A characterization of subshifts with computable language

Emmanuel Jeandel
LORIA, Campus Scientifique - BP 239
54506 VANDOEUVRE-LES-NANCY, FRANCE
emmanuel.jeandel@loria.fr

Pascal Vanier
Laboratoire d'Algorithmique, Complexité et Logique
Université de Paris-Est, LACL, UPEC, France
pascal.vanier@lacl.fr

Abstract

Subshifts are sets of colorings of $\mathbb{Z}^d$ by a finite alphabet that avoid some family of forbidden patterns. We investigate here some analogies with group theory that were first noticed by the first author. In particular we prove several theorems on subshifts inspired by Higman's embedding theorems of group theory, among which, the fact that subshifts with a computable language can be obtained as restrictions of minimal subshifts of finite type.

2012 ACM Subject Classification Theory of computation → Models of computation

Keywords and phrases subshifts, computability, Enumeration degree, Turing degree, minimal subshifts

Digital Object Identifier 10.4230/LIPIcs.STACS.2019.37

Funding *Pascal Vanier*: Sponsored by grant TARMAC ANR 12 BS02 007 01.

Acknowledgements The authors wish to thanks the anonymous referees for many helpful remarks and improvements.

36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019).
Editors: Rolf Niedermeier and Christophe Paul; Article No. 37; pp. 37:1–37:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Subshifts are colorings of $\mathbb{Z}^d$ by some finite alphabet Σ avoiding some family of forbidden patterns. They are closed shift invariant subsets of $\Sigma^{\mathbb{Z}^d}$. The most commonly studied family of subshifts are the subshifts of finite type (SFTs), those that can be defined via a finite family of forbidden patterns, which correspond to the sets of colorings by Wang tilesets.

It is well known since the work of Berger [5] that many problems or invariants in tiling theory, and therefore for subshifts of finite type, are not computable. A recent trend in multidimensional symbolic dynamics initiated by Hochman [16, 17] shows that computability is not a fluke but an integral part of the study of subshifts. Indeed, many recent results show precise correspondences between computability notions and invariants for subshifts [25, 19]. This has led to the study of another class of subshift, effective (or effectively closed) subshifts: subshifts which are defined by a recursively enumerable family of forbidden patterns.

Of particular interest is the embedding (simulation) theorem by Hochman [16], extended by Aubrun-Sablik and Durand-Romashchenko-Shen [2, 10], that characterizes effectively closed subshifts, as projections of higher dimensional subshifts of finite type,

This theorem is strikingly similar to theorems in combinatorial group theory and first order logic. The Higman embedding theorem [14] characterizes recursively presented groups, i.e. groups given by a *computable* set of relators, as subgroups of finitely presented groups, i.e. groups given by a *finite* set of relators. The Kleene-Craig-Vaught [21, 8] theorem characterizes recursively axiomatisable theories, i.e. theories given by a *computable* set of axioms, as syntactic restrictions of finitely axiomatisable theories, i.e. theories given by a *finite* set of axioms.

Based on this analogy, the first author described a general theory [18] in which many theorems of these three fields can be formulated using an unified framework, and a dictionary between similar notions can be established. The framework is quite abstract and it cannot be used to prove the embedding theorems above for all these theories at once: they rely after all in each case on properties of an encoding of Turing machines, and this encoding heavily depends on the theory under consideration. It suggests nonetheless that there is more than a similarity between these theorems, and that something deeper is to be found.

In this article, we study this by providing analogues in symbolic dynamics of the other embedding theorems of Higman:

- The relative Higman theorem [15] which, as its name indicates, is a relativized version of the classic Higman theorem
- The Boone-Higman-Thompson [6, 26] theorem that characterizes groups with computable word problem as subgroups of simple recursively presented groups.

The first theorem is presented in section 2.2. It is very similar to a theorem in a previous article by Aubrun and Sablik [1]. As we will explain, their article suffers however from unfortunate mistakes and the theorem they proved is regrettably wrong.

The second theorem is presented in section 2.3. The Boone-Higman-Theorem in our context, becomes: “A subshift has a computable language iff it is the restriction of a minimal subshift, itself a restriction of a subshift of finite type”. Using recent results from Durand and Romashchenko [11], this can be simplified to “A subshift has a computable language iff it is the restriction of a minimal subshift of finite type”. Whether such a simplification is possible for groups (i.e. whether any group with a computable word problem is a subgroup of a finitely presented simple group) is a long standing open question.

The article is organized as follows. We first start with defining the relevant notions from symbolic dynamics, computability theory, and group theory. We will then explain how concepts from group theory translate into notions of symbolic dynamics. The remaining part is devoted to the proof of the three Higman theorems for subshifts: the classic Higman theorem (a slight reformulation of the Hochman-Aubrun-Sablik-Durand-Romashchenko-Shen theorem), the relative Higman theorem and the Boone-Higman-Thompson theorem.

1 Preliminary definitions

1.1 Subshifts

The d-dimensional full shift is the set $\Sigma^{\mathbb{Z}^d}$ where Σ is a finite alphabet whose elements are called *letters* or *symbols*. Each element of the full shift may be seen as a coloring of $\mathbb{Z}^d$ with the letters of Σ. For $v \in \mathbb{Z}^d$, the shift function $\sigma_v : \Sigma^{\mathbb{Z}^d} \to \Sigma^{\mathbb{Z}^d}$ is defined by $\sigma_v(x_z) = x_{z+v}$. The full shift equipped with the distance $d(x, y) = 2^{-\min\{\|v\| \mid v\in\mathbb{Z}^d, x_v\neq y_v\}}$ forms a compact metric space on which the shift functions act as homeomorphisms. A closed shift invariant subset X of $\Sigma^{\mathbb{Z}^d}$ is called a *subshift* or *shift*. An element of a subshift X is called a *configuration* or *point*.

Subshifts are exactly the subsets of $\Sigma^{\mathbb{Z}^d}$ that avoid some *family of forbidden patterns*. A *pattern* of shape P, where P is a *4-connected*[1] finite subset of $\mathbb{Z}^d$, is an element of Σ^P or alternatively a function $p : P \to \Sigma$. A configuration x avoids a pattern p of shape P if $\forall z \in \mathbb{Z}^d$, $p \neq \sigma_z(x)_{|P}$.

Subshifts can thus be defined by some family of patterns they avoid. When a subshift can be defined this way by a finite family, it is called a *subshift of finite type*. When a subshift can be defined by a recursively enumerable family of forbidden patterns, it is called an *effectively closed subshift*.

If X is a subshift, we denote by $\mathcal{L}(X)$ its *language*, *i.e.* the set of patterns that appear somewhere in one of its points.

▶ **Example 1.** The set X_1 of all biinfinite words over the alphabet $\{a, b\}$ that do not contain the word aa is, by definition, a subshift. It is defined by the set of forbidden patterns $\mathcal{F} = \{aa\}$. Another possible defining set of forbidden patterns is $\mathcal{F} = \{aab, aaa\}$

▶ **Example 2.** The set X_2 of all biinfinite words over the alphabet $\{a, b\}$ where the letter a appears at most once is a subshift. It is defined e.g. by the set of forbidden patterns $\mathcal{F} = \{ab^na, n \in \mathbb{N}\}$. It can be proven that it is not a subshift of finite type, although it is certainly an effectively closed subshift.

We denote by $\Sigma^{d\star}$ the set of d-dimensional patterns over the alphabet Σ. For $d = 1$, we write this $\Sigma^\star$. As an abuse of notation, we consider a d-dimensional pattern to be also a k-dimensional pattern for $k > d$ along the d first dimensions: as an example if X is a d-dimensional subshift, $\mathcal{L}(X) \cap A^\star$ is the set of one dimensional patterns (i.e. horizontal words) over the alphabet A that appear in X.

▶ **Example 3.** Let X_3 be the two-dimensional subshift over the alphabet $\{0, 1\}$ defined with the set of forbidden patterns $\mathcal{F} = \{\left(\begin{smallmatrix}1\\1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & 1\end{smallmatrix}\right)\}$. X_3 is therefore the set of colorings of the plane with 0 and 1 s.t. no two symbols 1 can be put next to each other. It is easy to see that $\left(\begin{smallmatrix}0 & 1\\1 & 0\end{smallmatrix}\right) \in \mathcal{L}(X_3)$ but $\left(\begin{smallmatrix}1 & 0\\1 & 0\end{smallmatrix}\right) \notin \mathcal{L}(X_3)$.

Notice that any subshift X can always be defined by its set $\mathcal{L}(X)^c$. In particular X is an effectively closed subshift iff $\mathcal{L}(X)^c$ is recursively enumerable.

[1] The exact notion of connectedness we use is irrelevant. However it is crucial in what follows to look only at connected patterns.

1.2 Combinatorial Group Theory

We assume the reader has a passing familiarity with group theory, and will focus this brief description to the specifics of combinatorial group theory.

A good introduction to this particular aspect may be found in [22, 24]. The book by Higman and Scott [15] contains invaluable information about the interplay between group theory and computability.

A set of generators for a group G is a set S s.t. for any $g \in G$, there exist $s_1^{\pm 1}, \dots, s_n^{\pm 1} \in S$ such that $g = s_1 \cdots s_n$. A group is *finitely generated* if there exists a finite such S.

Let $a_1 \dots a_k$ be a set of generators for some finitely generated group G. The *word problem* for G, denoted $\mathcal{WP}(G, \{a_1 \dots a_k\})$ is the language of all formal words over the alphabet $\{a_1^{\pm 1} \dots a_k^{\pm 1}\}$ that evaluates to 1 (the identity element) in G. The computability properties of $\mathcal{WP}(G, \{a_1 \dots a_k\})$ do not depend on the set of generators (as long as it is finite), so that we will usually speak of the word problem as $\mathcal{WP}(G)$ without specifying the generators.

There is (up to isomorphism) a unique largest group generated by n elements, which is called the free group F_n on n generators. If the generators are written $a_1 \dots a_n$, F_n can be thought of as the set of all irreducible words over the alphabet $\{a_1^{\pm 1} \dots a_n^{\pm 1}\}$, i.e. all words that do not contain $a_i a_i^{-1}$ or $a_i^{-1} a_i$ as factors, with the obvious product operation.

F_n is the largest group with n generators $a_1 \dots a_n$ in the sense that if G is a group with n generators $s_1 \dots s_n$, then there is a unique onto morphism ϕ s.t. $\phi(a_i) = s_i$.

In particular any group with n generators can be seen as a quotient of a free group. This gives rise to the notion of groups given by generators and relations.

If $\mathcal{R}$ is a set of formal words over $\{a_1^{\pm 1} \dots a_n^{\pm 1}\}$, we denote by $\langle a_1, a_2, \dots a_n \mid \mathcal{R} \rangle$ the largest group G generated by n elements $a_1 \dots a_n$ s.t. all relations in $\mathcal{R}$ evaluate to 1 in the group G. Formally, G is the quotient of the free group F_n by the smallest normal subgroup N of F_n that contains all relations $\mathcal{R}$.

A finitely generated group G is finitely presented if $G = \langle S \mid \mathcal{R} \rangle$ for some finite S and $\mathcal{R}$, or more generally if G is isomorphic to such a group. G is *recursively presented* if $G = \langle S \mid \mathcal{R} \rangle$ for some finite[2] S and recursively enumerable set $\mathcal{R}$.

▶ **Example 4.** The group $G = \mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ is finitely presented. A possible finite presentation is $G = \left\langle a_1, a_2 \mid a_1 a_2 a_1^{-1} a_2^{-1}, a_2^3 \right\rangle$. There are of course other presentations with the same generators, for example $G = \left\langle a_1, a_2 \mid a_2 a_1 a_2 a_1^{-1} a_2, a_2^3 \right\rangle$.

For this group G, we have $a_1 a_2 a_1 a_2 \notin \mathcal{WP}(G, \{a_1, a_2\})$ and $a_1^2 a_2 a_1^{-1} a_2 a_1^{-1} a_2 \in \mathcal{WP}(G, \{a_1, a_2\})$.

Notice that for all groups G with generators S, we have that $G = \langle S \mid \mathcal{WP}(G, S) \rangle$ and that G is recursively presented iff $\mathcal{WP}(G)$ is recursively enumerable.

1.3 Subshifts as analogs of subgroups

There is a natural analogy between subshifts and subgroups, which is obtained in the following way: the alphabet plays the role of the generators and the forbidden patterns play the role of the relations.

If X is a d-dimensional subshift over the alphabet Σ given by the forbidden patterns $\mathcal{F}$, we will write $X = \langle \Sigma \mid \mathcal{F} \rangle^d$ to further stress the analogy between groups and subshifts.

Continuing the analogy, the word problem $\mathcal{WP}(G)$ of G correspond naturally to the complement of the language of X, $\mathcal{L}(X)^c$. In particular, if S is a set of generators, $G = \langle S \mid \mathcal{WP}(G, S) \rangle$. If X is a subshift over alphabet Σ, then $X = \langle \Sigma \mid \mathcal{L}(X)^c \rangle^d$.

---

[2] One could take more generally S to be recursively enumerable

To further the correspondence, we need an analogy in subshifts of the operations of adding/removing generators and relations. In terms of groups, H is obtained from G by adding relations iff H is a quotient of G. In terms of subshifts, Y is obtained from X by adding forbidden patterns iff $X \subseteq Y$. So taking a quotient corresponds to subshift containment.

If H is obtained from G by removing generators, it means that H is a subgroup of G (of course not all subgroups can be obtained this way). What the operations of removing symbols means for subshifts is discussed in the following section.

1.3.1 Removing symbols and dimensions

Removing symbols, or removing dimensions, is intuitively easy:

▶ **Definition 5.** *Let X' be a subshift over an alphabet Σ' of dimension d' and let $\Sigma \subseteq \Sigma'$ and $d < d'$, the (Σ^k, d)* restriction X of X' *is the set of d-dimensional configurations of width k over the alphabet Σ that appear in X'. We write $X \prec X'$ if X is some restriction of X'.*

By compactness X is exactly the subshift of dimension d over the alphabet Σ^k that forbids all patterns in $\mathcal{L}(X')^c \cap \left(\Sigma^k\right)^{\star d}$.

▶ **Example 6.** Let $X' = \langle a, b \mid (\,a\ a\,), (\,b\ b\,), \left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right), \left(\begin{smallmatrix} b \\ a \end{smallmatrix}\right)\rangle^2$, it is easy to see that X' contains only two configurations: a and b alternate on every row, and columns are uniform. That is every configuration locally look like figure 1.

…	…	…	…	…	…	…	…
…	a	b	a	b	a	b	…
…	a	b	a	b	a	b	…
…	a	b	a	b	a	b	…
…	a	b	a	b	a	b	…
…	a	b	a	b	a	b	…
…	…	…	…	…	…	…	…

Figure 1 Configurations of X'.

The $(\{a, b\}, 1)$ restriction of X' is therefore the one-dimensional subshift that contains all two configurations, that alternate a and b. The $(\{a\}, 1)$ restriction of X' is the empty subshift, and the $(\{a, b\}^2, 1)$ restriction contains exactly the two configurations that alternate $\left(\begin{smallmatrix} a \\ a \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} b \\ b \end{smallmatrix}\right)$.

In terms of computability, the restriction is significant : If $X \prec X'$ then X can be more complicated than X':

▶ **Proposition 7.** *If $X \prec X'$ then $\mathcal{L}(X)$ is corecursively enumerable in $\mathcal{L}(X')$.*

Indeed $P \in \mathcal{L}(X)$ iff for all n there exists a d' dimensional pattern of size n in $\mathcal{L}(X')$ with P at its center. (More precisely, $\mathcal{L}(X)^c$ is enumeration-reducible to $\mathcal{L}(X')^c$, see below for the definition.)

▶ **Proposition 8.** *There exist two subshifts $X \prec X'$ s.t. $\mathcal{L}(X')$ is computable and $\mathcal{L}(X)$ is not computable.*

Proof. Let X be any one-dimensional effectively closed subshift over the alphabet $\{a, b\}$ with a noncomputable language. It is well known that X can be given by a *computable* family of forbidden patterns $\mathcal{F}$ (see e.g. [3]).

Now let X' be the subshift over the alphabet $\{a, b, \#\}$ given by the same family of forbidden patterns. It is clear that $\mathcal{L}(X')$ is computable. Indeed, let $w \in \{a, b, \#\}^\star$ and write $w = \#u_1\#u_2 \dots \#u_k\#$ with $u_i \in \{a, b\}^\star$, with the # symbols at the ends possibly missing. Then $w \in \mathcal{L}(X')$ iff each u_i does not contain any element of $\mathcal{F}$. For the nontrivial direction, observe that in this case the biinfinite word ${}^\omega\#w\#^\omega$ does not contain any forbidden word of $\mathcal{F}$. As $\mathcal{F}$ is computable, we can test whether each u_i contains any element of $\mathcal{F}$, and therefore $\mathcal{L}(X')$ is computable.

On the other hand, the restriction of X' to the alphabet $\{a, b\}$ is our initial subshift X, which has an uncomputable language. ◀

This is in contrast with combinatorial group theory, where a (f.g.) subgroup of a group with a computable word problem has immediately a computable word problem. This is due to the fact that looking at subshifts makes us look at infinite objects given by finite words. To obtain theorems similar to Higman's, we will have to force an additional restriction:

▶ **Definition 9.** *Let X' be a subshift over an alphabet Σ' of dimension d' and X be a subshift over an alphabet $\Sigma \subseteq \Sigma'$ of dimension $d < d'$. We say that X is a* full restriction *of X', in symbols $X \sqsubseteq X'$ if $\mathcal{L}(X) = \mathcal{L}(X') \cap \Sigma^{\star d}$*

In other words, if X the (Σ, d) restriction of X', then every d-dimensional infinite word over Σ that can be found in X' is in X. Here we also ask that every *finite* word over Σ that can be found in X' is already in X. In this case:

▶ **Proposition 10.** *If $X \prec X'$ then $\mathcal{L}(X)$ is many-one reducible to $\mathcal{L}(X')$. In particular if $\mathcal{L}(X')$ is computable, then $\mathcal{L}(X)$ is computable.*

Proof. Obvious by definition: $\mathcal{L}(X) = \mathcal{L}(X') \cap \Sigma^{\star d}$ ◀

In this paper we will not be using restrictions of width more than 1.

1.3.2 Adding symbols and dimensions

The operation of adding a dimension is quite obvious.

▶ **Definition 11.** *Let X be a subshift of dimension d over the alphabet Σ. The* extension X' of X to dimension d' *is the subshift of dimension d' that avoids all patterns of $\mathcal{L}(X)^c$.*

A point of X' therefore looks like elements of X stacked in the additional dimensions[3]. Notice that by definition $X \sqsubseteq X'$.

Adding symbols is also easy to define:

▶ **Definition 12.** *Let X be a subshift of dimension d over the alphabet Σ. The* extension X' of X to alphabet $\Gamma \supset \Sigma$ *is the subshift over the alphabet Γ that avoid all patterns of $\mathcal{L}(X)^c$.*

[3] Note that this definition readily generalizes with X over an alphabet Σ^k and X' with alphabet Σ: every row of with k must avoid all patterns of $\mathcal{L}(X)^c$.

Notice that X' is defined using all patterns of $\mathcal{L}(X)^c$, not only a defining set of forbidden patterns. Notice also that $X \sqsubseteq X'$.

To understand what the points of X' look like, we will first look at an example where $\Gamma = \Sigma \cup \{\#\}$ and X is one-dimensional. In this case, a typical element of X' is of the form $\ldots\#u_{-1}\#u_0\#u_1\#u_2\ldots$ where each u_i is a finite word in $\mathcal{L}(X)$. Notice in particular that there is no relation between the word u_i and the word u_{i+1}.

If we look at a similar construction in dimension 2, we would see patterns of $\mathcal{L}(X)$ that are separated by # symbols, see Figure 2. The # symbol in the example is what is typically

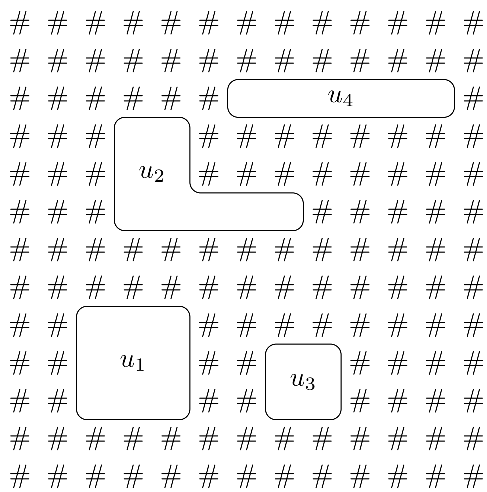

Figure 2 A configuration of X', the extension of $X \subseteq \Sigma^{\mathbb{Z}^2}$ to alphabet $\Gamma = \Sigma \cup \{\#\}$: any (connected) pattern of X can appear anywhere, as long as there are some # separating it from other patterns of X. The (unconnected) pattern consisting of u_1, u_2, u_3 and u_4 may not appear in a valid configuration of X.

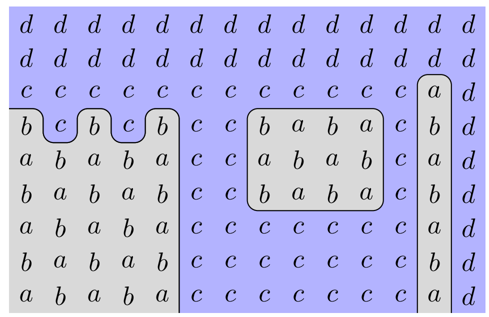

Figure 3 A portion of a valid configuration of the free product of $X = \langle a, b \mid (a\ a), (b\ b), \binom{a}{a}, \binom{b}{b}\rangle^2$ and $Y = \langle c, d \mid (c\ d), (d, c), \binom{c}{d}\rangle^2$, the 4-connected components of X and Y are gray and blue respectively.

called a *safe symbol* [7, 20] in symbolic dynamics, and is one of the typical conditions needed to obtain good mixing properties on subshifts.

More generally, we could define in the same way the free product of two subshifts:

▶ **Definition 13.** *Let X and Y be two subshifts of the same dimension on disjoint alphabets A and B respectively. The* free product *$X * Y$ is the subshift Z on $A \cup B$ with forbidden patterns $\mathcal{L}(X)^c \cup \mathcal{L}(Y)^c$*

A typical example of a point in Z is depicted in Figure 3. The discrete plane is divided into 4-connected zones that each correspond to a valid pattern of X or a valid pattern of Y.

We note that, for this construction to work, we need the language of a subshift to be defined in terms of *connected* patterns. If we took as the language of a subshift to be all patterns, connected or not, then the extension of X to alphabet $\Sigma \cup \{\#\}$ would merely consist in points of X with some symbols changed to #, which is a different beast altogether.

2 The three embedding theorems

In this section we prove the equivalent versions of the Higman embedding and Higman relative embedding theorems. To make the article easier to read, we will take some liberties

Group G	Subshift X
Group with n generators	Subshift on n symbols
Free group with n generators	Full shift on n symbols
Word problem $\mathcal{WP}(G)$	co-language $\mathcal{L}(X)^c$
Finitely presented group	SFT
Recursively presented group	Effectively closed subshift
Simple group	Minimal subshift
G_1 is a quotient of G_2	$X_1 \subseteq X_2$
$G_1 \subseteq G_2$	$X_1 \sqsubseteq X_2$ (Definition 9)

Table 1 Dictionary between groups and subshifts.

when stating the theorems of Higman. To obtain more exact statements, "G is a subgroup of H" should be replaced by "G is isomorphic to a subgroup of H".

Table 1 gives the correspondence we will use between the vocabulary of groups and the vocabulary of subshifts. It is based on the previous discussion and on the article [18]. The correspondence is not exact, but serves as an intuition for the theorems.

2.1 The Higman embedding theorem

We start with the first Higman embedding theorem:

▶ **Theorem 14** (Higman embedding theorem [14]). *A f.g. group G is recursively presented iff there exists a finitely presented group H s.t. $G \subseteq H$.*

▶ **Theorem 15** (Higman embedding theorem for subshifts). *A d-dimensional subshift X over an alphabet Σ is effectively closed iff there exists a $d+1$ dimensional SFT X' over an alphabet $\Gamma \supseteq \Sigma$ s.t. $X \sqsubseteq X'$*

As stated in the introduction, this theorem corresponds very closely to a result on subactions of subshifts first discovered by Hochman [16] and then improved by Aubrun-Sablik-Durand-Romashchenko-Shen. We first restate the theorem in a suitable form:

▶ **Theorem 16** ([2, 9]). *A d-dimensional subshift X over an alphabet Σ is effectively closed iff there exists a $(d+1)$-dimensional SFT $X' \subseteq (\Sigma \times \Gamma)^{\mathbb{Z}^{d+1}}$ such that*

$$X = \left\{x \mid (x^{\uparrow}, y) \in X'\right\}$$

where $x^{\uparrow}$ is the configuration where for any $z \in \mathbb{Z}^d$ and $j \in \mathbb{Z}$, $x^{\uparrow}_{z,j} = x_z$.

Proof of Theorem 15. All these constructions have one or several computation layers that check a layer on which the effectively closed subshift is written. In our case instead of superimposing the computation layer and the verified layer, we interleave them : if $c = (x, y) \in X \times Y$ in the original construction, the new configuration c' would be formed by $c'_{(i,2j)} = x_{(i,j)}$ and $c'_{(i,2j+1)} = y_{(i,j)}$. This remains an SFT.

We may further assume that the alphabet for the computation is disjoint from the alphabet of the checked subshift. Thus, by restricting the language to the words belonging to the alphabet of the checked layer, only this layer remains. ◀

Note that Higman's original theorem is valid non only for finitely generated groups but for general groups. To obtain a similar statement for subshifts, one would need to deal with subshifts over an infinite alphabet. We think that Hochman's original result [16] on effective dynamical systems provides such a generalization.

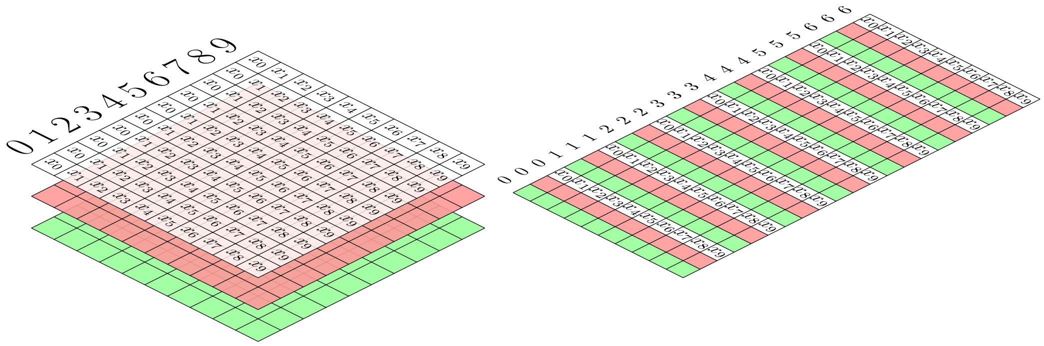

Figure 4 In [16, 2, 9] some layer contains a vertically repeated sequence $(x_i)_{i\in\mathbb{Z}}$ that is checked by some other layers which are superimposed as on the left. It is quite straigtforward to tranform a construction which has layers to an interleaving of the layers as seen on the right.

2.2 Higman's relative embedding theorem

The relative Higman theorem is, as its name indicates, the relativized version of the Higman embedding theorem, and states conditions on when a group G can be obtained as a subgroup of an extension of a group H. We first need a definition:

▶ **Definition 17** ([15]). *A group K is* finitely presented over *G if K can be obtained from G by adding finitely many generators and finitely many relations*

See [15, Definition 6.1] for the exact definition. The Higman relative embedding theorem then characterizes when a group G can be obtained as a subgroup of a group finitely presented in H. The classical relative embedding theorem correspond to the case where H is trivial. It turns out that the necessary computability criterion has to do with enumeration-reducibility, that we now define:

▶ **Definition 18** ([13]). *If L and M are two sets we say that L is* enumeration-reducible *to M, in symbols $L \leq_e M$ if there exists a partial computable function $f : \mathbb{N}\times\mathbb{N} \to P_f(\mathbb{N})$ where $P_f(\mathbb{N})$ is (a computable representation of) the set of all finite subsets of $\mathbb{N}$ s.t.*

$$x \in L \iff \exists n, f(n, x) \subseteq M$$

The definition might seem quite obtuse at first. Intuitively, $L \leq_e M$ if there is a computable procedure that can enumerate L from any enumeration of M.

The relative embedding theorem is then as follows:

▶ **Theorem 19** (The relative Higman embedding theorem [15]). *K is a subgroup of a group that is finitely presented over G iff $\mathcal{WP}(K) \leq_e \mathcal{WP}(G)$.*

We will now prove our version of the theorem. We first need an analog of "finitely presented over" in terms of subshift:

▶ **Definition 20.** *Let Y be a subshift over an alphabet Σ. U is* of finite type over *Y if U is obtained from Y by adding finitely many new symbols, dimensions, and finitely many new forbidden patterns.*

That is, $U = Y_1 \cap Y_2$, where Y_1 is an extension to a larger alphabet and higher dimension (in the sense of Definitions 11 and 12) of Y, and Y_2 is a subshift of finite type. To be consistent with the exact definition for groups, we also need that $Y \sqsubseteq U$, that is for none of the new forbidden patterns to contain only symbols of Σ.

This definition is straightforwardly extendable to effective subshifts:

▶ **Definition 21.** *Let Y be a subshift over an alphabet Σ. U is* effectively closed over Y *if U is obtained from Y by adding finitely many new symbols, dimensions and a recursively enumerable set of new forbidden patterns. As before, it is required that $Y \sqsubseteq U$.*

A straightforward corollary of Theorem 15 is the following:

▶ **Corollary 22.** *If Y is effectively closed over X, then there exists a subshift Z of finite type over Y such that $X \sqsubseteq Y \sqsubseteq Z$.*

We can now formulate our theorem.

▶ **Theorem 23** (The relative Higman embedding theorem for subshifts)**.** *Let X be a subshift over an alphabet A and Y be a subshift over an alphabet B disjoint from A.*

Then $\mathcal{L}(X)^c \leq_e \mathcal{L}(Y)^c$ iff there exists a subshift U of finite type over Y such that $X \sqsubseteq U$.

Let $Y = \emptyset$ be the empty subshift over the alphabet $\{0\}$. Then a subshift of finite type over Y is exactly the same as a subshift of finite type. Furthermore $\mathcal{L}(X)^c \leq_e \mathcal{L}(Y)^c$ means that $\mathcal{L}(X)^c$ is enumeration reducible over the full set, which is equivalent to saying that $\mathcal{L}(X)^c$ is recursively enumerable. In the case $Y = \emptyset$ this theorem is therefore equivalent to Theorem 15. Before going into the proof, we will give a few remarks.

First we want to state that this result is very similar, but incompatible with a result of Aubrun and Sablik[1]. The result of Aubrun and Sablik states that X can be obtained from Y using some operations (very similar to ours) iff $\mathcal{L}(X)^c \leq_s \mathcal{L}(Y)^c$ where $\leq_s$ is strong enumeration reducibility [13]. It turns out that there are many mistakes in the proofs so that the result as stated in their paper is actually provably wrong (the authors have been contacted and a corrigendum is being worked on). Problems arise in both directions in the proof. First, if X can be obtained from Y, then it is not true that $\mathcal{L}(X)^c \leq_s \mathcal{L}(Y)^c$. The authors use in their proof a lot of dovetailing arguments, but dovetailing arguments cannot be used for their reduction $\leq_s$. As an example, $A \leq_s B$ does not imply $A \times A \leq_s B$ or $A^\star \leq_s B$ [23]. In fact, the smallest reducibility relation that contains $\leq_s$ and that satisfy these statements *is* the reduction $\leq_e$ we used [23]. There are also some mistakes in the reverse direction that have been patched in Aubrun's PhD thesis, but only for the case of mixing subshifts. In fact, the set of operations the authors were taking is not sufficient to do the operations for general subshifts.

Proof. For simplicity, we focus on the case where the two subshifts are one-dimensional. Let $X \subseteq A^{\mathbb{Z}}$ and $Y \subseteq B^{\mathbb{Z}}$ be subshifts.

$\Leftarrow$: It is clear that if there exists U of finite type over Y with alphabet $C \subseteq A \cup B$ such that $\mathcal{L}(X) = \mathcal{L}(U) \cap A^\star$ then $\mathcal{L}(X)^c \leq_e \mathcal{L}(Y)^c$: it is clear that $\mathcal{L}(X)^c \leq_e \mathcal{L}(U)^c$, so we only need to prove that $\mathcal{L}(U)^c \leq_e \mathcal{L}(Y)^c$. Take an enumeration of $\mathcal{L}(Y)^c$ since U is of finite type over Y, any pattern not in $\mathcal{L}(Y)$ is not in $\mathcal{L}(U)$ and furthermore, to determine that a pattern p is in $\mathcal{L}(U)^c$, by compactness, one only needs to find some size at which it is impossible to form a valid pattern with p in its center. The procedure is the following: for every k, enumerate the k first patterns of $\mathcal{L}(Y)^c$ and check for all radiuses smaller than k whether each extension of p to this radius contains either some forbidden pattern enumerated this far or one of the patterns defining U from Y (which are in finite number). If there exists such a radius, it will be found at some step, p is then added to the enumeration. Thus $\mathcal{L}(X)^c \leq_e \mathcal{L}(Y)^c$.

$\Rightarrow$: Assume $\mathcal{L}(X)^c \leq_e \mathcal{L}(Y)^c$, we will construct a 2D subshift U effectively closed over Y such that $\mathcal{L}(X) = \mathcal{L}(U) \cap A^\star$, the result then follows by applying Corollary 22. In order to achieve this, from Y we will construct two intermediary subshifts:

- First we will construct Y_L: a $2D$ subshift in which the language of Y will be arranged in a dyadic like fashion. This subshift will be effective in Y. This subshift will serve as an "oracle" allowing to know whether a pattern is or is not in Y in a bounded manner: one configuration at least will contain all the patterns appearing in Y in bounded windows with computable sizes.
- From Y_L we can then construct U, in which one row out of two will be identical and belong to X and one row out of two will be in Y_L. This subshift is obtained by adding a recursively enumerable set of forbidden patterns and is thus effective over Y. By restricting the alphabet of this subshift we obtain X.

Let us now describe more precisely the different intermediate subshifts and how they are constructed, starting with Y_L:

- While Y is one dimensional, Y_L consists of two dimensions, its alphabet is $\Sigma_{Y_L} = B \cup \{\#\}$, with $\#$ a special symbol not belonging to B.

  Y_L will consist of rows, each of which will have a type: an integer $n \in \mathbb{N} \cup \{\infty\}$. A row of type $i \neq \infty$ has to be periodic. One row out of two will be of type 1, one row out of two in the remaining ones will be of type 2 and so on.

  Let us define inductively a row of type i: a row of type i consists of a sequence of $|B|^{2^i-1}$ words of length $2^i - 1$ each, separated by $\#$ and repeated periodically. All rows of some type in a configuration must be identical and a word appearing in a row of type i must be a subword of some word of a row of type $(i+1)$, see Figure 5. Thus, a word of some type i must appear as a subword of some word in a line of type k for any $k > i$ which does not contain a forbidden pattern of Y and thus appears in some configuration of Y.

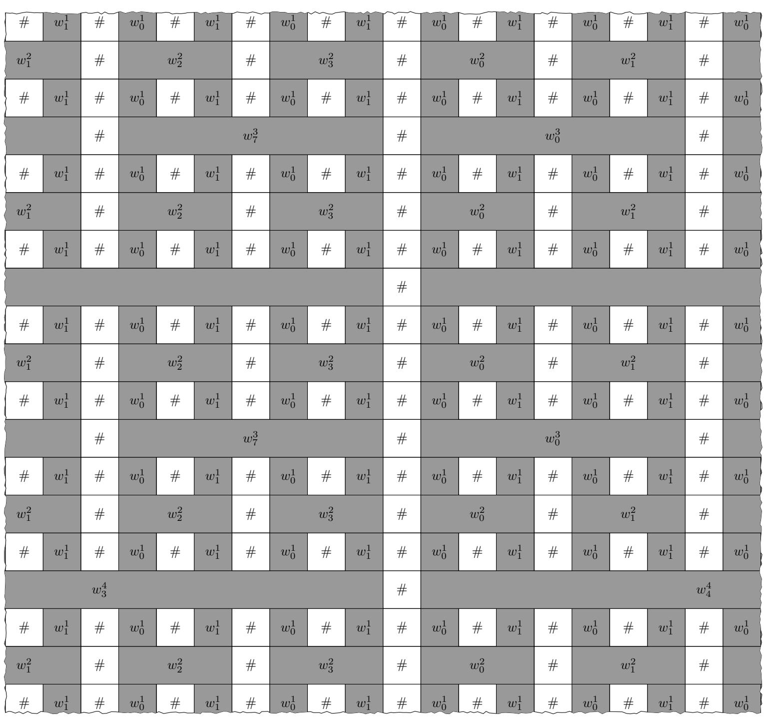

Figure 5 A typical point of Y_L : each line of type i is periodic of period $|B|^{2^{i-1}} \cdot 2^i$ and each word w^i_k in included in some word $w^j_{k'}$ for all $j > i$.

Thus Y_L is a 2D arrangement of words of $\mathcal{L}(Y)$ in a uniformly recurrent way, and there exists at least one configuration containing all of $\mathcal{L}(Y)$. Furthermore, Y_L is effective over Y.

- We describe how to construct U from Y_L: we know that $\mathcal{L}(X)^c \leq_e \mathcal{L}(Y)^c$. Thus, there exists a computable $f : \mathbb{N} \times \mathbb{N} \to \mathfrak{P}_{finite}(\mathbb{N})$ such that:

$$x \in \mathcal{L}(X)^c \text{ iff } \exists n \in \mathbb{N}, f(x,n) \subseteq \mathcal{L}(Y)^c$$

In other words, some word w is in $\mathcal{L}(X)$ iff for any $n \in \mathbb{N}$, there is some word of $f(x,n)$ in $\mathcal{L}(Y)$. That is to say, supposing $\mathcal{L}(Y)$ is given as an oracle, we have an enumerable way to check that a word w belongs to $\mathcal{L}(X)$: enumerate the $n \in \mathbb{N}$ and compute $f(w,n)$ and check that at least one element belongs to $\mathcal{L}(Y)$, if not halt. The computations that do not halt are the ones where w belongs to $\mathcal{L}(X)$.

Given Y_L, this can be implemented in an effective way: take $x \in A^{\mathbb{Z}}$ and $y \in Y_L$. We interleave x in y by using the same technique as in figure 4: we insert a copy of x between each pair of lines of y.

We now need to ensure that all words on the lines with alphabet A belong to $\mathcal{L}(X)$. This may also be done by adding a recursively enumerable set of forbidden patterns: in order to check that some subword w of x is in $\mathcal{L}(X)$, one needs to check that for each n, $f(w,n)$ appears in some line of type $i > |w|$: for every pattern w we forbid all patterns that contain w but no pattern of $f(w,n)$, since rows of type i appear every 2^{i+2} rows, for each w this constitutes a finite number of forbidden patterns for each n. Thus we may recursively enumerate the forbidden patterns for each $w \in A^\star$.

◀

2.3 The Boone-Higman-Thompson theorem

The Boone-Higman-Thompson theorem is a theorem that characterizes groups with a *computable word problem.* It turns out that the characterization is obtained with the notions of a simple group:

▶ **Theorem 24** (The Boone-Higman-Thompson theorem [6, 26])**.** *A group G has a computable word problem iff it is a subgroup of a simple recursively presented group.*

Recall that a simple group is a group with no proper (nontrivial) quotient. By Dictionary 1, the equivalent should be a subshift with no proper (nontrivial) subshift, i.e. what is called in the literature a minimal subshift. This seems to be indeed, the good analogy, as argued for in [18], and we will prove:

▶ **Theorem 25.** *Let X be a 1 dimensional subshift over an alphabet Σ. Then X has a computable language iff there exists a two dimensional minimal effective subshift Y over an alphabet $\Gamma \supset \Sigma$ such that $X \sqsubseteq Y$.*

Recently, Durand and Romashchenko [11] have proved that given a d-dimensional minimal effectively closed subshift, it can be realized as a subaction of a $(d+1)$-dimensional minimal SFT:

▶ **Theorem 26** ([11])**.** *Let X be a minimal effectively closed subshift. There exists a minimal SFT Y such that X is a subaction of Y: X is the projection by a letter to letter map of the lines of Y.*

This together with Theorem 25 gives us the subshift counterpart to the Boone-Higman-Thompson theorem:

▶ **Corollary 27** (The Boone-Higman-Thompson theorem for subshifts). *Let X be a 1 dimensional subshift over an alphabet Σ. Then X has a computable language iff there exists a three dimensional minimal subshift of finite type Y over an alphabet $\Gamma \supset \Sigma$ s.t. $X \sqsubseteq Y$.*

Both Theorem 25 and Corollary 27 translate to higher dimensions, the details of the proofs are left to the reader.

Before proving the Theorem 25, one needs a good intuition on what a minimal subshift looks like. Minimal subshifts are defined as subshifts that do not contain any nontrivial subshifts, but an equivalent, more palatable definition, is that minimal subshifts are uniformly recurrent subshifts, that is subshifts X where, for every pattern $u \in \mathcal{L}(X)$, there exists a size n s.t. the pattern u occurs in every pattern of X of size n. In particular, all configurations x of X have the same patterns, and every pattern that appear should appears everywhere, i.e. in any sufficiently large part of x.

We now proceed to the proof of the theorem. One direction is well known: A minimal effectively closed subshift has a computable language, see [4] for example. Therefore $\mathcal{L}(Y)$ is computable and therefore $\mathcal{L}(Y) \cap \Sigma^\star$ is computable.

The other direction essentially amounts to the following: Given a set of patterns L on an alphabet A, find a *minimal* subshift X that contains all patterns of L (and other patterns). Before reading the proof, the reader should try by itself as an exercise to find a two-dimensional minimal subshift X over an alphabet $\{a, b, c\}$ that contains all one-dimensional words over the alphabet $\{a, b\}$.

Our proof is quite similar to a construction by Elek and Monod [12] of a subshift with a non-amenable topological full group. Our construction is done however with more care to ensure that everything we are doing remains computable and that our subshift is already minimal, but the idea is essentially the same.

Let us now start with a 1 dimensional subshift X over an alphabet Σ with a computable language.

We define recursively a set $(w^i)_{i\in\mathbb{N}}$ of biinfinite rows. Each row will be periodic. We will denote by p_i the period of the row and by v_i the word that repeats, so that v_i is of length p_i and for all $k \in \mathbb{Z}$, $(w^i)_k = v_{k \mod p_i}$.

The row w^0 is the row of period $p_1 = 1$ corresponding to the word $v_0 = \#$. Suppose the row w^n is given, of period p_n.

Let $\{u_1, u_2 \dots u_{k_{n+1}}\}$ be the (computable) list of all words of length $2p_n - 1$ that appear in X. We define v_{n+1} to be the word consisting of all possible pairs of words of size $2p_n - 1$, separated by the # symbol

$$\#u_1\#u_1\#u_1\#u_2\#u_1\#u_3 \dots u_{k_{n+1}}\#u_{k_{n+1}-1}\#u_{k_{n+1}}\#u_{k_{n+1}}$$

and w_{n+1} is the biinfinite word where v_{n+1} repeats periodically. Notice that v_{n+1} is of size $p_{n+1} = 2k_{n+1}^2 p_n$ so that p_{n+1} is strictly greater than p_n and p_n divides p_{n+1} .

We repeat some properties of our set of rows:

- The row w^n is periodic of period p_n. Furthermore the symbol # appears in w^n only in positions multiple of $2p_{n-1}$.
- p_i divides p_j if $i < j$.
- $p_n > n$.

▶ **Lemma 28.** *Let u be a word of length k that appears in w^n for $n \geq k$ in position i. Then u appears in position $i + tp_{k-1}$ in w^k for some integer t.*

Proof. The result is clear for $n = k$. Now suppose that $n > k$. There are two cases for u: either $u = s_1\#s_2$ for two words $s_1, s_2 \in \Sigma^\star$ or $u = s$ for some word $s \in \Sigma^\star$.

We start with the first case, $u = s_1\#s_2$. The words s_1 and s_2 are words of size $< k$ that are factors of some word of size $2p_{(n-1)} - 1$ that appears in X. Therefore there are also respectively suffix and prefix of some words t_1, t_2 that appear in X, each of size $2p_{(k-1)} - 1$. By definition $t_1\#t_2$ appears in w^k therefore u appears in w^k. As every symbol # inside w^k appears at positions that are multiples of $2p_{k-1}$ and that it is also the case inside w^n (as p_k divides p_{n-1}), the position where u appears in w^k must be of the form $i + tp_{k-1}$ for some t.

Now the second case. Suppose that $u = s$ for some word s that appears in X of size k. u appears from position i to position $i + k - 1$ in w^n. Let $0 \le j < p_{k-1}$ so that $j = i - 1 \mod p_{k-1}$. u is a word of size k that appears in X and therefore can be completed as a word v of size $2p_{k-1} - 1$ that appears in X by adding j letters at the beginning and $2p_{k-1} - 1 - (j + k)$ letters at the end. This word v appears at position $tp_{k-1} + 1$ in w^k for some t and therefore u appears in position $tp_{k-1} + 1 + j = t'p_{k-1} + i$ in w^k.

◀

▶ **Definition 29.** *If i is an integer, the* level *of i, denoted by $lvl(i)$ is the greatest power of 2 that divides i, i.e. $i = k \times 2^{lvl(i)}$ with k odd. The level of 0 is $+\infty$ by convention.*

The two following lemmas are clear.

▶ **Lemma 30.** *Let $n > k$, then $lvl(i + 2^n) = lvl(i)$.*

▶ **Lemma 31.** *Let $i \neq j$ s.t. $lvl(i) \ge k$ and $lvl(j) \ge k$. Then $|i - j| \ge 2^k > k$.*

We now define a configuration y in the following way: the i-th row of y is the row w^j where j is the level of i.

For $i = 0$, we take any word w that is a limit point of $\{w^j, j \in \mathbb{N}\}$.

Notice that y is likely not computable, as the row 0 might be arbitrarily complex. However all other rows are computable.

To simplify notations, we will denote the rows of y in exponent, so that the symbol in the i-th row and j-th column of y is y^i_j and the i^{th} row of y is y^i. By definition, we therefore have $y^i_j = w^{lvl(i)}_j$ for $i \neq 0$.

▶ **Lemma 32.** *Let u be a pattern defined over $[1, k] \times [1, k]$ that appears in y.*

Then u also appears inside y at position $(i + 1, j + 1)$ with $i \in [0, 2^k - 1]$ and therefore u appears inside the first $2^{k+1} - 1$ rows of y (in the rows labeled 1 to $2^{k+1} - 1$).

Proof. Let u be a pattern defined on the square $[1, k] \times [1, k]$.

Suppose that u appears inside y at position $(i + 1, j + 1)$. That is: for all $(l, m) \in [1, k]^2, u^m_l = y^{i+m}_{j+l}$.

There are two cases. First, suppose that all of the integers $i + 1, i + 2, \dots i + k$ are of level strictly less than k. Then for all $l \in [i + 1, i + k]$ and all integers t, $y^{l+2^k t} = y^l$. We can therefore suppose wlog that $i \in [0, 2^{k-1}]$ and the result is proven.

Otherwise some of the integers $i + 1, \dots i + k$ is of level at least k. By Lemma 31, this happens only for one of the integers, say the integer $i + r = z \times 2^n$ for $n \ge k$.

The word u^r appears by definition in y^{i+r} which is of level at least k. By Lemma 28, it also appears in w^k at position $j + 1 + tp_{k-1}$ and therefore in y^{2^k} at position $j + 1 + tp_{k-1}$. (Lemma 28 also applies if $i + r = 0$, as the row 0 of u is the limit of rows of arbitrary large level). In other words for all $l \in [1, k], u^r_l = y^{2^k}_{j+tp_{k-1}+l}$.

We now claim that the word u appears in position $[2^k - r + 1, j + 1 + tp_{k-1}]$ inside y. That is, for all $l, m \in [1, k]^2, u^m_l = y^{2^k-r+m}_{j+tp_{k-1}+l}$. The result is clear for $m = r$. Now let $m \in [1, k], m \neq r$.

As $i+m$ is of level strictly less than k, $y^{i+m+t2^k} = y^{i+m}$ for all t by Lemma 30. In particular as $i+r$ is dividible by 2^k, we get that $y^{i+m} = y^{i+m+2^k-(i+r)} = y^{2^k-r+m}$. Furthermore the row y^{i+m} is periodic of period p_s for some $s < k$ and in particular it is periodic of period p_{k-1}. Therefore

$$u_l^m = y_{j+l}^{i+m} = y_{k+l+tp_{k-1}}^{i+m} = y_{k+l+tp_{k-1}}^{2^k-r+m}$$

◀

▶ **Corollary 33.** *Let u be a pattern of size $k \times k$ inside y. Then u appears in any window of size $2^{k+2} \times 2p_k$.*

Proof. Indeed, by the previous lemma, u appears inside y in position (i, j) for some $i \in [1, 2^k]$ However the rows from 1 to $2^{k+1} - 1$ are all periodic of period p_k, and repeat vertically with period 2^{k+1} by Lemma 30. Therefore the pattern u itself repeats horizontally with period p_k and repeats vertically with period 2^{k+1} and consequently appears in any window of size $2^{k+2} \times 2p_k$. ◀

▶ **Corollary 34.** *Let Y be the subshift that forbids all patterns of size $k \times k$ that do not appear in the square $[1, 2^{k+2}] \times [1, 2p_k]$ of y.*

Then Y is minimal, effectively closed, and $\mathcal{L}(X) = \mathcal{L}(Y) \cap \Sigma^\star$.

Proof. The first conclusions are immediate from the previous corollary. The second one comes from the fact that y (apart from the row 0) is computable. The third one is by definition of y. ◀

3 Discussion

In this article, we have introduced the analogues of the three Higman theorems, originally for groups, in terms of subshifts. This reinforces the convictions of the authors that symbolic dynamics has a deep connection with objects from combinatorial algebra. To obtain these theorems, we had to introduce the following concepts:

- An equivalent of the notion of free product for groups (Definitions 12 and 13).
- An equivalent of the notion of subgroup containement (Definition 9).

Compared to existing constructions, these two new ideas are rather *combinatorial* rather than *dynamic*. In particular, they cannot be defined easily in terms of the infinite words in the subshifts; they are defined in terms of the finite words that constitute the language of the subshift. This could be seen as a drawback of the construction, so we will give some arguments explaining why more "dynamical" constructions cannot work.

The concept of the free product of subshifts is used for the relativized Higman theorem. This operation does not appear in the work of Aubrun and Sablik [1] (which was flawed as we saw) and is probably mandatory; Suppose that a minimal subshift X is defined from a subshift Y by various dynamical constructions (cartesian product, factor, subactions, etc) as in [1]. Let Y' be the smallest subshift of Y that contains all uniformly recurrent points of Y. Then it is easy to see that Y' also defines X using the same constructions. However Y' may have very different computability properties than Y. In particular it is possible to have $\mathcal{L}(X)^c \leq_e \mathcal{L}(Y)^c$ but $\mathcal{L}(X)^c \not\leq_e \mathcal{L}(Y')^c$.

In fact, when looking at our whole construction (and the construction of [1]), we see that it is important to start with a subshift X with the property that, for every finite collection of words $u \in \mathcal{L}(X)$, there exists a *uniformly recurrent* point of X that contains all of them. So we either have to assume that our subshift has this property (this is what Aubrun did in her PhD thesis, by assuming some mixing properties), or use a free product: The free product of X with any (nonempty) subshift always has this property.

The full restriction operator $\sqsubseteq$, the analogue of subgroup containment, is mostly used in the equivalent of the Boone-Higman-Thompson theorem. In fact, it can be replaced in the other theorems by more traditional dynamical operators, like factor maps (the original Hochman theorem was indeed stated in terms of factor maps). It is not clear if we could obtain a analogue of Boone-Higman-Thompson theorem in terms of factor map. It is certainly true that factors of minimal subshifts of finite type have a computable language; However, they also have the additional property that the set of their uniformly recurrent points is dense, and therefore not all subshifts with computable language can be obtained this way.

References

1. Nathalie Aubrun and Mathieu Sablik. An order on sets of tilings corresponding to an order on languages. In *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, pages 99–110, 2009.
2. Nathalie Aubrun and Mathieu Sablik. Simulation of effective subshifts by two-dimensional subshifts of finite type. *Acta Applicandae Mathematicae*, 126(1):35–63, 2013. doi:10.1007/s10440-013-9808-5.
3. Alexis Ballier. *Propriétés structurelles, combinatoires et logiques des pavages*. PhD thesis, Aix-Marseille Université, 2009.
4. Alexis Ballier and Emmanuel Jeandel. Computing (or not) quasi-periodicity functions of tilings. In Jarkko Kari, editor, *Second Symposium on Cellular Automata "Journées Automates Cellulaires", JAC 2010, Turku, Finland, December 15-17, 2010. Proceedings*, pages 54–64. Turku Center for Computer Science, 2010.
5. Robert Berger. *The Undecidability of the Domino Problem*. Number 66 in Memoirs of the American Mathematical Society. The American Mathematical Society, 1966.
6. William W. Boone and Graham Higman. An algebraic characterization of groups with soluble word problem. *Journal of the Australian Mathematical Society*, 18(1):41–53, August 1974. doi:10.1017/S1446788700019108.
7. Mike Boyle, Ronnie Pavlov, and Michael Schraudner. Multidimensional sofic shifts without separation and their factors. *Transactions of the AMS*, 362(9):4617–4653, September 2010. doi:10.1090/S0002-9947-10-05003-8.
8. W. Craig and R. L. Vaught. Finite Axiomatizability Using Additional Predicates. *The Journal of Symbolic Logic*, 23(3):289–308, September 1958.
9. Bruno Durand, Andrei Romashchenko, and Alexander Shen. Effective Closed Subshifts in 1D Can Be Implemented in 2D. In *Fields of Logic and Computation*, number 6300 in Lecture Notes in Computer Science, pages 208–226. Springer, 2010. doi:10.1007/978-3-642-15025-8_12.
10. Bruno Durand, Andrei Romashchenko, and Alexander Shen. Fixed-point tile sets and their applications. *Journal of Computer and System Sciences*, 78(3):731–764, May 2012. doi:10.1016/j.jcss.2011.11.001.
11. Bruno Durand and Andrei E. Romashchenko. On the expressive power of quasiperiodic SFT. In *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 1–14, 2017.
12. Gábor Elek and Nicolas Monod. On the topological full group of a minimal cantor $\mathbb{Z}^2$-system. *Proceedings of the American Mathematical Society*, 141(10):3549–3552, October 2013.
13. Richard M. Friedberg and Hartley Rogers. Reducibility and Completeness for Sets of Integers. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 5:117–125, 1959.
14. Graham Higman. Subgroups of Finitely Presented Groups. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 262(1311):455–475, August 1961.
15. Graham Higman and Elizabeth Scott. *Existentially Closed Groups*. Oxford University Press, 1988.

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

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

18 Emmanuel Jeandel. Enumeration reducibility in closure spaces with applications to logic and algebra. In *ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–11, 2017.

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

20 Aimee Johnson and Kathleen Madden. Factoring higher-dimensional shifts of finite type onto the full shift. *Ergodic Theory and Dynamical Systems*, 25:811–822, 2005.

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

22 R.C. Lyndon and P.E. Schupp. *Combinatorial Group Theory*. Classics in Mathematics. Springer Berlin Heidelberg, 2001.

23 Daniele Marsibilio and Andrea Sorbi. Bounded enumeration reducibility and its degree structure. *Archive for Mathematical Logic*, 51:163–186, 2012.

24 Mark Sapir. *Combinatorial Algebra: Syntax and Semantics*. Springer Monographs in Mathematics. Springer, 2014.

25 Stephen G. Simpson. Mass problems associated with effectively closed sets. *Tohoku Mathematical Journal*, 63(4):489–517, 2011.

26 Richard J. Thompson. Embeddings into finitely generated simple groups which preserve the word problem. In Sergei I. Adian, William W. Boone, and Graham Higman, editors, *Word Problems II*, volume 95 of *Studies in Logic and the Foundations of Mathematics*, pages 401–441. North Holland, 1980.