



# Countable Sofic Shifts with a Periodic Direction

Ilkka Törmä<sup>1</sup> 

Published online: 27 December 2019  
© The Author(s) 2019

## Abstract

As a variant of the equal entropy cover problem, we ask whether all multidimensional sofic shifts with countably many configurations have SFT covers with countably many configurations. We answer this question in the negative by presenting explicit counterexamples. We formulate necessary conditions for a vertically periodic shift space to have a countable SFT cover, and prove that they are sufficient in a natural (but quite restricted) subclass of shift spaces.

**Keywords** Multidimensional symbolic dynamics · Sofic shifts · Countable shift spaces · Periodicity

## 1 Introduction

In the context of one-dimensional symbolic dynamics, the structural properties and connections between shifts of finite type (SFTs for short) and sofic shifts are quite well understood. They can be represented as sets of infinite walks on finite directed graphs, with labeled edges or vertices in the case of sofic shifts. Furthermore, to each sofic shift one can associate several SFT covers that are in some sense minimal: Fischer covers, Krieger covers, past set covers et cetera. In particular, these covers have the same topological entropy as the corresponding sofic shifts, and are topologically transitive if and only if the sofic shifts are. Several subclasses, such as shifts of almost finite type, can be defined via relations between a sofic shift and its set of SFT covers. See [11] for an overview of this topic.

In multidimensional symbolic dynamics, this connection is much weaker. While each sofic shift has SFT covers whose entropy is arbitrarily close to its own [3], it is not known whether equal entropy covers always exist. The class of sofic shifts is

---

Research supported by Academy of Finland grant 295095.

✉ Ilkka Törmä  
iatorm@utu.fi

<sup>1</sup> Department of Mathematics and Statistics, University of Turku, Matematiikan ja tilastotieteen laitos, 20014 Turun yliopisto, Finland

likewise more nebulous. Several sufficient and necessary conditions for soficity of a multidimensional shift space are known, but between them remains a rather large gap. These conditions are both computational [1, 5–7] and geometric [10, 13, 14], and recently a new class of non-sofic shift spaces was announced that fools all previously known soficness tests [4].

In this article, we concentrate on the subclass of countable shift spaces, that is, shift spaces with countably many configurations. The research of this particular class was initiated in [2] in the context of tiling systems. In one dimension, countable sofic shifts are exactly those with zero topological entropy, but in the multidimensional case not all zero entropy sofic shifts are countable. Our main result settles an analogue of the equal entropy cover problem of multidimensional sofic shifts: whether every countable sofic shift has a countable SFT cover. The answer is negative, and our surprisingly succinct topological proof presents an explicit counterexample (Example 1). The counterexample is vertically constant, meaning that each of its configurations has  $(0, 1)$  as a vector of periodicity, and this periodicity is an essential part of the proof. It readily generalizes to an arbitrary number of dimensions. We also formulate three necessary conditions, one geometric and two computational, for possessing a countable SFT cover, and prove that they are sufficient for the class of shift spaces whose horizontal rows come from a one-dimensional countable sofic shift. It is left as an open problem whether the conditions are sufficient in general, and whether the assumption of periodicity can be removed.

## 2 Definitions

### 2.1 Symbolic Dynamics

Let  $\Sigma$  be a finite alphabet and  $d \geq 1$ . The set  $\Sigma^{\mathbb{Z}^d}$  equipped with the product topology is the  $d$ -dimensional full shift, and its elements are called *configurations*. We will mostly concentrate on the cases  $d = 1$  and  $d = 2$ . The group  $\mathbb{Z}^d$  acts on  $\Sigma^{\mathbb{Z}^d}$  by shifting:  $\sigma^{\vec{v}}(x)_{\vec{w}} = x_{\vec{v}+\vec{w}}$ .

A *shift space* is a topologically closed subset of  $\Sigma^{\mathbb{Z}^d}$  which is invariant under the shift action. Every shift space is defined by a set of *forbidden finite patterns* as the set of configurations where none of these patterns occur. If the set of forbidden patterns is finite, the corresponding shift space is a *shift of finite type*, or SFT for short. If the set of forbidden patterns (equivalently, all patterns not occurring in the shift) is computably enumerable, the shift space is *effectively closed*. In our constructions,  $\Sigma$  will usually be a set of decorated square tiles, and the forbidden patterns are those where the decorations of neighboring tiles do not match. This model is commonly known as *Wang tiles*. Sometimes we will impose additional local constraints.

A configuration  $x \in \Sigma^{\mathbb{Z}^d}$  is *periodic* in a direction  $\vec{v} \in \mathbb{Z}^d$ , or that  $\vec{v}$  is a *period* of  $x$ , if  $\sigma^{\vec{v}}(x) = x$ . We say  $x$  is *non-horizontally constant*, if the basis vectors  $\vec{e}_2, \dots, \vec{e}_d$  are periods of  $x$ , and *non-horizontally periodic*, if some of their multiples are periods of  $x$ . For dimensions greater than 2, we thus view the first coordinate as ‘horizontal’. For  $d = 2$  we use the terms *vertically constant* and *vertically periodic*.

A one-dimensional configuration  $y \in \Sigma^{\mathbb{Z}}$  is *ultimately periodic to the right*, if there exists  $p > 0$  such that  $x_n = x_{n+p}$  for all large enough  $n \in \mathbb{Z}$ . Ultimate periodicity to the left is defined analogously. To help with the construction of non-horizontally constant sofic shifts, we use the following powerful result.

**Theorem 1** [1, 6] *For  $d \geq 2$ , every effectively closed non-horizontally constant  $\mathbb{Z}^d$ -shift space is sofic.*

Let  $V \subset \mathbb{Z}^d$  be a subgroup of translations. The orbit of a configuration  $x \in \Sigma^{\mathbb{Z}^d}$  under  $V$  is denoted  $\mathcal{O}_V(x) = \{\sigma^{\vec{v}}(x) \mid \vec{v} \in V\}$ . We say  $x$  is *uniformly recurrent along  $V$* , if the orbit closure  $\overline{\mathcal{O}_V(x)}$  is a minimal system under the action of  $V$ . This is equivalent to the following condition: for each finite pattern  $p$  that occurs in  $x$  at some  $\vec{w} \in \mathbb{Z}^d$ , there exists  $K \in \mathbb{N}$  such that for each  $\vec{v} \in V$ , there exists  $\vec{u} \in V \cap (\vec{v} + [-K, K]^d)$  such that  $p$  occurs in  $x$  at  $\vec{w} + \vec{u}$ . In particular, if  $x$  is  $\vec{v}$ -periodic for all  $\vec{v} \in V$ , then it is uniformly recurrent along  $V$ .

For  $e \leq d$ , the  $e$ -dimensional *projective subdynamics* (see [8, 15]) of a shift space  $X \subset \Sigma^{\mathbb{Z}^d}$  is denoted  $P_e(X) \subset \Sigma^{\mathbb{Z}^e}$ , and it consists of the restriction to  $\mathbb{Z}^e$  of each configuration of  $X$ ; we choose  $\mathbb{Z}^e$  as the subgroup of  $\mathbb{Z}^d$  where the last  $d - e$  coordinates have value 0. We denote  $P_1(X) = P(X)$ . In the two-dimensional case,  $P(X)$  consists of the central horizontal row of each configuration of  $X$ . If  $Y \subset \Sigma^{\mathbb{Z}^e}$  is a shift space, then  $P_e^\dagger(Y) \subset \Sigma^{\mathbb{Z}^d}$  is the non-horizontally constant shift space with  $P_e(P_e^\dagger(Y)) = Y$ .

A *block code* is a continuous map  $\psi : X \rightarrow Y$  between two  $\mathbb{Z}^d$ -shift spaces that intertwines the shift actions. It is defined by a finite *neighborhood*  $N \subset \mathbb{Z}^d$  and a *local rule*  $\Psi : \Sigma^N \rightarrow \Sigma$ , in the sense that  $\psi(x)_{\vec{v}} = \Psi(\sigma^{\vec{v}}(x)|_N)$  holds for all  $x \in X$  and  $\vec{v} \in \mathbb{Z}^d$ . The image of an SFT under a block code is a *sofic shift*, and the SFT and block code form its *SFT cover*.

An *edge shift* is a one-dimensional SFT  $X_G$  defined by a finite directed graph  $G = (V, E)$ . The configurations of the edge shift are the edges of all bi-infinite paths in the graph, so it is a subset of  $E^{\mathbb{Z}}$ . An edge labeling  $\lambda : E \rightarrow \Gamma$  induces a block map from  $X_G$  to  $\Gamma^{\mathbb{Z}}$  by acting on each edge, and  $\lambda$  is *right-resolving* if for each vertex  $v \in V$ ,  $\lambda$  is injective on the outgoing edges of  $v$ . Each one-dimensional sofic shift  $Y$  has an SFT cover consisting of a right-resolving labeling of an edge shift. We make use of the following result about the structure of one-dimensional countable sofic shifts. It is proved by noticing that the right-resolving cover must have disjoint cycles, and then transforming its graph into the desired form by duplicating vertices and “unrolling” parts of cycles.

**Lemma 1** (Lemma 4.8 in [15]) *Let  $X \subset \Gamma^{\mathbb{Z}}$  be a one-dimensional countable sofic shift. Then there exists an edge shift  $Y$  given by a graph  $G = (V, E)$  and a right-resolving factor map  $\psi : Y \rightarrow X$  such that  $G$  is a disjoint union of components  $K_1, \dots, K_n$ , where each  $K_i$  consists of a finite number of vertex-disjoint cycles  $c_{i,0}, \dots, c_{i,m_i}$  with one marked vertex, and vertex-disjoint nonempty paths  $p_{i,1}, \dots, p_{i,m_i}$ , where each  $p_{i,j}$  connects the marked vertex of  $c_{i,j-1}$  to the marked vertex of  $c_{i,j}$ .*

## 2.2 Counter Machines

In this article, we use several variants of counter machines to implement computation. We only describe their properties informally. For  $k \in \mathbb{N}$ , a deterministic  $k$ -counter machine  $M = (k, Q, \delta, q_0, q_f)$  consists of a finite set  $Q$  of internal states and  $k$  counters, each of which holds a natural number. A configuration of the machine is thus an element of  $Q \times \mathbb{N}^k$ . Some of the counters may be *input counters*, which means that their values are fixed at the beginning of the computation and cannot be modified; the others are *internal counters*. The machine is initialized in a state  $q_0 \in Q$  with the internal counters set to 0, and on each step of its computation, it may compare any internal counter for equality to 0 or an input counter, and based on that information and its internal state, it may increment or decrement one or more internal counters and update its internal state. The element  $\delta$  of the definition encodes the transition function of  $M$  in some suitable way.

An *integer  $k$ -counter machine* is similar to a  $k$ -counter machine, except that the values of the counters can be negative. An integer counter machine can have an *oracle counter*, which works as follows. There is a separate oracle alphabet  $\Sigma$ , and an oracle configuration  $x \in \Sigma^{\mathbb{Z}}$ . During the computation, the machine can modify the oracle counter normally, and it can query the symbol  $x_n$ , where  $n \in \mathbb{Z}$  is the current value of the oracle counter.

When a counter machine of any kind enters the final state  $q_f$ , it halts. What this means in the context of a machine simulated by a tiling depends on the application: it may result in a tiling error or produce some special tile.

Both types of counter machines can be simulated by countable  $\mathbb{Z}^2$  SFTs. We adapt the methods of [17] for this. Consider first a  $k$ -counter machine  $M = (k, Q, \delta, q_0, q_f)$ . The SFT  $X_M$  that simulates  $M$  has as its alphabet  $S_M = \{\#_L, \#_R\} \cup \left( (\{L, C, R\} \cup H_M) \times \prod_{i=1}^k \{P_i, Z_i\} \right)$ , where  $H_M$  is an auxiliary *head alphabet*. The non- $\#$  components of  $S_M$  are called the *head track* and *counter tracks*. The tracks of each horizontal row of  $X_M$  must have the form  $L^m h C^n R^k$  for the head track or  $P_i^{m_i} Z_i^{n_i}$  for the counter tracks for  $m, n, k, m_i, n_i \in \mathbb{N} \cup \{\infty\}$  and  $h \in H_M$ , or a degenerate version of such configuration where some of the regions don't exist. The tracks can be followed by regions of by  $\#_L$  on the left and  $\#_R$  on the right. The number  $m_i$  represents the value of counter  $i$ . The symbol  $h$  is the *zig-zag head*, which contains the internal state of  $M$  and some extra information that guides the computation. The region containing the tracks is the *computation cone*, and if its length is nonzero, then it grows by one cell to the right on the next row. If the computation cone is empty (meaning that the computation has not started yet), then the next row can be equal to the current one, or it can grow by one cell to contain the zig-zag head in the initial state  $q_0$ . The initial values of read-only counters are unrestricted and are 'hidden' until the computation cone extends sufficiently far to reveal them, but the internal counters are forced to have value 0 at the beginning of the computation. The zig-zag head travels back and forth in the computation cone at a speed of two cells per row, updating its internal state at the left border where it can see which counters are positive, and updating the internal counter values as it passes over them. As the head passes over an internal counter, it also remembers which read-only counters hold the

same value. The counter machine can introduce tiling errors by entering an internal state that has no outgoing transition in  $\delta$ . It can be shown that the resulting SFT  $X_M$  is countable.

See Fig. 1 for a sample simulation of a machine with two counters. The computation starts with a read-only (lower) counter initialized at value 5, and the internal (upper) counter at 0. This means that the counter machine is processing an “input value” of 5. On the first two computation steps, the machine increments the internal counter. This auxiliary counter is used for computation and its exact role is not important for this visualization. Note that the machine may compare the internal counter to the input counter already in the first step, even though the value of the latter is hidden; since internal counters are never hidden, they are known to be unequal to any hidden counter.

The simulation of an integer  $k$ -counter machine is similar, but there are three major differences. First, the computation cone grows both to the left and to the right, at the rate of one step per row. Second, there is an extra counter that is always initialized at 0 and never changes its value. It denotes the value 0, and other counters can be compared to it locally. Third, there is a separate vertically constant layer over the oracle alphabet  $\Sigma$ , and when the zig-zag head travels over the oracle counter, it can read the

#L	#L	P	P	Z	Z	Z	Z	Z	Z	→Z	Z
#L	#L	P	P	Z	Z	Z	Z	→Z	Z	Z	Z
#L	#L	P	P	Z	Z	→Z	Z	Z	Z	Z	Z
#L	#L	P	P	→Z	Z	Z	Z	Z	Z	Z	Z
#L	#L	→P	P	Z	Z	Z	Z	Z	Z	Z	#R
#L	#L	←P	Z	Z	Z	Z	Z	Z	Z	#R	#R
#L	#L	P	Z	←Z	Z	Z	Z	Z	#R	#R	#R
#L	#L	P	Z	Z	Z	←Z	Z	#R	#R	#R	#R
#L	#L	P	Z	Z	Z	→Z	#R	#R	#R	#R	#R
#L	#L	P	Z	→Z	Z	#R	#R	#R	#R	#R	#R
#L	#L	→P	Z	Z	#R	#R	#R	#R	#R	#R	#R
#L	#L	←P	Z	#R	#R	#R	#R	#R	#R	#R	#R
#L	#L	→Z	#R	#R	#R	#R	#R	#R	#R	#R	#R
#L	#L	#R	#R	#R	#R	#R	#R	#R	#R	#R	#R
#L	#L	#R	#R	#R	#R	#R	#R	#R	#R	#R	#R

Fig. 1 A configuration of  $X_M$ . The position and direction of the zig-zag and two counter tracks are shown, other states are not. The initial values of the counters are 0 and 5

symbol under it on this new layer. If this layer is unrestricted, then the resulting SFT will be uncountable, so in our construction we impose additional constraints on it to ensure countability.

### 3 Countable Cover Conditions

In this section we explore some necessary conditions for possessing a countable SFT cover, and provide examples of shift spaces that do or do not satisfy them. The following lemma, whose proof is topological, is the main workhorse behind our results.

**Lemma 2** *Let  $X \subset \Sigma^{\mathbb{Z}^d}$  and  $Y \subset \Gamma^{\mathbb{Z}^d}$  be shift spaces, and let  $\phi : X \rightarrow Y$  be a factor map. Let  $y \in Y$  be uniformly recurrent along a subgroup  $V \subset \mathbb{Z}^d$ . Then there exists  $x \in \phi^{-1}(y)$  that is uniformly recurrent along  $V$ . If  $X$  is countable, then  $x$  is periodic in each direction  $\vec{v} \in V$ .*

*Proof* Denote  $Z = \phi^{-1}(\overline{\mathcal{O}_V(y)}) \subset X$ , which is a compact set. We have  $\sigma^{\vec{v}}(Z) = Z$  for each  $\vec{v} \in V$ , and hence  $(Z, V)$  is a topological dynamical system. Let  $U \subset Z$  be a minimal subsystem of  $Z$ , which exists by a simple application of Zorn’s lemma. Then any  $z \in U$  is uniformly recurrent along  $V$ . Since  $\phi(U)$  is a subsystem of the orbit closure  $(\overline{\mathcal{O}_V(y)}, V)$ , which is minimal, we have  $\phi(U) = \overline{\mathcal{O}_V(y)}$  and hence  $y \in \phi(U)$ . Any element of  $U \cap \phi^{-1}(y)$  can be chosen as  $x$ .

If the shift space  $X$  is countable, then so is  $U$ . Since an infinite minimal dynamical system is uncountable,  $U$  must be finite. Then each  $x \in U$  and  $\vec{v} \in V$  satisfy  $\sigma^{k\vec{v}}(x) = x$  for some  $k \geq 1$ . □

**Proposition 1** *Let  $X \subset \Sigma^{\mathbb{Z}^d}$  be a countable SFT and  $Y \subset \Gamma^{\mathbb{Z}^d}$  a non-horizontally constant shift space, and let  $\phi : X \rightarrow Y$  be a factor map. Then each  $y \in Y$  has a  $\phi$ -preimage  $x \in X$  which is non-horizontally periodic, and each horizontal row of  $x$  is ultimately periodic in both directions.*

In particular, each horizontal row of each  $y \in Y$  is ultimately periodic to the left and right.

*Proof* Let  $p_2, \dots, p_d \geq 1$  and  $\vec{p} = (p_2, \dots, p_d) \in \mathbb{Z}^{d-1}$ , and consider the subset  $Z_{\vec{p}} \subset X$  of configurations that are  $p_i \vec{e}_i$ -periodic for all  $2 \leq i \leq d$ . We can define  $Z_{\vec{p}}$  by adding a finite number of forbidden patterns to those that define  $X$ , so it is an SFT. Denote  $P = \{0\} \times \prod_{i=2}^d \{0, \dots, p_i - 1\} \subset \mathbb{Z}^d$ . Define a map  $\pi_{\vec{p}} : Z_{\vec{p}} \rightarrow (\Sigma^P)^{\mathbb{Z}}$  by  $\pi_{\vec{p}}(z)_n = \sigma^n(z)|_P$  for each  $n \in \mathbb{N}$ . Then  $\pi_{\vec{p}}$  is injective on  $Z_{\vec{p}}$ , and the image  $\pi_{\vec{p}}(Z_{\vec{p}})$  is a one-dimensional shift space whose symbols are blocks of shape  $P$  drawn from  $X$ . Since  $X$  is countable, so is  $\pi_{\vec{p}}(Z_{\vec{p}})$ , and it is easily seen to be an SFT as well. Lemma 1 implies that each configuration of the countable SFT  $\pi_{\vec{p}}(Z_{\vec{p}})$ , and hence each row of  $Z_{\vec{p}}$ , is ultimately periodic to the left and right.

Let  $x \in X$  be some non-horizontally periodic  $\phi$ -preimage of  $y$ , given by Lemma 2 for the subgroup  $\{0\} \times \mathbb{Z}^{d-1} \subset \mathbb{Z}^d$ . Let  $\vec{p} \in \mathbb{Z}^{d-1}$  be the vector of its periods in the

directions  $\vec{e}_i$  as above. Then  $x \in Z_{\vec{p}}$ , and hence it is ultimately periodic to the left and right, and so is  $y$ .  $\square$

**Proposition 2** *Let  $Y \subset \Gamma^{\mathbb{Z}^d}$  be a countably covered non-horizontally constant sofic shift. Given three nonempty words  $u, v, w \in \Gamma^+$ , it is decidable whether  ${}^\infty uvw^\infty \in P(Y)$ .*

*Proof* Let  $X \subset \Sigma^{\mathbb{Z}^d}$  be a countable SFT and  $\phi : X \rightarrow Y$  a factor map. Let  $y \in \Gamma^{\mathbb{Z}^d}$  be the configuration whose every horizontal row is equal to  ${}^\infty uvw^\infty$ . We have  ${}^\infty uvw^\infty \in P(Y)$  if and only if  $y \in Y$ .

Let  $r \in \mathbb{N}$  be the radius of  $\phi$ . If  $y \notin Y$ , then by compactness there exists  $m \in \mathbb{N}$  such that there exists no  $(2m + 2r + 1)^d$ -shaped pattern  $Q$  over  $\Sigma$  which is locally valid with respect to the rules of  $X$ , and satisfies  $\phi(Q) = y|_{[-m, m]^d}$ . This  $m$  can be found algorithmically by enumerating locally valid patterns.

Suppose then that  $y \in Y$ . By Proposition 1, there exists  $x \in \phi^{-1}(y)$  which is non-horizontally periodic, and whose rows are ultimately periodic to the left and right. Such an  $x$  can be found algorithmically by enumerating all possible vertical periods, eventual horizontal periods and transient parts, and verifying whether  $x \in X$  and  $\phi(x) = y$ . The last two conditions can be verified in finite time since the configurations are non-horizontally periodic and ultimately periodic to the left and right.  $\square$

Propositions 1 and 2 give two properties that all non-horizontally constant countably covered sofic shifts satisfy, one of which is geometric and the other computational in nature. All sofic shifts are effectively closed, which gives another computational necessary condition. We formalize these results in a definition and combine them into a theorem.

**Definition 1** Let  $Y \subset \Gamma^{\mathbb{Z}^d}$  be a one-dimensional shift space. We say that  $Y$  satisfies the *countable cover conditions*, if

1.  $Y$  is effectively closed,
2. every  $y \in Y$  is ultimately periodic to the left and right, and
3. given  $u, v, w \in \Gamma^+$ , it is decidable whether  ${}^\infty uvw^\infty \in Y$ .

A  $d$ -dimensional shift space  $Y \subset \Gamma^{\mathbb{Z}^d}$  satisfies the countable cover conditions if it is non-horizontally constant and  $P(Y)$  satisfies the one-dimensional conditions.

**Theorem 2** *For  $d \geq 2$ , every countably covered non-horizontally constant  $\mathbb{Z}^d$ -sofic shift satisfies the countable cover conditions.*

Since the last two conditions allow us to enumerate the configurations of the shift space, we can also enumerate its language.

**Proposition 3** *Every shift space that satisfies the countable cover conditions has a computable language.*

We now give examples of non-horizontally constant sofic shifts that do not satisfy the countable cover conditions, showing that they are not countably covered.

*Example 1* (condition 2 not implied by others) Let  $\Gamma = \{0, 1\}$ , and consider the one-dimensional shift space  $Z \subset \Gamma^{\mathbb{Z}}$  defined by the forbidden patterns  $0^m 10^m 1$  for all  $m > 0$ , as well as  $10^n 11$  and  $10^n 10^{n+2}$  for all  $n \geq 0$ . It is the orbit closure of the configuration  $z = {}^\infty 0.11010010001 \dots$ , where the number of 0s between consecutive 1s always increases by one. Since  $Z$  is effectively closed, Theorem 1 implies that the  $d$ -dimensional non-horizontally constant shift space  $P^\dagger(Z) \subset \Gamma^{\mathbb{Z}^d}$ , whose horizontal rows come from  $Z$ , is sofic for each  $d \geq 2$ . It is also countable, since  $Z$  consists of the orbit of  $z$ , the orbit of  ${}^\infty 010^\infty$ , and the all-0 configuration. However, since the configuration  $z \in Z$  is not ultimately periodic to the right, the shift space  $Z$  does not satisfy the second countable cover condition, and Theorem 2 implies that  $P^\dagger(Z)$  is not countably covered.

*Example 2* (condition 3 not implied by others) Let  $\Gamma = \{0, 1, 2, 3\}$ , and consider the one-dimensional shift space  $Z \subset \Gamma^{\mathbb{Z}}$  defined by the forbidden patterns  $ab$  for  $a > b \in \Gamma$ , the pattern  $01^m 2^n 3$  for each  $m, n \in \mathbb{N}$  such that the  $m$ th Turing machine  $T_m$  does not halt after exactly  $n$  steps on empty input, as well as  $01^m 2^{n+1}$  for each  $m, n \in \mathbb{N}$  such that  $T_m$  halts after  $n$  steps. Then, for each  $m \in \mathbb{N}$ , the shift space  $Z$  contains the configuration  ${}^\infty 01^m 2^n 3^\infty$  if  $T_m$  halts after exactly  $n$  steps, and  ${}^\infty 01^m 2^\infty$  if  $T_m$  never halts. These are the only configurations of  $Z$  that contain both 0 and 2.

For  $d \geq 2$ , the  $d$ -dimensional non-horizontally constant shift space  $P^\dagger(Z) \subset \Gamma^{\mathbb{Z}^d}$  is again sofic by Theorem 1. Each  $y \in P^\dagger(Z)$  is ultimately periodic to the left and right. However, given  $m \in \mathbb{N}$  it is undecidable whether  ${}^\infty 01^m 2^\infty \in Z$ , since this is equivalent to the complement of the halting problem. Thus  $Z$  does not satisfy the third countable cover condition, and by Theorem 2,  $P^\dagger(Z)$  is not countably covered. It can also be shown that the language of  $Z$  is computable. This example shows that the first two conditions, together with a computable language, do not imply the third.

With a similar construction, we also show that the last two conditions do not imply the first one. Of course, since this example is not effectively closed, the corresponding non-horizontally periodic shift space is not sofic.

*Example 3* (condition 1 not implied by others) Let  $\Gamma = \{0, 1, 2, 3\}$ , and consider the one-dimensional shift space  $Z \subset \Gamma^{\mathbb{Z}}$  defined by the forbidden patterns  $ab$  for  $a > b \in \Gamma$ , the pattern  $01^m 2^{m+n} 3$  for each  $m, n \in \mathbb{N}$  such that  $T_m$  does not halt after  $n$  steps,  $01^m 2^{m+n+1}$  for each  $m, n \in \mathbb{N}$  such that  $T_m$  halts in exactly  $n$  steps, as well as  $01^m 2$  for each  $m \in \mathbb{N}$  such that  $T_m$  never halts. Then  $Z$  consists of the configurations  ${}^\infty 01^m 2^{m+n} 3^\infty$  where  $T_m$  halts after  $n$  steps, as well as those that do not contain both 0 and 2. The second countable cover condition is immediate, and the third is also easy to see. Since the pattern  $01^m 2$  occurs in  $Z$  if and only if  $T_m$  eventually halts, the complement of the language is not computably enumerable, and hence  $Z$  is not effectively closed.



A modified version of Example 1 shows that vertical periodicity is essential for our proof technique.

*Example 4* (countably covered version of Example 1) Let  $\Gamma = \{0, 1\}$ , and let  $Z \subset \Gamma^{\mathbb{Z}}$  be as in Example 1. Consider the two-dimensional configuration  $y \in \Gamma^{\mathbb{Z}^2}$  whose central horizontal row is  $z = {}^\infty 0.11010010001 \dots$  and all other rows are filled with 0s, and let  $Y \subset \Gamma^{\mathbb{Z}^2}$  be the orbit closure of  $y$ . It consists of the orbit of  $y$  and all configurations that contain at most one 1.

We claim that  $Y$  is a countably covered sofic shift. For this, let  $\Sigma$  be the tile set shown in Fig. 2a, and let  $X \subset \Sigma^{\mathbb{Z}^2}$  be the SFT consisting of those tilings where the contents of adjacent tiles match. The white regions with the symbols N and S are deemed to have distinct colors. This SFT is countable: if a configuration does not contain any parallel line segments, then it consists of finitely many infinite regions separated by straight line segments, and if it does contain parallel line segments, then it contains at least one finite square, and thus an infinite sequence of growing squares, and is a translated version of the configuration in Fig. 2b.

Let  $\pi : X \rightarrow \Gamma^{\mathbb{Z}^2}$  be the block code that sends each tile with a black dot to 1, and the others to 0. It is easy to see that  $\pi(X) = Y$ , and hence  $Y$  is a countably covered sofic shift.

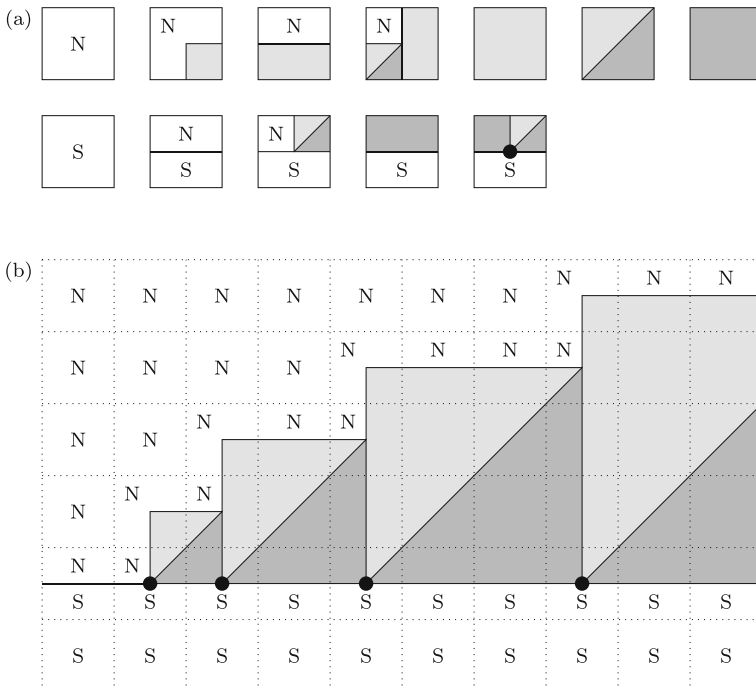
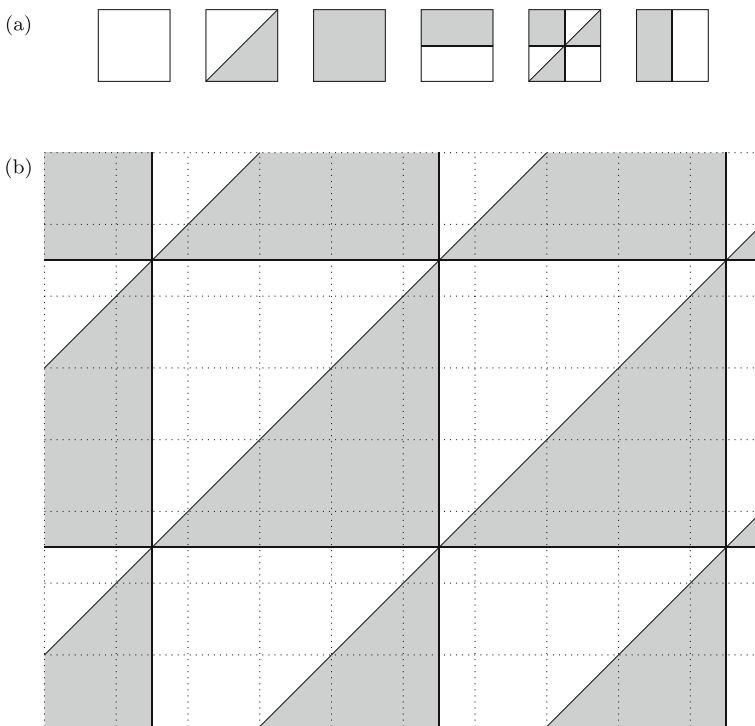


Fig. 2 a The tile set of Example 4, and b an example configuration. Dotted lines denote tile borders

We show that the converse of Theorem 2 holds in a particular subclass of shift spaces, namely, those whose projective subdynamics is contained in some one-dimensional countable sofic shift. In [17], such systems were said to have the *bounded signal property*. Before this, we give a few examples of two-dimensional countably covered sofic shifts that do not belong to this class, both to show that the result is not optimal, and to highlight some techniques of the construction.

*Example 5* (infinite number of periodic orbits) Consider the set of tiles shown in Fig. 3a, and the SFT  $X$  where the lines and colors of neighboring tiles must match. The configurations of  $X$  are grids formed by infinitely many squares of equal size, or ‘degenerate’ configurations containing infinite regions of the same color; see Fig. 3b for an example. It is easy to see that  $X$  is countable.

Consider the map  $\phi : X \rightarrow \{0, 1\}^{\mathbb{Z}^2}$  that sends to 1 those tiles that contain a vertical line (the two rightmost tiles in Fig. 3), and to 0 all other tiles. Then  $\phi(X)$  is a countably covered vertically constant sofic shift. The projective subdynamics  $P(\phi(X))$  consists of the periodic configurations  ${}^\infty(0^n 1)^\infty$  for all  $n \in \mathbb{N}$ , the ultimately periodic configuration  ${}^\infty 010^\infty$ , and the uniform configuration  ${}^\infty 0^\infty$ , as well

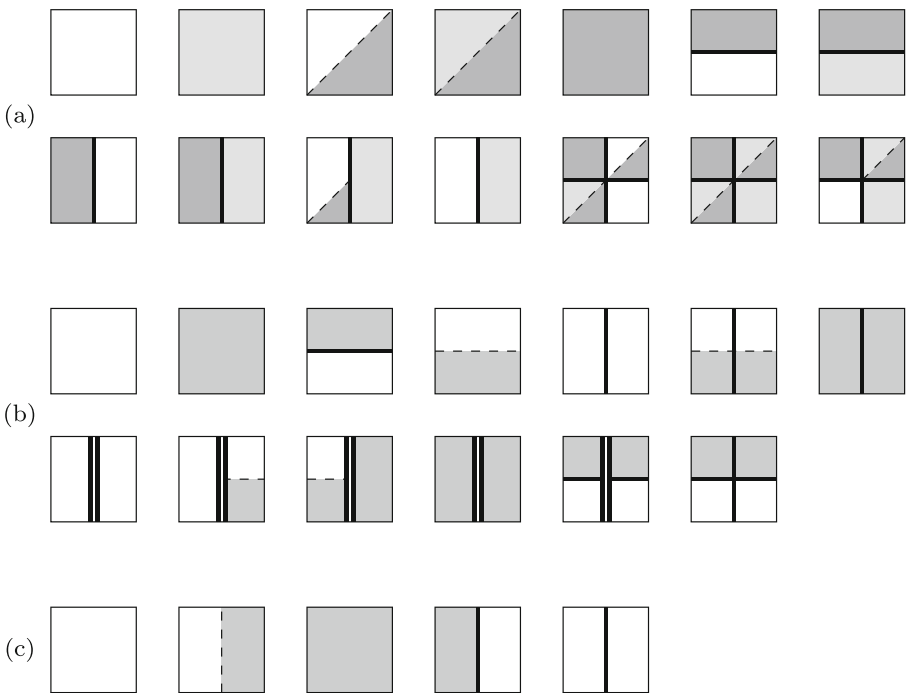


**Fig. 3** a The alphabet of the grid shift of Example 5, and b an example configuration. Dotted lines denote tile borders

as their shifts. In particular, Lemma 1 implies that  $P(\phi(X))$  is not contained in any one-dimensional countable sofic shift.

*Example 6* (unbounded number of “period breaker” patterns) Consider the one-dimensional shift space  $X \subset \{0, 1\}^{\mathbb{Z}}$  defined by forbidding those words over  $\{0, 1\}$  that, for some  $n \in \mathbb{N}$ , contain  $10^n 1$  as a subword and contain more than  $n$  occurrences of 1. Then  $X$  contains precisely those configurations where the number of 1s is strictly less than the distance between any two occurrences of 1. Since every configuration of  $X$  contains a finite number of 1s, it is a countable space. By Lemma 1, it is not contained in a countable one-dimensional sofic shift.

Let  $Y = P^\dagger(X) \subset \{0, 1\}^{\mathbb{Z}^2}$ . We show that  $Y$  is a countably covered sofic shift by constructing a countable SFT  $Z$  and a factor map  $\phi : Z \rightarrow Y$ . The alphabet of  $Z$  consists of three layers: the *grid layer* (which is somewhat similar to the SFT of Example 5 but more complex), the *counter layer*, and the *synchronization layer*. The idea is that the height of the grid cells on the grid layer is determined by the minimum distance between two 1s, the counter layer uses this information to globally bound the number of 1s, and the synchronization layer enforces some additional constraints to ensure countability. The tiles of the layers are shown in Fig. 4, and example configurations of the first two layers are shown in Figs. 5 and 6. The lines and colors of



**Fig. 4** The three layers of the alphabet of the shift space  $Z$  of Example 6: **a** the grid layer, **b** the counter layer, and **c** the synchronization layer

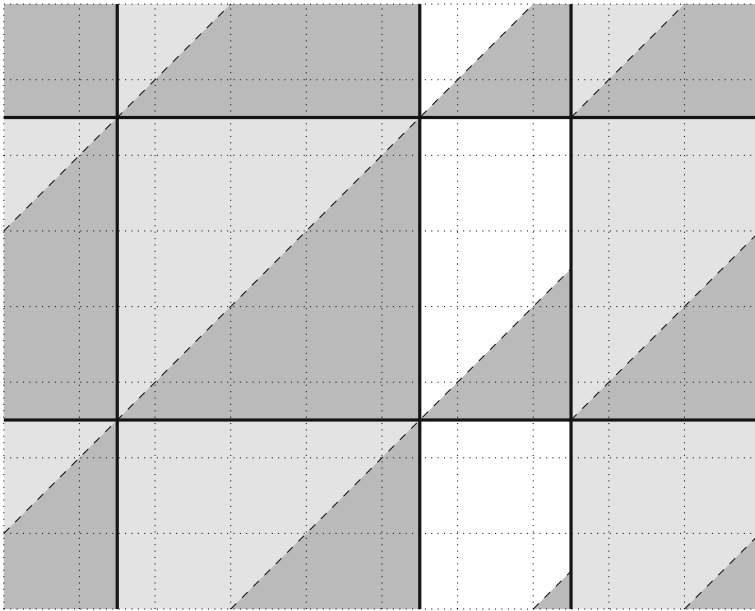


Fig. 5 An example configuration of the grid layer is Example 6

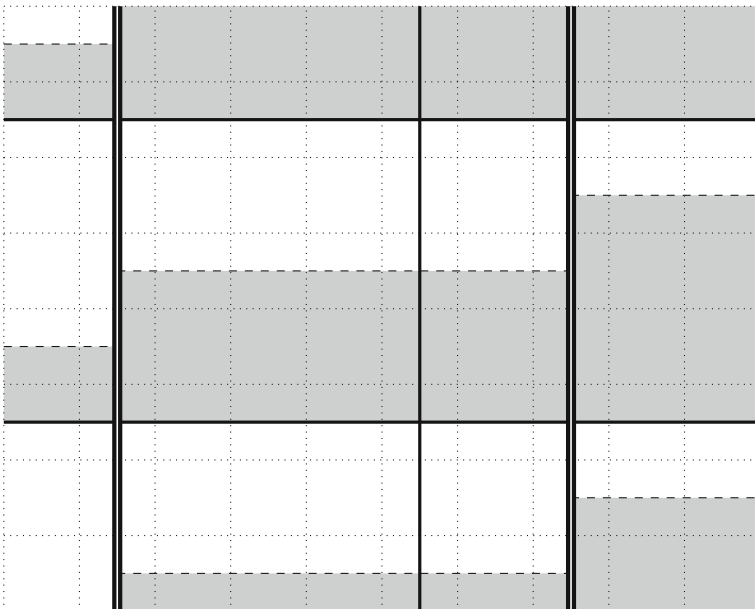


Fig. 6 An example configuration of the counter layer of Example 6. The thick lines coincide with those of Fig. 5

each layer must be correctly matched in neighboring tiles. Furthermore, we have the following rules between the layers:

1. The thick vertical lines of the grid layer must be placed on the thick vertical lines of the counter layer and the synchronization layer, and vice versa.
2. The thick horizontal lines of the grid layer must be placed on the thick horizontal lines of the counter layer, and vice versa.
3. If the dashed vertical line of the synchronization layer crosses the dashed diagonal line of the grid layer, then the dashed horizontal line of the counter layer must be present at that position as well.
4. A double vertical line on the counter layer must be paired with a tile of the grid layer that has a light gray region in its right half.
5. A vertical line of the grid layer with a white region on its left must be paired with a double vertical line on the counter layer.

Otherwise, the three layers are independent.

These rules force the following structure to  $Z$ . On the grid layer, we have an infinite grid of rectangular areas which are at least as high as they are wide (possibly with degenerate infinite grid cells). Their top halves are colored either white (if they are higher than they are wide) or light gray (if they are square-shaped). The colors of a single vertical column of grid cells must match. In particular, in a configuration with only finite grid cells and at least one light gray column, the heights of all grid cells are equal. On the counter layer, each horizontal row of grid cells of height  $h$  contains one counter, which may be incremented at most  $h - 1$  times at the vertical borders between grid cells. A vertical border is a doubled line if an increment happens at that position. By rule 3, the counters of different rows are synchronized by the vertical signals of the synchronization layer, and by rule 4, an incrementation of the counter causes the next column of grid cells to be light gray, and hence square-shaped. Conversely, rule 5 ensures that a non-square column of grid cells is accompanied by an increase of the counter, so only finitely many columns of grid cells can be non-square. In particular, since the counter can be incremented finitely many times, a configuration can contain only finitely many white columns.

We claim that the SFT  $Z$  is countable. Consider first a configuration that contains a finite grid cell, which forces the entire configuration to consist of a grid of finite cells. There are countably many choices of the height of the grid cells, and after that, countably many choices for the finitely many columns of non-square grid cells and the values of the counters. Other parts of the configuration are forced by these choices. An infinite grid cell cannot have finite width or height, so there are countably many choices for configurations containing them as well.

The block code  $\phi$  sends each double vertical line on the counter layer to 1, and all other tiles to 0. Thus, in the image of a configuration with height- $h$  rectangles, there can be at most  $h - 1$  columns of 1s (since counters can be incremented at most  $h - 1$  times), and the distance between consecutive columns of 1s is at least  $h$  (since each incrementation must be followed by a column of square grid cells). With the help of non-square grid cells, the distance can in fact be any number at least

$h$ , independently for each pair of consecutive 1s. Since  $h$  can be arbitrary, we have  $\phi(Z) = Y$ .

## 4 Partial Converse

We now prove a partial converse to Theorem 2 in a restricted class of shift spaces. The proof is a relatively complicated construction spanning most of the remaining article. We begin with some closure properties of the class of shift spaces that satisfy the countable cover conditions.

**Lemma 3** *Let  $W \subset \Sigma^{\mathbb{Z}}$  be an edge shift and  $Z \subset \Gamma^{\mathbb{Z}}$  a sofic shift, and let  $\psi : W \rightarrow Z$  be a right-resolving factor map. Let  $Y \subset Z$  be a shift space that satisfies the countable cover conditions. Then  $X = \psi^{-1}(Y)$  satisfies the countable cover conditions as well.*

*Proof* As a preimage of an effectively closed shift space,  $X$  is effectively closed.

Let  $x \in X$  be arbitrary, and denote its ultimately periodic image by  $\phi(x) = \infty uvw^{\infty}$ , where  $u, v, w \in \Gamma^+$ . Let  $G = (V, E)$  be the graph associated to the edge shift  $W$  (with  $E = \Sigma$ ). Denote by  $\rho_u : \Sigma \rightarrow \Sigma \cup \{\#\}$  the partial transition function induced by the word  $u$ , and similarly for  $w$ . For  $s \in \Sigma$ , the value  $\rho_u(s)$  is the final edge of the path in  $G$  that starts from  $s$  and has label  $u$ , or  $\#$  if no such path exists. The partial function is well-defined by the right-resolvingness of  $\psi$ . If  $x_i = s \in \Sigma$  and  $\phi(x)_{[i, i+|G||u|-1]} = u^{|G|}$ , then there exist  $m, n \leq |G|$  such that  $x_{i+m|u|} = \rho_u^m(s) = \rho_u^{m+kn}(s) = x_{i+(m+kn)|u|}$  for all  $k \in \mathbb{N}$ . This implies that  $x$  is ultimately periodic to the left with period  $n|u|$ . Similarly, it is ultimately periodic to the right with period  $n'|w|$  for some  $n' \leq |G|$ . This implies that  $X$  satisfies the second countable cover condition.

Let  $u, v, w \in \Gamma^+$  be arbitrary. The condition  $\infty uvw^{\infty} \in X$  is equivalent to  $\psi(\infty uvw^{\infty}) \in Y$ . This configuration is ultimately periodic, and the periods and transient part are easily computable from  $u, v$  and  $w$ . Hence  $X$  satisfies the third countable cover condition.  $\square$

**Lemma 4** *Let  $Y \subset \Gamma^{\mathbb{Z}}$  be a shift space that satisfies the countable cover conditions, and let  $Z \subset \Gamma^{\mathbb{Z}}$  be an SFT. Then  $X = Y \cap Z$  satisfies the countable cover conditions as well.*

*Proof* The first countable cover condition holds, since effectively closed shifts are closed under finite intersection. The second condition is immediate, since  $X \subset Y$ . Given  $u, v, w \in \Gamma^+$ , we can decide  $\infty uvw^{\infty} \in Y$  by assumption, and  $\infty uvw^{\infty} \in Z$  is decidable since  $Z$  is defined by finitely many local rules. Hence  $X$  satisfies the third countable cover condition.  $\square$

**Theorem 3** *Let  $Z \subset \Gamma^{\mathbb{Z}}$  be a countable sofic shift, let  $d \geq 2$ , and let  $Y \subset \Gamma^{\mathbb{Z}^d}$  be a non-horizontally constant countable shift space with  $P(Y) \subset Z$ . If  $Y$  satisfies the countable cover conditions, then it is countably covered.*

The high-level idea of the proof is the following. By Lemma 1 and the above lemmas, we may assume  $Z$  is a one-dimensional SFT whose configurations consist of periodic regions separated by a bounded number of short intermediate words. Then the vertical periodicity and the local constraints of  $Z$  can be enforced by local rules. For each pair of the intermediate words, the SFT cover contains a component that periodically initializes a simulated counter machine computation between those words whenever both of them occur in the image. The counter machine reads the part of the configuration between the two words (and a short region to its left and right) and verifies its local correctness. Any non-horizontally periodic configuration whose rows come from  $Z$  contains an outermost pair of intermediate words, and the computation related to this pair effectively verifies the entire configuration. The tricky part of the construction is to guarantee that the computation is restarted periodically while preserving countability, which is achieved by simulating deterministic tile sets that tile large rectangles but not the infinite plane.

*Proof* We prove the result in the case  $d = 2$ . The general case follows easily by considering an SFT cover which is constant in each dimension except the first two, and a factor map whose neighborhood is contained in  $\mathbb{Z}^2 \times \{0\}^{d-2}$ .

We first simplify the structure of  $Z$  in order to make it easier to handle in the construction. Let  $W \subset \Sigma^{\mathbb{Z}}$  be an edge shift and  $\psi : W \rightarrow Z$  a right-resolving factor map as given by Lemma 1. By Lemma 3, the  $\psi$ -preimage of  $P(Y)$  satisfies the countable cover conditions. Let  $Y' \subset \Sigma^{\mathbb{Z}^2}$  be a vertically constant shift space with  $P(Y') = \psi^{-1}(P(Y))$ . If  $Y'$  is countably covered, then so is  $Y = \psi(Y')$ . Hence we can safely assume that  $Z$  is an edge shift satisfying the conditions of Lemma 1 by replacing it with  $W$  if necessary. Thus  $Z$  is a finite (not necessarily disjoint) union of SFTs of the form

$$Cl(\{\infty u_0 v_1 u_1^{n_1} v_2 u_2^{n_2} \cdots v_k u_k^\infty \mid n_1, \dots, n_k \in \mathbb{N}\}) \tag{1}$$

for some rank  $k \in \mathbb{N}$  and words  $u_i, v_i \in \Gamma^+$ , where the  $u_i$  consist of the edges in the disjoint cycles in the graph and hence do not share any symbols. The  $v_i$  consist of the paths between these cycles, so they do not share symbols with any other  $v_j$  or  $u_j$ . The  $u_i$  are called the *period words* and the  $v_i$  are called the *period breakers*. By Lemma 4, we can consider each of these components separately, and take the disjoint union of the resulting two-dimensional countable SFTs to obtain a countable cover for  $Y$ . From now on, we assume that  $Z$  has the form (1) for some rank  $k \in \mathbb{N}$  and words  $u_i, v_i \in \Gamma^+$  that share no symbols.

We now begin the construction of the countable SFT  $X$ . It contains a number of layers, the first of which is a vertically constant layer whose rows come from  $Z$ , called the *image layer*. The block code  $\phi : X \rightarrow \Gamma^{\mathbb{Z}^2}$  is simply a projection onto the image layer. For each pair of indices  $1 \leq i < j \leq k$ , there is also a layer corresponding to the period breakers  $v_i$  and  $v_j$ . This layer is denoted by  $X_{i,j}$ , and its purpose is to run a computation verifying the correctness of a segment of the image layer that contains  $v_i$  and  $v_j$ , if both occur there.

Each layer  $X_{i,j}$  has the same basic structure, which we break into three sub-layers. The first layer is called the *verification machine layer*. Let  $M_{i,j} = (k, Q, \Sigma, \delta, q_0, q_f)$  be a deterministic integer  $k$ -counter machine with one oracle

counter. Let  $V_{i,j}$  be the SFT defined in Section 2.2 that simulates the machine  $M_{i,j}$ . It contains a computation cone that expands to the left and right, a special read-only counter that always maintains the value 0, and an oracle layer that can be probed by the oracle counter. We modify  $V_{i,j}$  as follows. First, the oracle layer is replaced by the image layer of  $X$ , so that the simulated machine is able to read its contents. When the zig-zag head enters the final state  $q_f$ , it no longer makes a new sweep, but stays indefinitely at the position of its anchor counter, at the central column of its computation cone. We also introduce a new symbol  $\#$ , which is required to fill an entire horizontal row of the layer wherever it is present. If the next row contains the zig-zag head, it must be in the initial state  $q_0$ , with each counter set to 0 and the computation cone having width 1. If the previous row contains the zig-zag head, it must be in the final state. There are no other restrictions on  $\#$ . This means that the new symbol can effectively reset the computation after it has halted. On each horizontal row of the verification machine layer (except those that are filled with  $\#$ ), we require that the anchor counter is placed on the leftmost symbol of  $v_i$  on the image track. Since this symbol occurs only once in  $v_i$ , and does not occur in any word  $v_p$  for  $p \neq i$  or  $u_p$  for  $1 \leq p \leq k$ , this uniquely determines the position of the anchor counter as a function of the image layer.

The counter machine  $M_{i,j}$  behaves as follows. First, it reads the entire string  $w = v_i u_i^{n_i} v_{i+1} \cdots v_j \in \Sigma^+$  between the words  $v_i$  and  $v_j$  on the image layer (if  $v_j$  is not present, this computation never halts). It then checks whether  ${}^\infty u_{i-1} w u_j^\infty \in P(Y)$ , which is computable by assumption. If this holds, then  $M_{i,j}$  enters the final state and suspends the computation. If  ${}^\infty u_{i-1} w u_j^\infty \notin P(Y)$ , then  $M_{i,j}$  computes a number  $p \in \mathbb{N}$  such that  $u_{i-1}^p w u_j^p$  does not occur in  $P(Y)$ , which is possible since  $P(Y)$  has a computable language. Then,  $M_{i,j}$  reads the  $|u_{i-1}|^p$  symbols to the left of  $v_i$ , and the  $|u_j|^p$  symbols to the right of  $v_j$ , obtaining an extended string  $u w v \in \Sigma^+$ . If  $u w v$  occurs in  $P(Y)$ , then  $M_{i,j}$  enters the final state and suspends the computation. If  $u w v$  does not occur in  $P(Y)$ , then  $M_{i,j}$  halts and produces a tiling error.

A moment's reflection should convince the reader that if each machine  $M_{i,j}$  for  $1 \leq i < j \leq k$  is able to complete its computation in a configuration of  $X$ , then the image layer is a configuration of  $P^\dagger(Y)$ . However, at this point of the construction it is not guaranteed that every valid configuration contains such computations, since the machines may stay in the final state  $q_f$  on each row. Another issue is that the machines may perform their computations infinitely many times, with a row of  $\#$ -symbols between each run, and since the zig-zag head may wait in the final state  $q_f$  an indeterminate number of steps before each reset, the number of such configurations is uncountable. The remaining layers of  $X_{i,j}$  will fix these issues by forcing the  $\#$ -rows to occur periodically.

The second sub-layer of  $X_{i,j}$  is the *machine grid*, which is exactly equivalent to the SFT of Example 5. We require that the leftmost symbol of  $v_i$  and the leftmost symbol of  $v_j$  (which are unique to these words and occur in them only once) are paired with vertical lines of the machine grid, and none of the symbols between them (the remaining symbols of  $v_i$ , as well as all symbols of  $u_p$  for  $i \leq p < j$  and  $v_p$  for  $i < p < j$ ) are paired with vertical lines of the machine grid. This completely determines the locations of the vertical lines of the machine grid as a function of

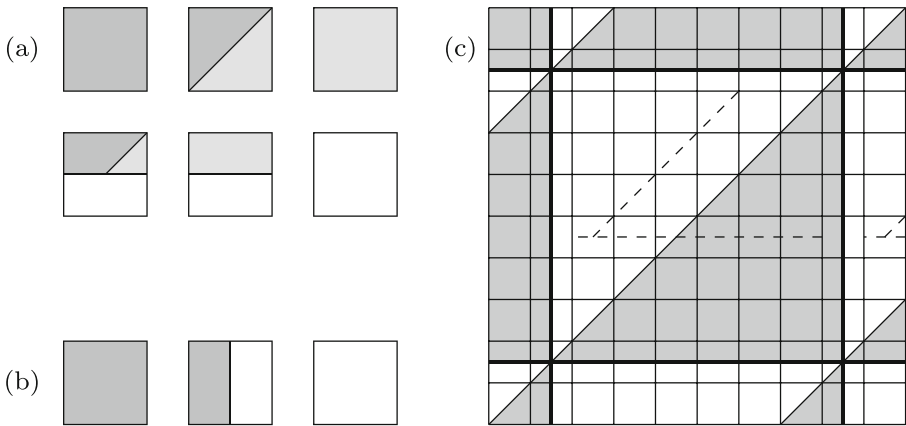


the image layer. Those grid cells that are bordered by  $v_i$  and  $v_j$  are called *central*. The purpose of the machine grid is to provide a skeleton for another computational layer consisting of counter machines. If the image layer contains at least two period breakers, then at least one of the layers  $X_{i,j}$  has a non-degenerate machine grid, on top of which we can force a computation.

The final sub-layer is called the *tile machine layer*, and it is used to run a computation inside each cell of the machine grid. These computations will be linked together in order to simulate a tiling in the machine grid. The purpose of the simulated tiling is to periodically send a signal to the verification machine layer, which in turn forces a row of  $\#$ -symbols that resets the computation of  $M_{i,j}$ .

The computations of the tile machine layer are carried out by deterministic counter machines, which we embed into the machine grid. The simulation of computation will take place in the upper left half of each grid cell (copies of the solid white tile of Fig. 3). For this, we take a deterministic  $k$ -counter machine  $M = (k, Q, \delta, q_0, q_f)$  and modify the SFT  $X_M$  defined in Section 2.2. We join the symbols  $\#_L$  and  $\#_R$  into one symbol,  $\#$ , which is overlaid on the tiles of the machine grid layer that are not solid white. On the solid white tiles we overlay other tiles of  $S_M$ . The bottom-most white tile initializes the computation in state  $q_0$  with all internal counters set to 0 (the input counters can have any value that fits in the grid cell). Once the machine enters the final state  $q_f$ , the zig-zag head stops and stays at the left end of the computation cone, similarly to the simulation of  $M_{i,j}$ . We require that the top left corner of each machine grid cell contains a zig-zag head in the state  $q_f$ .

Note that the behavior of each simulated copy of  $M$  is a priori completely independent from the others. We now enhance the construction by allowing the machines to communicate with their neighbors. This is reminiscent of the fixed point construction of [6], where a tiling simulates a grid of Turing machines, which in turn collaborate to simulate another tileset. Each copy of  $M$  has four read-only input counters:  $C_{\text{north}}$ ,  $C_{\text{south}}$ ,  $C_{\text{east}}$  and  $C_{\text{west}}$ , called the *north, south, east and west counters*. We would like to enforce by local rules that the north counter of each copy of  $M$  is equal to the south counter of the copy to its north, and analogously for the east and west counters. In order to implement these constraints, we introduce new signals into the machine grid cells. We define new tile sets  $S_{\text{hsig}}$  and  $S_{\text{vsig}}$  in Fig. 7. To each interior tile of a grid cell we overlay an element of the product alphabet  $S_{\text{sig}} = S_{\text{hsig}}^2 \times S_{\text{vsig}}$ , called the *east, west and south signal layers*. In the interior of each grid cell, for each track of  $S_{\text{sig}}$ , the lines and colors of adjacent tiles must match, and the horizontal or vertical line must be present (this is enforced by requiring the corners of the grid cell to be adjacent to certain colors on each track). Also, the leftmost tile on the horizontal line must be the bottom left tile in Fig. 7a. In each grid cell, the line on the west signal track must be at the same height as the line on the east signal track of the eastern neighboring grid cell. The east and west counters of each copy of  $M$  must be positioned where the corresponding diagonal signals of  $S_{\text{sig}}$  hit the top edge of the grid cell. The vertical line of the south signal layer must be positioned on the south counter, and the south signal must hit the southern border of the grid cell at the same position as the north signal of the southern neighboring grid cell. In this way, the desired couplings between the input counters of neighboring grid cells are achieved.



**Fig. 7** **a** The alphabet  $S_{\text{hsig}}$ , **b** the alphabet  $S_{\text{vsig}}$ , and **c** an illustration of the coupling between the west and east signals of neighboring grid cells. Counters and other signals are not shown

Each simulated copy of  $M$  executes the same program, which features a computable sequence of Wang tile sets  $(T_n)_{n \in \mathbb{N}}$ . The edge colors of each  $T_n$  are represented as integers, and the sets of colors of different tile sets are disjoint. Each machine checks that its north, south, east and west counters form a valid tile in one of the sets  $T_n$ , and if this is the case, enters the final state  $q_f$ . We say that the machine and the associated machine grid cell *simulate* said tile. If the quadruple of numbers is not a valid tile, the machine halts and produces a tiling error.

In this way, each grid cell acts as a tile of some  $T_n$  (for the same  $n$ , since the colors are disjoint), and an infinite machine grid simulates a valid tiling over this set. Note that the number of computation steps generally depends on the values of the input counters, even if they represent tiles of the same set  $T_n$ . Since the simulated copies of  $M$  are not required to halt at the same time and can wait in the state  $q_f$  until the computation cone terminates, this is not an issue as long as the grid cells are large enough.

We now describe the tile sets  $T_n$  in more detail. Let  $g : \mathbb{N} \rightarrow \mathbb{N}$  be a computable function depending on  $Y$ , which we fix later. The tiles of  $T_n$  are similar to those of the grid shift in Example 5, and we call the structure they form in a valid configuration the *simulated grid*. In addition to the markings in Fig. 3, on each tile not containing a horizontal or vertical line we superimpose a tile of an auxiliary tile set  $R_n$ . The tile sets  $R_n$  have the following properties.

- Each  $R_n$  is a north-west deterministic tile set.
- The tile set  $R_n$  can tile a square region of the shape  $[0, g(n) - 1]^2$  so that the west border is colored uniformly with a color  $W_n$ , and the north border is colored uniformly with  $N_n$ .
- The tile set  $R_n$  does not tile the infinite plane.
- The tile sets  $R_n$  and the colors  $W_n$  and  $N_n$  are computable from  $n$  by a counter machine in time  $\exp(O(n^q))$  for some constant  $q \in \mathbb{N}$ , independently of the choice of  $g$ .

Such tile sets can be constructed as variants of the Robinson tiles of [16], as was done in [9]. Note the requirement on the time complexity, which can be satisfied as follows. The construction relies on embedding a Turing machine computation in the Robinson tiling. If the machine halts after  $m$  steps when initialized on an empty tape, the tile set can tile a square of shape  $[0, m - 1]^2$ , but does not tile the infinite plane. We can implement the tile sets  $R_n$  using an embedded Turing machine that writes the number  $n$  to the tape, counts to  $g(n)$  and halts. Then  $R_n$ ,  $W_n$  and  $N_n$  can be computed from  $n$  by a polynomial time algorithm, which a counter machine can in turn simulate with exponential slowdown by storing the contents of the binary tape of a Turing machine into the base-2 representations of the values of its counters [12, Section 11.2]. This establishes the desired properties of the  $R_n$ .

We require that the tiles of  $R_n$  that are superimposed on tiles directly to the east of a vertical grid line of  $T_n$  have west color  $W_n$ , and those that lie directly to the south of a horizontal grid line have north color  $N_n$ . The tiles of  $R_n$  that lie to the north or west of the grid lines have no additional restrictions on their edge colors. The idea is that the tileset forces the occurrence of grid cells which are filled by tiles drawn from  $R_n$ , and hence have width and height at most  $g(n)$ . Because the tile set  $R_n$  is north-west deterministic, each simulated grid cell has identical contents. Thus the SFT defined by  $T_n$  is countable.

*Remark 1* At this point of the construction we have defined two different grids associated to the layer  $X_{i,j}$ , and the reader should take note of their distinct roles. The first one is the machine grid on the second sub-layer of  $X_{i,j}$ . Its horizontal lines are synchronized to the words  $v_i$  and  $v_j$ , and the counter machines of the tile machine layer are simulated inside its grid cells.

The second one is the simulated grid. It is found in the SFTs defined by the tile sets  $T_n$ , which the counter machines of the tile machine layer simulate. Each grid cell of the machine grid corresponds to a tile of  $T_n$  for some choice of  $n$ , so the simulated grid cells are formed by collections of machine grid cells. Inside each simulated grid cell we have a (simulated) locally valid pattern over the auxiliary tile set  $R_n$ .

We will use the horizontal lines of the simulated grid to control the #-rows of the verification machine layer.

For this, we relate the verification machine layer of  $X_{i,j}$  to the machine grid and tile machine layers in the following way. The symbol # of the verification machine layer can only be paired with horizontal lines of the machine grid. If the zig-zag head of  $M_{i,j}$  is in the final state  $q_f$  and is directly below a horizontal line of the machine grid, then that horizontal line must contain #-symbols on the verification machine layer. We then allow each simulated copy of  $M$  to read the contents of the cells below its zig-zag head that contain a horizontal line of the machine grid when it begins its computation. If those cells contain # on the verification machine layer, then  $M$  must simulate a tile of the simulated grid that contains a horizontal line; otherwise, it cannot simulate such tile. This change requires  $M$  to remember one more bit of information, which can be incorporated to its finite internal state.

These modifications have the following effects on  $X_{i,j}$ . The simulation of the counter machine  $M_{i,j}$  is reset at the bottom edge of each horizontal line of the simulated grid, and before this reset, the machine must have finished its computation and entered the final state. Conversely, when the machine  $M_{i,j}$  has entered its final state, the next horizontal line of the machine grid must reset the computation, and the corresponding tiles of the simulated grid must contain horizontal lines. Hence the computations of  $M_{i,j}$  are synchronized with the simulated horizontal lines of  $T_n$ . Since  $M_{i,j}$  is deterministic, it cannot change its behavior after a reset, so each configuration of  $X_{i,j}$  containing a central column of machine grid cells is vertically periodic.

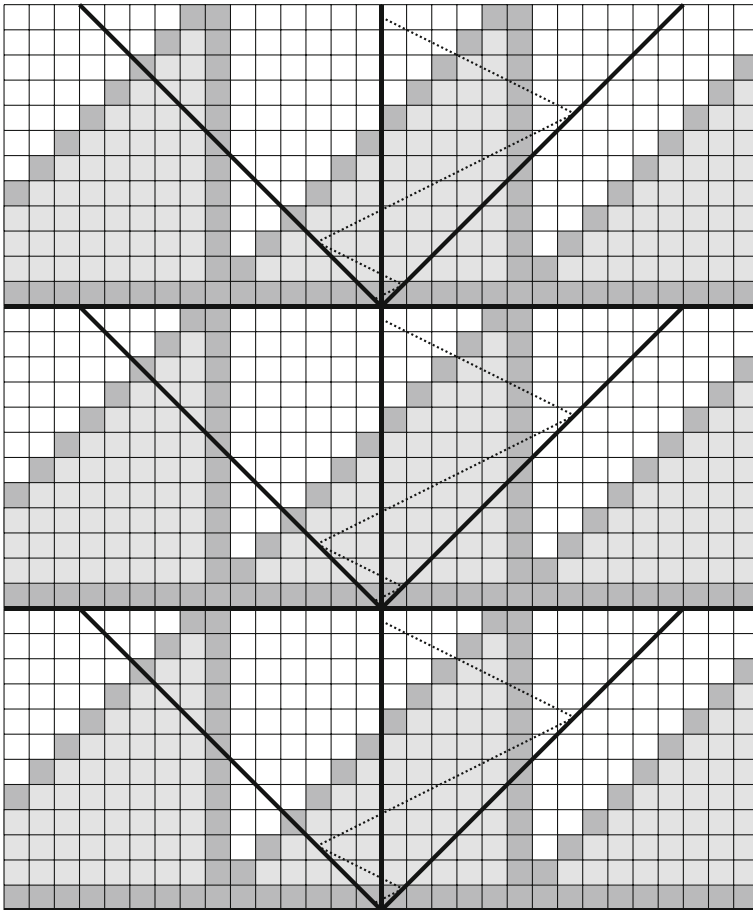
We now choose the computable function  $g : \mathbb{N} \rightarrow \mathbb{N}$  in the definition of the tile sets  $T_n$ , which gives a bound for the vertical distance between two resets. For  $m \in \mathbb{N}$ , let  $H(m)$  be the maximal amount of vertical space, measured in number of cells, needed for the simulation of the machine  $M_{i,j}$  in a configuration where the distance between  $v_i$  and  $v_j$  (and thus the size of the machine grid cells) is  $m$ . In such a configuration, each simulated machine  $M$  can perform at most  $\log_3(m+1)$  computation steps before halting: the machine grid cells have height  $m$ , each simulated computation step of  $M$  consists of one back-and-forth sweep of the zig-zag head, and the amount of vertical space required for the  $i$ th sweep is  $2 \cdot 3^i$  tiles, so  $n$  sweeps require  $\sum_{i=0}^{n-1} 2 \cdot 3^i = 3^n - 1$  tiles. For  $n \in \mathbb{N}$ , let  $G(n)$  be the maximal number of steps required for  $M$  to simulate a tile of  $T_n$ . The functions  $G, H : \mathbb{N} \rightarrow \mathbb{N}$  are computable, and by our construction of  $T_n$  we have  $G(n) = \exp(O(n^q))$ , where the constant in the  $O$ -notation may depend on  $g$ . We may also assume that  $G$  and  $H$  are non-decreasing. We require

$$g(n) \geq H(3^{G(n)} - 1) \quad (2)$$

for all large enough  $n \in \mathbb{N}$ ; such a  $g$  can be found since the right hand side is computable, and  $H$  and  $q$  are independent of the choice of  $g$ . For simplicity, we choose the same function  $g$  for each choice of  $i$  and  $j$ . Let  $n_0 \in \mathbb{N}$  be such that (2) holds for all  $n \geq n_0$ , and denote  $m_0 = 3^{G(n_0)} - 1$ .

Figure 8 is a schematic diagram of a configuration of  $X_{i,j}$ . The small squares are machine grid cells, each of which contains a simulated copy of  $M$  (not shown). Dark gray squares simulate the horizontal, vertical and/or diagonal lines of the simulated grid, while light gray and white squares simulate the uniformly colored tiles. The machine grid cells inside the large simulated grid cells also simulate the superimposed tiles of  $R_n$ . The thick horizontal lines are rows of  $\#$ -symbols on the verification machine layer. The thick vertical line is the anchor of the verification machine layer, and the thick diagonal lines are the borders of the computation cone. The dotted line is the zig-zag head. Counters are not shown. The pattern repeats periodically to the north and south, and ultimately periodically to the east and west.

We make one final modification to  $X_{i,j}$ . Suppose that the distance between the leftmost symbols of  $v_i$  and  $v_j$ , or equivalently the size of the machine grid cells, is some  $m \leq m_0$ . In this case we allow the tile machine layer to be filled with a special blank symbol, so that no computation takes place and no tiling is simulated. If the blank symbol is present, we require that for every  $H(m)$  consecutive horizontal rows there is exactly one row of  $\#$ -symbols on the verification machine layer, which means



**Fig. 8** A schematic diagram of a configuration of  $X_{i,j}$

that the machine  $M_{i,j}$  is guaranteed to be present and have enough time to finish its computation before restarting periodically. These modifications can be realized by local rules, since they concern finitely many machine grid sizes and finitely many local patterns of the layers. The reason for these additions is that in small machine grid cells, there might not be enough space for the machines  $M$  to simulate a suitable tile set  $T_n$ , so that no valid configuration of the tile machine layer exists. We still want to allow  $M_{i,j}$  to perform its computation and produce a configuration of  $Y$  on the image layer, and this is what the blank symbol achieves. This concludes the definition of the SFT  $X$ .

It remains to prove two claims: that the SFT  $X$  is countable, and that  $\phi$  is a factor map from  $X$  onto  $Y$ . We begin with the former claim. Let  $z = \infty u_{i-1} v_i u_i^{n_i} v_{i+1} \cdots v_j u_j^\infty \in Z$  be a configuration whose outermost period breakers are  $v_i$  and  $v_j$  for some  $i < j$ , and let  $x \in X$  be a configuration whose image layer

contains  $z$ . There are countably many choices for  $z$ , and we claim that there are countably many choices for  $x$  once  $z$  is fixed. For each  $i \leq a < b \leq j$ , the size of the machine grid cells and the positions of their vertical grid lines in  $X_{a,b}$  are determined by the positions of the words  $v_a$  and  $v_b$ . For other choices of  $a < b$ , the machine grid of  $X_{a,b}$  has only one or zero vertical and horizontal lines, since at most one of the words  $v_a$  and  $v_b$  is present on the image layer. We call these layers  $X_{a,b}$  *degenerate*. For the positions of the horizontal lines of each machine grid (degenerate or not) we have a countable number of choices.

The tile machine layer and verification machine layer of each  $X_{a,b}$  work in unison to guarantee countability. If  $X_{a,b}$  is degenerate, then there are only finitely many copies of the machines  $M$  and  $M_{a,b}$ , so there are countably many choices for the computations. Due to our method of implementing the simulations, the set of degenerate configurations where no computation takes place is likewise countable. Suppose then that  $X_{a,b}$  is not degenerate, and that the position of the machine grid has been fixed. If the machine grid is sparse enough, then the tile machine layer is not blank. Each machine grid cell hosts a computation of  $M$ , which simulates a tile of  $T_n$ , for some  $n \in \mathbb{N}$  that all the machines agree on. The number of valid tilings of  $T_n$  is countable, and each of them contains a simulated grid of side length at most  $g(n)$ . Each horizontal line of  $T_n$  causes a new simulation of  $M_{a,b}$  to be initialized at the lower left corner of the machine grid cell on the central column. Since  $M_{a,b}$  is deterministic, each simulated copy performs the same computation. If the machine grid is dense, then the tile machine layer is blank, and  $M_{a,b}$  is forced to perform its computation periodically due to the additional rules we imposed in this case. In any case,  $x$  is completely determined by the positions of the  $\#$ -rows on the verification machine layer, for which we have countably many choices. Hence there are countably many choices for  $x$ .

Finally, if the image layer contains a degenerate configuration of the type  ${}^\infty u_{i-1} v_i u_i^\infty$  or  ${}^\infty u_i^\infty$ , then each layer  $X_{a,b}$  is also degenerate, and has countably many choices for its contents as above. We have shown that  $X$  is a countable SFT.

Next, we claim that  $\phi(x) \in Y$  for each  $x \in X$ . Suppose that  $x$  has an image layer whose rows have the form  $z = {}^\infty u_{i-1} v_i u_i^{n_i} v_{i+1} \cdots v_j u_j^\infty \in Z$ . If the machine grid cells of  $X_{i,j}$  have width less than  $m_0$ , then the tile machine layer is blank, and the verification machine layer periodically simulates a computation of  $M_{i,j}$ . If the grid cells have width at least  $m_0$ , then the tile machine layer is not blank. In particular, the copies of  $M$  on the tile machine layer simulate a tiling of some  $T_n$ , which periodically initializes a simulated computation of  $M_{i,j}$  on the verification machine layer. In both cases,  $M_{i,j}$  performs a successful computation, so we have two possibilities: either  $z \in P(Y)$ , in which case we are done, or  $z \notin P(Y)$ , in which case the machine finds a number  $p \in \mathbb{N}$  such that the word  $w' = u_{i-1}^p v_i u_i^{n_i} v_{i+1} \cdots v_j u_j^p$  does not occur in  $P(Y)$ . After this, it scans  $|u_{i-1}|^p$  cells to the left of  $v_i$  and  $|u_j|^p$  steps to the right of  $v_j$ , finds the word  $w'$ , and halts with a tiling error. This is a contradiction, so we must have  $z \in P(Y)$ . We have shown that  $\phi(x) \in Y$ .

We claim that  $\phi : X \rightarrow Y$  is surjective. Let  $y \in Y$  be arbitrary, and let  $z = {}^\infty u_{i-1} v_i u_i^{n_i} v_{i+1} \cdots v_j u_j^\infty \in P(Y) \subset Z$  be its central row. We construct a configuration  $x \in X$  that has  $y$  on its image layer. Let  $1 \leq a < b \leq k$  be arbitrary. If  $a < i$  or  $j < b$ , then the layer  $X_{a,b}$  of  $x$  can be degenerate: the machine grid layer contains a single vertical line on the leftmost symbol of  $v_a$  or  $v_b$ , if either is present,

or no vertical line at all. It contains no horizontal line. The tile machine layer and the verification machine layer contain no machines, just infinite regions tiled with copies of a single tile.

Suppose then that  $i \leq a < b \leq j$ . On the machine grid layer of  $X_{a,b}$ , we have two vertical lines on the leftmost symbols of  $v_a$  and  $v_b$  with distance  $m > 0$ , and no other vertical lines between them. If  $m \leq m_0$ , then we fill the tile machine layer with blank symbols, and place rows of  $\#$ -symbols at regular intervals of length  $H(m)$  on the verification machine layer. Suppose now  $m > m_0 = 3^{G(n_0)} - 1$ , and let  $n \geq n_0$  be the largest natural number with  $m > 3^{G(n)} - 1$ . On the tile machine layer of  $X_{a,b}$ , we can simulate a copy of  $M$  in each machine grid cell, and they have enough space to perform  $\log_3(m + 1) > G(n)$  computation steps, which suffices for the simulation of the tile set  $T_n$  and the formation of the simulated grid. Furthermore, we have  $m \leq 3^{G(n-1)} - 1$ , and from (2) we obtain  $g(n) \geq H(3^{G(n)} - 1) \geq H(3^{G(n-1)} - 1) \geq H(m)$  since  $G$  and  $H$  are non-decreasing. Thus the distance between two  $\#$ -rows on the verification machine layer of  $X_{a,b}$  can be at least  $H(m)$  cells.

We claim that in both cases, the spaces between these rows can be filled with valid computations of  $M_{a,b}$ . If  ${}^\infty u_{a-1} v_a u_a^{n_a} \cdots v_b u_b^\infty \in P(Y)$ , then the machine scans this word and enters the final state before  $H(m)$  steps. Otherwise, there exists  $p \in \mathbb{N}$  such that  $w = u_{a-1}^p v_a u_a^{n_a} \cdots v_b u_b^p$  does not occur in  $P(Y)$ , and hence does not occur in  $z$ . The machine computes this  $p$ , reads the word of  $z$  containing  $v_a u_a^{n_a} \cdots v_b$ , verifies that it is not equal to  $w$ , and enters the final state before  $H(m)$  steps.

We have shown that these layers form a valid configuration of  $X_{a,b}$  consistent with the image layer  $y$ . Thus  $\phi$  is surjective, and hence  $X$  is a countable SFT cover of  $Y$ . □

## 5 Conclusions

We have provided examples of countable sofic shifts that have no countable SFT covers, the simplest one being Example 1. All of them are non-horizontally constant, and rely on Lemma 2 for the proof of this property. It seems that periodicity is an essential component of our proof technique and cannot be dispensed with easily. Example 4 shows that if we remove the periodicity from Example 1, the resulting sofic shift is countably covered. In fact we don't know whether there exist two-dimensional countable sofic shifts without countable SFT covers where this property does not follow from the existence of a periodic component of some kind.

In [18] it was shown that all effectively closed subsystems of the *distinct square shift* are sofic. The distinct square shift is the two-dimensional binary shift space  $X$  whose configurations consist of square patterns of 1s with disjoint borders and distinct sizes in a background of 0s. It seems likely that some sufficiently complex countable subshifts of  $X$  would have no countable SFT cover, but we are currently unable to prove this.

On the other hand, we do not know whether the countable cover conditions are sufficient for the existence of a countable SFT cover in all non-horizontally constant shift spaces. Let us explicitly state this as an open problem.

*Question 1* Is it true that a non-horizontally constant shift space is a countably covered sofic shift if and only if it satisfies the countable cover conditions?

Theorem 3 provides a positive answer for shift spaces whose rows come from a one-dimensional countable sofic shift, and we have provided examples outside of this class. The construction depends heavily on the simple structure of one-dimensional countable sofic shifts, namely the bounded number of period breakers. We used this property to construct a finite number of semi-independent layers, one for each pair of period breakers, which together verify the correctness of the image configuration. It is not clear whether the construction can be extended to a larger class of shift spaces.

Our proof of Lemma 2 uses Zorn's lemma, and thus relies on some version of the Axiom of Choice. We are not certain whether a different approach could avoid its use in the special case of countable SFTs.

Finally, as stated in the Introduction, this research was motivated by the equal entropy SFT cover problem for multidimensional sofic shifts, and in particular the question whether each zero entropy sofic shift has a zero entropy cover. As our approach relies on periodicity and the topological properties of countable shift spaces, it cannot be directly applied to all zero entropy sofic shifts. Furthermore, the constructions of [1, 6] give zero entropy SFT covers to all vertically constant sofic shifts, so counterexamples cannot be found inside this class.

**Acknowledgements** Open access funding provided by University of Turku (UTU) including Turku University Central Hospital. The author is thankful to Andrei Romashchenko, Bruno Durand and Pierre Guillon for discussions and comments on this article, to Ville Salo for Example 3, and to the anonymous reviewers for their comments.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Aubrun, N., Sablik, M.: Simulation of effective subshifts by two-dimensional subshifts of finite type. *Acta Appl. Math.* **126**(1), 35–63 (2013)
2. Ballier, A., Durand, B., Jeandel, E.: Structural aspects of tilings. In: Albers, P.W.S. (ed.) *Proceedings of the 25th Annual Symposium on the Theoretical Aspects of Computer Science*, p. 11. IBFI Schloss Dagstuhl, Bordeaux (2008)
3. Desai, A.: Subsystem entropy for  $\mathbb{Z}^d$  sofic shifts. *Indag. Math. (N.S.)* **17**(3), 353–359 (2006)
4. Destombes, J., Romashchenko, A.: Resource-Bounded Kolmogorov Complexity Provides an Obstacle to Soficness of Multidimensional Shifts. *Arxiv e-prints* (2018)
5. Durand, B., Levin, L.A., Shen, A.: Complex tilings. *J. Symbol. Log.* **73**, 593–613, 6 (2008)
6. Durand, B., Romashchenko, A., Shen, A.: Fixed-point tile sets and their applications. *J. Comput. System Sci.* **78**(3), 731–764 (2012)



7. Hochman, M.: On the dynamics and recursive properties of multidimensional symbolic systems. *Invent. Math.* **176**(1), 131–167 (2009)
8. Johnson, A., Kass, S., Madden, K.: Projectional entropy in higher dimensional shifts of finite type. *Compl. Syst.* **17**(3), 243–257 (2007)
9. Kari, J.: The nilpotency problem of one-dimensional cellular automata. *SIAM J. Comput.* **21**(3), 571–586 (1992)
10. Kass, S., Madden, K.: A sufficient condition for non-soficness of higher-dimensional subshifts. *Proc. Amer. Math. Soc.* **141**(11), 3803–3816 (2013)
11. Lind, D., Marcus, B.: *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, Cambridge (1995)
12. Minsky, M.L.: *Computation: Finite and Infinite Machines*. Prentice-hall, Inc., Upper Saddle River (1967)
13. Ormes, N., Pavlov, R.: Extender sets and multidimensional subshifts. *Ergodic Theory Dyn. Syst.* **36**(3), 908–923 (2016)
14. Pavlov, R.: A class of nonsofic multidimensional shift spaces. *Proc. Amer. Math. Soc.* **141**(3), 987–996 (2013)
15. Pavlov, R., Schraudner, M.: Classification of sofic projective subdynamics of multidimensional shifts of finite type. *Trans. Amer. Math. Soc.* **367**(5), 3371–3421 (2015)
16. Robinson, R.M.: Undecidability and nonperiodicity for tilings of the plane. *Invent. Math.* **12**, 177–209 (1971)
17. Salo, V., Törmä, I.: Constructions with countable subshifts of finite type. *Fund. Inf.* **126**(2-3), 263–300 (2013)
18. Westrick, L.B.: Seas of squares with sizes from a  $\pi_1^0$  set. *Israel J. Math.* **222**(1), 431–462 (2017)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.