halting problem. Given a Turing machine M we construct a SFT Y_M such that Y_M is conjugate to X iff M halts.

Let R_M be Robinson's SFT [15] encoding computations of M: R_M is empty iff M halts[1].

Now take the full shift on one more symbol than X, note it F. Let Y_M be now the disjoint union of X and $R_M \times F$.

If M halts, $Y_M = X$ and hence is conjugate to X. In the other direction, suppose M does not halt, then $R_M \times F$ has entropy strictly greater than that of X and hence Y_M is not conjugate to X. □

Corollary 2.5. *Given two SFTs X, Y of dimension $d \geq 2$ as an input, it is Σ^0_1-hard to decide whether $X = Y$.*

2.2. Sofic and effective subshifts

For effective and sofic subshifts, the complexity becomes higher: checking whether a pattern is admissible or whether it is extensible is the same complexity-wise. It is Π^0_1 in both cases, which disallows us from using the same compactness tricks as in Lemma 2.1.

Lemma 2.6. *Given two effective subshifts X, Y and a local function F, deciding if $F(X) \subseteq Y$ is Π^0_2.*

[1] Robinson's SFT is in dimension 2 of course, for higher dimensions, we take the iterated lift: we take the rules that the symbol in $x \pm e_i$ equals the symbol in x, for $i > 2$.

Proof. $F(X) \subseteq Y$ if and only if the image of every extensible pattern of X is an extensible pattern of Y, which is equivalent to the following logical sentence: *For every pattern M of radius r, M is extensible for $X \Rightarrow F(M)$ is extensible for Y.* Which is clearly Π^0_2. □

Corollary 2.7. *Given two effective subshifts X, Y it is Π^0_2 to decide whether $X = Y$.*

Let us now prove the Π^0_2-hardness of the equality problem in order to obtain the Π^0_2-completeness of it.

Theorem 2.8. *Given two effective subshifts X, Y, deciding whether $X = Y$ is Π^0_2-hard.*

Proof. To show that the problem is Π^0_2-hard, we start from the TOTAL problem which is Π^0_2-complete. Take the following two one-dimensional effective subshifts:

- Let M be a Turing machine, X_M is the subshift on two symbols $\{\#, 0\}$ where we forbid all words $\#0^n\#$ such that $M(n)\downarrow$.
- Y is the subshift on the alphabet $\{\#, 0\}$ where we forbid the words $\#0^n\#$ for all $n \in \mathbb{N}$ such that the subshift is composed only of the orbits the two following points:

$$\cdots 000000 \cdots \text{ and } \cdots 000\#000 \cdots$$

These subshifts are effective, since for any Turing machine M one can enumerate with a Turing machine all n's such that $M(n)\downarrow$.

The two subshifts, X_M and Y are equal if and only if the Turing machine M halts on all inputs : if the machine M does not halt on n, then the subshift X_M contains the periodic point $\cdot\#0^n\#0^n\#\cdot$ which is not in Y. □

Now using Theorem 1.1 allowing to lift an effective subshift of dimension 1 to a sofic one of dimension 2, we obtain the following corollary:

Corollary 2.9. *Given two sofic subshifts X, Y of dimension $d \geq 2$, knowing whether $X = Y$ is Π^0_2-complete.*

We may now head back to the conjugacy problem: a straightforward adaptation of the proof of Theorem 2.3 leads to the following upper bound:

Theorem 2.10. *Given two effective subshifts X, Y as an input, it is Σ^0_3 to decide whether they are conjugate.*

Only remains the hardness part, which we prove by reducing to COFIN, the set of Turing machines that do not halt on a finite number of input only.

Theorem 2.11. *Given two effective subshifts X, Y (resp. sofic of dimension $d \geq 2$) as an input, deciding whether they are conjugate is Σ^0_3-hard.*

Proof. We give a construction for effective subshifts of dimension 1 which can again be lifted to sofic subshifts of dimension 2 or higher.

Given a Turing machine M, we construct two subshifts X_M and Y_M on the alphabet $\{\#, 0, 1\}$:

- X_M: we forbid the words $\#1, 1\#, 10, 0\#$, the words $\#0^k1$ when k is **not** of the form 2^{i+1} with $i \in \mathbb{N}$ and the words $\#0^{2^{n+1}}1$ for all n such that $M(n)\downarrow$. The subshift is formed of the following biinfinite words:

$$\cdots \#\#\#0^{2^{n+1}}111\cdots \qquad \text{with } M(n)\uparrow$$

$$\begin{gathered}\cdots \#\#\#\#\#\#\cdots \\ \cdots \#\#\#000\cdots \\ \cdots 000000\cdots \\ \cdots 111111\cdots \\ \cdots 000111\cdots\end{gathered}$$

- Y_M: we forbid the words $\#1, 1\#, 10, 0\#$, the words $\#0^k1$ when k is not of the form $2^{i+1}+2^i$ with $i \in \mathbb{N}$ and the words $\#0^{2^{n+1}}1$ for all n such that $M(n)\downarrow$. The subshift Y_M is formed of the following words:

$$\cdots \#\#\#0^{2^{n+1}+2^n}111\cdots \qquad \text{with } M(n)\uparrow$$

$$\begin{gathered}\cdots \#\#\#\#\#\#\cdots \\ \cdots \#\#\#000\cdots \\ \cdots 000000\cdots \\ \cdots 111111\cdots \\ \cdots 000111\cdots\end{gathered}$$

Let us now prove that X_M and Y_M are conjugate if and only if the set $H_M = \{n \mid M(n)\uparrow\}$ is finite:

$\Rightarrow$ If H_M is finite, then there exists some integer N that bounds all its elements. Then there clearly exists a conjugacy function F with radius $r_F > 2^{N+1} + 2^N$ which consists only in shifting right-infinite sequence of ones by 2^n and adding 2^n ones at the beginning.

$\Leftarrow$ If H_M is infinite, suppose there exists a conjugacy function $F : Y_M \to X_M$. First note that $\#^{r_F}, 0^{r_F}, 1^{r_F}$ respectively have $\#, 0, 1$ as images. If this were not the case, then the words $\cdots \#\#\#0^k111\cdots$ would have the same image image for $k > r_F$ and this would contradict the injectivity of F. Now take $n \in H_M$ such that $2^n > 2r_F + 1$. The point

$$\cdots \#\#\#0^{2^{n+1}+2^n}111\cdots$$

has an image that does not belong to the subshift X_M because it is of the form

$$\cdots \#\#\#w_1 \cdots w_{2r_F} 0^{2^{n+1}+2^n-2r_F} w'_1 \cdots w'_{2r_F} 111\cdots$$

with $w_i, w'_i \in \{\#, 0, 1\}$.

$\square$

Note 2.12. *In the previous proof, it was only made use of the injectivity of F for the reciprocal. This will be used in Corollary 4.5.*

3. Factorization

We will prove here that factorization is Σ^0_3-complete for SFTs, sofic and effective subshifts. To do this we will prove that the upper bound is Σ^0_3 in the effective case and that the SFT case is Σ^0_3-hard, thus leading to the completeness result for all classes.

We start with two small examples to see why factorization is more complex than conjugacy in the SFT case. Here the examples are the simplest possible: we fix the SFT to which we factor in a very simple way, thus making the factor map known in advance.

Theorem 3.1. *Let Y be the SFT containing exactly one configuration, a uniform configuration. Given an effective subshift X as an input, it is Π^0_1-complete to know whether X factors onto Y.*

Proof. In this case the factor map is forced: it has to send everything to the only symbol of $\mathcal{A}_Y$. And the problem is hence equivalent to knowing whether a SFT is *not* empty, which is Π^0_1-complete. □

Theorem 3.2. *Let Y be the empty SFT. Given an effective subshift X as an input, it is Σ^0_1-complete to know whether X factors onto Y.*

Proof. Here any factor map is suitable, the problem is equivalent to knowing whether X is empty, which is Σ^0_1-complete. □

We study now the hardness of factorization in the general case, that is to say when two SFTs are given as inputs and we want to know whether one is a factor of the other. We prove here with Theorems 3.3 and 3.10 the Σ^0_3-completeness of the factorization problem.

3.1. Factorization is in Σ^0_3

Theorem 3.3. *Given two effective subshifts X, Y as an input, deciding whether X factors onto Y is Σ^0_3.*

Proof. The subshift X factors onto Y iff there exists a factor map F, a local function, such that $F(X) = Y$. This forces one existential quantifier, and the result follows from the next lemma and Lemma 2.6 which prove that deciding whether $F(X) = Y$ is Π^0_2. □

Lemma 3.4. *Given two effective subshifts X, Y and a local map F as an input, deciding if $Y \subseteq F(X)$ is Π^0_2.*

Proof. We prove here that the statement $Y \subseteq F(X)$, that is to say, *for every point $y \in Y$, there exists a point $x \in X$ such that $F(x) = y$*, is equivalent to the following Π^0_2 statement: *for any pattern m, if m is extensible for Y, then $F^{-1}(m)$ contains an extensible pattern for X*. This statement is Π^0_2 since checking that m is extensible is Π^0_1.

We now prove the equivalence. Suppose that $Y \subseteq F(X)$, then any extensible pattern m of Y appears in a configuration $y \in Y$ which has a preimage $x \in X$. Thus m has an extensible preimage in X. This proves the first direction.

Conversely, suppose all extensible patterns m of Y have extensible preimages in X. Let y be a point of Y, then we have an increasing sequence m_i of extensible patterns converging to y. All of them have at least one extensible preimage m'_i. By compactness, we can extract from this sequence a converging subsequence, note x its limit. By construction x is a point of X and a preimage of y.

□

3.2. Factorization is Σ^0_3-hard

We give two proofs here for the Σ^0_3-hardness of factorization : one for effective subshifts in dimension one and one for SFTs in dimension $d \geq 2$. The proof for SFTs gives us completeness for all classes, SFTs, sofic and effective subshifts, but only for dimensions $d \geq 2$. Also, the proof for effective subshifts in dimension one gives the ideas that will be refined to get the proof for SFTs.

3.2.1. Effective subshifts

Theorem 3.5. *Given two effective subshifts X, Y as an input, it is Σ^0_3-hard to decide whether X factors onto Y.*

Proof. We reduce the problem to COFIN, the set of Turing machines that do not halt on a finite set of inputs.

Given a Turing machine M, we construct two effective subshifts X_M and Y_M such that X_M factors onto Y_M if and only if the set of inputs on which M does not halt is finite:

- X_M is defined on the alphabet[2] $\{\#, W, R, 0, 1, B\}$ and is constituted of the following points and their orbits:

$$\begin{array}{ll} \cdots \#\#\#W^{n+1}0BBB\cdots & \text{with } M(n)\uparrow \\ \cdots \#\#\#W^{n+1}1BBB\cdots & \text{with } M(n)\uparrow \\ \\ \cdots WWWbBBB\cdots & \text{with } b \in \{0,1\} \\ \cdots \#\#\#WWW\cdots & \\ \cdots \#\#\#RRR\cdots & \\ \cdots \#\#\#\#\#\#\cdots & \\ \cdots WWWWWW\cdots & \\ \cdots RRRRRR\cdots & \\ \cdots BBBBBB\cdots & \end{array}$$

  X_M is effective since the only "complex" forbidden patterns to enumerate are those of the form $\#W^nbB$ with $b \in \{0,1\}$ with $M(n)\downarrow$ and the other forbidden patterns are the two symbol ones that follow:

$$\begin{array}{c} 0\#, 1\#, W\#, B\#, R\#, \\ 0W, 1W, BW, RW, \\ WR, 0R, 1R, BR, \\ 00, \#0, R0, 10, B0, \\ 11, \#1, R1, 01, B1, \\ \#B, WB, RB \end{array}$$

- Y_M is defined on the alphabet $\{\#, W, 0, 1, B\}$ and consists of the following points

[2] W stands for white, B for blue and R for red.

and their orbits:

$$\begin{array}{ll} \cdots\#\#\#0W^{n+1}BBB\cdots & \text{with } M(n)\uparrow \\ \cdots\#\#\#1W^{n+1}BBB\cdots & \text{with } M(n)\uparrow \\ \\ \cdots\#\#\#bWWW\cdots & \text{with } b\in\{0,1\} \\ \cdots WWWBBB\cdots & \\ \cdots\#\#\#\#\#\# & \\ \cdots BBBBBB\cdots & \\ \cdots WWWWWW\cdots & \end{array}$$

It is quite clear that Y_M is an effective subshift.

We will call the $\{0,1\}$ symbols *decorations*: they will appear at most once in a configuration and only in some configurations of X_M. Hence when we will talk about the *decoration* of a configuration, we will mean the only $\{0,1\}$ symbol of this configuration.

Now let us check that X_M factors onto Y_M if and only if the set of inputs on which M does not halt is finite:

- $\Rightarrow$ Suppose the number of inputs on which M does not halt is finite, then there exists an upper bound N on all of these inputs. We may take a factor map of radius $N+3$, which shifts the decoration that is at the end of the W's to the beginning (*i.e.* just after the # symbols). It also maps the $\cdots\#\#\#WWW\cdots$ configuration to the $\cdots\#\#\#0WWW\cdots$ configuration and the $\cdots\#\#\#RRR\cdots$ configuration to the $\cdots\#\#\#1WWW\cdots$ configuration, we can deduce the other mappings easily from these. Note that we had to add the configuration $\cdots\#\#\#RRR\cdots$ to X_M in order to palliate to the missing decoration when the word $W\cdots W$ becomes infinite, *i.e.* the limit case.

- $\Leftarrow$ Suppose the number of inputs on which M does not halt is infinite and that there exists a factor map F from X_M onto Y_M of radius r. Since there are points of the form $\cdots\#\#\#bW^{n+1}BBB\cdots$ in Y_M for n's that are arbitrarily large, the r-blocks $\#\cdots\#$, $W\cdots W$ and $B\cdots B$ must have #, W and B as images. But since the factor map is of radius r, then for all $n>r$ such that $M(n)\uparrow$, the points $\cdots\#\#\#W^{n+1}0BBB\cdots$ and $\cdots\#\#\#W^{n+1}1BBB\cdots$ of X_M necessarily have the same image: $\cdots\#\#\#bW^{n+1}BBB\cdots$. Thus, the points $\cdots\#\#\#\bar{b}W^{n+1}BBB\cdots$ of Y_M have no preimage in X_M by F and F cannot be a factor map.

□

3.2.2. Finite type

The idea of the proof for SFTs of dimension $d\geq 2$ is similar, but this time we have to explicitly encode the computations in the SFT, since there is no way anymore to "hide" them in the forbidden patterns.

To do this, we use a simpler version of the construction introduced in [9], which could actually have already been used in it[3]. This construction introduced a new way to put

[3]It would in particular have led to a lower constant in Theorem 4.4.

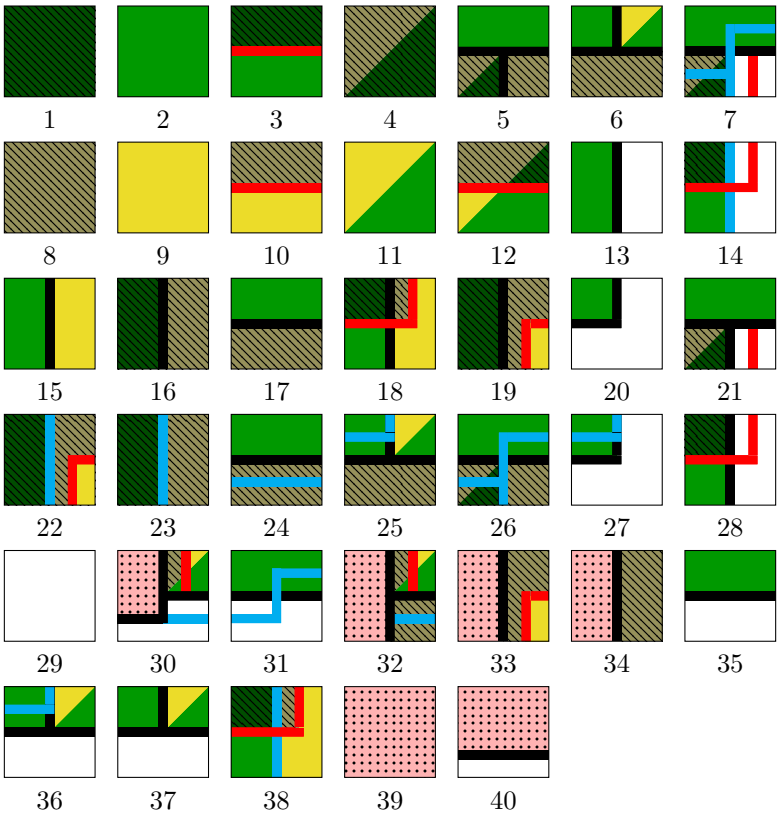


Figure 2: The tileset T.

Turing machine computations in SFTs. In particular, the base construction has exactly one point (up to shift) in which computations may be encoded. It will in particular behave nicely when submitted to local maps.

T-structures. We construct in this section an SFT in which exactly one configuration up to shift forms a sparse grid. It has the property that whenever two subshifts are based on it, and one factors onto or embeds into the other one, then the grid configurations have to be mapped to grid configurations. This is achieved by using the tileset T of Figure 2.

First, note that the configuration represented in figure 3, that we will call α, is in T. Configuration α forms a grid in which we can encode computations as we will see later. We will also see that it is the only one containing an infinite grid and how to encode computations in it.

Before proving anything, let us set the vocabulary that will be used to describe the SFT $\mathcal{X}_T$.

- Tile 30 is the *corner tile*.
- Tiles 20 and 27 are the *end tiles*.

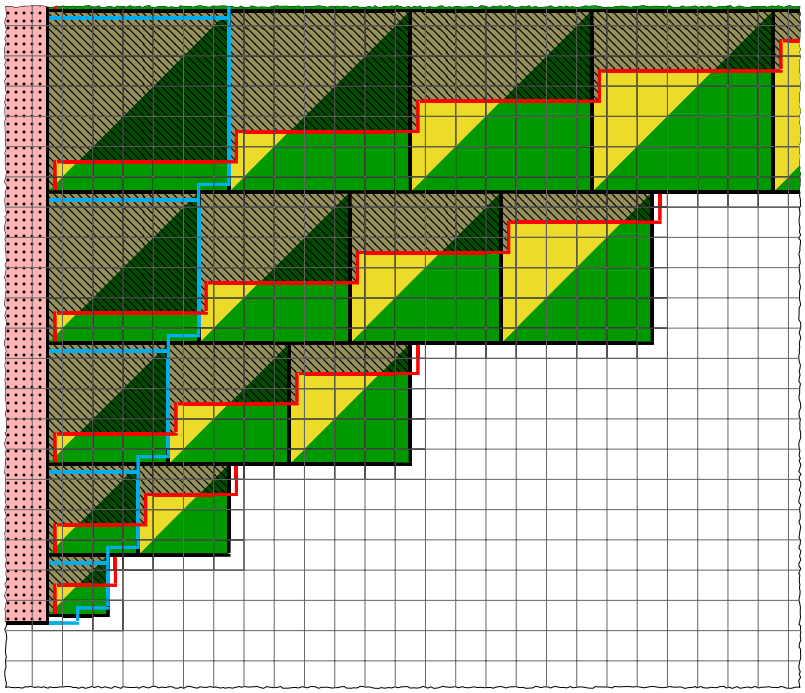


Figure 3: Configuration α: the unique valid tiling by T in which there are 2 or more vertical lines.

- Tiles 30, 32, 33 and 34 are the *start tiles*.
- A *horizontal line* is connex horizontal alignment of tiles containing a vertical black line (tiles 5, 6, 7, 17, 21, 24, 25, 26, 31, 35, 36, 37). It may be terminated by start tiles on its left and by end tiles on its right.
- A *vertical line* is a connex vertical alignment of tiles containing a black or blue vertical line (tiles 13, 14, 15, 16, 18, 19, 22, 23, 28, 32, 33, 34, 38). It may be terminated on top by tiles 5, 21, 26 and on bottom by tiles 6, 20, 25, 27, 30, 36, 37.
- A *diagonal* is a connex diagonal alignment (positions $(i,j),(i+1,j+1),\dots$) of tiles among 4, 11, 12.
- A *square* of size k is an extensible pattern of support $[\![0,k+1]\!]^2$ such that $\{0\}\times[\![1,k]\!]$ and $\{k+1\}\times[\![1,k]\!]$ are vertical lines and $[\![1,k]\!]\times\{0\}$ and $[\![1,k]\!]\times\{k+1\}$ horizontal lines. A square does not contain a horizontal/vertical line in the subpattern of support $[\![1,k]\!]^2$. Remark that the colors above and below a horizontal line differ and that this forces a square to contain a diagonal at positions (i,i), for $0<i<k+1$ and a counting signal (see next bullet point) somewhere in between the two horizontal lines. A *row* of squares is a horizontal alignment of squares.
- A *counting signal* is a connex path of tiles among 3, 7, 10, 12, 14, 19, 22, 32, 33, 38, such that the red signal is connected. It may be started (on the left) only by tiles 30, 32 and ended (on the right) by tiles 7, 21. The counting signal counts the

number of squares on each row and forces this number to be exactly the height of these squares.

- An *increase signal* is formed by a path of tiles among 7, 14, 22, 23, 24, 25, 26, 27, 30, 31, 32, 36, 38, such that the blue signal is connected. This signal forces squares to increase their size by exactly one on the row right above. And thus to increase their number by one also.

Let us first notice that whenever the corner tile appears in a point, this point is necessarily a shifted copy of α, the point of figure 3: the corner tile forces tile 33 to appear above it and tiles 31 and then 27 to appear on its right. These tiles enforce the existence of the first square of size 1, *i.e.* the first row. The increase signal forces the first square of the row above to be of size 2, and so on...

Lemma 3.6. *SFT $\mathcal{X}_T$ admits at most one point, up to translation, with two or more vertical lines. This is the α configuration represented in figure 3.*

Proof. Let x be a point containing two horizontal lines, these two lines necessarily face each other: either they are infinite, or they end on the left, in which case they end on the same column, start tiles being necessarily all in the same column, start tiles being all in the same column because of the color on their left. We may suppose that there is no other horizontal line in between and that they are at distance $k+1$.

Since the sides above and below a horizontal line do not have the same color, there must necessarily be one or more diagonals in between. Each diagonal forces vertical lines thus forming squares of size k. Furthermore, these squares are necessarily cut horizontally by a counting signal, which moves above exactly once each time it crosses a vertical line. This guarantees that there are exactly k squares in this row and thus that it is not infinite.

The increase signal necessarily appears on the vertical line formed by the right side of the bottommost square and forces the existence of squares of size $k+1$ in the row above and of size $k-1 \geq 1$ in the row below. The increase signal also forces squares to appear. The corner tile will appear at the bottom left corner of the only square of size 1 of the bottommost row. □

One may notice that if the corner tile appears at position $(0,1)$ then there are horizontal lines of length $(k+1)k+1$ which start at positions $\left(0, \frac{k(k+1)}{2}\right)$: theses lines form the bottom border of a row of k squares, one may also see that the increase signal draws a parabola.

Lemma 3.7. *Besides the α configuration, the SFT $\mathcal{X}_T$ contains points formed of uniform zones (one tile only) except for three infinite strips of finite width : a vertical strip, a horizontal strip and a diagonal SW-NE strip. See figure 5.*

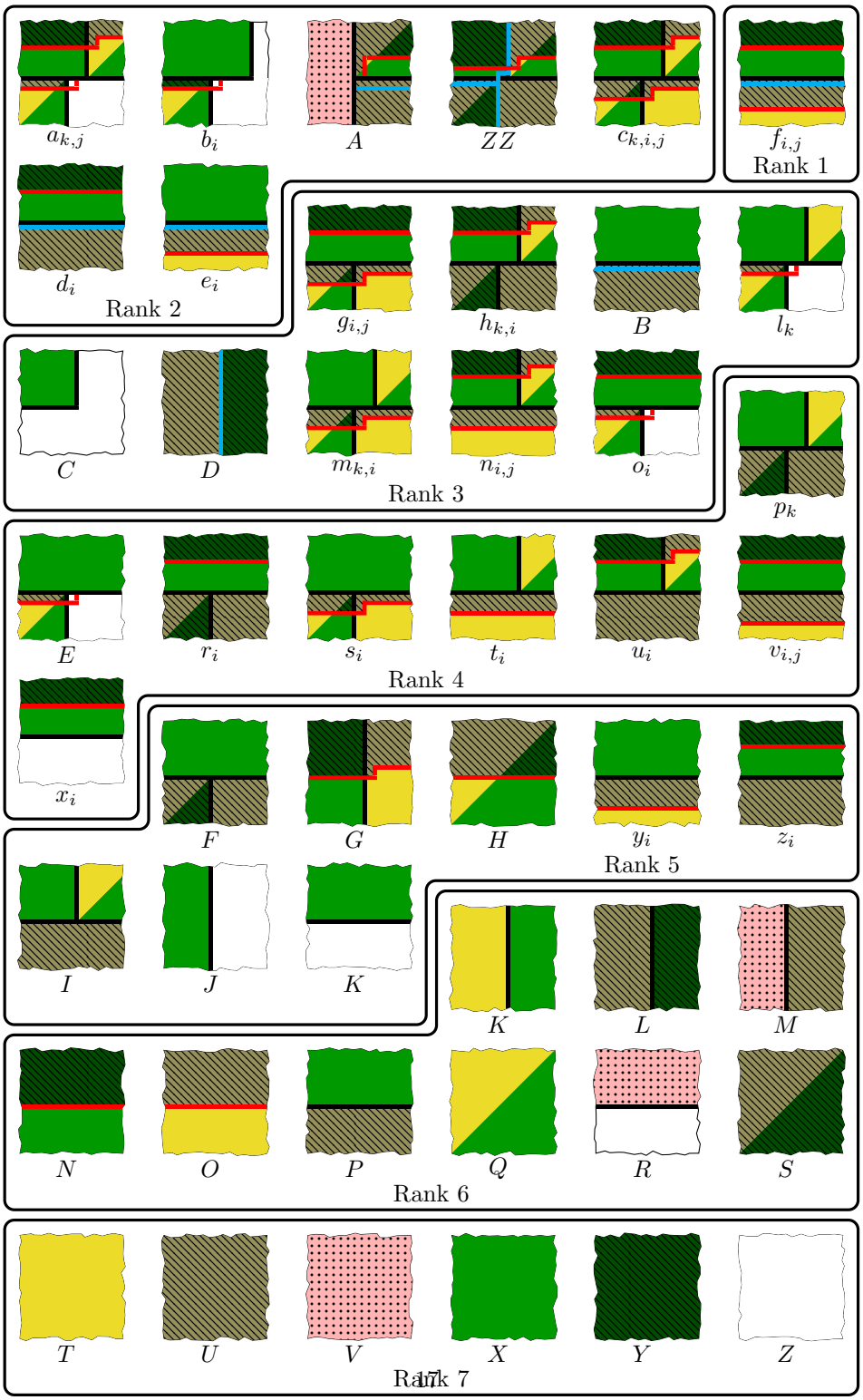


Figure 4: The configurations of $\mathcal{X}_T \setminus \{\alpha\}$, the subscripts indicate that vertical lines, horizontal lines and signals may be at different distances.

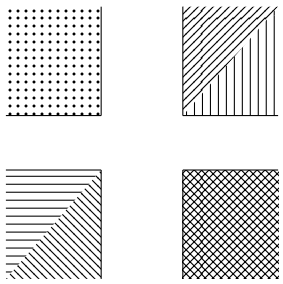


Figure 5: Uniform quarter- and eighth-planes in non-α-configurations.

Proof. Lemma 3.6 states that there is one configuration at most (up to shift) that has two horizontal lines or more. The other configurations necessarily have one of the following shapes:

- There is a horizontal line, in which case there can be at most one vertical line above and/or one vertical line below, otherwise there would necessarily be squares and hence two horizontal lines. There can also be a counting signal arbitrarily far above and/or below.
- There is no horizontal line, then there is at most one vertical line. An increase signal may again appear on the left and/or on the right.

All the points of $\mathcal{X}_T$ are shown in figures 3 and 4. □

Corollary 3.8. *SFT $\mathcal{X}_T$ is countable and all its configurations are computable from a single Turing machine.*

Let's see now how to encode computations inside the α configuration. First note that on each square's bottom line, there is at most one vertical line ending: suppose that the corner tile is at abscissa 0, then on the kth row, containing exactly k squares of size k, the vertical lines are at abscissa $i(k+1)$, for $0 \leq i \leq k$ and in the row above the vertical lines are at abscissa $i(k+2)$ with $0 \leq i \leq k+1$. Since

$$(i-1)(k+2) < i(k+1) < i(k+2) \qquad \text{for } 0 < i \leq k+1,$$

this is true for all squares, except for the leftmost ones for which the first vertical line is the same. So one may see α as a grid, for which the number of intersections increases of 1 for each row, see Figure 6.

Now, if we want to encode computations in T, we can use a classical encoding of Turing machines as Wang tiles, with the tile starting the computation on the corner. Since the grid grows, the Turing machine will never run out of space.

Our reductions will use SFTs based on this construction, they will be feature a different tilings on its grid. We qualify an SFT which is basically T with a tiling on its grid as having T-structure.

Definition 3.1 (T-structure). *We say an SFT X has T-*structure *if it is a copy of T to which we superimposed new symbols only on the symbols representing the horizontal/vertical lines and their crossings.*

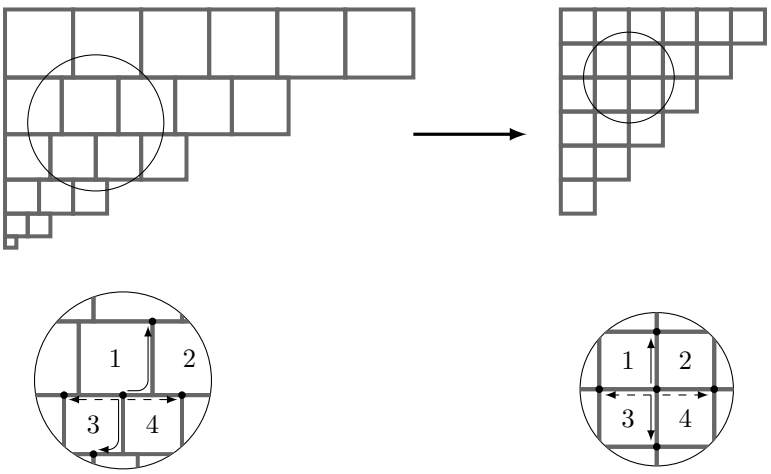


Figure 6: How the grid of α may be seen as a regular grid. In particular, one can see how information may be transmitted from one intersection to its neighbors.

Note that an SFT may have T-structure while having no α-configuration: for instance if you put a encode computations of a Turing machine that always halts and produce an error when it halts.

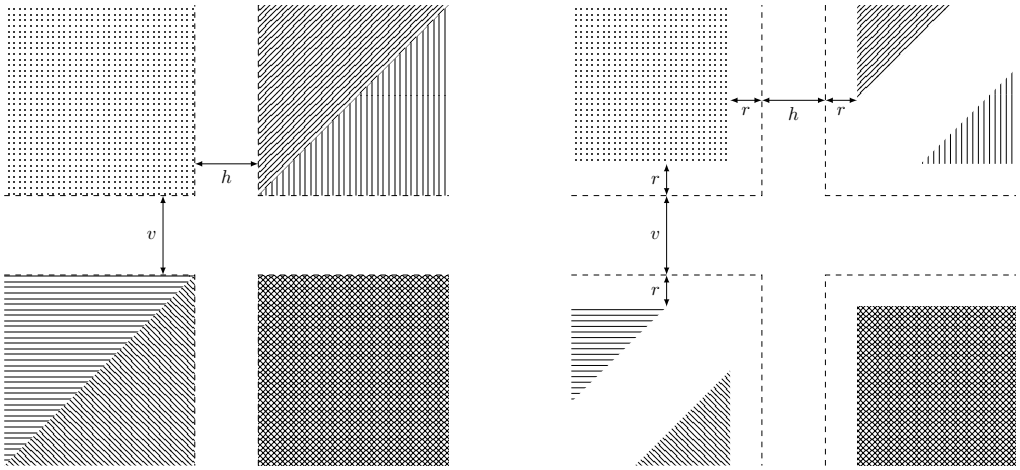
The reduction. The next lemma states a very intuitive result, that will be used later, namely that if an SFT with T-structure factors to another one, then the structure of each point is preserved by factorization. Furthermore, it shows that the factor map can only send a cell to its corresponding one, that is to say cell of the preimage has to be in the window of the image.

Lemma 3.9. *Let X, Y be two SFTs with T-structure, such that X factors onto Y. Let r be the radius of the factor map, then any α-configuration of Y is factored on by an α-configuration of X shifted by v, with $\|v\|_\infty \leq r$.*

Proof. By Lemma 3.7 we know that non-α configurations have two uniform (same symbols everywhere) quarter-planes and four uniform eighth-planes, as seen on Figure 7. The two north east eighth-planes are not uniform in configuration α. Thus these configurations cannot be factored on α.

It remains to prove the second part: that in the factoring process the α-structure is at most shifted by the radius of the factorization. We do that by *reductio ad absurdum*, suppose that an α-configuration x of X is mapped to an α-configuration y of Y and shifts it by $v = (v_x, v_y)$, with $\|v\|_\infty > r$. Without loss of generality we may suppose that $v_x > r$ and $v_y > 0$ and that the corner tile of the preimage is at position $(0, 1)$. We are now going to show that this is not possible.

For all $k \in \mathbb{N}^*$ there is a square with lower left corner at $(2k^2 + k, 2k^2 + k)$, see Figure 8 on the left. Inside this square, there are two $(k-1) \times (k-1)$ uniform smaller square subpatterns, see Figure 8 on the right. Now take k such that $k > (\|v\|_\infty + 2r + 1)$. By hypothesis, there is a vertical line symbol t at $z_p = (2k^2 + 2k + 1, 2k^2 + k)$ on x, and thus



(a) The uniform quarters and eights of planes of the non-α configurations. The horizontal and vertical distances h and v between these planes correspond to the distances between lines and signals.

(b) The image by F of these uniform subplanes still is uniform, note that it has been amputated of strips of width r, the radius of F.

Figure 7: The images of non-α configurations are necessarily non-α configurations. The white zones represent the zones which are not uniform.

at $z_i = (2k^2 + 2k + 1 + v_x, 2k^2 + k + v_y)$ on y. We know $x_{|z_i+B_r}$ has image t, and by what precedes that $x_{|z_i+B_r} = x_{|z_i+(1,0)+B_r}$ since they are both uniform, therefore, there should be two t symbols next to each other in y at z_i and $z_i + (1, 0)$. This is impossible.

□

Theorem 3.10. *Given two SFTs X, Y as an input, deciding whether X factors onto Y is Σ^0_3-hard.*

For this proof, we will reduce from the problem COFIN, which is known to be Σ^0_3-complete, see Kozen [11]. COFIN is the set of Turing machines which run infinitely only on a finite set of inputs, as stated earlier.

Proof. Given a Turing machine M, we construct two SFTs X_M and Y_M such that X_M factors on Y_M iff the set of inputs on which M does not halt is finite. We first introduce an SFT Z_M on which both will be based. It will have T structure. Above the T base, we allow the cells of the grid to be either white or blue according to the following rules:

- All cells on a same horizontal line are of the same color.
- A blue horizontal line may be above a white horizontal line, but not the contrary.

We now allow computation on blue cells only. The Turing machine M is launched on the input formed by the size of the first blue line (in number of cells). We forbid the machine to halt.

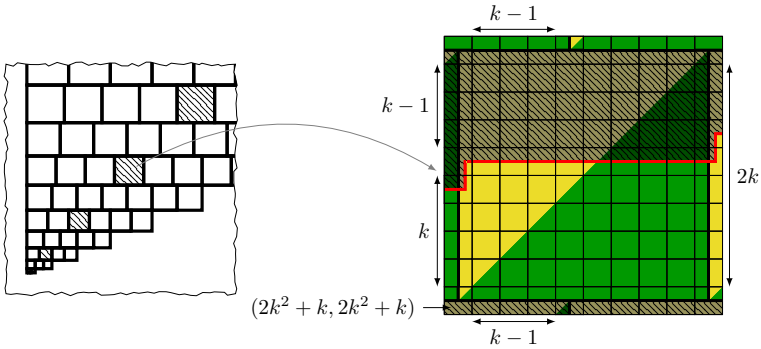


Figure 8: For every $k \in \mathbb{N}^*$, the square starting at position $(2k^2+k, 2k^2+k)$ is of the form on the right. We can see that there are two uniform $(k-1) \times (k-1)$ square subpatterns at $(2k^2+2k+2, 2k^2+k+1)$ and $(2k^2+k+1, 2k^2+2k+2)$ respectively.

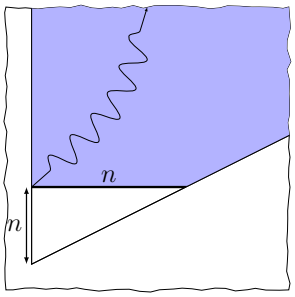


Figure 9: Computation on input n in the SFT Z, the blue zone contains the computation, and its distance from the corner tile corresponds to the input.

So for each n on which M does not halt, there is a configuration with white cells until the first blue diagonal appears, then computation occurs inside the blue cone, see Figure 9 for a schematic view. If M halts on n, then there is no configuration where the first blue line codes n. By compactness, there is of course a configuration with only white lines. If M is total, then the only α-configuration in Z_M is the one with no blue horizontal lines.

Now from Z_M, we can give X_M and Y_M:

- X_M: Let Z'_M be a copy of Z_M to which we add two decorations 0 and 1 on the blue cells only, and all blue cells in a configuration must have the same decoration. Now X_M is Z'_M to which we add a third color, red, that may only appear alone, instead of white and blue (one can see this as adding a copy of the configurations with only white horizontal lines). No computation is superimposed on red.

- Y_M is a copy of Z_M where we decorated only the corner tile with two symbols 0 and 1.

We now check that X_M factors onto Y_M iff M does not halt on a finite set of inputs:

$\Rightarrow$ Suppose the set of inputs on which M does not halt is finite: there exists N such that M halts on every input greater than N. The following factor map F works:

- F is the identity on Z_M. Note that the additional copy of T is also sent to the component Z_M.
- F has a radius big enough so that if its window is centered on the corner tile, it would cover the beginning of a computation on input N, that is $r_F > N^2 + N$.
- An α-configuration x of X_M is sent on the same α-configuration y in Y_M. For the decorations, when there is a computation on x, the factor map can see it and gives the same decoration to the corner tile of y. When there is no computation, the factor map doesn't see a computation zone and gives decoration 0 to the corner tile. The configuration with only white diagonals and decoration 1 of Y_M is factored on by the α-configuration colored in red contained in X_M.

Note that this also works when M is total.

$\Leftarrow$ Conversely, suppose M does not halt on an infinite set of inputs, and that there exists a factor map F with radius r: Lemma 3.9 states that all α-configurations of Y_M are factored on by α-configurations of X_M. Now, there is an infinite number of α-configurations with corner tile decorated with 0 (resp. 1) in Y_M, they all must be factored on by some α-configuration of X_M. Still by Lemma 3.9, the corner tile of the preimage must be in the window of the corner tile of the image. However, there can only be a finite number of configurations in which the symbols in this window differ. So the α-configurations of X_M factor to a finite number of α-configurations of Y_M with one of the decorations. This is impossible.

Note that the construction of X_M and Y_M from the description of M is computable and uniform. The reduction is thus many-one. □

4. Embedding

4.1. SFTs

We prove now Theorem 0.5 stating that the embedding problem for SFTs is Σ^0_1-complete for SFTs. We start with an analogue of Lemma 3.9 :

Lemma 4.1. *Let X, Y be two SFTs with T-structure, such that X embeds into Y. Let r be the radius of the embedding, then any α-configuration of X is mapped to an α-configuration of Y shifted by v, with $\|v\|_\infty \leq r$.*

Proof. First note that the uniform points of X must be mapped to uniform points of Y. So all different uniform points, and thus all uniform patterns of support B_r, have different images. Now an α-configuration of X has arbitrarily large uniform areas, as seen in Lemma 3.9, see also Figure 8. These uniform areas alternate, so their image also alternates when they are sufficiently large. The only configurations that have increasingly large alternating uniform areas are α-configurations. So α-configurations of X are mapped to α-configurations of Y. The proof that these mappings do not shift the T-structure by more than r is exactly the same as in Lemma 3.9. □

Lemma 4.2. *Let X and Y be two SFTs, it is Σ^0_1 to check whether X embeds into Y.*

Proof. To decide whether X embeds into Y, we have to check if there exists an injective local function $F : X \to Y$. Such a function being local, it can be guessed with a first order existential quantifier. To check that it is an embedding, we have to check that $F(X) \subseteq Y$ and that for all $x_1, x_2 \in X$, $x_1 \neq x_2 \Rightarrow F(x_1) \neq F(x_2)$. We know from Lemma 2.1 that checking $F(X) \subseteq Y$ is Σ_1^0. We now show that the second part is also Σ_1^0 by showing that the two following statements are equivalent.

- There exist $x_1, x_2 \in X$ such that $x_1 \neq x_2$ and $F(x_1) = F(x_2)$.
- For all $r > \max(r_F, r_X)$, there exist two r-blocks M_1 and M_2 such that M_1, M_2 are extensible and $(M_1)_0 \neq (M_2)_0$ and $F(M_1) = F(M_2)$.

It is clear that the second statement is Π_1^0 and that the first statement is the negation of the definition of injectivity. Now to the proof of their equivalence:

- Suppose there exist two different points $x_1, x_2 \in X$ such that $F(x_1) \neq F(x_2)$, we may assume x_1 and x_2 differ in 0 by shifting. For all $r > \max(r_F, r_X)$, the central r-blocks M_1, M_2 of x_1, x_2 are extensible and differ in 0
- Suppose now that for all $r > \max(r_F, r_X)$ there exist two extensible r-blocks M_1^r, M_2^r differing in 0 and such that $F(M_1^r) = F(M_2^r)$. By the pigeonhole principle, there is an infinity of M_1^r which have the same symbol in 0 and thus of M_2^r without this symbol in 0. Take these subsequences of M_1^r and M_2^r, by compactness we can extract converging subsequences from them which converge to two points $x_1, x_2 \in X$ with different symbols in 0. These two points have the same image, by construction.

□

Lemma 4.3. *Given two SFTs X, Y as an input, deciding whether X embeds into Y is Σ_1^0-hard.*

We will use a reduction from the halting problem, the set of Turing machines that halt on a blank input, and a construction based on a T-structure, as before.

Proof. Given a Turing machine M, we construct two SFTs X_M and Y_M such that X_M embeds into Y_M iff the Turing machine M halts. Both SFTs have as a base an SFT Z_M with a T-structure, in which we encode computations of M. Let us describe Z_M : Z_M is only T on which we directly encode the computation of M, it may eventually reach a halting state in which case the remaining space is given a new color, say blue. So our SFT Z_M can take two different forms : if the machine M halts, then a blue zone appears, if it does not halt, then this zone does not appear.

- Now X_M is Z_M for which we add a decoration to the corner tile, 0 or 1, so there are two different grid points in any case, whether the machine M halts or not.
- Y_M is Z_M for which we add a decoration to the halting state only (it appears at most once), there are two different grid points only when the machine M halts.

Let us check now that X_M embeds into Y_M if and only if M halts.

$\Rightarrow$ When the machine M halts, X_M embeds into Y_M : the radius of the embedding r is the distance between the halting state and the corner, the decoration of the corner is just translated to the halting state. All the rest remains unchanged. Note that there are less non α-configurations in X_M than in Y_M : they are the same except for the configurations containing exactly one horizontal line and two vertical lines with a halting state at their crossing. They have different decorations in Y_M but not in X_M.

$\Leftarrow$ When the machine M does not halt, there are two different α-configurations in X_M up to shift, while there is only one in Y_M, so they must have the same image.

□

4.2. Effective and sofic shifts

Lastly, we prove Theorem 0.6:

Lemma 4.4. *It is Σ^0_3 to decide given two effective subshifts X, Y whether X embeds in Y.*

Proof. To decide whether X embeds into Y one needs to decide whether there exists F such that $F(X) \subseteq Y$ and for all $x_1, x_2 \in X$, $x_1 \neq x_2 \Rightarrow F(x_1) \neq F(x_2)$. Guessing F and checking whether $F(X) \subseteq Y$ is Σ^0_3, as a consequence of Lemma 2.6, while checking the injectivity part remains Σ^0_1: as for the SFT case, this is equivalent to negating the following statement which remains Σ^0_1 in the effective case:

- For all $r > \max(r_F, r_X)$, there exist two r-blocks M_1 and M_2 such that M_1, M_2 are extensible and $(M_1)_0 \neq (M_2)_0$ and $F(M_1) = F(M_2)$.

□

And a corollary of the proof of Theorem 2.11, we obtain the Σ^0_3-hardness for effective subshifts.

Corollary 4.5. *Given X, Y two effective subshifts, it is Σ^0_3-hard to decide whether X embeds in Y.*

[1] Aubrun, N., Sablik, M., 2013. Simulation of effective subshifts by two-dimensional subshifts of finite type. Acta Applicandae Mathematicae.

[2] Berger, R., 1964. The Undecidability of the Domino Problem. Ph.D. thesis, Harvard University.

[3] Boyer, L., Theyssier, G., 2010. On factor universality in symbolic spaces. In: MFCS. pp. 209–220.

[4] Boyle, M., 1983. Lower entropy factors of sofic systems. Ergodic Theory and Dynamical Systems 3, 541–551.

[5] Boyle, M., 2008. Open Problems in Symbolic Dynamics. Contemporary Mathematics 469, 69–118.

[6] Hedlund, G. A., 1969. Endomorphisms and automorphisms of the shift dynamical system. Theory of Computing Systems 3 (4), 320–375.

[7] Hochman, M., 2009. A note on universality in multidimensional symbolic dynamics. Discrete and Continuous Dynamical Systems S 2 (2).

[8] Hochman, M., Apr. 2009. On the dynamics and recursive properties of multidimensional symbolic systems. Inventiones Mathematicae 176 (1), 2009.

[9] Jeandel, E., Vanier, P., 2011. Π^0_1 sets and tilings. In: Theory and Applications of Models of Computation (TAMC). Vol. 6648 of Lecture Notes in Computer Science. pp. 230–239.

[10] Jeandel, E., Vanier, P., 2013. Hardness of conjugacy, embedding and factorization of multidimensional subshifts of finite type. In: Portier, N., Wilke, T. (Eds.), STACS. Vol. 20 of LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 490–501.

[11] Kozen, D., 2006. Theory of Computation. Springer, New York.

[12] Krieger, W., 1982. On the subsystems of topological markov chains. Ergodic Theory and Dynamical Systems 2 (02), 195–202.
URL http://dx.doi.org/10.1017/S0143385700001516

[13] Lind, D., Marcus, B., 1995. An introduction to symbolic dynamics and coding. Cambridge University Press, New York, NY, USA.

[14] Lind, D. A., 2004. Multi-Dimensional Symbolic Dynamics. In: Williams, S. G. (Ed.), Symbolic Dynamics and its Applications. No. 60 in Proceedings of Symposia in Applied Mathematics. American Mathematical Society, pp. 61–79.

[15] Robinson, R. M., 1971. Undecidability and Nonperiodicity for Tilings of the Plane. Inventiones Math. 12.

[16] Rogers, Jr., H., 1987. Theory of recursive functions and effective computability. MIT Press, Cambridge, MA, USA.

[17] Wang, H., 1961. Proving theorems by Pattern Recognition II. Bell Systems technical journal 40, 1–41.

[18] Williams, R. F., 1973. Classification of subshifts of finite type. Annals of Mathematics 98, 120–153.