Hardness of Conjugacy, Embedding and Factorization of multidimensional Subshifts of Finite Type*

Emmanuel Jeandel[1] and Pascal Vanier[2]

1 **LORIA**
Campus Scientifique - BP 239
54506 Vandoeuvre-les-Nancy
France
`emmanuel.jeandel@loria.fr`
2 **Einstein Institute of Mathematics**
Hebrew University
Givat Ram
Jerusalem 91904
Israel
`pascal.vanier@lif.univ-mrs.fr`

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 of finite type (SFTs) in dimension $d > 1$. In particular, we prove that the factorization problem is Σ_3^0-complete and that the conjugacy and embedding problems are Σ_1^0-complete in the arithmetical hierarchy.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases Subshifts, Computability, Factorization, Embedding, Conjugacy

Digital Object Identifier 10.4230/LIPIcs.STACS.2013.490

A d-dimensional Subshift of Finite Type (SFT) is the set of colorings of $\mathbb{Z}^d$ by a finite set of colors in which a finite set of forbidden patterns never appear. One can also see them as tilings of $\mathbb{Z}^d$, and in dimension 2 they are equivalent to the usual notion of tilings introduced by Wang [13]. SFTs 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 $\Sigma = \{1, \ldots, n\}$ 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 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 [4], although has been proved decidable in the particular case of one-sided SFTs on $\mathbb{N}$, see [14]. 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 emptyness problem which is Σ_1^0-complete [1]. 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:

* This work was sponsored by the ANR grant ANR-09-BLAN-0164 and the ISF grant number 1409/11.

30th Symposium on Theoretical Aspects of Computer Science (STACS'13).
Editors: Natacha Portier and Thomas Wilke; pp. 490–501
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

▶ **Theorem 1.** *For any fixed X, given Y as an input, it is Σ^0_1-complete to decide if 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 [3]. 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 [2, 5]. 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 2.** *Given two SFTs X, Y as inputs, it is Σ^0_3-complete to decide if 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 $h(X) > h(Y)$ [8]. We prove here that the problem is Σ^0_1-complete:

▶ **Theorem 3.** *Given two SFTs X, Y as inputs, it is Σ^0_1-complete to decide if X embeds into Y.*

The paper is organised as follows: first we give the necessary definitions and fix the notation is section 1, after what we give the proofs of Theorems 1, 2 and 3 in Sections 2, 3 and 4 respectively.

1 Preliminary definitions

1.1 Subshifts of finite type

We give here some standard definitions and facts about multidimensional subshifts, one may consult Lind [10] or Lind/Marcus [9] for more details.

Let Σ be a finite alphabet, its elements are called *symbols*, the d-dimensional full shift on Σ is the set $\Sigma^{\mathbb{Z}^d}$ of all maps (colorings) from $\mathbb{Z}^d$ to the Σ (the colors). For $v \in \mathbb{Z}^d$, the shift functions $\sigma_v : \Sigma^{\mathbb{Z}^d} \to \Sigma^{\mathbb{Z}^d}$, 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 $\Sigma^{\mathbb{Z}^d}$ is called a *configuration*.

Every closed shift-invariant (invariant by application of any σ_v) subset X of $\Sigma^{\mathbb{Z}^d}$ is called a *subshift*. 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 $p : P \to \Sigma$, 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 $\Sigma^{\mathbb{Z}^d}$ containing the configurations having nowhere a pattern of F. More formally, $X_{\mathcal{F}}$ is defined by

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

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.

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^0_1 to be precise) in general to know whether a pattern is extensible while it is always decidable efficiently (polytime) to know if a pattern is admissible.

As we said before, SFTs are compact spaces, this gives a link between admissible and extensible: 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 pattern, then we can extract from it a sequence converging to some point of the SFT.

Note that instead of using the formalism of SFTs for the constructions 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 Σ_X for the alphabet of the subshift X.

Let $X \subseteq \Sigma_X^{\mathbb{Z}^2}$ and $Y \subseteq \Sigma_Y^{\mathbb{Z}^2}$ be two subshifts a function $F : X \to Y$ is a block code if there exists a finite set $V = \{v_1, \dots, v_k\} \subset \mathbb{Z}^2$, the *window*, and a local map $f : \Sigma_X^{|V|} \to \Sigma_Y$, such that for any point $x \in X$ and $y = F(x)$, for all $z \in \mathbb{Z}^d$, $y_z = f(x_{z+v_1}, \dots, x_{z+v_k})$. 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.

When the map F is bijective and invertible and its inverse is also a block code, the subshifts X and 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 $[\![0,n]\!]^d$ where d is the dimension. 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 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 [12].

In computability, 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 $\forall$ or 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, $B \leq_m A$. The problem $0^{(n)}$ is Σ^0_n-complete. Furthermore, it is Σ^0_n-complete if it is in Σ^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.

2 Conjugacy

We prove here the Σ^0_1-completeness of the conjugacy problem in dimension $d \geq 2$, even for a fixed SFT. We first prove the following lemma, which is the first step to show that conjugacy is Σ^0_1 and also proves that equality is Σ^0_1.

▶ **Lemma 4.** *Given F, X, Y as an input, deciding if $F(X) \subseteq Y$ is Σ^0_1.*

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^0_1 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, hence 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, there exists a configuration $x \in X$ such that $F(x)$ contains a forbidden pattern in its center. ◀

▶ **Corollary 5.** *Given two SFTs X, Y as an input, it is Σ_1^0 to decide if $X = Y$.*

▶ **Theorem 6.** *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 $F : \Sigma_X^{B_{r_F}} \to \Sigma_Y$ and $G : \Sigma_Y^{B_{r_G}} \to \Sigma_X$ 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 Σ_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 4 that checking whether $F(X) \subseteq Y$ is Σ_1^0, we have the desired result. ◀

▶ **Theorem 7.** *For any X, given Y as an input, it is Σ_1^0-hard to decide if X and Y are conjugate (resp. equal).*

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 [11] 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 8.** *Given two SFTs X, Y as an input, it is Σ_1^0-hard to decide if $X = Y$.*

3 Factorization

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

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

▶ **Theorem 9.** *Let Y be the SFT containing exactly one configuration, a uniform configuration. Given X as an input, it is Π_1^0-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 Σ_Y. And the problem is hence equivalent to knowing whether a SFT is *not* empty, which is Π_1^0-complete. ◀

▶ **Theorem 10.** *Let Y be the empty SFT. Given X as an input, it is Σ_1^0-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_1^0-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 11 and 15 the Σ_3^0-completeness of the factorization problem.

3.1 Factorization is in Σ_3^0

▶ **Theorem 11.** *Given two SFTs X, Y as an input, deciding whether X factors onto Y is in Σ_3^0.*

Proof. The shift X factors onto Y iff there exists a factor map F, a local function, such that $F(X) = Y$. This is the first existential quantifier.The result follows from the next lemma and Lemma 4. ◀

▶ **Lemma 12.** *Given two SFTs X, Y and a local map F as an input, deciding if $Y \subseteq F(X)$ is Π_2^0.*

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_2^0 statement: *for any admissible pattern m of Y, if m is extensible, then $F^{-1}(m)$ contains an admissible pattern.* This statement is Π_2^0 since checking that m is *not* extensible is Σ_1^0, that is to say: there exists a radius r such that all r-blocks containing m are not admissible.

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 $x \in X$. A preimage of m being extensible, it is also admissible. This proves the first direction.

Conversely, suppose all extensible patterns m of Y have an admissible preimage. 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 admissible 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_3^0-hard

To prove the hardness, we use the base construction that we introduced in [6]: we note it T. This construction introduces a new way to put Turing machine computations in SFTs, in particular, the base construction has exactly one point (up to shift) in which computations may be encoded. We call this point *configuration α*, its schematic view is shown in Figure 2a. The computation is encoded in the inner grid which is sparse. Each crossing between a horizontal line and a vertical one forms a *cell*. The constraints are carried along the vertical and horizontal lines, so that we may view the encoding of the Turing machine as a tiling on the grid. For each time step, the tape of the Turing machine is encoded in the NW-SE diagonals and the size of the diagonal steadily increases in size when going north-east. At each growth of the diagonal size, it gains two cells.

Configuration α is made of two layers: one producing the horizontal lines and the other the vertical ones. The layer producing the vertical lines is shown in Figure 1, the vertical lines are the black vertical lines. The configuration producing the horizontal lines is its exact symmetric along the south-west/north-east diagonal. The key property of these layers is that when a corner tile (the tile in the lower left corner of the first square) appears, then the point is necessarily of this form.

In the original construction, corner tiles of the horizontal and vertical layers could only be superimposed to each other. We just change this so that instead, the corner tile of the vertical layer has to be at position $(1, -1)$ relative to the corner tile of the horizontal one. This change does not impact any of the properties of T, but simplifies a bit the proof of Lemma 14.

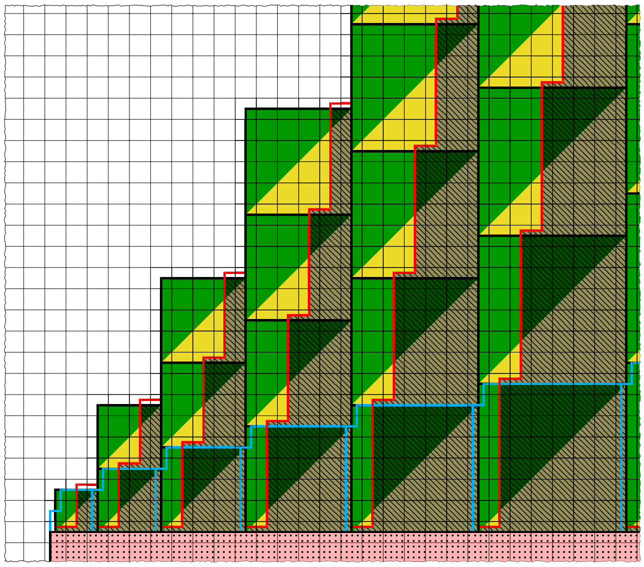

Figure 1 The vertical layer of point α, the meaningful point of X_T. The corner tile may be seen on the first non all-white column: it is the lower left corner of the square.

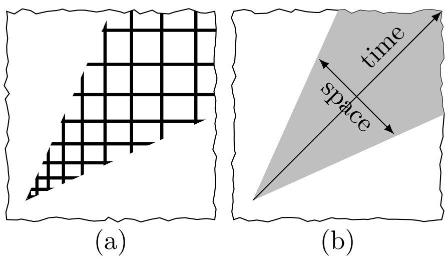

Figure 2 (a) The skeleton of configuration α. (b) How the computation is superimposed to α.

Our reduction will use two SFTs based on this construction, both of them will be feature a different tiling on its grid. We will say that an SFT which is basically T with a tiling on its grid as having T-structure.

▶ **Definition 13** (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.

Note that an SFT may have T-structure while having no α-configuration: for instance if you put a computation of a Turing machine that produces an error whenever it halts.

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 14.** *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 [6, Lemma 1], we know that non-α-configurations have at most one vertical line and one horizontal line. And therefore that they have two uniform (same symbols everywhere) quarter-planes and four uniform eighth-planes, as seen on Figure 3. The two north east eighth-planes are not uniform in α. Thus they cannot be factored on α.

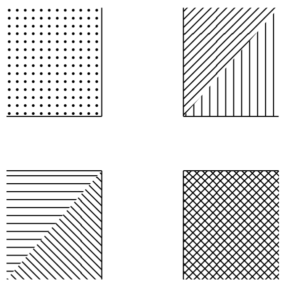

Figure 3 Uniform quarter- and eighth-planes in non-α-configurations.

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 contradiction, 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 vertical and horizontal corner tiles of the preimage are at positions $(0, 1)$ and $(1, 0)$ respectively. We are now going to show that this is not possible.

On the horizontal layer, for all $k \in \mathbb{N}^*$ there is a square with lower left corner at $(2k^2 + k, 2k^2 + k)$, see Figure 1. Inside this square, there are two $(k-1) \times (k-1)$ uniform smaller squares, see Figure 4. This being also true for the vertical layer, these squares remain uniform when they are superimposed. 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 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 eachother in y at z_i and $z_i + (1, 0)$. This is impossible. ◀

▶ **Theorem 15.** *Given two SFTs X, Y as an input, deciding whether X factors onto Y is Σ_3^0-hard.*

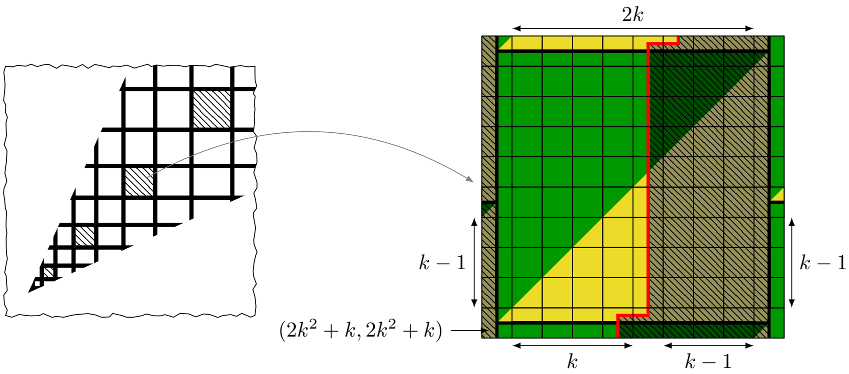

Figure 4 For every $k \in \mathbb{N}^*$, the square starting at position $(2k^2+k, 2k^2+k)$ is of the form on the right on the component producing the vertical lines (and is the symmetric along the diagonal for the one producing the horizontal lines). We can see that there are two uniform $(k-1) \times (k-1)$ squares at $(2k^2+2k+2, 2k^2+k+1)$ and $(2k^2+k+1, 2k^2+2k+2)$ respectively.

For this proof, we will reduce from the problem **COFINITE**, which is known to be Σ_3^0-complete, see Kozen [7]. **COFINITE** is the set of Turing machines which run infinitely only on a finite set of inputs.

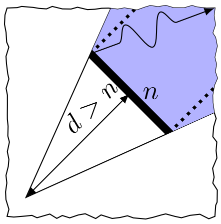

Figure 5 Computation on input n in the SFT Z, the number of white diagonals d preceeding the computation is strigtly greater than the input n.

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 NW-SE diagonal are of the same color.
- A blue diagonal may follow (along direction SW-NE) a white diagonal, but not the contrary.
- A transition from white diagonal to blue may only appear when the grid grows.

We now allow computation on blue cells only. Only the diagonals after the growth of the grid may contain computations. 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.

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 5 for a schematic view. If M halts on n, then there is no tiling where the first blue line codes n. By compactness, there is of course a configuration with only white diagonals. If M is total, then the only α-configuration in Z_M is the one with only white diagonals.

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. No computation is superimposed on red.
- Y_M is a copy of Z_M where we decorated only the horizontal 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 M does not halt on a finite set of inputs: 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 when its window is centered on the corner tile, it would cover the beginning of the computation on input 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 14 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 14, 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

We prove now Theorem 3 stating that the embedding problem is Σ^0_1-complete. We start with an analogue of Lemma 14 :

▶ **Lemma 16.** *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 14, see also Figure 4. These uniform areas alternate, so their image also alternates when they are sufficiently large. The only configurations that have growingly 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 14. ◀

▶ **Lemma 17.** *Let X and Y be two SFTs, it is Σ_1^0 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 4 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 admissible r-blocks M_1, M_2 such that $(M_1)_0 \neq (M_2)_0$ 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 :

- 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 admissible and differ in 0
- Suppose now that for all $r > \max(r_F, r_X)$ there exist two admissible 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 18.** *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.

⇒ 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 : these are the configurations containing an infinite cross of black lines with a halting state on top. They have different decorations in Y_M but not in X_M.

⇐ 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 there are two that must have the same image.

◀

References

1 Robert Berger. *The Undecidability of the Domino Problem.* PhD thesis, Harvard University, 1964.

2 Laurent Boyer and Guillaume Theyssier. On factor universality in symbolic spaces. In *MFCS*, pages 209–220, 2010.

3 Mike Boyle. Lower entropy factors of sofic systems. *Ergodic Theory and Dynamical Systems*, 3:541–551, 1983.

4 Mike Boyle. Open Problems in Symbolic Dynamics. *Contemporary Mathematics*, 469:69–118, 2008.

5 Michael Hochman. A note on universality in multidimensional symbolic dynamics. *Discrete and Continuous Dynamical Systems S*, 2(2), 2009.

6 Emmanuel Jeandel and Pascal Vanier. Π_1^0 sets and tilings. In *Theory and Applications of Models of Computation (TAMC)*, volume 6648 of *Lecture Notes in Computer Science*, pages 230–239, 2011.

7 Dexter Kozen. *Theory of Computation.* Springer, New York, 2006.

8 Wolfgang Krieger. On the subsystems of topological markov chains. *Ergodic Theory and Dynamical Systems*, 2(02):195–202, 1982.

9 Douglas Lind and Brian Marcus. *An introduction to symbolic dynamics and coding.* Cambridge University Press, New York, NY, USA, 1995.

10 Douglas A. Lind. Multi-Dimensional Symbolic Dynamics. In Susan G. Williams, editor, *Symbolic Dynamics and its Applications*, number 60 in Proceedings of Symposia in Applied Mathematics, pages 61–79. American Mathematical Society, 2004.

11 Raphael M. Robinson. Undecidability and Nonperiodicity for Tilings of the Plane. *Inventiones Math.*, 12, 1971.

12 Hartley Rogers, Jr. *Theory of recursive functions and effective computability.* MIT Press, Cambridge, MA, USA, 1987.

13 Hao Wang. Proving theorems by Pattern Recognition II. *Bell Systems technical journal*, 40:1–41, 1961.

14 R. F. Williams. Classification of subshifts of finite type. *Annals of Mathematics*, 98:120–153, 1973.