TUCS

Ilkka Törmä

Structural and Computational Existence Results for Multidimensional Subshifts

Turku Centre for Computer Science

TUCS Dissertations
No 200, July 2015

Structural and Computational Existence Results for Multidimensional Subshifts

Ilkka Törmä

To be presented, with the permission of the Faculty of Mathematics and Statistics of the University of Turku, for public criticism in Auditorium Calonia 1 on July 31, 2015, at 12 noon.

University of Turku
Department of Mathematics and Statistics
FI-20014 Turku
Finland

2015

Supervisors

Jarkko Kari
Department of Mathematics and Statistics
University of Turku
FI-20014 Turku
Finland

Reviewers

Emmanuel Jeandel
Lorraine Research Laboratory in Computer Science and its Applications
University of Lorraine
Campus Scientifique - BP 239
F-54506 Vandœuvre-lès-Nancy
France

Stephen G. Simpson
Department of Mathematics
Pennsylvania State University
McAllister Building, Pollock Road
State College, PA 16802
USA

Opponent

Alexander Shen
Department of Computer Science
University of Montpellier
LIRMM
161 Rue Ada
F-34095 Montpellier Cedex 5
France

ISBN 978-952-12-3237-4
ISSN 1239-1883

Abstract

Symbolic dynamics is a branch of mathematics that studies the structure of infinite sequences of symbols, or in the multidimensional case, infinite grids of symbols. Classes of such sequences and grids defined by collections of forbidden patterns are called subshifts, and subshifts of finite type are defined by finitely many forbidden patterns. The simplest examples of multidimensional subshifts are sets of Wang tilings, infinite arrangements of square tiles with colored edges, where adjacent edges must have the same color. Multidimensional symbolic dynamics has strong connections to computability theory, since most of the basic properties of subshifts cannot be recognized by computer programs, but are instead characterized by some higher-level notion of computability.

This dissertation focuses on the structure of multidimensional subshifts, and the ways in which it relates to their computational properties. In the first part, we study the subpattern posets and Cantor-Bendixson ranks of countable subshifts of finite type, which can be seen as measures of their structural complexity. We show, by explicitly constructing subshifts with the desired properties, that both notions are essentially restricted only by computability conditions. In the second part of the dissertation, we study different methods of defining (classes of) multidimensional subshifts, and how they relate to each other and existing methods. We present definitions that use monadic second-order logic, a more restricted kind of logical quantification called quantifier extension, and multi-headed finite state machines. Two of the definitions give rise to hierarchies of subshift classes, which are a priori infinite, but which we show to collapse into finitely many levels. The quantifier extension provides insight to the somewhat mysterious class of multidimensional sofic subshifts, since we prove a characterization for the class of subshifts that can extend a sofic subshift into a nonsofic one.

Tiivistelmä

Symbolidynamiikka on matematiikan ala, joka tutkii äärettömän pituisten symbolijonojen ominaisuuksia, tai moniulotteisessa tapauksessa äärettömän laajoja symbolihiloja. Siirtoavaruudet ovat tällaisten jonojen tai hilojen kokoelmia, jotka on määritelty kieltämällä jokin joukko äärellisen kokoisia kuvioita, ja äärellisen tyypin siirtoavaruudet saadaan kieltämällä vain äärellisen monta kuviota. Wangin tiilitykset ovat yksinkertaisin esimerkki moniulotteisista siirtoavaruuksista. Ne ovat värillisistä neliöistä muodostettuja tiilityksiä, joissa kaikkien vierekkäisten sivujen on oltava samanvärisiä. Moniulotteinen symbolidynamiikka on vahvasti yhteydessä laskettavuuden teoriaan, sillä monia siirtoavaruuksien perusominaisuuksia ei ole mahdollista tunnistaa tietokoneohjelmilla, vaan korkeamman tason laskennallisilla malleilla.

Väitöskirjassani tutkin moniulotteisten siirtoavaruuksien rakennetta ja sen suhdetta niiden laskennallisiin ominaisuuksiin. Ensimmäisessä osassa keskityn tiettyihin äärellisen tyypin siirtoavaruuksien rakenteellisiin ominaisuuksiin: äärellisten kuvioiden muodostamaan järjestykseen ja Cantor-Bendixsonin astelukuun. Halutunlaisia siirtoavaruuksia rakentamalla osoitan, että molemmat ominaisuudet ovat olennaisesti laskennallisten ehtojen rajoittamia. Väitöskirjan toisessa osassa tutkin erilaisia tapoja määritellä moniulotteisia siirtoavaruuksia, sekä sitä, miten nämä tavat vertautuvat toisiinsa ja tunnettuihin siirtoavaruuksien luokkiin. Käsittelen määritelmiä, jotka perustuvat toisen kertaluvun logiikkaan, kvanttorilaajennukseksi kutsuttuun rajoitettuun loogiseen kvantifiointiin, sekä monipäisiin äärellisiin automaatteihin. Näistä kolmesta määritelmästä kahteen liittyy erilliset siirtoavaruuksien hierarkiat, joiden todistan romahtavan äärellisen korkuisiksi. Kvanttorilaajennuksen tutkimus valottaa myös niin kutsuttujen sofisten siirtoavaruuksien rakennetta, jota ei vielä tunneta hyvin: kyseisessä luvussa selvitän tarkasti, mitkä siirtoavaruudet voivat laajentaa sofisen avaruuden ei-sofiseksi.

Acknowledgements

I'll begin by thanking my advisor, Professor Jarkko Kari, who has given me both excellent guidance and a substantial amount of freedom in conducting my PhD research. As an undergraduate student, his lectures on cellular automata and tiling systems sparked my interest in the mathematical research of these topics, and at the time I was beginning to write my Master's thesis, he gave me the opportunity of joining his research group as an intern. Doing research under Jarkko has been an amazing experience, and I have learned a lot from him.

Secondly, I'd like to thank Ville Salo, my colleague, co-author, and friend. I started my PhD studies one year after Ville, and for most of that time I have shared an office with him. The majority of our research has been a joint effort, and many fun days have passed while we figuratively banged our heads together over various problems. Without Ville, this dissertation would be quite different.

Several other people have also had an influence on my research. I'd like to thank Charalampos "Babis" Zinoviadis, also an occupant of our office, for introducing me to several interesting research directions, and for moral support during my studies. Thanks are due to Jyrki Lahtonen and Roope Vehkalahti for engaging seminar sessions on group theory and other topics, and for being fun people in general. Professors Juhani Karhumäki and Luca Zamboni introduced me to the exciting and useful fields of computational complexity and combinatorics on words. My knowledge of topology and measure theory is largely due to the courses of Professor Kari Ylinen, who also showed me the importance of mathematical rigor. I've had the pleasure to discuss mathematics and other things with several scholars from French universities that have visited or been a part of our research group, including Pierre Guillon (who suggested me the topic of my Master's thesis), Timo Jolivet, Nathalie Aubrun and Alexis Ballier. I'd also like to thank Professor Michael Schraudner for inviting me to visit the Center of Mathematical Modeling in University of Chile, and all the great people I met there.

Of course, I'm grateful to Professor Emmanuel Jeandel and Professor Stephen G. Simpson for reviewing my dissertation, and to Professor Alexander Shen for agreeing to act as my opponent.

An essential part of my studies has been the friendly atmosphere of the Department of Mathematics and Statistics (which was called just the Department of Mathematics when I started there). I owe a thanks to all my colleagues there, especially the administrative staff, Tuire and Sonja, who made my life much easier by reminding me of all the important stuff I tend to forget.

I'm thankful to Turku Centre for Computer Science, University of Turku, Turku University Foundation, and Nokia Foundation for providing financial support during my PhD studies.

Last but not least, I want to thank all my friends and family for their support. I'm grateful to my mother for everything she has done for me, and to Juho and Sampo for keeping my feet on the ground. A special thanks goes to Suvi, my spouse and best friend, for being the co-author of our everyday life and many adventures.

Turku, June 2015
Ilkka Törmä

Contents

Chapter 1

Introduction

1.1 Multidimensional Symbolic Dynamics

Consider a system of some kind that evolves with time, like a swinging pendulum or traffic flowing through a city, whose behavior we wish to analyze. Most of the time, we cannot describe the system perfectly, but can only observe some parts of it; also, the system probably cannot be monitored constantly, but only at certain intervals. One way of studying the evolution of the system is to divide its possible states into some finitely many classes, and recording its class at certain intervals. For example, we could describe the traffic flow at a given time as 'low', 'normal' or 'high', and write down this description once every 10 minutes. Given enough observations, this sequence may then provide us with a wealth of information about the traffic system. This method of encoding the evolution of a dynamical system into a sequence of discrete values is the motivation behind the field of mathematics known as *(one-dimensional) symbolic dynamics*. The associated theory deals with efficient encodings of different systems, and the intrinsic properties of the symbolic sequences themselves.

The field of symbolic dynamics is often said to originate from the 1898 article of Hadamard [Had98], where he studies the properties of certain geometrical objects called geodesics by encoding them into sequences of symbols. In 1938 and 1940, Morse and Hedlund published a two-part article titled 'Symbolic Dynamics' [MH38, MH40], where one-dimensional symbolic dynamics was treated from an abstract point of view, focusing mostly on dynamical notions of symbolic sequences. Hedlund's influential article [Hed69] on certain natural functions between *subshifts of finite type (SFTs)*, which are classes of sequences defined by finitely many forbidden patterns that they must avoid, took a more combinatorial approach. The field has since grown significantly, and different classes of subshifts have been studied from the dynamical, algebraic and computational viewpoints by numerous au-

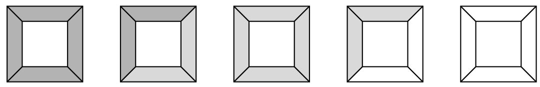

Figure 1.1: A set of five Wang tiles with three colors: white, light gray, and dark gray.

thors. The books [LM95, Kit98, Kůr03] provide a comprehensive overview of the subject.

Multidimensional symbolic dynamics is a generalization of the theory from sequences of symbols to *tilings on multidimensional grids*. The simplest and most intuitive examples of multidimensional symbolic dynamical systems are given by sets of *Wang tiles*. A Wang tile is a square whose four edges are colored by some colors drawn from a finite set of choices; an example is shown in Figure 1.1. We can place two such tiles next to each other, if their adjacent edges have the same color. If we have a suitable collection of Wang tiles, we can cover a large region with them, but it may also be that the coloring forces every sufficiently large patch to contain a mismatched pair of tiles. This concept was introduced by Hao Wang in [Wan61] for the purpose of encoding certain problems in mathematical logic into problems of geometry. In the article, he stated the famous *tiling problem*, sometimes known as the *domino problem*: given a set of Wang tiles, decide whether they can tile the infinite plane (or equivalently, an arbitrarily large square region). In other words, the tiling problem asks whether the subshift of finite type defined by forbidding tiles with mismatched edges is nonempty. It was proved by Berger in [Ber66] (and simplified by Robinson in [Rob71]) that the tiling problem is *undecidable*, or that it cannot be solved by a fixed algorithm. The result also shows that there exist sets of Wang tiles that only produce *aperiodic* tilings. This is something that cannot happen in the one-dimensional setting, since any nonempty one-dimensional subshift of finite type contains a periodic sequence. The quest for finding the smallest set of Wang tiles that only produces aperiodic tilings has attracted some attention in the past. Berger's original tileset consisted of thousands of tiles, Robinson's simplified solution only requires 56 tiles, and the current record is 13, achieved in [Cul96] with a modification of the set of 14 tiles of [Kar96]. Very recently, an aperiodic tileset of size 11 was found, but the result has not been published yet. This settles the question of the aperiodic tileset of minimal size, since the same group also proved that all tilesets of size at most 10 either do not tile the plane, or produce a periodic tiling [JR15].

Because of the added freedom provided by additional 'directions', multidimensional subshifts are generally speaking much harder to analyze than their one-dimensional counterparts. As a simple example, consider the one-dimensional *golden mean shift*, which is the set of those infinite sequences of zeros and ones where two ones cannot occur next to each other. The first of the two sequences

$$\cdots 0010100011010010 \cdots$$

and

$$\cdots 0010100001010010 \cdots$$

is not a member of the golden mean shift, since it contains the forbidden pattern $1\,1$, but the second one is (unless, of course, $1\,1$ occurs somewhere outside the shown segment). To every subshift we can associate a nonnegative real number, called its *topological entropy*, that in some sense measures its size and complexity. More explicitly, the entropy of a subshift is the asymptotic growth rate of the number of patterns occurring in it. The entropy of the golden mean shift is not very hard to compute explicitly using just elementary combinatorics and calculus, and its value is exactly $\log \frac{1+\sqrt{5}}{2}$, the logarithm of the golden mean $\phi = \frac{1+\sqrt{5}}{2}$ (which gives the subshift its name). On the other hand, consider the *two-dimensional golden mean shift*, also known as the *hard squares model*, which is the set of those infinite two-dimensional grids where every grid square has been labeled by either 0 or 1, and neither of the two-dimensional patterns $\begin{smallmatrix}1 & 1\end{smallmatrix}$ and $\begin{smallmatrix}1 \\ 1\end{smallmatrix}$ occurs anywhere. An example of a member of the two-dimensional golden mean shift shift is the following grid:

$$\begin{array}{ccccccccccc}
 & & & & & \vdots & & & & & \\
 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & \\
 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \\
 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
\cdots & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots \\
 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & \\
 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & \\
 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \\
 & & & & & \vdots & & & & &
\end{array}$$

Again, we assume that neither of the forbidden patterns occurs anywhere outside the finite rectangle shown here. Even though the two-dimensional golden mean shift is one of the simplest multidimensional subshifts, with only two forbidden patterns of size 2, no exact formula for its entropy is currently known; in fact, it is an open problem whether such a formula even exists! This subshift has been thoroughly investigated in the past. For example, it is known from [Bax99] that its entropy is approximately

$$1.503048082475332264322066329475553689385781\,0$$

and in [Pav12], it was proved that good approximations for its entropy are relatively easy to produce by a computer program.

Another difficult open problem in multidimensional symbolic dynamics is a possible characterization of *sofic shifts*. A subshift is sofic, if it can be obtained from an SFT by re-labeling its tilings using a local rule. For example, the one-dimensional golden mean shift can be re-labeled by the rules $0\,0 \mapsto 1$, $0\,1 \mapsto 0$ and $1\,0 \mapsto 0$ to create a sofic shift known as the *even shift*. As the name implies, the even shift is the set of sequences where there are an even number of 0's between any two consecutive 1's, and because of their long-range constraints, the even shift is not an SFT. For example, the sequence

$$\cdots 1000011100001100 \cdots$$

is a member of the even shift. It is known that one-dimensional sofic shifts are exactly those whose admissible patterns can be described by a regular language (see Section 4 of [LM95]), which implies that they are computationally very simple and (for the most part) easy to analyze. By contrast, the class of two-dimensional sofic shifts is still badly understood. There are several nontrivial conditions that every multidimensional sofic shift necessarily satisfies [Des06, HM10, KM13, Pav13], and conversely, some conditions that guarantee soficness [Moz89, DRS10, AS13, OP15], but none of them gives a complete characterization for the class.

1.2 Connections to Computability Theory

The connection of multidimensional symbolic dynamics to computability and logic has remained strong, and many results on one-dimensional subshifts have multidimensional versions with a computational conclusion instead of an algebraic of combinatorial one. For example, the topological entropy of a one-dimensional subshift of finite type is always of the form $r \log \alpha$, where $r \in \mathbb{Q}_{\geq 0}$ is a nonnegative rational number and $\alpha > 1$ is a *Perron number*, an algebraic integer largest in absolute value among its algebraic conjugates [Lin84]. Conversely, every such number is the entropy of some one-dimensional SFT. On the other hand, the entropies of multidimensional SFTs are exactly the *right recursively enumerable* nonnegative real numbers, that is, the limits of recursively enumerable descending sequences of rational numbers [HM10]. It was earlier proved in [HKC92] that the entropy of a given one-dimensional cellular automaton, and thus a two-dimensional SFT, is uncomputable even up to a constant error. This means that even the mysterious two-dimensional golden mean shift mentioned above is much better behaved than multidimensional subshifts in general. The uncomputability of entropy and the undecidability of the tiling problem are examples of the more general phenomenon that most properties of multidimensional SFTs and sofic shifts cannot be checked effectively.

Another example of the computational nature of multidimensional subshifts is given by the geometric notion of *projective subdynamics*. The projective subdynamics of a two-dimensional subshift is the collection of all horizontal rows that occur in its infinite grids. It is always a one-dimensional subshift, and its structure may or may not reflect that of the original two-dimensional subshift. For example, it is not very hard to see that the projective subdynamics of the two-dimensional golden mean shift is exactly the one-dimensional golden mean shift, so in this case the two subshifts are very closely related. In general, however, the projective subdynamics of two-dimensional sofic shifts can be extremely complicated, since they are restricted only by a computational condition. Namely, it was independently proved in [DRS10] and [AS13], following a similar but weaker result in [Hoc09], that a one-dimensional subshift can be realized as the projective subdynamics of a two-dimensional sofic shift if and only if there is a computer program that outputs an infinite sequence of forbidden patterns defining it. The projective subdynamics of two-dimensional SFTs are computationally equally complex, but there are some additional geometric restrictions [PS14, Gui12].

The phenomena of having computational characterizations for properties of multidimensional subshifts, and of most of their properties being undecidable, are due to the fact that it is very easy to embed arbitrary computation into a multidimensional SFT. For example, most proofs of the undecidability of the tiling problem reduce it to the halting problem of Turing machines by constructing a set of Wang tiles that simulates the computation of a given Turing machine, and produces an infinite valid tiling only if the machine never halts. Usually, the most difficult aspect of such constructions is to guarantee that every valid tiling contains a simulated computation, and this is achieved by forcing longer and longer computations to occur at every region of a tiling.

1.3 The Structure of This Dissertation

As the title suggests, the main results of this dissertation are constructions of multidimensional subshifts with various structural and computational properties. Chapter 2 presents the notation and mathematical formalism used here. We formally define subshifts and their topological and combinatorial structure, together with different notions of computability and the ways in which they relate to symbolic dynamics. We also present some simple examples of the concepts. The chapter is meant to be mostly self-contained, although we present some basic results of symbolic dynamics without proof.

Chapter 3 contains a collection of results that we use repeatedly through the course of this dissertation. Many of them are original, others have been collected from various sources. Section 3.1 contains more examples of simple subshifts, and its purpose is to give some intuition on the structural concepts of *determinism*, the *bounded signal property*, and countability. Intuitively, a two-dimensional subshift is deterministic if the upper half of any of its tilings uniquely determines the lower half.[1] For example, the set of Wang tiles in Figure 1.1 defines a deterministic subshift, since all valid tilings consist of infinite 'stripes' stretching from southwest to northeast. The bounded signal property, on the other hand, means that the set of horizontal rows occurring in the tilings of the subshift is structurally simple (in a certain precise sense), and countability means that the number of tilings in the subshift is at most countably infinite. Especially in the context of multidimensional SFTs, countability is a very interesting and nontrivial restriction, and we study it more thoroughly in Chapter 4. Section 3.2 contains some examples and general results on the *Cantor-Bendixson derivative*, which is a notion arising from abstract topology that can be applied to subshifts. Section 3.3 contains our first construction of a complicated subshift: the simulation of arbitrary counter machines in countable SFTs. It provides us with a flexible 'template' for embedding computation in SFTs and sofic shifts that we can modify as needed, and we make use of some variant of the construction in every chapter of this dissertation. Finally, in Section 3.4 we discuss multidimensional sofic shifts, in particular the few known necessary conditions for a subshift to be sofic.

The following four chapters are based on the original publications [ST13, Tör14b, Tör14a, ST14], but they have been extended and heavily modified to fit the common theme and formalism of this dissertation. In Chapter 4, we study the structural properties of countable SFTs and sofic shifts, concentrating on the aforementioned Cantor-Bendixson derivative, the related Cantor-Bendoxson rank, and the *subpattern poset*. The subpattern poset is a partial ordering of the tilings of a subshift based on the finite patterns they contain: tilings with many distinct patterns are 'greater' than simpler ones. The subpattern poset and Cantor-Bendixson derivative operation are connected in subtle ways, and especially in the countable case, both can be seen as indicators of the structural complexity of a given subshift. Section 4.2 is an extended introduction to the topic through the simpler case of one-dimensional subshifts. In Section 4.3, we investigate the Cantor-Bendixson ranks of countable SFTs, and the computational complexity of their derivatives. The original content of this section consists of the characterization of the set of possible ranks of countable SFTs, and the minimal rank of

[1] Recall that a two-dimensional subshift is simply a certain kind of collection of infinite tilings.

a countable SFT containing an uncomputable tiling. Some related upper and lower bounds were obtained earlier in [BDJ08, JV11, ST13, BJ13] using various constructions, the main idea of which has been to find the structurally simplest way of embedding computation into a countable SFT. In our construction, we use an enhanced version of a finite state machine with one counter, which gives an optimal solution, at least from the point of view of the Cantor-Bendixson rank. In Section 4.4, we direct our interest to subpattern posets of countable SFTs and sofic shifts. In the most involved construction of this dissertation, we give a near-characterization for the possible subpattern posets of countable SFTs and sofic shifts.[2] As is usually the case with properties of multidimensional subshifts, it turns out that the characterization is computational, so a vast collection of partially ordered sets can be realized as subpattern posets. Finally, in Section 4.5, we show that the subpattern posets of those countable SFTs that possess the bounded signal property are much more restricted.

Chapters 5 and 6 are variations of the same idea: defining multidimensional subshifts using logical operations. In Chapter 5, we consider logical formulas that express local or global restrictions on tilings, and define their associated subshifts as the sets of those tilings that satisfy the restrictions. Our approach is analogous to defining classes of finite graphs by logical formulas such as $\forall \mathsf{v} \exists \mathsf{w}\ \mathsf{e}(\mathsf{v}, \mathsf{w})$, where the variables v and w represent vertices, and the binary relation e denotes the existence of an edge; this formula corresponds to the class of directed graphs without sink vertices. Intuitively, we define classes of tilings by quantifying over the tilings of other subshifts, and checking a local (or at least finitary) condition. We continue the research started in [JT13], and study certain hierarchies of formulas obtained by counting quantifier alternations. The main results of the chapter state that the hierarchies collapse to at most four distinct levels, the highest of which has a simple computational characterization. In other words, every formula $\exists \mathsf{x}_1 \forall \mathsf{x}_2 \cdots \exists \mathsf{x}_n\ \phi$ is equivalent to one of the form $\forall \mathsf{x} \exists \mathsf{y}\ \psi$, where the variables denote tilings, in the sense that they define the same subshift. We also obtain characterizations for the lower classes of the hierarchies.

In Chapter 6, we investigate the effect of quantification on the class of sofic shifts in the slightly more concrete context of *quantifier extensions*, which produce new subshifts from existing ones through universal and existential quantification. The concept is a generalization of the *multi-choice shift spaces* in [LMP13]: while multi-choice shift spaces intuitively represent the condition "the tiles at *these* coordinates can be chosen freely", quantifier extensions correspond to the more general "the tiles at *these* coordinates can

[2] The word *near-characterization* here means the following. We show that the subpattern posets of multidimensional countable SFTs are exactly those partially ordered sets that satisfy a certain computability condition, up to a collection of 'degenerate' elements whose behavior we can mostly control.

be chosen freely, as long as *these* additional restrictions are respected". In the main result of the section, we show that for all nontrivial instances of the 'additional restrictions', the class of multidimensional sofic shifts is not closed under the universal quantifier extension. In some special cases, this can be proved by a computability-theoretic argument, but the general result requires combinatorial tools.

In Chapter 7, we study another unusual method of defining multidimensional subshifts, this time using automata theory. We define a class of finite state machines, called *plane-walking automata*, that can walk freely on tilings, and may reject it by entering a special rejecting state. The collection of tilings that the machine never rejects, no matter where it is initialized, forms a subshift. This is somewhat similar to the definition of regular languages using two-way finite automata, except that we view infinite computations as accepting. We study the hierarchy of subshift classes obtained by increasing the number of heads of the plane-walking automata. This has been studied before in [DM02], but with a slightly different acceptance model, and using immobile pebbles instead of several mobile heads. In view of the results of Chapter 5, the reader may not be surprised that the hierarchy we define collapses to the class of three-headed automata in all dimensions, and that this class has a computational characterization, which is due to the fact that three-headed finite automata can simulate arbitrary computation. We also prove some limitations of one- and two-headed automata, which depend heavily on the dimension of the tiling they are walking on.

Finally, Chapter 8 contains some reflections on our results and possible future directions. In particular, we discuss the nature of the collapsing hierarchies in Chapters 5 and 7, and possible generalizations of our results to the more abstract context of *subshifts on groups*, as well as state several open problems.

Chapter 2

Definitions

2.1 Multidimensional Symbolic Dynamics

In this chapter, we present the notation and formalism used in this dissertation. It is assumed that the reader is familiar with elementary set theory, topology, automata theory, and computability theory, but other than that, the presentation is largely self-contained. We adopt the convention that the set of natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$ contains 0.

To begin with, let A be a finite set of *symbols* or *letters*, called the *alphabet*, endowed with the discrete topology. The set of finite *words* over A is denoted A^*, and the length of a word $w \in A^*$ by $|w|$. The empty word of length 0 is denoted by ϵ. Fix an integer dimension $d \geq 1$. A *d-dimensional pattern over S* is a pair $P = (D, s)$, where $D = D(P) \subset \mathbb{Z}^d$ is the *domain* of P, and $s : D \to A$ is the *arrangement* of symbols in it. We denote $P_{\vec{v}} = s(\vec{v})$ for $\vec{v} \in D$. The restriction of P to a smaller domain $E \subset \mathbb{Z}^d$ is denoted $P|_E$. The set of d-dimensional finite patterns over A is denoted $\mathcal{P}_d(A)$, and the set of patterns with domain $D \subset \mathbb{Z}^d$ is denoted A^D. A full pattern with domain $\mathbb{Z}^d$ is called a *configuration*, and the set $A^{\mathbb{Z}^d}$ of all configurations is called the *d-dimensional full shift on A*. A configuration $x \in A^{\mathbb{Z}^d}$ is *uniform* if there exists $a \in A$ with $x_{\vec{n}} = a$ for all $\vec{n} \in \mathbb{Z}^d$. For a dimension $d' < d$, a d'-dimensional vector $\vec{v} \in \mathbb{Z}^{d'}$ and a d-dimensional configuration $x \in A^{\mathbb{Z}^d}$, we denote by $x_{\vec{v}}$ the restriction of x to the hyperplane $\mathbb{Z}^{d-d'} \times \{\vec{v}\}$, that is, the $(d - d')$-dimensional configuration $y \in A^{\mathbb{Z}^{d-d'}}$ defined by $y_{\vec{n}} = x_{(\vec{n},\vec{v})}$ for all $n \in \mathbb{Z}^{d-d'}$. In particular, if $x \in A^{\mathbb{Z}^2}$ and $i \in \mathbb{Z}$, then $x_i \in A^{\mathbb{Z}}$ is the i'th horizontal row of x.

We make the full shift a topological space by giving it the product topology. A clopen base for it is given by *cylinder sets*, which are sets of the form $\{x \in A^{\mathbb{Z}^d} \mid x|_{D(P)} = P\}$ for finite patterns $P \in \mathcal{P}_d(A)$. The clopen subsets of $A^{\mathbb{Z}^d}$ are exactly the finite unions of cylinder sets. When we want to stress the topological properties of configurations, we may call them *points* of $A^{\mathbb{Z}^d}$.

For a pattern P over A and $\vec{n} \in \mathbb{Z}^d$, we define the *translation of P by $\vec{n}$*, denoted $\sigma^{\vec{n}}(P)$, as the pattern with domain $D(P) - \vec{n}$ and arrangement function $\vec{v} \mapsto P_{\vec{v}-\vec{n}}$. The restriction $\sigma^{\vec{n}} : A^{\mathbb{Z}^d} \to A^{\mathbb{Z}^d}$ of $\sigma^{\vec{n}}$ to the full shift is simply called the *translation by $\vec{n}$*, and it defines an action of $\mathbb{Z}^d$ on $A^{\mathbb{Z}^d}$. If $d = 1$, this is also called the *shift action*. For a configuration $x \in A^{\mathbb{Z}^d}$, we denote its *orbit* by $\mathcal{O}(x) = \{\sigma^{\vec{n}}(x) \mid \vec{n} \in \mathbb{Z}^d\}$. For two patterns P and Q, we say that P *occurs in Q at $\vec{n} \in \mathbb{Z}^d$* if $\sigma^{\vec{n}}(Q)_{D(P)} = P$. If P occurs in Q at some coordinate, we denote $P \sqsubset Q$.

Remark 2.1. We sometimes regard two finite patterns equal if they are translates of each other, and sometimes not. The choice should always be clear from the context. Similarly, the alphabet A is sometimes identified with $A^{\{\vec{0}\}}$, the set of singleton patterns.

In the one-dimensional setting $d = 1$, where configurations are bi-infinite sequences of symbols, the shift action defines a *dynamical system* on the full shift. Intuitively, if the shift action is applied to such a sequence repeatedly, we can think of it as shifting to the left at a steady pace of, say, one step each second. This intuition is especially fitting, if the sequence was obtained by observing another system at certain time intervals, and recording its behavior. In the multidimensional case, we have more than one degree of freedom, so it make less sense to think of the full shift as evolving in time, which is why we think of σ as a spatial translation operation, instead of a temporal one.

A *d-dimensional subshift over A* is a topologically closed subset $X \subset A^{\mathbb{Z}^d}$ satisfying $\sigma^{\vec{n}}(X) = X$ for all $\vec{n} \in \mathbb{Z}^d$. Alternatively, all subshifts X can be defined by a set $\mathcal{F} \subset \mathcal{P}_d(A)$ of *forbidden patterns* as $\mathcal{X}_{\mathcal{F}} = \{x \in A^{\mathbb{Z}^d} \mid \forall P \in \mathcal{F} : P \not\sqsubset x\}$. If $\mathcal{F}$ is finite, then $\mathcal{X}_{\mathcal{F}}$ is a *subshift of finite type* (SFT for short), and if the domain of every pattern of $\mathcal{F}$ is of the form $\{\vec{0}, \vec{e}_i\}$, where $e_1, \ldots, e_d$ is the natural basis of $\mathbb{Z}^d$, then $\mathcal{X}_{\mathcal{F}}$ is a *tiling system*. The *language* of a subshift $X \subset A^{\mathbb{Z}^d}$ is $\mathcal{B}(X) = \{P \in \mathcal{P}_d(A) \mid x \in X, P \sqsubset x\}$, the set of finite patterns occurring in its configurations. For a domain $D \subset \mathbb{Z}^d$, we denote $\mathcal{B}_D(X) = \mathcal{B}(X) \cap A^D$, and the set of symbols occurring in X is denoted $\mathcal{A}(X) = \mathcal{B}_{\{\vec{0}\}}(\mathsf{X})$. If $d = 1$ and $n \in \mathbb{N}$, we also denote $\mathcal{B}_n(X) = \mathcal{B}(X) \cap A^n$. The *orbit closure* $\overline{\mathcal{O}(x)}$ of a configuration $x \in A^{\mathbb{Z}^d}$ is a subshift, and we denote $\mathcal{B}(x) = \mathcal{B}(\overline{\mathcal{O}(x)})$. A subshift $X \subset A^{\mathbb{Z}^d}$ is *strongly irreducible (with constant $M > 0$)* if for any two domains $D, D' \subset \mathbb{Z}^d$ such that $\min\{\|\vec{n} - \vec{m}\| \mid \vec{n} \in D, \vec{m} \in D'\} \geq M$ where $\| \cdot \|$ denotes the maximum norm, and any two patterns $P \in \mathcal{B}_D(X)$ and $P' \in \mathcal{B}_{D'}(X)$, there exists a configuration $x \in X$ with $x|_D = P$ and $x|_{D'} = P'$.

Let X and Y be d-dimensional subshifts, possibly over different alphabets. A *block map* is a continuous function $f : X \to Y$ which intertwines the shift maps on X and Y: $f \circ \sigma^{\vec{n}}|_X = \sigma^{\vec{n}}|_Y \circ f$ for all $\vec{n} \in \mathbb{Z}^d$. Alternatively, a block map f can be defined by a *local function* $F : \mathcal{B}_D(X) \to \mathcal{A}(Y)$ by

$f(x)_{\vec{n}} = F(x|_{D+\vec{n}})$ for all $x \in X$ and $\vec{n} \in \mathbb{Z}^d$, where $D \subset \mathbb{Z}^d$ is a finite domain, called the *neighborhood* of f [Hed69]. Block maps with neighborhood $\{\vec{0}\}$ are called *symbol maps*. An image of a subshift under a block map is a subshift, and images of SFTs are called *sofic shifts*. Two subshifts are called *conjugate* if there is a bijective block map between them. It is known that every SFT is conjugate to a tiling system, and every sofic shift is the image of a tiling system under a symbol map. Block maps are the 'natural' morphisms between subshifts, and we treat conjugate subshifts as essentially equivalent. In particular, conjugacies respect the properties of being an SFT and being a sofic shift, as well as most interesting notions of computability.

Example 2.2. Let $A = \{0, 1\}$, and let $\mathcal{F} \subset \mathcal{P}_2(A)$ be the set of patterns where the letter 1 occurs twice. Then $\mathcal{X}_\mathcal{F} \subset A^{\mathbb{Z}^2}$ is the set of two-dimensional configurations containing at most one letter 1. This subshift, or some version of it, is sometimes called the *sunny side up shift* (see [PS14]). It is a sofic shift.

A famous example of an SFT is the *two-dimensional golden mean shift* on the same alphabet, defined by the forbidden patterns $\begin{smallmatrix} 1 & 1 \end{smallmatrix}$ and $\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}$. Since the patterns are of size 2×1 and 1×2, respectively, X is even a tiling system. In its configurations, no two letters 1 can be adjacent, but there are no other restrictions.

For a subshift $X \subset A^{\mathbb{Z}^d}$, the block maps from X to itself are called *cellular automata*. The set of *limit spacetime diagrams* of a cellular automaton $f : X \to X$ is the $(d + 1)$-dimensional subshift

$$\{x \in A^{\mathbb{Z}^{d+1}} \mid \forall i \in \mathbb{Z} : x_i \in X \wedge x_i = f(x_{i+1})\}.$$

The reason we do not call such configurations simply the spacetime diagrams of f is that f might not be surjective, and in this case only rows with an infinite chain of preimages appear in limit spacetime diagrams. The set of such rows is usually called the *limit set* of f.

To define a one-dimensional SFT or sofic shift, instead of supplying the forbidden patterns, we usually use the notation

$$\mathcal{B}^{-1}(L) = \{x \in A^{\mathbb{Z}} \mid \forall r \in \mathbb{N} : \exists u \in L : x_{[-r,r]} \sqsubset u\}$$

where $L \subset A^*$ is a regular language. As the notation x_i for $x \in A^{\mathbb{Z}^2}$ and $i \in \mathbb{Z}$ implies, we sometimes think of a two-dimensional configuration as being a one-dimensional configuration with legal rows as the alphabet. In particular, we may use the notation $x_{[i,j]}$ to extract a finite list of rows from x.

See [LM95, Section 13.10] for a short survey on multidimensional symbolic dynamics.

2.2 Some Discrete Geometry

We call a vector $(i, j) \in \mathbb{Z}^2$ with $\gcd(i, j) = 1$ a *direction*, and we denote by $SL_d(\mathbb{Z})$ the restriction of the special linear group $SL_d(\mathbb{R})$ to those functions that map $\mathbb{Z}^d$ bijectively to itself. In the two-dimensional case, $SL_2(\mathbb{Z})$ is the group 2×2 integer matrices of determinant ± 1. It follows from Bezout's identity that for any two directions $\vec{d}, \vec{e} \in \mathbb{Z}^2$, there exists an element $L \in SL_2(\mathbb{Z})$ with $L(\vec{d}) = \vec{e}$. For a configuration $x \in A^{\mathbb{Z}^d}$ and $L \in SL_d(\mathbb{Z})$, we define $L(x) \in S^{\mathbb{Z}^d}$ by $L(x)_{\vec{n}} = x_{L^{-1}(\vec{n})}$ for all $\vec{n} \in \mathbb{Z}^d$, and for a subshift $X \subset A^{\mathbb{Z}^d}$, we define $L(X) = \{L(x) \mid x \in X\}$. From the linearity of L it follows that $L(X)$ is also a subshift. We fix a special element $L_{\frac{\pi}{2}} = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right) \in SL_2(\mathbb{Z})$ which performs a counterclockwise rotation by $\frac{\pi}{2}$. For simplicity, we denote $L_{\frac{\pi}{2}}^2 = L_\pi$.

We say that a subshift $X \subset A^{\mathbb{Z}^d}$ *has a period* if there exists a vector $\vec{n} \in \mathbb{Z}^d \setminus \{(0,0)\}$ such that $x = \sigma^{\vec{n}}(x)$ holds for all $x \in X$, and $\vec{n}$ is then called a *period* of X. If X has a period $\vec{n}$, and every period it has is a rational multiple of $\vec{n}$, then we say X is *singly periodic*. If X has d periods which are linearly independent over $\mathbb{R}$, then we say X is *totally periodic*. We define periods, single periodicity and total periodicity of individual configurations analogously. If a configuration has no period, it is *aperiodic*, and a subshift is aperiodic is all of its configurations are. A one-dimensional configuration $y \in A^{\mathbb{Z}}$ is periodic if $y = \sigma^p(y)$ for some $p > 0$, and we say y is *eventually periodic* if for some $n_0, p > 0$, we have $y_i = y_{i+p}$ for all $i > n_0$ and $y_i = y_{i-p}$ for all $i < -n_0$. A two-dimensional configuration is horizontally (vertically) eventually periodic if its rows (columns, respectively) are eventually periodic.

We define the *projective subdynamics* of a subshift $X \subset A^{\mathbb{Z}^2}$ as the set of horizontal rows appearing in its configurations: $\mathrm{Proj}(X) = \{x_0 \mid x \in X\}$. For a general direction $\vec{d} \in \mathbb{Z}^2$, the *$\vec{d}$-projective subdynamics* $\mathrm{Proj}_{\vec{d}}(X)$ of X is defined as $\mathrm{Proj}(L(X))$, where $L \in SL_2(\mathbb{Z})$ is such that $L(L_{\frac{\pi}{2}}(\vec{d})) = (0,1)$. This means that we examine the contents of discrete lines perpendicular to $\vec{d}$, and in particular, $\mathrm{Proj}(X) = \mathrm{Proj}_{(0,1)}(X)$. For a configuration $x \in S^{\mathbb{Z}^2}$, we define $\mathrm{Proj}(x) = \{x_i \mid i \in \mathbb{Z}\}$ as the set of rows of x, and generalize this to $\mathrm{Proj}_{\vec{d}}(x)$ analogously.

We say that a subshift $X \subset A^{\mathbb{Z}^2}$ has the *bounded signal property* in direction $\vec{d}$, if for some subshift $Y \subset B^{\mathbb{Z}^2}$ conjugate to X, there exists a countable one-dimensional SFT $Z \subset B^{\mathbb{Z}}$ such that $\mathrm{Proj}_{\vec{d}}(Y) \subset Z$ (see Lemma 3.1). When the direction is not given, the default value $\vec{d} = (0,1)$ is assumed. Intuitively, having the bounded signal property in some direction means that the subshift can send information in that direction only using a bounded number of signals. The bounded signal property in a fixed direction is, by definition, invariant under conjugacy. However, it is certainly possible that the rows of X are not contained in a countable SFT but those of Y

are, while X and Y are conjugate. In our constructions, when we state that a subshift we construct has the bounded signal property, we always mean that its set of rows is actually contained in a countable SFT.

Let $X \subset A^{\mathbb{Z}^2}$ be a two-dimensional subshift, and denote the upper half-plane by $H = \{(i, j) \mid i \in \mathbb{Z}, j \geq 1\}$. We say that X is *southward deterministic*, if whenever $x, y \in X$ are such that $x|_H = y|_H$, we have $x = y$. If X is an SFT defined by a finite set $\mathcal{F} \subset \mathcal{P}_d(A)$ of forbidden patterns, we say X is *extendably southward deterministic*, if it is southward deterministic, and whenever $x \in A^{\mathbb{Z}^2}$ is such that no pattern of $\mathcal{F}$ occurs in $x|_H$, there exists $y \in X$ with $x|_H = y|_H$. Note that extendable southward determinism depends on the set $\mathcal{F}$, and is not necessarily preserved by conjugacy. By the translation invariance of X and a compactness argument, south determinism is equivalent to the existence of an integer $i \geq 1$ such that $x_{\vec{n}} = y_{\vec{n}}$ for all $\vec{n} \in [-i, i] \times [1, i]$ implies $x_{\vec{0}} = y_{\vec{0}}$, when $x, y \in X$. We say X is *deterministic in the direction* $\vec{d}$ if $L(X)$ is southward deterministic, where $L \in SL_2(\mathbb{Z})$ is such that $L(L_{\frac{\pi}{2}}(\vec{d})) = (1, 0)$. Note that for Wang tiles, it is common to define northwest determinism as the property that a tile is uniquely defined by its north and west neighbors, whereas we would call the corresponding SFT southeast deterministic, since the tiles to the southeast of a known half plane are determined. The definitions of determinism and the bounded signal property via orthogonal directions are similar to the notion of *slicing* as defined in [DFW13], although we consider SFTs rather than cellular automata.

Remark 2.3. The definitions of projective subdynamics, the bounded signal property and determinism could be extended to arbitrary dimensions, but for simplicity, we only use them in the two-dimensional context.

2.3 Partial Orders and Cantor-Bendixson Derivatives

By a *preorder* we mean a reflexive and transitive binary relation on a set P. If the preorder is antisymmetric, then P with this order is called a *poset*, or a partially ordered set. If it is also total, then P is a *chain*. From any preorder $\leq$ on a set P, we obtain a poset $(\tilde{P}, \leq)$ where $\tilde{P}$ is the set of $\leq$-equivalence classes of P (that is, the equivalence classes of the equivalence relation $(p \leq q) \wedge (q \leq p)$) and $\leq$ is the natural induced order among these classes. We often use the elements of P and their $\leq$-equivalence classes interchangeably.

We obtain a natural preorder on the full shift $A^{\mathbb{Z}^d}$ by defining $x \leq y$ by $\mathcal{B}(x) \subset \mathcal{B}(y)$. This is equivalent to $\overline{\mathcal{O}(x)} \subset \overline{\mathcal{O}(y)}$. We denote the condition $x \leq y$ and $y \leq x$ by $x \approx y$. To a subshift $X \subset A^{\mathbb{Z}^2}$, we associate its *subpattern poset* $\mathrm{SP}(X) = (\tilde{X}, \leq)$. Comparisons between points of X refer to comparisons between the associated elements of the subpattern poset. In particular, chains in X refer to chains in $\tilde{X}$.

We say a poset P has the *ascending chain condition*, or ACC, if $p_1 \leq p_2 \leq \cdots$ implies that for some $n \in \mathbb{N}$, $p_i = p_j$ for all $i, j \geq n$. That is, every ascending chain is eventually constant. Symmetrically, we define the *descending chain condition* DCC. For posets $(P, \leq), (Q, \preccurlyeq)$, we denote by $P \boxtimes Q$ the poset $(P \dot\cup Q, (\leq) \dot\cup (\preccurlyeq))$, that is, the disjoint union of P and Q where the order among elements of P is $\leq$, the order among elements of Q is $\preccurlyeq$, and there are no relations between elements of P and Q. For posets $((P_i, \leq_i))_{i \in I}$, we similarly write $\boxtimes_i P_i$ to denote the disjoint union of all posets P_i. For a poset $(P, \leq)$, we write $-P$ for the poset $(P, \geq)$, where $p \geq q$ is defined by $q \leq p$. Note that a poset P has the ACC if and only if $-P$ has the DCC. An *order-embedding* of a poset $(P, \leq)$ to a poset $(Q, \preccurlyeq)$ is an injective function $\phi : P \to Q$ such that $p \leq q$ if and only if $\phi(p) \preccurlyeq \phi(q)$ for all $p, q \in P$. If ϕ is bijective, it is called an *order-isomorphism*. The *height* of an element $p \in P$ is the maximal cardinality of a chain that can be embedded in the sub-poset $\{q \in P \mid q \leq p\}$ of P, minus one. In particular, the height of a minimal element is 0.

Let X be a topological space. For every ordinal λ, we define the *Cantor-Bendixson derivative of order* λ of X, denoted by $X^{(\lambda)}$, by transfinite induction:

- $X^{(0)} = X$,
- $X^{(\alpha+1)} = \{x \in X^{(\alpha)} \mid x \text{ is not isolated in } X^{(\alpha)}\}$, and
- $X^{(\alpha)} = \bigcap_{\beta < \alpha} X^{(\beta)}$, if α is a limit ordinal.

There must exist an ordinal λ such that $X^{(\lambda)} = X^{(\lambda+1)}$, as X is a set. The lowest such λ is called the *Cantor-Bendixson rank* of X, and is denoted $\mathrm{rank}(X)$. From the definition of the derivative operator, it is clear that then $X^{(\mathrm{rank}(X))}$ is a perfect space. We say that a topological space X is *ranked* if and only if $X^{(\mathrm{rank}(X))} = \emptyset$. The *rank* $\mathrm{rank}_X(x)$ *of a point* x in a ranked topological space X is the smallest ordinal λ such that $x \notin X^{(\lambda)}$.

2.4 Computability and Logic

Let ϕ be a formula in first-order arithmetic. If ϕ contains only bounded quantifiers, then we say ϕ is Σ^0_0 and Π^0_0. For all $n > 0$, we say ϕ is Σ^0_n if it is equivalent to a formula of the form $\exists k : \psi$ where ψ is Π^0_{n-1}, and ϕ is Π^0_n, if it is equivalent to a formula of the form $\forall k : \psi$ where ψ is Σ^0_{n-1}. This classification is called the *arithmetical hierarchy* (see e.g. [Odi89, Chapter IV.1] for an introduction to the topic). A subset X of $\mathbb{N}$ is Σ^0_n or Π^0_n, if $X = \{x \in \mathbb{N} \mid \phi(x)\}$ for some formula ϕ with the corresponding classification.

It is known that the Σ_1^0 sets are exactly the recursively enumerable sets, and the Π_1^0 sets their complements. When classifying sets of objects other than natural numbers (e.g. finite patterns), we assume that the objects are in some natural and computable bijection with $\mathbb{N}$. Also, a subshift $X \subset A^{\mathbb{Z}^d}$ is given the same classification as its language, and analogously, we say that X is *recursive* if its language is. In a slightly more general way, any topologically closed set $X \subset A^{\mathbb{N}}$ is given the same classification as the set $\{x_{[0,n-1]} \mid x \in X, n \in \mathbb{N}\} \subset A^*$ of finite prefixes of elements of X. See [CR98] for a general survey on Π_1^0 sets. The nonstandard quantifier $\exists^\infty n : \phi(n)$ has the meaning 'there exist infinitely many $n \in \mathbb{N}$ such that $\phi(n)$.'

A subset $X \subset \mathbb{N}$ is *many-one reducible* (or simply *reducible*) to another set $Y \subset \mathbb{N}$, if there exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that $n \in X$ if and only if $f(n) \in Y$ for all $n \in \mathbb{N}$. We also say that X is *Turing-reducible* to Y, denoted $X \leq_{\mathrm{T}} Y$, if there exists an oracle Turing machine that can decide the membership problem of X using Y as an oracle. If $X \leq_{\mathrm{T}} Y$ and $Y \leq_{\mathrm{T}} X$ both hold, then X is *Turing equivalent to* Y, and we denote $X \equiv_{\mathrm{T}} Y$. If every set in a class $\mathcal{C} \subset 2^{\mathbb{N}}$ is reducible to $X \subset \mathbb{N}$, then X is said to be $\mathcal{C}$*-hard*. If, in addition, X is in $\mathcal{C}$, then X is $\mathcal{C}$*-complete*. These definitions are understood with respect to many-one reductions, not Turing reductions, unless explicitly stated otherwise.

Using Turing reductions, the arithmetical hierarchy can be extended into the *hyperarithmetical hierarchy*. We will not define it rigorously, as the definition is somewhat tedious and we only use it in Section 4.4, but we give a short outline of the features that are relevant to us. A *computable ordinal* is a chain $\alpha = (\mathbb{N}, R)$, where $R \subset \mathbb{N}^2$ is a computable well-ordering of $\mathbb{N}$. For each computable ordinal α, there exists a set $0^{(\alpha)} \subset \mathbb{N}$ such that $0^{(0)} = \emptyset$, and $\alpha < \beta$ implies $0^{(\alpha)} \leq_{\mathrm{T}} 0^{(\beta)}$. Furthermore, for successor ordinals, $0^{(\alpha+1)}$ contains exactly those numbers $n \in \mathbb{N}$ for which the oracle Turing machine with index n never halts on input n using $0^{(\alpha)}$ as an oracle. A set $X \subset \mathbb{N}$ is *hyperarithmetical*, if it is Turing equivalent to $0^{(\alpha)}$ for some computable ordinal α. A more complete treatment of this subject can be found in [Sac90].

In some of our constructions, we need a computational device, and counter machines fit the bill. A k*-counter machine* is defined as a quintuple $M = (k, Q, \delta, q_0, q_f)$, where $k \in \mathbb{N}$ is the number of counters Q is a finite *state set*, $q_0, q_f \in Q$ the *initial and final states* and

$$\delta \subset Q \times [1, k] \times \{\mathrm{Z}, \mathrm{P}, -1, 0, 1\} \times Q$$

the *transition relation*. An *instantaneous description (ID) of* M is an element of $Q \times \mathbb{N}^k$, with the interpretation of $(q, n_1, \ldots, n_k)$ being that the machine is in state q with counter values $n_1, \ldots, n_k$. The machine operates in possibly nondeterministic steps as directed by δ. If we have $(p, i, \mathrm{Z}, q) \in \delta$

($(p, i, \mathrm{P}, q) \in \delta$), then from any ID $(p, n_1, \ldots, n_k)$ such that $n_i = 0$ ($n_i > 0$, respectively), the machine M may move to the ID $(q, n_1, \ldots, n_k)$. If we have $(p, i, r, q) \in \delta$ with $r \in \{-1, 0, 1\}$, then from any ID $(p, n_1, \ldots, n_i, \ldots, n_k)$, the machine M may move to the ID $(q, n_1, \ldots, n_i + r, \ldots, n_k)$. We denote the possibility of a move from an ID I_1 to an ID I_2 via a transition $d \in \delta$ by $I_1 \stackrel{d}{\to} I_2$. We may assume that counter machines never try to decrement a counter below 0. The machine is initialized from the ID $(q_0, n, 0, \ldots, 0)$ for an input $n \in \mathbb{N}$, and it halts when it reaches the final state q_f, *accepting* the input. The set of inputs that the machine eventually accepts is called its *language*. The machine M is called *reversible* if at most one transition of δ is applicable to any given ID. Note that the transition relation of a reversible counter machine need not be bijective, only injective. The classical reference for counter machines is [Min67], although our precise definition comes from [Mor96].

There is a convenient way of converting an arbitrary counter machine into a reversible machine with only two counters, and we use this in several constructions. The following result can be extracted from the proofs in [Mor96], and we especially note that while the original results concern only deterministic machines, the following lemmas hold even for nondeterministic ones with exactly the same proofs.

Lemma 2.4 (Proved as Theorem 3.1 of [Mor96]). *For any k-counter machine $M = (k, Q, \delta, q_0, q_f)$ there exists a reversible $(k+2)$-counter machine $M' = (k+2, Q \cup Q', \delta', q_0, q_f)$ such that for all $m_i, n_i, h \in \mathbb{N}$ and $q, p \in Q$,*

$$(q, m_1, \ldots, m_k) \Rightarrow_M (p, n_1, \ldots, n_k)$$

holds if and only if there exists $\ell \in \mathbb{N}$ with

$$(q, m_1, \ldots, m_k, h, 0) \Rightarrow^*_{M'} (p, n_1, \ldots, n_k, \ell, 0)$$

where the intermediate states of the computation are in Q'. Also, all transitions of δ' whose first component is in Q' are deterministic.

Note in particular that the countability of the set of infinite computations is preserved by the transformations.

In addition to the standard counter machines, we also use *counter machines with string input*, shortened CMS. We are not aware of such machines being defined before. A CMS is a tuple $M = (k, k', A, Q, \delta, q_0, q_f)$, where k, Q, q_0 and q_f have the same meaning as for conventional counter machines, while $k' < k$ is the number of *output counters*, A is the finite *input alphabet*, and the transition relation has the type

$$\delta \subset Q \times \big(([1, k] \times \{\mathrm{Z}, \mathrm{P}, -1, 0, 1\}) \cup \bar{A}\big) \times Q,$$

where $\bar{A} = A \cup \{\#\}$ for a new symbol $\# \notin A$. An ID of M is now an element $(q, w, n_1, \ldots, n_k)$ of $Q \times A^* \times \mathbb{N}^k$, with the interpretation that M is in the state $q \in Q$ reading the input word $w \in A^*$ and has the counter values $n_1, \ldots, n_k \in \mathbb{N}$. The operation of a CMS M is identical to a conventional counter machine, except for transitions of the form $(p, a, q) \in \delta$ for $a \in \bar{A}$. If such a transition exists, then from any ID $(p, w, n_1, \ldots, n_k)$ with $w_{n_k} = a$ in the case $a \in A$, or $|w| \leq n_k$ in the case $a = \#$, M may move to the ID $(q, w, n_1, \ldots, n_k)$.

The CMS is initialized from an ID $(q_0, w, 0, \ldots, 0)$ for $w \in A^*$, and halts when it reaches the final state q_f. The tuple $(n_1, \ldots, n_{k'}) \in \mathbb{N}^{k'}$ of the first k' counter values in the final state q_f is the *output* of M on w, and is denoted $M(w)$. Note that n_1 is the value of the first counter, not the greatest counter value. If M never halts or reaches an ID from which no valid transition exists, no output is generated.

The intuitive meaning of the above definition is that in one step, a CMS may increment or decrement one of its counters, check whether a counter is zero, or check the input symbol under the last counter, if one exists. When it halts, the counter values of the final ID are considered as outputs. Thus a CMS can be interpreted as a partial function from A^* to $\mathbb{N}^{k'}$. Conventional counter machines are computationally universal, and it is not hard to see that the same holds for these devices.

Especially in Chapter 4 and Chapter 7, we also use the following variant of a counter machine, called an *arithmetical program*. This is essentially the model MP1RM (More Powerful One-Register Machine) defined in [Sch72]. An arithmetical program is a tuple $M = (Q, \delta, q_0, q_f)$, where the transition relation

$$\delta \subset Q \times \left((\{+, -, \cdot, /\} \times \mathbb{N}) \cup \mathbb{N} \cup \mathbb{N}^2 \right) \times Q$$

is finite.

The machine has one counter, so its IDs have the form $(q, n) \in \Sigma \times \mathbb{N}$, and it operates on them as follows. If $(q, (\mathrm{op}, m), p) \in \delta$ for $\mathrm{op} \in \{+, -, \cdot, /\}$, the machine M can move to to ID $(p, n \,\mathrm{op}\, m)$. If $(q, n, p) \in \delta$, or if $(q, (m, k), p) \in \delta$ for some $k, m \in \mathbb{N}$ such that $n \equiv m \bmod k$, then the machine can move to (p, n). We may assume that the program never divides the counter value by a number unless it has checked that the division would be exact, and never subtracts m unless the counter value is at least m. An arithmetical program is essentially a one-counter machine augmented with the ability to multiply and divide the counter by, and check its congruence class modulo, some constant values. The machine is initialized in the ID (q_0, n) for an *input value* $n \in \mathbb{N}$, which is *accepted* if the final state q_f is reached during the computation. The determinism and reversibility of arithmetical programs are defined analogously to counter machines. It is easy to see that arithmetical programs can simulate counter machines in the following sense.

Lemma 2.5. *Let $M = (k, Q, \delta, q_0, q_f)$ be a k-counter machine. Then there exists an arithmetical program $M' = (Q, \delta', q_0, q_f)$ such that for all $q, p \in Q$ and $m_1, \ldots, m_k, n_1, \ldots, n_k \in \mathbb{N}$,*

$$(q, m_1, \ldots, m_k) \Rightarrow_M (p, n_1, \ldots, n_k)$$

holds if and only if

$$(q, p_1^{m_1}, \ldots, p_k^{m_k}) \Rightarrow_{M'} (p, p_1^{n_1}, \ldots, p_k^{n_k})$$

does, where p_i is the i'th prime number. Also, M' is deterministic and/or reversible if and only if M is.

However, it is also known that a deterministic two-counter machine can be simulated in such a way that the counter values need not be encoded as a product of prime powers.

Lemma 2.6 ([Sch72])**.** *Deterministic arithmetical programs and deterministic two-counter machines accept the same set of languages. In particular, if a set $L \subset \mathbb{N}$ is recursively enumerable, then the set $\{2^n \mid n \in L\}$ is accepted by some deterministic arithmetical program.*

However, no characterization is known for the class of languages accepted by arithmetical programs. It has been shown that the set of prime numbers is not accepted by any arithmetical program [IT93].

2.5 A Few Words on the Figures and Proofs

As most of our results are constructions of different subshifts with some desired properties, whose configurations are infinite two-dimensional configurations, it should not surprise the reader that this dissertation contains a lot of pictures. For example, in Chapter 4, the configurations of our SFTs have large uniformly colored triangles and rectangles, and we specify which edges of these shapes can be glued together. However, in some of our general proofs we assume that SFTs are given a tiling systems, but this will be explicitly stated.

Multiple levels of abstraction are used in figures. In Figure 3.1, an actual partial configuration of an SFT is given. However, most figures are projections of the actual SFTs, leaving out some details which are explained in the text. Figure 3.5 is an example of a projection where most of the detail is left in. Some of the different overlapping regions are not shown (for example, the area to the left of the computation head should be colored differently from the area on its right), and only partial information about the state of the computation head is shown. On many occasions, we take this further, drawing discrete versions of rational lines as straight lines and leaving out the contours of tiles to reduce clutter, and the details are given in the text. An example of this approach is Figure 4.9.

Most of the subshifts appearing in this thesis are two-dimensional. In all informal descriptions of two-dimensional subshifts and configurations, the y-axis is vertical and increases upward, while the x-axis is horizontal increases to the right. In other words, the notions 'up' and 'north' refer to the vector $(0, 1)$, and 'right' and 'east' refer to $(1, 0)$. Also, rows are horizontal and columns are vertical.

As is common practice, we do not go in too much detail on statements whose proof is a straightforward geometrical case analysis, and we do not show the specifics of the counter machines used in the proofs, but instead just give the algorithms they execute in pseudocode.

Chapter 3

Preliminary Results and Constructions

3.1 Determinism, Bounded Signal Property, and Countability

Throughout the later chapters of this dissertation, especially in Chapter 4, we discuss the properties of determinism, countability and the bounded signal property for an SFT. We show by the examples listed in Table 3.1 that essentially all combinations of the three are possible. We do not know examples of uncountable SFTs with the bounded signal property in every direction. Thus, in the examples, we are satisfied with the downward bounded signal property for uncountable SFTs. Similarly, as it is known that only periodic SFTs are deterministic in every direction [BL97], we usually give only one direction of determinism. The nondeterministic SFTs we present are deterministic in no direction. Example 3.4 is essentially a simplified version of Construction 3.11, but otherwise, the results of this section serve mainly as an introduction to the different concepts. We begin with the following structure lemma of countable sofic shifts (and in particular SFTs) in order to make the nature of the bounded signal property more explicit. The version we give is Lemma 1 in [ST12a] (see also Lemma 4.8 of [PS14]).

Lemma 3.1. *Let $X \subset A^{\mathbb{Z}}$ be a one-dimensional countable sofic shift. Then there exists a finite set T of tuples of words in $\mathcal{B}(X)$ such that every configuration $x \in X$ is representable as*

$$x = {}^{\infty}u_0 v_1 u_1^{n_1} \cdots u_{m-1}^{n_{m-1}} v_m u_m^{\infty}$$

for a unique $t = (u_0, \ldots, u_m, v_1, \ldots, v_m) \in T$. In particular, X has only finitely many periodic configurations.

Thus, the configurations of one-dimensional countable sofic shifts consist of long segments with a common small period, separated by a bounded number of short period-breaking patterns. If a two-dimensional subshift has the bounded signal property in the vertical direction, then its horizontal rows are of this simple form. The name "bounded signal property" refers to the short period-breaking patterns, which can be seen as 'signals' traveling in the two-dimensional configurations.

Next, we move on to the examples mentioned above.

Example 3.2. We give an example of a countable SFT which does not have the bounded signal property in any direction but is deterministic in all but the six directions $(1,0), (-1,0), (0,1), (0,-1), (1,-1)$ and $(-1,1)$. This is the grid shift from [ST12b]. It consists of infinite horizontal and vertical lines that divide each configuration into rectangles. The rectangles are forced to be squares by coloring their northwest half differently from the southeast half, using a diagonal signal as a separator. The grid shift is over the alphabet $\{0,1,2\}$ and is defined by the allowed patterns of size 2×2 which occur in Figure 3.1.

1	1	1	1	2	1	1	1	1	2
1	1	1	2	0	1	1	1	2	0
1	1	2	0	0	1	1	2	0	0
1	2	0	0	0	1	2	0	0	0
2	0	0	0	0	2	0	0	0	0
1	1	1	1	2	1	1	1	1	2
1	1	1	2	0	1	1	1	2	0
1	1	2	0	0	1	1	2	0	0
1	2	0	0	0	1	2	0	0	0
2	0	0	0	0	2	0	0	0	0

Figure 3.1: A partial configuration of the grid shift containing exactly the allowed 2×2 patterns.

Example 3.3. We give an example of a countable SFT which is deterministic in no direction, but has the bounded signal property in every direction. First, consider the SFT $X \subset \{0,1,2\}^{\mathbb{Z}^2}$ defined by the allowed patterns of size 2×2 which occur in Figure 3.2. This SFT is not deterministic in any direction $(x,y) \in \mathbb{Z}^2$ with $y \geq 0$. Analogously, the rotated version $L_\pi(X)$ is not deterministic in any direction $(x,y) \in \mathbb{Z}^2$ with $y \leq 0$, so $X \times L_\pi(X)$ is not deterministic in any direction. However, it is countable and has the bounded signal property in every direction.

0	0	0	0	2	2	2	2
0	0	0	0	2	2	2	2
0	0	0	0	2	2	2	2
0	0	0	0	2	2	2	2
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Figure 3.2: A partial configuration of the quarter plane shift containing exactly the allowed 2×2 patterns.

Example 3.4. We give an example of a downward deterministic uncountable SFT with the bounded signal property. An indirect way to find such an example is to use Construction 3.11, presented later in this chapter: Let M be a nondeterministic counter machine with a single counter that it can choose to increment or preserve on each step, use Lemma 2.4 to make it reversible, and plug the resulting machine in Construction 3.11. The resulting SFT is then south deterministic and has the bounded signal property, but it is uncountable as the choice of whether the counter is incremented or not at each step is visible in the configuration.

We also give a direct construction, which can be regarded as a 'broken' version of Construction 3.11: There, a bouncing signal moves upward in an expanding cone, and its movement is extendably north deterministic. We did not make this choice of direction only because the counter machine intuitively computes in this direction. This example reverses the direction of determinism, making the SFT extendably south deterministic, and illustrates what can go wrong with this choice.

The two-dimensional SFT X has the alphabet depicted in Figure 3.3, and its forbidden patterns are exactly those 2×2 patterns where the lines of adjacent or diagonally adjacent tiles do not match. An example configuration is depicted in Figure 3.4.

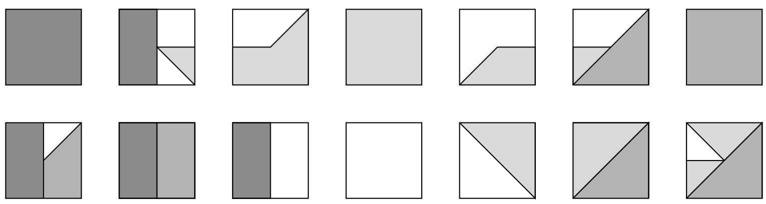

Figure 3.3: The alphabet of X in Example 3.4.

A configuration of X may contain at most one occurrence of the bottom left tile in Figure 3.3, which then forces an infinite cone to extend to the northeast. Inside this cone, a signal travels back and forth with speed 2, bouncing between the walls.

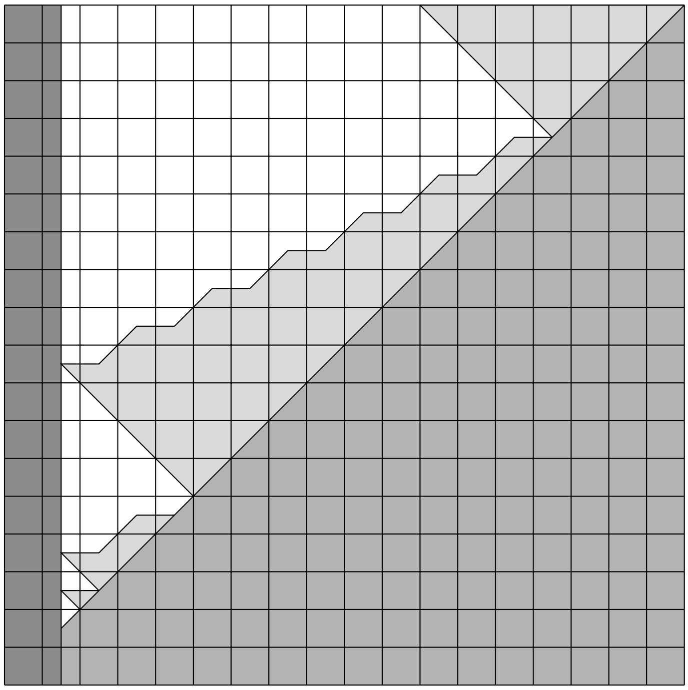

Figure 3.4: An example configuration of X in Example 3.4.

The resulting SFT X is south deterministic (every row is the image of the one above it under a cellular automaton of radius 2), and it clearly has the downward bounded signal property. It is also uncountable, since for every sequence $c = (c_i)_{i \in \mathbb{N}} \in \{0,1\}^{\mathbb{N}}$, we can construct a configuration of X where, on its i'th visit to the right border of the cone, the signal bounces back using the tile on the bottom right in Figure 3.3 if $c_i = 0$, and the tile on the top right if $c_i = 1$. For example, the configuration in Figure 3.4 corresponds to a sequence beginning with 010. These configurations are all distinct, and their number is uncountable.

An interesting property of SFTs that are both deterministic and have the bounded signal property is that they, in a sense, come from the type of cellular automata studied in [ST12a] (cellular automata on one-dimensional countable sofic shifts). We state the following theorems only for downward determinism, as one can apply a transformation in $GL_2(\mathbb{Z})$ to rotate the direction of determinism.

Table 3.1: The 8 different combinations of determinism, countability and the bounded signal property. The 'yes' cases are emphasized. In the first column, we use the auxiliary subshifts $X = \{x \in \{0,1\}^{\mathbb{Z}^2} \mid x = \sigma^{(1,0)}(x)\}$ and $Y = \{x \in \{0,1\}^{\mathbb{Z}^2} \mid x \in \{\sigma^{(1,0)}(x), \sigma^{(1,1)}(x)\}\}$.

Subshift	Deterministic	Bounded signal property	Countable
$\{0,1\}^{\mathbb{Z}^d}$	no	in no direction	no
Ex. 3.2 $\times$ Ex. 3.3	no	in no direction	*yes*
Ex. 3.3 $\times X$	no	*downward*	no
Ex. 3.3	no	*in all directions*	*yes*
Y	*downward*	in no direction	no
Ex. 3.2	*southwest*	in no direction	*yes*
Ex. 3.4	*downward*	*downward*	no
$\{0\}^{\mathbb{Z}^2}$	*in all directions*	*in all directions*	*yes*

Proposition 3.5. *Let $X \subset A^{\mathbb{Z}^2}$ be a deterministic countable SFT with the bounded signal property. Then there is a cellular automaton f on a countable one-dimensional SFT $Y \subset (B\dot\cup\{\#\})^{\mathbb{Z}}$ such that X is conjugate to the set of limit spacetime diagrams of f not containing $\#$.*

The need for the special symbol # comes from the fact that our definition of determinism allows for SFTs where half-planes can be locally legal, while not having legal extensions (so a half-plane can have 0 or 1 extensions to the full plane). If we assume that all half-planes extend to a unique configuration, we have the following more natural result.

Proposition 3.6. *Let $X \subset A^{\mathbb{Z}^2}$ be an extendably deterministic countable SFT with the bounded signal property. Then there is a cellular automaton f on a countable one-dimensional SFT $Y \subset B^{\mathbb{Z}}$ such that X is conjugate to the set of limit spacetime diagrams of f. Furthermore, the alphabet B, the forbidden patterns of Y, and the local function of f are computable from the forbidden patterns of X.*

A similar result is true for SFTs without the bounded signal property, but with Y not necessarily countable. The constructions in this thesis are deterministic only in the weaker sense, and thus do not directly come from cellular automata on countable SFTs. Furthermore, even if one restricts to the limit spacetime diagrams where the symbol # does not occur, the direction of determinism in our constructions is usually not a very interesting one: Most of our deterministic constructions are cones extending northeast from a seed pattern, and everything of interest happens within the cone. The direction of determinism, however, usually has a negative y-component. That is, the cellular automaton can only recreate a finite initial pattern from the infinite tail of the cone.

While our examples imply that all the combinations of determinism, countability and the bounded signal property are possible, there are some interesting connections between them, as for instance, Lemma 4.48 holds for countable SFTs with the bounded signal property and deterministic SFTs with the bounded signal property for slightly different reasons, but not for general SFTs with the bounded signal property.

3.2 Cantor-Bendixson Derivatives of Subshifts

In this section, we give some intuition about the Cantor-Bendixson derivatives of subshifts, and prove preliminary results about it for general topological spaces. Recall that the Cantor-Bendixson derivative of a topological space is obtained by removing all isolated points from it. We note here that in a subshift $X \subset A^{\mathbb{Z}^d}$, a configuration x is isolated if and only if there exists a pattern $(D, s) \in \mathcal{B}(X)$ such that x is the only element of X with $x|_D = s$. Conversely, the configuration x is not isolated if and only if for all $n \in \mathbb{N}$, there exists another configuration $y \in X$ which is distinct from x but satisfies $x|_{[-n,n]^d} = y|_{[-n,n]^d}$. We use this fact without any explicit mention in many of our proofs. As for the subpattern order, for two configurations $x, y \in X$ we have $x \leq y$ if and only if there exists a sequence $(\vec{v}_n)_{n \in \mathbb{N}}$ of translation vectors in $\mathbb{Z}^d$ such that $x = \lim_{n\to\infty} \sigma^{\vec{v}_n}(y)$.

Example 3.7. Consider the two-dimensional SFT $X \subset \{0,1\}^{\mathbb{Z}^2}$ defined by the three forbidden patterns 10, $\substack{0\\1}$ and $\substack{1\\0}$. The subshift X contains the all-0 and all-1 uniform configurations $x^0, x^1 \in X$, together with an infinite number of configurations with a left half plane filled with 0, and a right half plane filled with 1. If $x \in X$ is a configuration of the latter kind, then the pattern $P = 01$ occurs in it at some coordinate $\vec{n} \in \mathbb{Z}^2$, and x is the only configuration of X with this property. Consequently, x is isolated in X. On the other hand, the uniform configurations are not isolated, since for all $n \in \mathbb{N}$, there exist non-uniform configurations that agree with them on the square domain $[-n, n]^2$. This implies that the Cantor-Bendixson derivative of X is $X^{(1)} = \{x^0, x^1\}$. We can similarly prove that $X^{(2)} = \emptyset$, so that X is ranked and $\mathrm{rank}(X) = 2$.

As for the subpattern poset of X, note first that the uniform configurations are incomparable, since the only pattern they share is the empty one. On the other hand, any non-uniform configuration $x \in X$ satisfies $x > x^0$ and $x > x^1$, since we have $x^0 = \lim_{n\to\infty} \sigma^{(-n,0)}(x)$ and $x^1 =$

$\lim_{n\to\infty} \sigma^{(n,0)}(x)$. All non-uniform configurations are translates of each other, so they lie in the same $\leq$-equivalence class. Thus the subpattern poset of X is order-isomorphic to the three-element poset $\{2, 3, 6\}$ ordered by divisibility.

Note that in the above example, the ranked subshift X was countable. This is no coincidence, and in fact the following result shows that exactly the countable subshifts are ranked.

Lemma 3.8 ([BDJ08])**.** *A subshift $X \subset A^{\mathbb{Z}^d}$ is ranked if and only if it is countable.*

The proofs of the following general results are straightforward, but we give them for completeness.

Lemma 3.9. *A subspace of a ranked topological space is ranked. Furthermore, taking the Cantor-Bendixson rank is an order-preserving operation from ranked topological spaces to ordinals.*

Proof. Clearly, $X \subset Y$ implies $X^{(\lambda)} \subset Y^{(\lambda)}$ for all ordinals λ by transfinite induction. Because Y is ranked, $Y^{(\lambda)} = \emptyset$ holds for the Cantor-Bendixson rank λ of Y. Then also $X^{(\lambda)} = \emptyset$, so the Cantor-Bendixson rank of X cannot be more than λ. □

Lemma 3.10. *Let X be a topological space and Y an open subset of X. Then, for all ordinals λ, we have $X^{(\lambda)} \cap Y = Y^{(\lambda)}$.*

Proof. It is clear that $Y^{(\lambda)} \subset X^{(\lambda)}$ and $Y^{(\lambda)} \subset Y$, so it is enough to show that $X^{(\lambda)} \cap Y \subset Y^{(\lambda)}$ for every ordinal λ. We prove the claim by transfinite induction. As a base case, we have $X^{(0)} \cap Y = Y = Y^{(0)}$. Now, let $\lambda = \lambda' + 1$ for some ordinal λ'. Let $x \in X^{(\lambda)} \cap Y$, and let U be an open neighborhood of x in Y. Since U is also open in X, there exists

$$y \in X^{(\lambda')} \cap U = Y^{(\lambda')} \cap U$$

with $y \neq x$. As for limit ordinals λ, we have $X^{(\lambda')} \cap Y \subset Y^{(\lambda')}$, for all $\lambda' < \lambda$, so

$$X^{(\lambda)} \cap Y = \bigcap_{\lambda' < \lambda} X^{(\lambda')} \cap Y \subset \bigcap_{\lambda' < \lambda} Y^{(\lambda')} = Y^{(\lambda)},$$

which concludes the proof. □

3.3 Simulation of Counter Machines in Countable SFTs

We present a general construction for embedding the computations of a counter machine into a countable SFT. We can also enforce the bounded signal property and – if the counter machine is reversible – determinism. This construction, or some variant of it, will be used in all of the remaining chapters of this thesis.

Construction 3.11 (Embedding computations in a countable SFT with the bounded signal property)**.** Suppose we are given a nondeterministic counter machine $M = (k, Q, \delta, q_0, q_f)$ with

$$\delta \subset Q \times [1, k] \times \{\mathrm{Z}, \mathrm{P}, -1, 0, 1\} \times Q.$$

Further suppose that the set of infinite computations of M is countable; note that the set of halting computations is countable for every counter machine, as such computations are finite. We construct a countable SFT X_M with the bounded signal property whose configurations correspond to computation histories of M in a concrete way. The basic structure of X_M is very similar to that of the subshift in Example 3.4. A single horizontal row of X_M consists of a computation zone, which contains segments of special counter symbols whose lengths encode the counter values, together with a 'zig zag head' that holds a state of M. The head sweeps back and forth, updating its state and the counter values, and the computation area increases in size, so that X_M will contain an infinite computation cone extending to the northeast. If M is reversible, then X_M will be south deterministic. However, as the counter machine is running upward, we describe how rows evolve bottom-up.

Since the alphabet of X_M depends on that of M and the local rules are somewhat complex, we do not define X_M with decorated tiles as in Example 3.4. Instead, we first define a set of legal rows Y, and then add constraints that state which configurations may occur on top of each other (which may further reduce the set of legal rows). That is, we first define a subshift Y_M such that $\mathrm{Proj}(X_M) \subset Y_M$, and then describe a relation $R \subset Y_M^2$ which is a subSFT of Y_M^2 such that $X_M = \{x \in S^{\mathbb{Z}^2} \mid \forall i : (x_i, x_{i+1}) \in R\}$.

We begin by defining the alphabet of X_M. First, we define the auxiliary *head alphabet* $H_M = \{\overset{-1}{\leftarrow}, \overset{0}{\leftarrow}, \overset{1}{\leftarrow}, \overset{-1}{\rightarrow}, \overset{1}{\rightarrow}\} \times Q \times [1, k]$. The alphabet of X_M is then $A_M = \{0, 1\} \cup ((\{\ell, r\} \cup H_M) \times \prod_{i=1}^{k} \{\mathrm{P}_i, \mathrm{Z}_i\})$. The intuition is that H_M is the 'state' of the zig zag head – the arrow determines its direction, and the other components its Q-state and the counter it is modifying – the k components in the end of A_M encode the counter values as strings of the form $\mathrm{P}_i^m \mathrm{Z}_i^n$, and the other values of A_M are used to control the geometry of X_M.

We define a (partial) function $\phi : \mathbb{N} \times H_M \times \mathbb{N} \cup \times \mathbb{N}^k \to A_M^{\mathbb{Z}}$ to encode the configurations of Y_M in a concise form. For all $n_\ell, n_r, n_1, \ldots, n_k \in \mathbb{N} \cup \infty$ such that $n_i \leq n_\ell + n_r + 1$ for each i, and all $h \in H_M$, let

$$\phi((n_\ell, h, n_r), (n_1, \ldots, n_k)) = {}^{\infty}0.(\ell^{n_\ell} h r^{n_r} \times \prod_{i=1}^{k} \mathrm{P}_i^{n_i} \mathrm{Z}_i^{n_\ell + n_r + 1 - n_i}) 1^{\infty}.$$

The interpretation is that each n_i is the value in the ith counter, the width of the computation area of the current row is $n_\ell + 1 + n_r$, and the zig zag head is represented by h. The interpretation of $h = (\xrightarrow{b}, q, i)$ is that the zig zag head is moving to the right in order to reach the right border of the computation zone. The interpretation of $h = (\xleftarrow{b}, q, i)$ for $b \in \{-1, 0, 1\}$ is that the zig zag head is moving left, and trying to increment the counter i by b. The subshift Y_M is the orbit closure of the image of ϕ and configurations ${}^\infty 01^\infty$ where computation has not started yet.

For the degenerate configurations $x = {}^\infty 01^\infty$, we allow $(x, x) \in R$ and $(x, y) \in R$ for $y = \phi((0, h, n), (n, 0, \ldots, 0))$, where $n \in \mathbb{N}$ is arbitrary and $h = (\xleftarrow{0}, q_0, 1)$. The choice of h here (other than the initial state $q_0 \in Q$) is somewhat arbitrary. We also take the orbit closure of R, which puts into it the degenerate configurations with $n = \infty$. In our applications, we may further restrict the initial state by requiring, for example, that $n = 0$.

Configurations $\phi((m_\ell, h, m_r), (n_1, \ldots, n_k))$ are followed by configurations of the same form. Note in particular that the left border of the computation area does not move. If $h = (\xrightarrow{b}, q, i)$ for $b = \pm 1$, the zig zag head is moving to the right, so the values (m_ℓ, h, m_r) are updated to $(m_\ell + 2, h, m_r - 1)$ if $m_r \geq 1$, and to $(m_\ell, (\xleftarrow{b}, q, i), m_r + 1)$ if $m_r = 0$. The counter values are not changed at this point.

If $h = (\xleftarrow{b}, q, i)$ for $b \in \{-1, 0, 1\}$, the zig zag head is moving to the left, so the values (m_ℓ, h, m_r) are updated to $(m_\ell - 1, h', m_r + 2)$ if $m_\ell \geq 1$, and $(m_\ell, h_{\text{next}}, m_r + 1)$ if $m_\ell = b = 0$. The case $m_\ell = 0, b = \pm 1$ leads to an error. Define $h' = h$ if $n_i \neq m_\ell - 1$, and $h' = (\xleftarrow{0}, q, i)$ otherwise. In the latter case, the value n_i is also updated to $n_i + b$, and we set the bit b to 0 in order to prevent the zig zag head from updating the same counter multiple times. The element $h_{\text{next}} \in H_M$ can be chosen as $(\xrightarrow{b}, p, j)$ if there exists a transition $(q, j, b, p) \in \delta$ for $b \in \{-1, 0, 1\}$. The case where $q = q_f$ is the halting state is handled separately. Depending on the application of X_M, it may cause a tiling error, a loop where $h_{\text{next}} = h$, or something else entirely.

We remark here that, in contrast to Example 3.4, the movement of the head is north deterministic, so the possible uncountability of X_M can only be the result of M having uncountably many computation histories. The beginning of a computation is shown in Figure 3.5.

We claim that there are only countably many configurations in X_M. Namely, if one of the rows is in the orbit of $\phi((m_\ell, h, m_r), (n_1, \ldots, n_k))$ for finite m_ℓ and m_r, then the configuration contains a full computation cone, and thus corresponds to an infinite computation of M. If none of the rows are of this form, then the configuration contains at most one back-and-forth sweep of the zig zag head, and it is easy to see that there are only countably many such degenerate configurations.

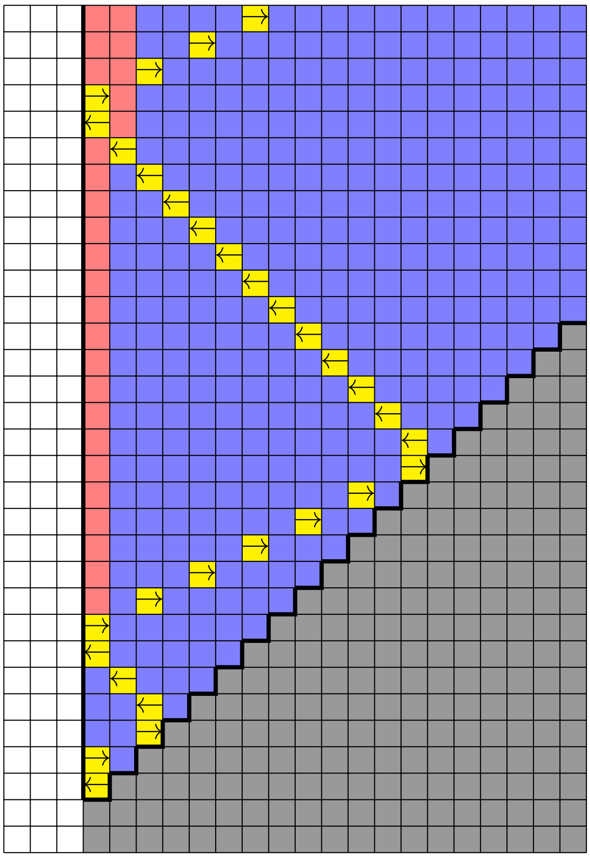

Figure 3.5: The base of a computation cone, with some details omitted. The limits of the cone are indicated by thick lines, and the zig-zag head by a yellow tile with an arrow indicating its direction. The red counter is incremented in the first two computation steps.

Finally, south determinism is proved as follows if M is reversible. The movement of the zig zag head is reversible and not influenced by the actual transitions of M. It is a simple case analysis that the operation of updating a counter during the sweep is reversible. Consider then the step where the zig zag head is situated at the left border of the computation cone, and nondeterministically chooses a transition from δ. The inverse step can be deterministically chosen depending on which counters contain the value 0.

In [JV11], a similar encoding of 'computation in a cone' is used, but instead of counter machines, computation histories of Turing machines are embedded into configurations of countable SFTs. While both approaches have their merits, we feel that it is slightly more obvious how the counter machine construction works and why the resulting subshift is countable.

Also, in our approach, the resulting subshift has the bounded signal property, and its Cantor-Bendixson rank is much lower.

Now, it is relatively easy to modify this construction to simulate a counter machine with string input, or an arithmetical program. However, there is a minor caveat in the case of CMSs: their input is given as a string, and the construction given here does not forbid giving an infinite string as input, so unless the inputs are restricted by the machine itself or some additional structure, the resulting SFT is not countable.

Construction 3.12 (Embedding CMSs and arithmetical programs in a countable SFT)**.** Recall Construction 3.11 and its notation, and let $M = (k, k', A, Q, \delta, q_0, q_f)$ be a CMS. To simulate M in an SFT X_M with the bounded signal property, we modify Construction 3.11 as follows. First, the alphabet of the zig zag head is $H_M = \{\overset{0}{\leftarrow}, \overset{1}{\leftarrow}, \overset{-1}{\leftarrow}, \rightarrow\} \times Q \times [1, k] \times A$, and that of X_M is $A_M = \{0, 1\} \cup ((A \cup \{\#\}) \times (\{\ell, r\} \cup H_M) \times \prod_{i=1}^{k} \{\mathrm{P}_i, \mathrm{Z}_i\})$. The zig zag head effectively remembers also the A-symbol under the first counter, updating it every time the first counter is moved, and each row of the computation cone contains a word $w \in (A \cup \{\#\})^*$ representing the input to M. We guarantee by 2×2 forbidden patterns that $w \in A^* \#^*$, and that it is a prefix of the word above it. It it intuitively clear that the CMS M is correctly simulated on configurations that contain the infinite computation cone. The shortcoming of this construction is that the input word may be infinite, so the SFT X_M is not countable unless the set of possible inputs is restricted in some way.

Now, let $M = (Q, \delta, q_0, q_f)$ be an arithmetical program. In this case, the alphabet of X_M will look somewhat simpler than in Construction 3.11, but the movement of the zig zag head is more complicated. The alphabet of X_M is simply $A_M = \{0, \mathrm{Z}, \ell, r\} \cup Q_M \cup Q'_M$, and the non-degenerate rows of X_M have the form ${}^\infty 0.\ell^{n_\ell} q r^{n_r} q' \mathrm{Z}^\infty$, so that the computation zone is infinite and the subset $\{\ell, r\} \cup Q_M \subset A_M$ plays the role of the positive counter symbol P. The sets Q_M and Q'_M should me thought of as the state sets of two finite state automata, the first of which is the zig zag head and the second is called the *counter head*, and we do not specify them explicitly. Each ID (q, n) of M corresponds to a configuration with $n_\ell = n$ and $n_r = 0$, where the zig zag head lies next to the counter head.

If the transition requires a number $m \in \mathbb{N}$ to be added to or subtracted from the counter, the two heads simply move m steps to the right or left, respectively. Multiplication by $m \geq 2$ are handled by the following procedure. First, the zig zag head starts moving to the left with speed $m + 1$, and the counter head to the right with speed $m - 1$. When the zig zag head reaches the left end of the computation zone, it turns to the right. Eventually, the zig zag head reaches the counter head, and if the counter value was $n \in \mathbb{N}$, this happens at the coordinate mn. The procedure for division is similar, except that the counter head moves to the left. Equality to m is easy to check with an SFT rule, and to check the congruence class of the counter modulo m, the zig zag head walks to the left border of the computation zone, and walks back by steps of length m. See Figure 3.6 for a visualization of the construction.

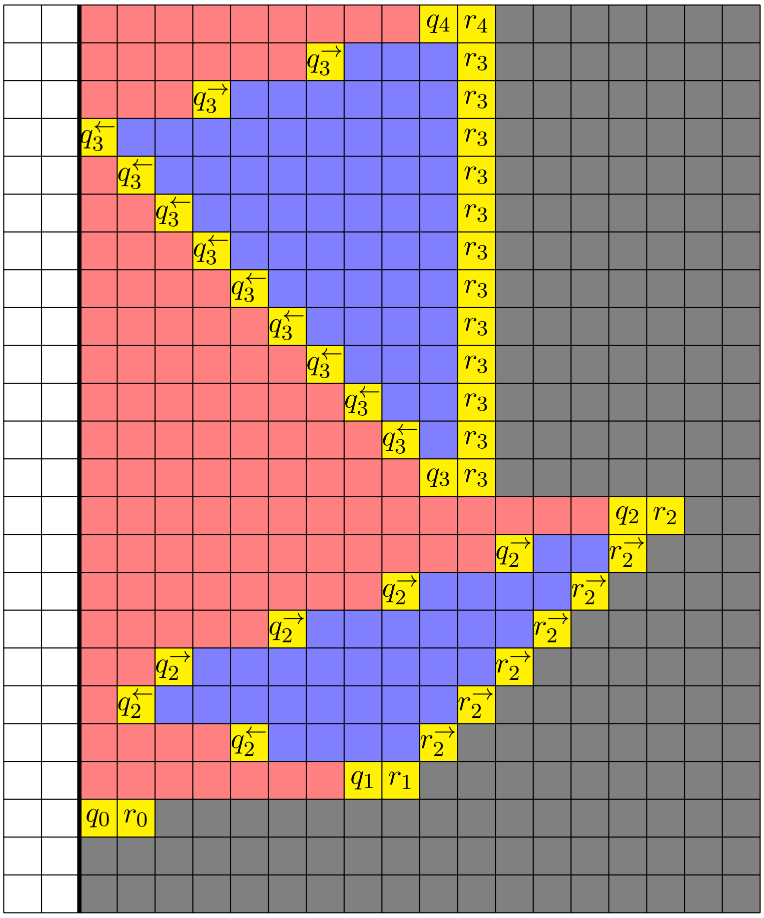

Figure 3.6: The simulation of an arithmetical program that first adds 7 to the counter, then multiplies it by 2, subtracts 5, and checks its parity modulo 3.

The above procedure for simulating an arithmetical program can also be applied directly to two independent finite state machines and an unmoving 'anchor' (here, the left end of the computation zone). This is one of the main tools of Chapter 7, where we construct subshifts using multihead finite automata. Since the Cantor-Bendixson rank of the simulating subshift is also very low, it is used in several optimality results in Chapter 4. In general, arithmetical programs seem to be the simplest computationally universal devices that can be simulated by local rules, in the sense of having as few 'moving parts' as possible.

3.4 Multidimensional Sofic Shifts

A major part of this thesis, especially Chapter 6, is devoted to the study of multidimensional sofic shifts and their properties. In this section, we discuss some known facts about them, and present the class of *countably covered sofic shifts*, which are given special attention.

Definition 3.13. A sofic shift $X \subset A^{\mathbb{Z}^d}$ is *countably covered* if it is the image of a countable SFT via a block map.

This notion is not standard in the literature. Of course, all countably covered sofic shifts are themselves countable, and all countable SFTs are also countably covered sofic shifts. We also remark that it is not known whether every countable sofic shift is countably covered. The following lemma allows us to easily construct new countably covered sofic shifts from existing ones.

Lemma 3.14. *Let $X \subset A^{\mathbb{Z}^2}$ be a countably covered sofic shift, and let $Y \subset (A \times B)^{\mathbb{Z}^2}$ be an SFT. If the subshift $Y \cap (X \times B^{\mathbb{Z}^2})$ is countable, then it is a countably covered sofic shift.*

Proof. Since X is countably covered, there exists a countable SFT $Z \subset C^{\mathbb{Z}^2}$ and a surjective block map $f : Z \to X$. Let $N \subset \mathbb{Z}^2$ be the neighborhood of f. Let $D \subset \mathbb{Z}^2$ be a finite domain such that the SFTs Z and Y are defined by sets of forbidden patterns $F \subset C^D$ and $G \subset (A \times B)^D$, respectively.

Consider now the set

$$F' = F \times B^D \cup \{(P, Q) \in (C \times B)^{D+N} \mid (f(P), Q|_D) \in G\}$$

of finite patterns over $C \times B$. This set defines an SFT $Y' \subset Z \times B^{\mathbb{Z}^2}$, since every pattern whose first component is in F is forbidden. Thus we can define a block map $g : Y' \to X \times B^{\mathbb{Z}^2}$ by $g(z, z') = (f(z), z')$. Since Z is countable, every configuration $x \in X \times B^{\mathbb{Z}^2}$ has at most countably many g-preimages. Furthermore, for all $y' \in Y'$ we have $g(y') \in Y$, since every pattern that would cause some $Q \in G$ to occur in the image is forbidden in Y. In fact, we have

$$Y' = \{(z, z') \in Z \times B^{\mathbb{Z}^2} \mid (f(z), z') \in Y\} = g^{-1}(Y \cap (X \times B^{\mathbb{Z}^2})),$$

and Y' is countable, since $Y \cap (X \times B^{\mathbb{Z}^2})$ is countable and g is countable-to-one. The image of Y' under the block map g is exactly $Y \cap (X \times B^{\mathbb{Z}^2})$, and the claim follows. $\square$

In a typical use case of Lemma 3.14, we have a countably covered sofic shift $X \subset A^{\mathbb{Z}^2}$ and a subset $A' \subset A$, and we wish to superimpose a label from another alphabet L on top of each occurrence of a symbol of A'. In this case, we would define $B = L \cup \{\#\}$, where $\# \notin L$ is a new symbol denoting 'no label,' and define Y by the forbidden symbols (a, ℓ) for all $a \in A \setminus A'$ and $\ell \in L$, and $(a, \#)$ for all $a \in A'$. Of course, if there are configurations in X with infinitely many occurrences of some element of A', we need to impose some additional constraints on Y to ensure the countability of $Y \cap (X \times B^{\mathbb{Z}^2})$.

The class of one-dimensional sofic shifts has a simple and useful characterization, which we will present as Lemma 4.3. It is very much related to the Myhill-Nerode theorem on regular languages. On the other hand, the class of multidimensional sofic shifts lacks such a description, and remains somewhat mysterious. The following combinatorial lemma is one of the few known methods for showing a multidimensional subshift to be nonsofic, and it is actually a partial analogue of Lemma 4.3. We use it in the two-dimensional case, but (a suitable modification of) it holds in all dimensions. A proof of the result has appeared explicitly at least in [KM13], but the technique is much older, presumably originating from the theory of picture languages. In the proof, the notation ∂D for a domain $D \subset \mathbb{Z}^2$ stands for the set

$$\{\vec{n} \in \mathbb{Z}^2 \setminus D \mid (\vec{n} + \{\pm\vec{e}_1, \pm\vec{e}_2\}) \cap D \neq \emptyset\} \subset \mathbb{Z}^2.$$

Lemma 3.15. *Let $X \subset A^{\mathbb{Z}^2}$ be a sofic shift. Then there exists a number $C > 0$ such that for all $n \geq 1$ and all sets $\Lambda \subset X$ of size more than C^n, there exist $x \neq y \in \Lambda$ such that $c(x, y, n) \in X$, where $c(x, y, n) \in S^{\mathbb{Z}^2}$ is defined by $c(x, y, n)|_{[0,n-1]^2} = x|_{[0,n-1]^2}$ and $c(x, y, n)|_{\mathbb{Z}^2 \setminus [0,n-1]^2} = y|_{\mathbb{Z}^2 \setminus [0,n-1]^2}$.*

Proof. Since X is sofic, there exist a tiling system $Y \subset B^{\mathbb{Z}^2}$ and a surjective symbol map $\phi : Y \to X$. We claim that the bound $C = |B|^4$ suffices. Namely, let $n \geq 1$ and $\Lambda \subset X$ with $|\Lambda| > C^n$ be arbitrary, and for all $x \in \Lambda$, choose a preimage $\tilde{x} \in \phi^{-1}(x)$, and denote $f(x) = \tilde{x}|_{\partial[0,n-1]^2}$. Since $|\partial[0, n-1]^2| = 4n$, the number of different values $f(x)$ for $x \in \Lambda$ is at most $|B|^{4n} = C^n$. Thus there exist distinct $x, y \in \Lambda$ with $\tilde{x}|_{\partial[0,n-1]^2} = \tilde{y}|_{\partial[0,n-1]^2}$. But then the configuration $c(\tilde{x}, \tilde{y}, n) \in B^{\mathbb{Z}^2}$ is in Y, since it satisfies the local constraints of Y. It follows that $\phi(c(\tilde{x}, \tilde{y}, n)) = c(x, y, n) \in X$. $\square$

Another useful condition for a multidimensional subshift to be sofic is a computational one. It is a special case of the following general result on the computational complexity of subshifts.

Lemma 3.16. *Let $X \subset A^{\mathbb{Z}^d}$ be a subshift, and let $k \in \mathbb{N}$. If X is Π^0_k, then it can be defined by a Σ^0_k set of forbidden patterns. Conversely, a subshift defined by a Σ^0_k set of forbidden patterns is Π^0_{k+1}. In particular, SFTs and sofic shifts are Π^0_1 subshifts.*

Proof. For the first claim, we simply note that the complement of $\mathcal{B}(X)$ is a Σ_k^0 set of forbidden patterns for X. Let then $\mathcal{F} \subset \mathcal{P}_d(A)$ be a Σ_k^0 set of finite patterns, and let $X = \mathcal{X}_\mathcal{F}$. Denote by $\mathcal{B}_\mathcal{F} \subset \mathcal{P}_d(A)$ the set of patterns where no pattern of $\mathcal{F}$ occurs. For any given pattern $P \in \mathcal{P}_d(A)$, we have $P \in \mathcal{B}(X)$ if and only if the condition

$$\forall n \in \mathbb{N} : (D(P) \subset [-n,n]^d \Rightarrow \exists Q \in \mathcal{B}_\mathcal{F} \cap A^{[-n,n]^d} : P \sqsubset Q)$$

holds. Here we consider the domain of P to be fixed. Since $\mathcal{B}_\mathcal{F}$ is a Σ_k^0 set, we see that $\mathcal{B}(X)$ is a Π_{k+1}^0 set. $\square$

In addition to the combinatorial Lemma 3.15 and the computational Lemma 3.16, there exists a dynamical soficness condition. It is not used in this dissertation, but we include the statement for completeness. Recall that the *(topological) entropy* of a d-dimensional subshift $X \subset A^{\mathbb{Z}^d}$ is defined as the nonnegative real number

$$h(X) = \lim_{n \to \infty} \frac{1}{n^d} \log |\mathcal{B}_{[0,n-1]^d}(X)|,$$

which can be shown to always exist. Intuitively, it is a measure of the 'size' of X, or the asymptotic amount of freedom we have in constructing an arbitrary configuration $x \in X$. As already noted in Chapter 1, it was proved in [HM10] that the entropies of multidimensional SFTs are exactly the right recursively enumerable nonnegative reals, which shows another connection between multidimensional symbolic dynamics and recursion theory. Now, the dynamical condition states the following.

Proposition 3.17 ([Des06])**.** *Let $X \subset A^{\mathbb{Z}^d}$ be a sofic shift. Then the set of entropies $\{h(Y) \mid Y \subset X \text{ sofic}\}$ of all sofic subshifts of X is dense in the interval $[0, h(X)]$.*

As a corollary, for every $a \in [0, h(X)]$ there exists a (not necessarily sofic) subshift $Y \subset X$ with $h(Y) = a$. The proposition is not vacuous, as there exist subshifts $X \subset A^{\mathbb{Z}^d}$ with positive entropy such that every subshift $Y \subset X$ satisfies either $h(Y) = h(X)$ or $h(Y) = 0$. This follows, for example, from the multidimensional analogue of the Jewett-Krieger theorem, announced in [Wei85] (see [Ros87] for a more detailed proof). The original Jewett-Krieger theorem states that for any measure-theoretic dynamical system (X, T, μ), there exists a minimal subshift $Y \subset A^\mathbb{Z}$ with a unique shift-invariant measure ν such that (X, T, μ) is measure-theoretically isomorphic to (Y, σ, ν). In particular, the topological entropy of Y equals the measure-theoretic entropy of (X, T, μ), which can be any nonnegative real number.

Complementing the above set of results that restrict the class of multidimensional SFTs and sofic shifts, there are numerous constructions of such objects that push these boundaries and show that, in some sense or other, they are close to being the best we can obtain. For example, while Proposition 3.17 states that positive-entropy sofic shifts have many subsystems, Boyle, Pavlov and Schraudner constructed in [BPS10] a positive-entropy sofic shift in which these subsystems are 'poorly separated,' in that their intersection is nonempty and consists of a single uniform configuration. In the one-dimensional case, this is impossible.

The following powerful result, which we will use in some of our constructions, complements Lemma 3.16 by showing that any one-dimensional Π_1^0 subshift can be implemented as the projective subdynamics of a two-dimensional sofic shift. Hochman presented the result in [Hoc09] for a three-dimensional sofic shift, and the two-dimensional case was independently proved in [DRS10] and [AS13]. In both proofs, an infinite hierarchy of computations running in the SFT cover processes longer and longer words of the one-dimensional configuration, and produces a tiling error if a forbidden one is found. Their differences lie in the actual implementation of this hierarchy.

Theorem 3.18 ([DRS10, AS13])**.** *Let $X \in A^{\mathbb{Z}}$ be a one-dimensional Π_1^0 subshift. Then the vertically periodic two-dimensional subshift*

$$\{y \in A^{\mathbb{Z}^2} \mid \exists x \in X : \forall (i,j) \in \mathbb{Z}^2 : y_{(i,j)} = x_i\}$$

is sofic.

Chapter 4

Structural Properties of Countable Two-Dimensional SFTs

4.1 Introduction

In this section, we study the geometric and hierarchical structure of two-dimensional subshifts. More explicitly, we analyze the abstract structure of their subpattern posets, and the possible Cantor-Bendixson ranks they may possess. The former notion was introduced in the context of quasiperiodic configurations in [Dur99], and applied to countable two-dimensional SFTs in [BDJ08]. The notion of Cantor-Bendixson rank, on the other hand, originates from abstract topology. The ranks of general Π^0_1 subsets of the one-sided full shift have been studied extensively [CCS$^+$86, CDJS93], but in the context of subshifts, the notion was also first studied in [BDJ08]. Since the notion of Cantor-Bendixson rank is particularly suitable in the context of countable subshifts, and has resulted in very fruitful research in the past, we also restrict our attention to this subclass.

Another reason for restricting to countable SFTs is that the study of multidimensional symbolic dynamics has traditionally placed much emphasis on SFTs with aperiodic configurations, and several complicated constructions of such objects with highly nontrivial properties exist [Rob71, Moz89, Hoc09, DRS10, DRS12]. Intuitively, it seems that few things can be achieved using multidimensional Π^0_1 subshifts but not multidimensional SFTs. In order to find limitations to what SFTs can do, one approach is to restrict to a subclass of SFTs with an additional limitation, and prove that some further restricting properties must follow. The results of [BDJ08] can be considered to follow this schema: if we restrict ourselves to the class of nonempty countable SFTs, then we can always find doubly periodic configurations.

In [BDJ08], some structural limitations on the subpattern poset of a countable SFT were proved, including the ascending chain condition (Theorem 4.20 in this chapter). The main motivation for studying these objects was the following result.

Theorem 4.1 (Theorem 3.11 in [BDJ08])**.** *Every infinite two-dimensional countable SFT contains a singly periodic configuration.*

The results of [BDJ08] were complemented in [ST13] with constructions of countable SFTs with an infinite descending chain, and a subpattern poset containing an order-embedded copy of every finite poset. In this section, we present a generalization of these results, giving an almost complete characterization of the possible subpattern posets of countable SFTs. This class of posets is defined by a very weak computability condition, implying that the restriction of countability for SFTs is, in the end, rather weak. Except for the complicated main construction of this section, we have tried to make our constructions deterministic and having countable projective subdynamics, if possible. Interestingly, the construction of an SFT with high Cantor-Bendixson rank is possible with any one of the three properties of determinism, countability and the bounded signal property, but we show that determinism *or* countability, in conjunction with the bounded signal property, already forbids an infinite descending chain.

The problem of finding an infinite Cantor-Bendixson rank was solved in [JV11] by drawing runs of a Turing machine on the configurations of the SFTs. Once it is shown that the SFT is countable, the existence of an infinite Cantor-Bendixson rank follows easily, as the ranks obtained are cofinal in the Cantor-Bendixson ranks of countable Π^0_1 sets. In [ST13], we gave an alternative example of rank more than ω with a more geometric construction, and also showed a perhaps simpler alternative for embedding computations, using a counter machine instead of a Turing machine. In the constructions presented here, we rely heavily on counter machines and Construction 3.11, and also prove the complete characterization of the possible Cantor-Bendixson ranks of countable SFTs using arithmetical programs.

In addition to the constructions presented above, we include the construction of a countable SFT whose derivatives grow in complexity from [ST13], and an SFT with infinite rank which is not of computational nature.

This chapter is based on the article [ST13].

4.2 The One-Dimensional Case

In this section we look at the Cantor-Bendixson ranks of one-dimensional (not necessarily countable) sofic shifts to illustrate the difference between the one-dimensional and the multidimensional case. It is more natural to consider sofic shifts instead of SFTs, as sofic shifts are closed under the derivative operation by Proposition 4.5, but the derivative of an SFT can be a proper sofic shift, by Example 4.7.

One-dimensional sofic shifts have a useful and well-known characterization, Lemma 4.3, in terms of the different contexts of words that appear in them. We give the characterization without proof, but for example, it easily follows from Theorem 3.2.10 of [LM95].

Definition 4.2. Let $X \subset A^{\mathbb{Z}}$ be a subshift with $\mathcal{A}(X) = A$. The *extender set* of a word $v \in A^*$ in X is $E_X(v) = \{(w, w') \in (A^*)^2 \mid wvw' \sqsubset X\}$.

The assumption $\mathcal{A}(X) = A$ is only for definiteness, as we can then discuss the extender sets of words in a subshift X without specifying the alphabet A. The following fundamental result should be compared to Lemma 3.15, which is its multidimensional analogue.

Lemma 4.3. *A one-dimensional subshift is sofic if and only if it has a finite number of different extender sets.*

We now relate the extender sets of words in a subshift and its derivative.

Lemma 4.4. *Let $X \subset A^{\mathbb{Z}}$ be a subshift, and let $u, v \in A^*$. If we have $E_X(u) = E_X(v)$, then $E_{X^{(1)}}(u) = E_{X^{(1)}}(v)$.*

Proof. Let $E_X(u) = E_X(v)$, and suppose that $(w, w') \in E_{X^{(1)}}(u) \setminus E_{X^{(1)}}(v)$. Then $wvw' \not\sqsubset X^{(1)}$, so the set of points $x \in X$ with $x_{[0,|wvw'|-1]} = wvw'$ is finite. But since we have $E_X(u) = E_X(v)$, these points are in a bijective correspondence with the points $y \in X$ such that $y_{[0,|wuw'|-1]} = wuw'$, which implies that $wuw' \not\sqsubset X^{(1)}$, a contradiction. □

This implies that the number of different extender sets cannot increase in the derivative, so the two previous lemmas give the following:

Proposition 4.5. *The derivative of a one-dimensional sofic shift is sofic.*

A simple further analysis proves the following result.

Proposition 4.6. *All one-dimensional sofic shifts have finite rank. More explicitly, the rank of a one-dimensional sofic shift is bounded by the number of different extender sets in it.*

Proof. Let $X \subset A^{\mathbb{Z}}$ be a sofic shift, and let $k \in \mathbb{N}$ be the number of different extender sets in X. If $X \neq X^{(1)}$, then necessarily $X \neq A^{\mathbb{Z}}$, so we may choose $u \notin \mathcal{B}(X)$. Further, choose $v \in \mathcal{B}(X) \setminus \mathcal{B}(X^{(1)})$. We have $E_X(u) \neq E_X(v)$ and $E_{X^{(1)}}(u) = \emptyset = E_{X^{(1)}}(v)$. By Lemma 4.4, $X^{(1)}$ has at most $k-1$ different extender sets. By induction, $X^{(i)}$ has at most $k - i$ different extender sets, and it is then clear that $X^{(i)} = X^{(i+1)}$ for some $i \leq k$. □

We now show an example of an SFT whose derivative is proper sofic (sofic but not an SFT). Compare this with the SFT constructed in Theorem 4.13, which is in particular a two-dimensional SFT whose derivative is not an SFT.

Example 4.7. The subshift $X = \mathcal{B}^{-1}(a^* b^* c^*)$ is a countable SFT. The isolated points in it are all configurations where both letters a and c occur. Thus, its derivative is $X^{(1)} = \mathcal{B}^{-1}(a^* b^* + b^* c^*)$, which is a proper sofic shift.

Finally, we show that a countable one-dimensional sofic shift has a rather uninteresting subpattern poset.

Lemma 4.8. *Let $X \subset A^{\mathbb{Z}}$ be a sofic shift. If a word $w \in A^*$ occurs infinitely many times in some configuration $x \in X$, then X has a periodic configuration in which w occurs.*

Lemma 4.9. *Let $X \subset A^{\mathbb{Z}}$ be a countable sofic shift, and let $x, y \in X$. We have $x < y$ if and only if $x = {}^\infty u^\infty$ for some word $u \in A^+$, and y is an eventually periodic point with $y_{(-\infty,i]} = {}^\infty u$ or $y_{[i,\infty)} = u^\infty$ for some $i \in \mathbb{Z}$.*

Proof. Since x and y are in X, it follows from Lemma 3.1 that they are eventually periodic. Suppose first that $x = {}^\infty u^\infty$ is periodic. One of the infinite tails of y, say $y_{[0,\infty)}$, contains arbitrarily long subwords of x. By the theorem of Fine and Wilf [FW65], we then have $y_{[i,\infty)} = u^\infty$ for some $i \in \mathbb{N}$.

Suppose then that x is not periodic, and let $w_i = x_{[-i,i]}$. Since $w_i \sqsubset y$ but $x \neq y$, the configuration y must contain infinitely many copies of w_i for all $i \in \mathbb{N}$. By Lemma 4.8, every word w_i occurs in a periodic point of X. Since x is itself not periodic and the lengths of the w_i grow without bound, there are infinitely many periodic points in X, a contradiction with Lemma 3.1. $\square$

In particular, the maximum cardinality of a chain that can be embedded in the subpattern poset of a one-dimensional countable sofic shift is 2.

4.3 Cantor-Bendixson Ranks and Complexity of Derivatives

The two main constructions of this section concern the computational complexity of the k'th derivative of a countable SFT, and the set of all attainable Cantor-Bendixson ranks.

Theorem 4.13 shows that there exists an SFT whose derivatives of order less than ω have maximal possible computational complexity. This is essentially Theorem 1 in [ST12b], although we make the bounded signal property explicit, and determinize the construction.

Theorem 4.14 is a slight improvement on Theorem 4.3 in [JV11] and Theorem 4.5 in [ST13], which state that Cantor-Bendixson ranks of countable two-dimensional SFTs are cofinal in the Cantor-Bendixson ranks of countable Π^0_1 sets. More specifically, the theorem of [JV11] states that for every countable Π^0_1 set with Cantor-Bendixson rank λ, there is a countable SFT with rank $\lambda + 11$, and the proof used a method for simulating Turing machines in countable SFTs. In [ST13], this was improved to $\lambda + 4$ using Construction 3.11. The construction in Theorem 4.3 in [JV11] could be made deterministic, but counter machines are essential for the bounded signal property. Furthermore, the bounded signal property forces the binary track (which contains a point of the Π^0_1 set) used in the proof of Theorem 4.14 to be simulated 'in software', while [JV11] uses an actual tape. In this thesis, we further lower the simulation overhead from 4 to 3 using Construction 3.12 for arithmetical programs.

We start with an upper bound for the computational complexity of derivatives, which we then reach with a construction. A generalization of the following lemma was proved in [CCS$^+$86, Lemma 1.2 (3)], but we include a proof for completeness.

Lemma 4.10. *Fix a d-dimensional Π^0_k subshift $X \subset A^{\mathbb{Z}^d}$. Given a finite pattern $P \in \mathcal{P}_d(A)$, it is Π^0_{k+2} whether $P \in \mathcal{B}(X^{(1)})$.*

Proof. Let $m \in \mathbb{N}$ be such that $D(P) \subset [-m, m]^d$. Then P occurs in the derivative $X^{(1)}$ if and only if $P \in \mathcal{B}(X)$ and for all $n \geq m$, there exist two distinct patterns $Q \neq R \in \mathcal{B}(X)$ such that $[-n, n]^d \subset D(Q) = D(R)$, $Q|_{[-n,n]^d} = R|_{[-n,n]^d}$, and $Q|_{D(P)} = P$. This condition is clearly Π^0_{k+2}. □

The following construction shows that the bound given by Lemma 4.10 on the complexity of k'th derivatives of Π^0_1 subshifts is strict, and can be attained by a single deterministic countable SFT with the bounded signal property. In particular, it implies that Proposition 4.5 is far from true in dimension two, since two-dimensional sofic shifts are Π^0_1, while their derivatives may be Π^0_3-complete, and thus highly nonsofic by Lemma 3.16. The rank of the subshift we construct is $\omega + k$ for some finite k. We start with a definition, and a classical recursion-theoretic lemma.

Definition 4.11. For $k \in \mathbb{N}$, denote by Φ_k the set of first-order arithmetical formulas with k free variables and only bounded quantifiers. For $k, \ell \in \mathbb{N}$, denote by ϕ^k_ℓ the ℓ'th formula in Φ_k, ordered first by length and then lexicographically.

Lemma 4.12 (Lemma 2 in [KSW60])**.** *Let $k \in \mathbb{N}$, and let $\phi \in \Phi_{2k+1}$ be an arithmetical formula. Then there exists a formula $\psi \in \Phi_{k+1}$, uniformly computable from ϕ and k, such that*

$$\forall n_1 \; \exists n_2 \; \cdots \forall n_{2k-1} \; \exists n_{2k} \; \forall n_{2k+1} \; \phi(n_1, \ldots, n_{2k+1})$$

is equivalent to

$$\exists^\infty n_1 \, \exists^\infty n_2 \, \cdots \exists^\infty n_k \, \forall n_{k+1} \, \psi(n_1, \ldots, n_{k+1}).$$

We denote $\psi = I(\phi)$ in the above lemma. With this result, we can transform alternating quantifiers into infinitary ones, and the application to derivatives is rather straightforward.

Theorem 4.13. *There exists a deterministic countable two-dimensional SFT $X \subset A^{\mathbb{Z}^2}$ with the bounded signal property such that the problem whether $P \sqsubset X^{(k)}$ for a given pattern $P \in \mathcal{P}_2(A)$ is Π^0_{2k+1}-complete, for all $k \in \mathbb{N}$.*

Proof. As an illustration of the idea, consider the closure of the subset of $\{0,1\}^{\mathbb{N}}$ consisting of points of the form

$$0^\ell 1 0^k 1 0^{n_1} 1 0^{n_2} \cdots 0^{n_k} 1 0^\infty$$

where $I(\phi^{2k+1}_\ell)(n_1, n_2, \ldots, n_k, n_{k+1})$ is true for all $n_{k+1} \in \mathbb{N}$. This set is Π^0_1-complete. Clearly, the derivative of this closed set contains only those points of the form

$$0^\ell 1 0^k 1 0^{n_1} 1 0^{n_2} \cdots 0^{n_{k-1}} 1 0^\infty,$$

where $I(\phi^{2k+1}_\ell)(n_1, n_2, \ldots, n_{k-1}, n_k, n_{k+1})$ holds for infinitely many $n_k \in \mathbb{N}$ and all $n_{k+1} \in \mathbb{N}$. Thus the derivative is Π^0_3-complete, and we could verify by induction that the n'th derivative is Π^0_{2n+1}-complete. The construction of X is an implementation of the same idea using a suitable counter machine and Construction 3.11.

Let M be a nondeterministic counter machine that operates as per Algorithm 1. The machine M simply guesses the parameters $k \in \mathbb{N}$ and $\ell \in \mathbb{N}$, then guesses the k numbers $n_1, \ldots, n_k \in \mathbb{N}$, and finally checks in an infinite loop that $I(\phi^{2k+1}_\ell)(n_1, n_2, \ldots, n_{k-1}, n_k, n_{k+1})$ holds for all $n_{k+1} \in \mathbb{N}$. By Lemma 2.4, we may assume that M is reversible. Note that the set of infinite computations of M is countable, since the choice $b = 1$ can be made only finitely many times during a computation. We plug M in Construction 3.11 to obtain the corresponding countable deterministic SFT $X = X_M$ with the bounded signal property. In this case, the final state of M results in a tiling error in X_M. Even after applying Lemma 2.4, the nondeterministic guesses of M are visible in the SFT, so a finite initial part of a computation cone where the parameters k and ℓ have been chosen occurs in $X^{(k)}$ if and only if the formula

$$\exists^\infty n_1 \, \cdots \exists^\infty n_k \, \forall n_{k+1} \, I(\phi^{2k+1}_l)(n_1, \ldots, n_{k+1})$$

is true. But by Lemma 4.12, this is equivalent to

$$\forall n_1 \, \exists n_2 \, \cdots \forall n_{2k-1} \, \exists n_{2k} \, \forall n_{2k+1} \, \phi^{2k+1}_l(n_1, \ldots, n_{2k+1}),$$

Algorithm 1 The program of the counter machine M

```
k ← −1
repeat
    k ← k + 1
    choose b ∈ {0, 1}
until b = 1
ℓ ← −1
repeat
    ℓ ← ℓ + 1
    choose b ∈ {0, 1}
until b = 1
for all i ∈ {1, ..., k} do
    n_i ← −1
    repeat
        n_i ← n_i + 1
        choose b ∈ {0, 1}
    until b = 1
for all n_{k+1} ∈ ℕ do
    if not I(φ_ℓ^{2k+1})(n_1, ..., n_{k+1}) then
        reject
```

and thus the subshift $X^{(k)}$ is Π^0_{2k+1}-hard in the sense of the claim. By Lemma 4.10, it is actually Π^0_{2k+1}-complete. $\square$

In this dissertation, we have not tried to extend Theorem 4.13 beyond the first infinite ordinal ω, which would of course be a natural strengthening of the result. The complexity of $X^{(\lambda)}$ for countable SFTs X and computable ordinals λ probably covers the hyperarithmetical hierarchy. However, there cannot exist a single countable SFT X such that $X^{(\lambda)}$ has maximal computational complexity for all computable ordinals λ, simply because the rank of X is a computable ordinal.

Next, we prove that deterministic countable SFTs with the bounded signal property can have arbitrarily high computable Cantor-Bendixson ranks. We use both Lemma 2.4 and Lemma 2.5 to make the construction deterministic, and to optimize the simulation overhead to 3 (from the value 11 in Theorem 4.3 of [JV11], and 4 in [ST13]). Even without these optimization steps, we would obtain arbitrarily high computable Cantor-Bendixson ranks with countable SFTs with the bounded signal property.

Theorem 4.14. *For any countable Π^0_1 set $S \subset \{0,1\}^{\mathbb{N}}$, there is a countable deterministic SFT $X \subset A^{\mathbb{Z}^2}$ with the bounded signal property for which* $\mathrm{rank}(X) = \mathrm{rank}(S) + 3$ *holds.*

Proof. Without loss of generality we assume that $S \neq \emptyset$. Since S is Π^0_1, there exists a Turing machine T_S which outputs a potentially infinite list of words $F \subset \{0,1\}^*$ such that

$$S = \{x \in \{0,1\}^{\mathbb{N}} \mid \forall n \in \mathbb{N} : x_{[0,n-1]} \notin F\}.$$

We denote by $T_S(n)$ the (finite) list of words produced by T_S after n computation steps. We define a counter machine M_S by Algorithm 2.

Algorithm 2 The program of the counter machine M_S

$n \leftarrow 0$
$W = \{\lambda\}$
loop
  choose $w_n \in \{0,1\}$
  $W \leftarrow W \cup \{w_0 \cdots w_n\}$
  if $W \cap T_S(n) \neq \emptyset$ **then reject**
  $n \leftarrow n + 1$

By Lemma 2.4, we can assume that M_S is reversible, and by Lemma 2.5, there exists a reversible arithmetical program M'_S that simulates it. It is clear that the infinite computation histories of M'_S form a countable set, since each corresponds to an element of S. We then apply Construction 3.12 to M'_S to obtain our SFT X. Here, the single counter of M'_S is always initialized to 1 in the SFT, and rejection results in a tiling error.

For all coordinates $\vec{n} \in \mathbb{Z}^2$, the set $X_{\vec{n}}$ of configurations of X where the computation starts at $\vec{n}$ is homeomorphic to S, since the choices of the bits $w_n \in \{0,1\}$ are visible in the configurations and are the only source of nondeterminism in M_S. Furthermore, each set $X_{\vec{n}}$ is open in X, so by Lemma 3.10, we have

$$X^{(\lambda)} = \bigcup_{\vec{n} \in \mathbb{Z}^2} X^{(\lambda)}_{\vec{n}} \cup Y$$

for every ordinal $\lambda \leq \mathrm{rank}(S)$, where $Y \subset X$ contains only configurations where the computation does not start.

Now, we describe the set Y, and show that its Cantor-Bendixson rank is exactly 3. Let $y \in Y$ be arbitrary, so that y does not contain the start of a computation, and let $\alpha = \mathrm{rank}(Y)$. If y contains the left border of the computation zone, we claim it cannot contain both the zig zag head and the counter head. This is because M is executing an algorithm whose memory consumption increases with time (it remembers the set W which increases in size), so for all $m \geq 0$ there exists $t \geq 0$ such that after t computation steps, the value of the counter is at least m. Thus, if y contains the left border, it may also contain a single infinite sweep of the zig zag head, but nothing more, and then any pattern containing both the zig zag head and the left border is isolating for y. Conversely, if y contains the counter head, it may also contain a single sweep of the zig zag head, but nothing more. If y is not isolated in Y, then it is periodic in some direction, and thus $\alpha \leq 3$. Finally, Figure 4.1 shows an example configuration of rank $\alpha - 3$, which contains the left border of a computation cone and the zig zag head. Such a configuration exists in Y, since Algorithm 2 requires the zig zag head to visit the left border infinitely many times. This shows that $\alpha = 3$, and the theorem is proved. □

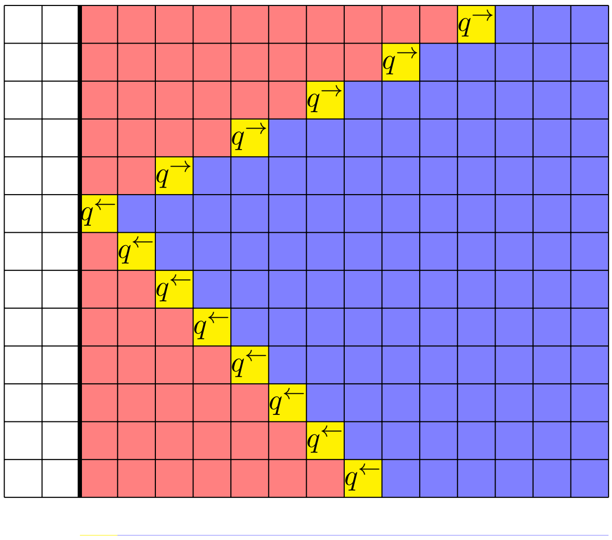

Figure 4.1: A configuration of minimal Cantor-Bendixson rank in the space Y of Theorem 4.14.

In [CCS+86] it is proved that the rank of a nonempty countable Π^0_1 set can be any recursive successor ordinal. Thus, Theorem 4.14 shows that for any recursive ordinal α *not* of the form $\lambda + n$ for a limit ordinal λ and $n \in \{0, 1, 2\}$, there exists a countable deterministic two-dimensional SFT with Cantor-Bendixson rank exactly λ. In [BDJ08], it it shown that the ranks λ and $\lambda + 1$ cannot be achieved for any countable subshift. In the context of countable SFTs, the remaining case of $\lambda + 2$ has been settled in the preprint [BJ13].

Theorem 4.15 (Theorem 5.3 in [BJ13])**.** *For a limit ordinal λ, there exists no countable SFT with Cantor-Bendixson rank $\lambda + 2$.*

Corollary 4.16. *The Cantor-Bendixson ranks of countable SFTs are exactly the finite ordinals, and those of the form $\lambda + n$, where λ is a computable limit ordinal and $n \geq 3$.*

With the same construction, we also obtain the upper bound 5 for the smallest possible rank of a countable SFT with an uncomputable configuration. For Π_1^0 subshifts (and closed sets in general), the smallest such rank is known to be 2 [CDTW12]. Applied to such a set, the construction in Theorem 4.14 gives a countable SFT with rank 5, and clearly preserves the computability and uncomputability of non-degenerate points. Furthermore, the following result can be found in [BJ13].

Proposition 4.17 (Corollary 4 in [BJ13])**.** *An SFT of Cantor-Bendixson rank at most* 4 *contains only computable configurations.*

This proves that our bound is strict.

Corollary 4.18. *The minimal Cantor-Bendixson rank of an SFT containing an uncomputable configuration is* 5*.*

Finally, we present our geometric construction of an infinite Cantor-Bendixson rank as another, perhaps more natural, example of how infinite ranks might arise in countable SFTs. The construction has the bounded signal property, but it is not deterministic.

Example 4.19. We give an example of a countable two-dimensional SFT X of rank at least ω with the bounded signal property. For a motivating example, consider the one-dimensional subshift containing points of the form

$$^\infty 0 a^k 0^{m_1} a^{k-1} b 0^{m_2} a^{k-2} b^2 0^{m_3} \cdots 0^{m_k} b^k 0^\infty,$$

where $k \in \mathbb{N}$ and $m_i \in \mathbb{N}$ for all $i \in \{1, \ldots, k\}$ are arbitrary. For all $k \in \mathbb{N}$, the subshift contains configurations with k 'islands' floating in a sea of 0's, but no configuration contains an infinite number of islands. This is a countable subshift with infinite rank, and in the following, we construct a two-dimensional SFT X that uses exactly the same idea.

Define an SFT X over the alphabet in Figure 4.2, and the obvious 2×2 patterns (the gray areas labeled T and B are of different colors). A configuration of X may contain one infinite horizontal *dedicated line* (the thick line in the figure), whose top and bottom halves are colored differently. On the two sides of the line one may have (perhaps infinite) right triangles. The top and bottom triangles must be located on the same coordinates, meaning that their right angles are vertically opposite.

From the top corner of every triangle above the dedicated line, an *increment signal* is sent to the right, and it is absorbed by the tile to the southwest of the top corner of the next triangle on the right. The symmetric condition holds for the triangles below the dedicated line. This causes the heights of the higher triangles to increase to to right, and those of the lower triangles to increase to the left. See Figure 4.3 for a clarifying picture.

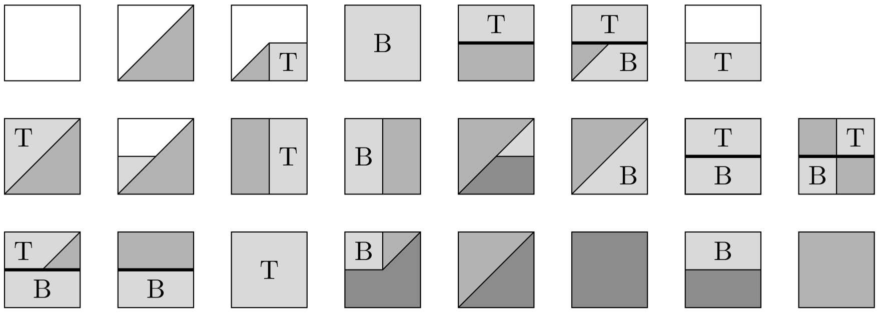

Figure 4.2: The alphabet of the SFT in Example 4.19.

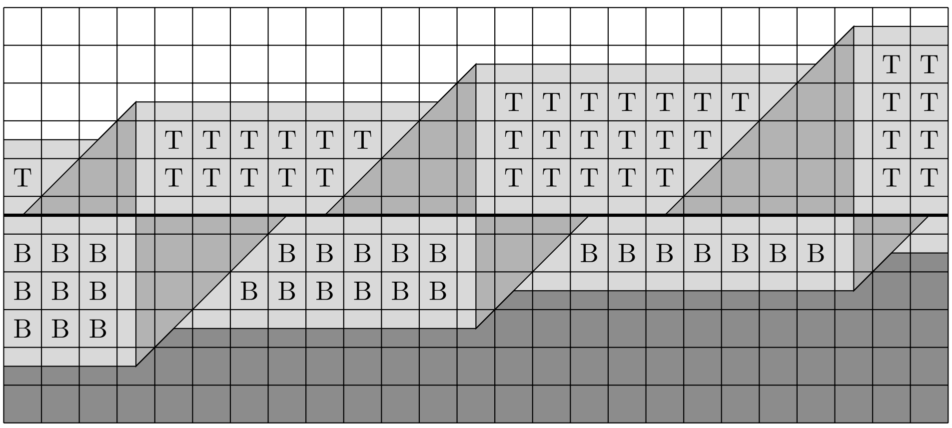

Figure 4.3: An example configuration of the SFT X.

We first claim that X is countable. Indeed, for each $(n, m) \in \mathbb{N}^2$, if a configuration $x \in X$ contains a pair of opposite triangles of sizes n and m, then there are at most $n + m - 1$ pairs of triangles in x, since the size of the higher (lower) triangles decreases to the left (right, respectively). The number of ways to arrange these points triangles is countable. One can also check that the number of exceptional points (ones containing, say, an infinite triangle or just signals) is countable.

Next, we show that $X^{(\omega)}$ is a nonempty set of finite rank. First, the isolated points of X are exactly those that only contain finite triangles, and whose rightmost lower and leftmost higher triangles are of size 2 (the height of a tile being 1). In general, if $x \in X$ contains only finite triangles, we say that x *has type* (n, m) if the rightmost lower triangle and leftmost higher triangle have sizes n and m, respectively. It is then easy to see that for all $k \in \mathbb{N}$, the set $X^{(k)}$ will contain all of X, except for the configurations of type (n, m) with $n + m < k + 4$. Then $X^{(\omega)}$ is nonempty, but will consist of only exceptional points, and clearly $X^{(\omega+k)} = \emptyset$ for some finite k.

4.4 Structure of Subpattern Posets

4.4.1 Infinite Chains and Antichains

In this section, we investigate the subpattern posets of countable SFTs, in particular their structure as abstract posets. It is known that the subpattern poset of a countable subshift has the ACC, meaning that it contains no infinite increasing chain. A proof of this result can be found in [BDJ08, Theorem 3.7] using the theory of Cantor-Bendixson derivatives, and in [ST12b] with a more concrete approach. We repeat the latter proof here for completeness.

Theorem 4.20. *The subpattern poset of a countable subshift has the ACC.*

Proof. Let $X \subset A^{\mathbb{Z}^d}$ be a subshift, and let $(x_i)_{i \in \mathbb{N}}$ be a sequence of configurations of X with $x_i < x_{i+1}$ for all $i \in \mathbb{N}$. We will show that X is uncountable. For each binary word $w \in \{0,1\}^*$, we define a number $\ell_w \geq |w|$, and a finite pattern $P_w \sqsubset x_{|w|}$ with domain $[-\ell_w, \ell_w]^d \subset \mathbb{Z}^d$, as follows. First, let $\ell_\epsilon = 0$, and let $P_\epsilon = x_0|_{\vec{0}}$. Next, suppose that we have defined the pattern $P_w \sqsubset x_i$, where $i = |w| > 0$. Since $x_i < x_{i+1}$, there exists a number $\ell > \ell_w$ and two distinct coordinates $\vec{m} \neq \vec{n} \in \mathbb{Z}^d$ such that $x_{i+1}|_{[-\ell_w, \ell_w] + \vec{m}} = x_{i+1}|_{[-\ell_w, \ell_w] + \vec{n}} = P_w$, but $x_{i+1}|_{[-\ell, \ell] + \vec{m}} \neq x_{i+1}|_{[-\ell, \ell] + \vec{n}}$. Then we define $\ell_{w0} = \ell_{w1} = \ell$, $P_{w0} = x_{i+1}|_{[-\ell, \ell] + \vec{m}}$ and $P_{w1} = x_{i+1}|_{[-\ell, \ell] + \vec{n}}$.

Note that the above construction guarantees that $P_w \sqsubset P_v$, if $w \in \{0,1\}^*$ is a prefix of $v \in \{0,1\}^*$. Furthermore, the domain of P_w only depends on the length $|w|$. Then, each infinite word $u \in \{0,1\}^{\mathbb{N}}$ defines a configuration $x_u \in X$ by $x_u = \lim_{i \to \infty} P_{u_{[0,i]}}$, where the patterns have been centered to the origin, and the configurations are all distinct by the construction of the patterns P_w. We have described an injection from $\{0,1\}^{\mathbb{N}}$ to X, which means that X is uncountable. $\square$

A subpattern poset of a countable SFT may contain infinite antichains, and in fact they are harder to avoid than produce. The grid subshift of Example 3.2 has this property, but even the one-dimensional countable SFT $\mathcal{B}^{-1}(0^*1^*2^*)$ contains an infinite antichain. This leaves us with the case of infinite downward chains, which was left open in [BDJ08]. It was later shown in [ST13], with a complicated construction, that the subpattern posets of countable SFTs may have infinite downward chains. In the next subsection, we generalize this result, and provide an almost complete description of the isomorphism classes of subpattern posets of countable SFTs.

4.4.2 Hyperarithmetical Subpattern Posets

In this section, we show that the subpattern posets of countable SFTs can be almost characterized, up to a subset of bounded height, by a computability condition. For this, recall the definition of hyperarithmetical sets from Section 2.4.

Definition 4.21. A *hyperarithmetical poset* is a poset $(P, \preceq)$ such that both $P \subset \mathbb{N}$ and $\preceq \subset \mathbb{N}^2$ are hyperarithmetical sets.

We will prove that the subpattern posets of countable SFTs are, up to order-isomorphism, almost exactly the hyperarithmetical posets. One direction of our result, that subpattern posets are always hyperarithmetical, follows rather easily from general computability theory.

Lemma 4.22 (Part of Theorem 1.1 of [CCS$^+$86])**.** *Let $X \subset \{0,1\}^{\mathbb{N}}$ be a countable Π^0_1 set, and let $x \in X$ have rank $\lambda + n$ in X, where λ is a computable limit ordinal and $n < \omega$. Then $x \leq_T \emptyset^{(\lambda+2n)}$.*

Theorem 4.23. *Let $X \subset A^{\mathbb{Z}^d}$ be a countable two-dimensional Π^0_1 subshift with Cantor-Bendixson rank $\lambda + n$, where λ is a computable limit ordinal and $n < \omega$. Then there exists a hyperarithmetical poset $(P, \preceq)$, with P and $\preceq$ Turing-reducible to $\emptyset^{(\lambda+2n+2)}$, which is order-isomorphic to $\mathrm{SP}(X)$.*

Proof. In this proof, for a computable ordinal α and $i \in \mathbb{N}$, we denote by $M_i^{(\alpha)}$ the oracle Turing machine with index i and oracle $0^{(\alpha)}$, which may compute numbers, finite patterns, other Turing machines, or whatever is convenient for us at that point. First, every configuration $x \in X$ has rank at most $\lambda + n$ in X, so Lemma 4.22 implies that $x \leq_T \emptyset^{(\lambda+2n)}$. This means that there exists an index $i \in \mathbb{N}$ such that $M_i^{(\lambda+2n)}(m) = x_{[-m,m]^d} \in \mathcal{P}_d(A)$ for every $m \in \mathbb{N}$. In the above case, we denote $x = x^i$.

Consider the problem of determining whether $x^i \in X$ for a given index $i \in \mathbb{N}$. There is an index $e \in \mathbb{N}$ such that, given $i \in \mathbb{N}$, the oracle machine $M_e^{(\lambda+2n)}$ checks for each $k \in \mathbb{N}$ whether $\left(M_i^{(\lambda+2n)}(m)\right)_{m=0}^{k}$ is a consistent sequence of cocentric square patterns on A, none of which contains any of the forbidden patterns of X, and halts as soon as this is not the case. The divergence of $M_e^{(\lambda+2n)}$ on input i, which is equivalent to $x^i \in X$, is Turing reducible to $\emptyset^{(\lambda+2n+1)}$.

Next, consider the problem whether $x^i \leq x^j$ holds for given indices $i, j \in \mathbb{N}$. There exists an index $f \in \mathbb{N}$ such that, given an index $i \in \mathbb{N}$ and a pattern $P \in \mathcal{P}_d(A)$, the machine $M_f^{(\lambda+2n)}$ checks for each $k \in \mathbb{N}$ whether P occurs in $x^i_{[-k,k]^d}$, halting if it does, and diverging if the computation of $x^i_{[-k,k]^d}$ by $M_i^{(\lambda+2n)}$ diverges. We can also find an index $g \in \mathbb{N}$ such that, given $i \in \mathbb{N}$ and $P \in \mathcal{P}_d(A)$, the machine $M_g^{(\lambda+2n+1)}$ checks for all $k \in \mathbb{N}$ that the computation of $x^i_{[-k,k]^d}$ halts but $M_f^{(\lambda+2n)}(j, x^i_{[-k,k]^d})$ diverges; if this is not the case, the machine halts. In the case that $x^j \in X$, the divergence of $M_g^{(\lambda+2n+1)}$ is equivalent to $x^i \leq x^j$, which is thus Turing reducible to $\emptyset^{(\lambda+2n+2)}$. Determining $x^i \approx x^j$ is then also Turing reducible to $\emptyset^{(\lambda+2n+2)}$.

We now construct the poset $(P, \preceq)$, where $P \subset \mathbb{N}$ and $P, \preceq \leq_T \emptyset^{(\lambda+2n+2)}$. The set P is defined inductively by

$$i \in P \iff x^i \in X \wedge \forall j < i : (j \in P \implies x^i \not\approx x^j)$$

for all indices $i \in \mathbb{N}$. This means that $i \in P$ if and only if $x^i \in X$ is the lowest-indexed member of its $\approx$-equivalence class, and thus P is in a natural bijection with $\tilde{X}$, the underlying set of $\mathrm{SP}(X)$. The ordering $\preceq$ is simply defined by

$$i \preceq j \iff x^i \leq x^j$$

for all $i, j \in P$, which makes it a hyperarithmetical relation Turing reducible to $\emptyset^{(\lambda+2n+2)}$. The aforementioned bijection becomes an order-isomorphism between $\mathrm{SP}(X)$ and $(P, \preceq)$, which finishes the proof. □

Corollary 4.24. *The subpattern poset of a two-dimensional countable Π_1^0 subshift is order-isomorphic to a hyperarithmetical poset.*

We proceed to the converse claim, that all hyperarithmetical posets can be realized as subpattern posets of countable SFTs, up to a subset of elements of bounded height. For our construction, we need the following useful result.

Theorem 4.25 (Part of Theorem 4.14 of [JM69])**.** *Let α be a computable ordinal. Then there exists a set $N \subset \mathbb{N}$ with $N \equiv_{\mathrm{T}} 0^{(\alpha)}$ and a computable function $f : \mathbb{N} \to \mathbb{N}$ such that, denoting $N = \{n_0, n_1, \ldots\}$ with $n_i < n_{i+1}$, we have*

1. *$f(n_0) = n_0$, and $f(n_{i+1}) = n_i$ for all $i \in \mathbb{N}$, and*

2. *N is the only infinite subset of $\mathbb{N}$ with property 1.*

In the above situation, we denote $N = \mathrm{Tr}(f)$, and call it the trace of *f.*

This result is usually stated as 'hyperarithmetical sets are exactly the unique infinite branches of computable trees.'

Theorem 4.26. *Let $(P, \preceq)$ be a hyperarithmetical poset with the ascending chain condition. Then there exists a countable two-dimensional SFT $X \subset A^{\mathbb{Z}^2}$ and an order-embedding $\phi : P \to \mathrm{SP}(X)$ such that every element in $\mathrm{SP}(X) \setminus \phi(P)$ has height at most 3.*

Intuitively, the above theorem states that a very large class of posets, including all finite posets and the reversals of all computable ordinals, can be 'almost' realized as the subpattern posets of countable SFTs. The rest of this section is devoted to the proof of this result, which takes the form of an explicit construction of the SFT X. For this, fix the hyperarithmetical poset $(P, \preceq)$, and let $f : \mathbb{N} \to \mathbb{N}$ be a computable function such that P and $\preceq$ are both Turing reducible to the trace $\mathrm{Tr}(f)$. The existence of such an f is guaranteed by Theorem 4.25. We may assume without loss of generality that $0 \notin P$. The countable SFT $X \subset X_s \times X_h \times X_c$ consists of three layers, each of which is an SFT itself, and we construct them separately.

The intuition behind the construction is the following. We will construct a geometric structure that resembles a one-way infinite chain in a directed graph, and each vertex corresponds to a computation step of a counter machine. The machine stores an element p of the poset P, and performs an infinite computation, during which it slowly enumerates the set P and the relation $\preceq$. Conversely, to each element $p \in P$ corresponds exactly one such computation chain, say $(x_i^p)_{i \in \mathbb{N}}$. Now, since the counter machine enumerates the relation $\preceq$, it can also enumerate the set $c(p) = \{q \in P \mid q \prec p\}$. We construct the chain in such a way that for all $q \in c(p)$ and for infinitely many indices $i \in \mathbb{N}$, a copy of a long initial segment of the chain $(x_j^q)_{j \in \mathbb{N}}$ terminates at the vertex x_i^p. Thus the chain $(x_i^p)_{i \in \mathbb{N}}$ is actually part of an infinite rootless tree T_p, which contains all finite subtrees of all T_q for $q \in c(p)$. The tree T_p corresponds to the configuration $\phi(p) \in X$, and the above condition on T_p and T_q corresponds exactly to $\phi(q) \preceq \phi(p)$.

The main difficulties in the construction are various synchronization and timing constraints, and the condition of countability. First of all, the counter machine simulated in the chains is nondeterministic: it constructs a hyperarithmetical set $N \subset \mathbb{N}$ one guess at a time, and uses the computable function $f : \mathbb{N} \to \mathbb{N}$ given by Theorem 4.25 to guarantee that $N = \mathrm{Tr}(f)$. However, this requires that the computation run for infinitely many steps, but the initial segments attached to the chain $(x_i^p)_{i \in \mathbb{N}}$ are all finite. Because of this, we need to synchronize the computations of the different branches, that is, guarantee that they all make the exact same nondeterministic guesses, and this is enforced locally by the vertices that connect the initial segments to the chain. This causes another issue, since the vertices are implemented in the SFT X as finite triangular regions of tiles, and the synchronization condition is complex enough that we check it by a simulated computation, which must fit inside the region. Fortunately, this problem can be solved by computing the above set $N \subset \mathbb{N}$ very slowly, allowing the triangular regions to grow to accommodate the computations. Finally, the countability of X is achieved partially as a side product of the above considerations (in particular the guarantee that apart from the choice of the element of P, the simulated counter machines have a unique infinite computation path), and partially by the choice of geometry and suitable forbidden patterns that restrict the shape of its configurations.

4.4.3 The Skeleton Layer

The first layer, X_s, is called the *skeleton layer*, and it provides a rigid geometric structure to X. The remaining layers contain simulations of counter machines, which we anchor to the structure of the skeleton layer. Throughout the construction and the proof of its correctness, we also define several classes of infinite rootless trees, since configurations of the skeleton layer are structured as such objects.

Definition 4.27. A *skeleton tree* is an infinite directed tree $T = (V, E, \pi)$, where $\pi : V \to \mathbb{N}$ is a labeling of the vertices, satisfying the following conditions.

- It is connected, and $\pi(v) \in \{2^n \mid n \geq 2\}$ for all $v \in V$.
- The set of edges E is partitioned into two disjoint subsets, the *successor edges* E_s and the *child edges* E_c.
- If $(v, w) \in E_s$, then $\pi(w) = 2\pi(v)$. We say that v is a *successor* of w, and w a *predecessor* of v.
- If $(v, w) \in E_c$, then $\pi(w) = 8\pi(v)$. We say that v is a *child* of w, and w a *parent* of v.
- Every vertex has either one predecessor, or one parent, but not both.
- Every vertex has at most one child, and at most one successor.
- A vertex $v \in V$ has a successor if and only if $\pi(v) > 4$.

The value $\pi(v)$ is called the *width* of the vertex $v \in V$ (the reason for this terminology will soon be clear).

The alphabet of X_s is depicted in Figure 4.4, and its forbidden patterns are the obvious 2×2 ones, plus a couple of other patterns that we describe below. An example configuration is shown in Figure 4.5. As can be seen in the latter figure, some configurations of X_s contain a southeast half-plane colored with H, whose border we call the *dedicated line*, and chains of shrinking triangular shapes (areas colored with the two lighter grays), which we call *vertex triangles*. We say that a vertex triangle is *rooted* at the coordinate of the dedicated line that contains its southeast corner. Each vertex triangle consists of a light and dark triangular region, which we call its *outer and inner triangles*, respectively. For example, in Figure 4.5, the triangle ACD is a vertex triangle rooted at C, and its outer and inner triangles are ABD and BCD, respectively. Note that the discrete line from B to D has slope 8. The dark gray area between the dedicated line and a vertex triangle is called its *shadow*.

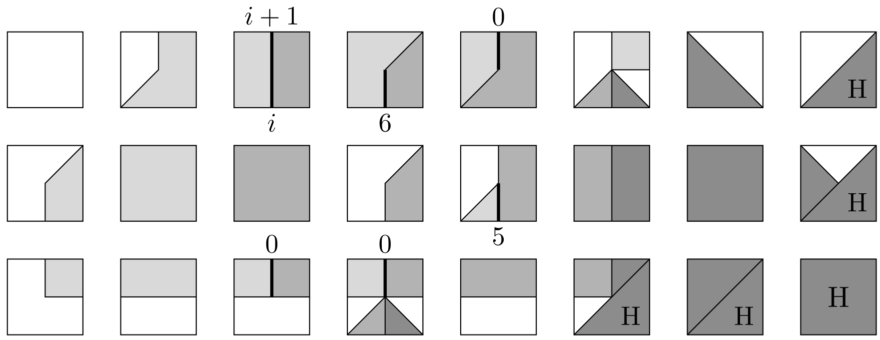

Figure 4.4: The alphabet of the skeleton layer X_s. The endpoints of the thick lines are labeled by integers from 0 to 6, and i ranges from 0 to 5.

Now, the configurations of X_s are closely related to skeleton trees. Notice that the north corner of a vertex triangle Δ is always attached to another vertex triangle Δ', by the west corner of its inner or outer triangle. In the former case, we call Δ' the *parent of* Δ, and in the latter, the *predecessor of* Δ. In Figure 4.5, the large triangle above ACD is its parent, while ACD itself is the predecessor of the smaller triangle below it. In order to make the connection between the skeleton layer and the set of skeleton trees even more explicit, we add some new forbidden patterns to X_s. First, analogously to the last condition of Definition 4.27, we require that a vertex tree must have a successor, unless its width is exactly 4. This can be implemented by forbidding every 4×4 pattern whose southwest corner coincides with that of a large vertex triangle without a successor. Second, we require that if the tile at the bottom left corner of Figure 4.4 (the southwest corner of a vertex triangle with no successor) occurs at some coordinate $(i, j) \in \mathbb{Z}^2$, then the

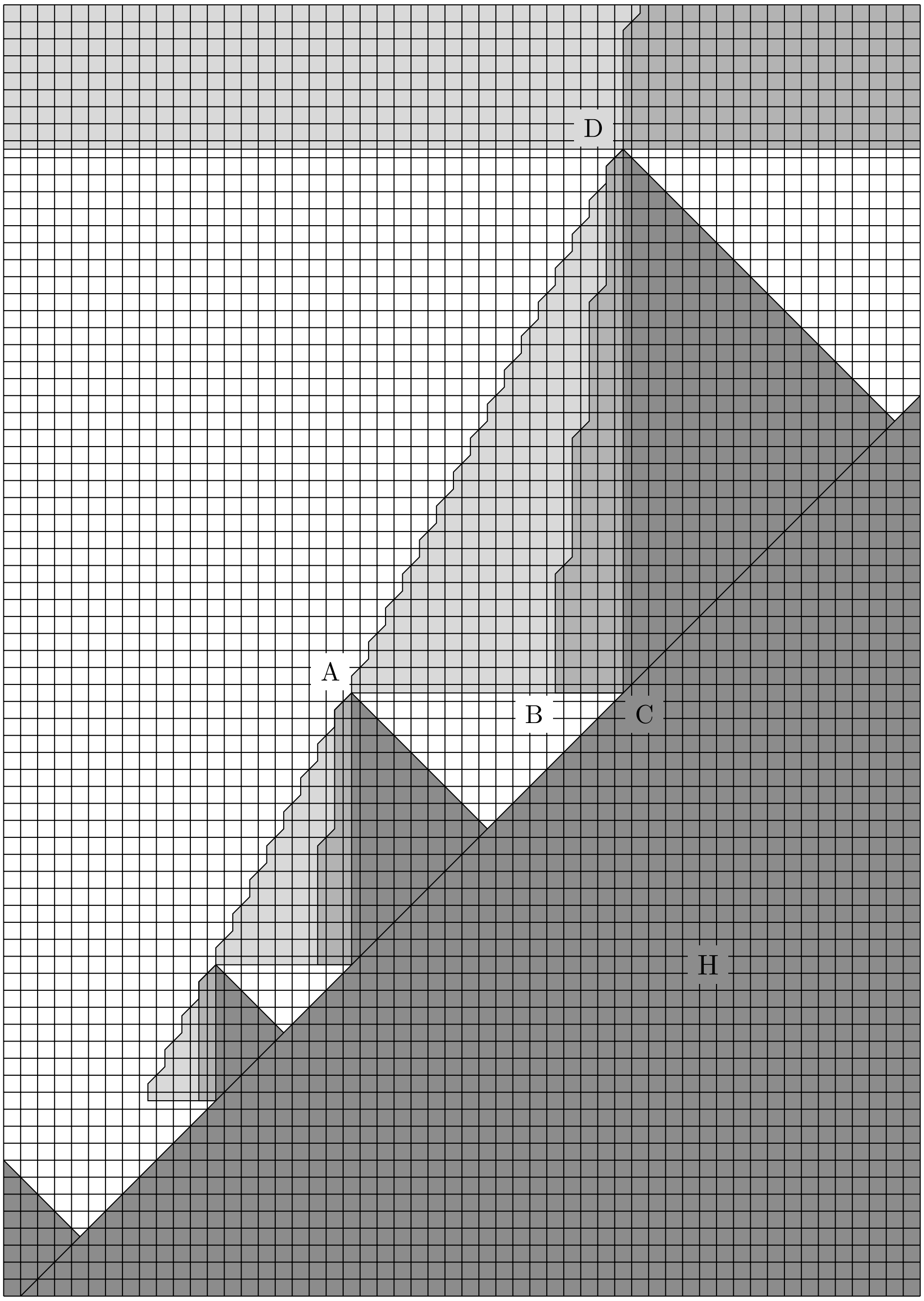

Figure 4.5: A configuration of the skeleton layer X_s. The letter H refers to the color of the half-plane, while A, B, C and D denote intersection points of the discrete lines.

middle tile of the rightmost column (the east corner of a shadow) occurs at $(i-4, j-8)$. We call this the *shadow condition*, and it concludes the definition of X_s.

Now, it is easy to see that every configuration $x \in X_s$ containing a finite vertex triangle Δ gives rise to a unique skeleton tree $T_s(x) = (V, E, \pi)$: the vertex set V consists of the vertex triangles of x, the edge set E inherits their parent and predecessor relations, and the labeling $\pi : V \to \mathbb{N}$ is defined by their widths. Note that since x contains a dedicated line, it cannot contain an infinite vertex triangle: the southeast corner of that triangle would lie on some coordinate $\vec{n}$ of the dedicated line, and then its shadow would cover all coordinates to the northeast of $\vec{n}$, contradicting the fact that Δ must have an infinite chain of predecessors and parents. The converse of this observation also holds.

Lemma 4.28. *Let $T = (V, E, \pi)$ be a skeleton tree. Then there exists a configuration $x \in X_s$ such that $T_s(x)$ is isomorphic to T. Moreover, x is unique up to translation.*

Proof. Let $v \in V$ be an arbitrary vertex. We construct the configuration x so that the vertex triangle corresponding to v is rooted at the origin. For each vertex $w \in V$, we define the number $n(w) \in \mathbb{Z}$ so that $n(w) = n(u) + 2\pi(u)$ holds for all $(u, w) \in E$. We first set $n(v) = 0$, and since T is a connected tree, this unambiguously defines $n(w)$ for all $w \in V$.

Now, we claim that there exists a configuration $x \in X_s$ such that for all $w \in V$, there is a vertex triangle $\Delta(w)$ of width $\pi(w)$ rooted at $(n(w), n(w))$ in x. We first prove that if we place such vertex triangles on the infinite plane, no triangle overlaps another one or its shadow. For this, it suffices to show that no two vertices $u \neq w \in V$ satisfy $n(u) \leq n(w) \leq n(u) + \pi(u) + \min\left(\pi(u), \frac{1}{2}\pi(w)\right)$, unless $(u, w) \in E$. We call this the *overlap condition*; see Figure 4.6 for a visualization of it.

To prove that the overlap condition holds, assume $n(u) \leq n(w)$, and let $u = v_0, v_1, \ldots, v_k = w$ be the unique minimal-length undirected path from u to w in the tree T. By the properties of skeleton trees, there is an index $m \in \{1, \ldots, k\}$ such that $(v_i, v_{i+1}) \in E$ for all $i < m$ and $(v_{i+1}, v_i) \in E$ for all $i \geq m$. Note that $n(v_i) \geq n(u) + 2(2^i - 1)\pi(u)$ holds for $i \leq m$. Thus, in the case $m = k$, we have

$$n(w) = n(v_m) \geq n(u) + 2(2^m - 1)\pi(u) \geq n(u) + 2\pi(u),$$

with equality only if $m = 1$, that is, $(u, w) \in E$. Since $\frac{1}{2}\pi(w) \geq \pi(u)$ also holds, the overlap condition is satisfied in this case.

Suppose then that $m < k$, so that there are two possibilities for the number $n(v_{m+1})$. First, if v_m is the predecessor of v_{m+1}, then we have $n(w) < n(u)$, so this case cannot hold. Thus v_m is the parent of v_{m+1} and

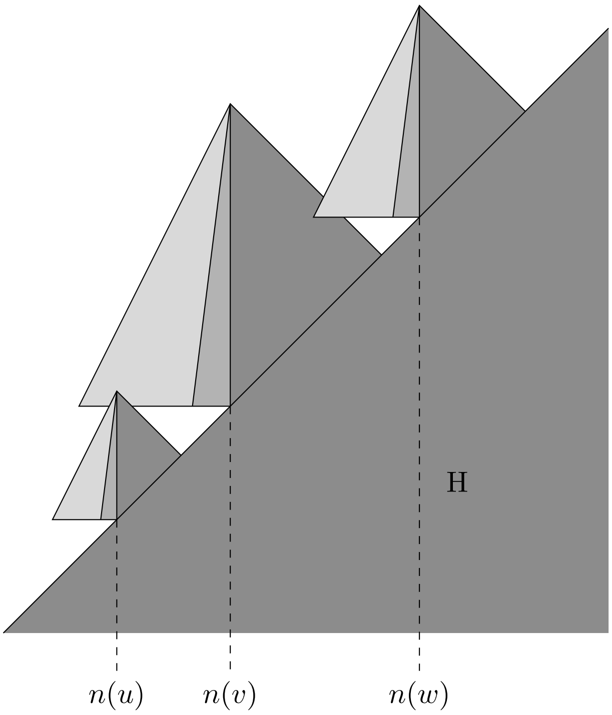

Figure 4.6: The overlap condition can be violated in two ways. On the left, we have $n(u) \leq n(v) < n(u) + 2\pi(u) < n(u) + \frac{1}{2}\pi(v)$, so that the east border of $\Delta(u)$ crosses the south border of $\Delta(v)$. On the right, we have $n(v) \leq n(w) < n(v) + \pi(v) + \frac{1}{2}\pi(w) < \pi(v)$, so that the south border of $\Delta(w)$ intersects the shadow of $\Delta(v)$.

the predecessor of v_{m-1}, and we have $\pi(v_m) = 8\pi(v_{m+1}) = 2\pi(v_{m-1})$. Using this, we obtain that

$$n(v_{m+1}) = n(v_m) - 2\pi(v_{m+1}) = n(v_{m-1}) + \frac{3}{2}\pi(v_{m-1}).$$

Since $n(v_{i+1}) = n(v_i) - \pi(v_i)$ and $\pi(v_i) = 2^{m-i-2}\pi(v_m)$ hold for all $i \geq m+1$, this implies that

$$\begin{aligned} n(w) &= n(v_{m-1}) + \frac{3}{2}\pi(v_{m-1}) - \sum_{i=m+1}^{k-1} \pi(v_i) \\ &= n(u) + 2(2^{m-1} - 1)\pi(u) + \frac{3}{2}\pi(v_{m-1}) - \frac{1}{4}\pi(v_m) + 2\pi(w) \\ &= n(u) + 2(2^{m-1} - 1)\pi(u) + \pi(v_{m-1}) + 2\pi(w). \end{aligned}$$

In the case $m > 1$, the last term is greater than $n(u) + 2\pi(u)$, and if $m = 1$, it is greater than $n(u) + \pi(u) + 2\pi(w)$, so the overlap condition holds in both cases. Furthermore, if w is a leaf, then $\pi(w) = 4$, and the above calculation implies $n(w) - n(v_{m-1}) = \pi(v_{m-1}) + 8$, so that the east corner of the shadow of $\Delta(v_{m-1})$ lies at the coordinate $(n(w) - 8, n(w) - 8)$. Thus, the shadow condition also holds for $\Delta(w)$.

Now, we can construct the configuration x by placing the dedicated line at $\{(a, a) \mid a \in \mathbb{Z}\}$, adding a vertex triangle $\Delta(w)$ of width $\pi(w)$ rooted at $(n(w), n(w))$ for all vertices $w \in V$, adding the shadows of these triangles, and filling the rest of x with the white tiles. If there is a leaf vertex $w \in V$ with an infinite chain of predecessors, then $\Delta(w)$ is the leftmost vertex triangle in x, and we must additionally place an infinite shadow whose east corner lies at $(n(w) - 8, n(w) - 8)$ in order for $\Delta(w)$ to satisfy the shadow condition. Since no two vertex triangles overlap, they inherit their parent and predecessor relations from the skeleton tree T, and the shadow condition is satisfied for all width-4 triangles, the configuration x contains no forbidden patterns of X_s, and thus $x \in X_s$. The condition that $T_s(x)$ is isomorphic to T holds by construction, and it is also easy to see that the position of $\Delta(v)$ completely determines x, so that the configuration is unique up to translation. □

Before moving on to the second layer of X, we note that the skeleton layer X_s is not countable, since there exist uncountably many different skeleton trees. However, the two remaining layers will restrict the structure of X to ensure its countability.

4.4.4 The Hyperarithmetical Layer

The second layer of X is called the *hyperarithmetical layer*, and we denote it by X_h. Its purpose is to simulate the computation of a function $f : \mathbb{N} \to \mathbb{N}$

given by Theorem 4.25, which can then be used to compute the structure of the hyperarithmetical poset $(P, \preceq)$. For this, we define another kind of tree.

Definition 4.29. Let $M = (k, Q, \delta, q_0, q_f)$ be a nondeterministic counter machine. An *M-tree* is a directed tree $T = (V, E, \pi, I)$, where $\pi : V \to \mathbb{N}$ and $I : V \to Q \times \mathbb{N}^k$ are labelings of the vertices, satisfying the following conditions.

- (V, E, π) is a skeleton tree.
- The label of a leaf $v \in V$ is the initial ID $I(v) = (q_0, 0, \ldots, 0)$ of M.
- For every vertex $v \in V$ that has a predecessor $w \in V$, there exists a transition $d \in \delta$ with $I(v) \xrightarrow{d} I(w)$.

Intuitively, an M-tree is simply a skeleton tree that simulates a computation of M in each of its successor chains. Analogously to the skeleton layer, the well-behaved configurations of the hyperarithmetical layer X_h basically encode M-trees into geometric shapes.

Before proceeding to the definition of X_h, we fix the behavior of the counter machine M, the idea for which is the following. The machine will have three special counters, the *poset counter* n_{poset}, the *set counter* n_{set}, and the *branch counter* n_{br}, along with a special *branch state* q_{br}. The poset counter holds an element of the hyperarithmetical poset P, and the highest value of that counter in an M-tree $T = (V, E, \pi, I)$ (with respect to the order $\preceq$) is the *poset value* of that tree. The control layer will later ensure that if $v \in V$ is a vertex of T and $w \in V$ is its child, then the poset counter value of $I(w)$ is strictly lower than that of $I(v)$ in the order $\preceq$. The branch counter and branch state interact with the control layer to achieve this. The set counter, on the other hand, encodes an initial segment of the trace set $\mathrm{Tr}(f) \subset \mathbb{N}$, which is used to compute the set P and the relation $\preceq$.

The program of M is shown in Algorithm 3, and it contains some unconventional notation, which we now explain. First, the notation **in parallel** means that the first block of code should be executed at the same time as the second; the program takes turns executing one step of the first block, one step of the second, one of the first again, etc. The reason for this is that we want the first block to be 'synchronized' across any two computation histories of M, if they make the same nondeterministic choices in the first block. Second, the notation $n_i \hookleftarrow m$ for a counter n_i and a value $m \in \mathbb{N}$, first used on line 3.11, signifies that the counter n_i is given the value m *instantaneously*: the value m is first computed into an auxiliary counter, whose value is then copied to n_i in one step. An ordinary counter machine does not have this functionality, but we can implement is as a special kind of transition. Third, we denote by $2^{\mathbb{N}}_*$ the set of pairs (ℓ, K), where $\ell \in \mathbb{N}$ and $K \subset \{0, 1, \ldots, \ell\}$. The function $e : 2^{\mathbb{N}}_* \to \mathbb{N}$, first used on the same line as the instantaneous assignment operation, encodes an element of $2^{\mathbb{N}}_*$ into a natural number. The details of the encoding function are irrelevant, as long as it is computable, has a computable inverse, and satisfies $e(\ell_1, K_1) \leq e(\ell_2, K_2)$ whenever $\ell_1 \leq \ell_2$. Then there exists an increasing computable function $t : \mathbb{N} \to \mathbb{N}$ with the following property. Given two encodings of pairs $(\ell_1, K_1), (\ell_2, K_2) \in 2^{\mathbb{N}}_*$, a counter machine simulated by the SFT of Construction 3.11 can decide in at most $2^{t(\ell_1+\ell_2)}$ horizontal rows whether the conditions $\ell_1 \leq \ell_2$ and $K_1 = K_2 \cap \{0, 1, \ldots, \ell_1\}$ hold. This function is used on line 3.10, and the reason for including it will become clear in the next subsection.

Finally, we define two oracle Turing machines M_P and $M_\prec$ with the following properties. If the machine M_P is given a number $p \in \mathbb{N}$ as input and the set $\mathrm{Tr}(f) \subset \mathbb{N}$ as an oracle, it decides whether $p \in P$. Similarly, if the machine $M_\prec$ is given two numbers $r, p \in \mathbb{N}$ as input and $\mathrm{Tr}(f)$ as an oracle, it decides whether $r \prec p$ holds. The existence of the machines follows from the definition of the function f. Now, given a finite set $K \subset \mathbb{N}$ and $\ell, p \in \mathbb{N}$, we denote by $M_P^K \vdash_\ell p \in P$ the condition that the machine M_P accepts the input p using the oracle K, and only queries the oracle with values from $\{0, \ldots, \ell\}$. We define $M_P^K \vdash_\ell p \notin P$ and $M_\prec^K \vdash_\ell r \prec p$ similarly, and denote by $M_\prec^K(\ell, p)$ the set of those numbers $r \in \{0, \ldots, \ell\}$ for which $M_\prec^K \vdash_\ell r \prec p$ holds. Note that these notions are computable from ℓ, p, r and K. We also note that on line 3.21, it may be that neither of $M_P^K \vdash_\ell p \notin P$ or $M_P^K \vdash_\ell p \in P$ holds, if the machine M_P needs to query the oracle with some value $t > \ell$.

Let $v \in V$ be a vertex of an M-tree (V, E, π, I) with label $I(v) = (q, n_1, \ldots, n_k)$. We call the state $q \in Q$ the *M-state* of v. Also, the value of the poset counter in $I(v)$ is called the *poset element of* v, and the element of $2^{\mathbb{N}}_*$ encoded in its set counter is the *initial segment computed at* v.

We now move on to the definition of X_h. Its alphabet is depicted in Figure 4.7, and an example configuration in Figure 4.8. In the alphabet, the labels I and I_n of the lines range over the set $Q \times \{0, 1\}^k$.

We now list the forbidden patterns of the hyperarithmetical layer X_h. First, naturally, every 2×2 pattern of tiles where the lines of neighboring tiles do not match is forbidden. Next, in a pattern of shape 2×1 consisting of two tiles with a vertical signal (bottom row, second tile from the right), if the labels of the signals are $(p, a_1, \ldots, a_k)$ and $(q, b_1, \ldots, b_k)$, where $p, q \in Q$ and $a_i, b_i \in \{0, 1\}$, then we require that $p = q$ and $a_i \leq b_i$ for all $i \in \{1, \ldots, k\}$. This means that the labels of a group of vertical lines are decreasing (not necessarily strictly) to the west, and we can use such a group to model an ID of M: its state is the common state of the signals, and the value of counter i is the number of tiles whose i'th bit is 1. Analogously, we require that the labels of a group of horizontal signals are decreasing to the north, so that it also models an ID of M.

Algorithm 3 The program of the counter machine M used in X_h

in parallel
  $\ell = -1$
  $K \leftarrow \emptyset$
  loop
    $\ell \leftarrow \ell + 1$
    choose $b \in \{0, 1\}$
    if $b = 1$ **then**
      if $f(\ell) \neq \min(\ell, \max K)$ **then reject** ▷ Here, $\max \emptyset = \infty$
      $K \leftarrow K \cup \{\ell\}$
    wait for $t(2\ell)$ **steps**
    $n_{\mathrm{set}} \hookleftarrow e(\ell, K)$

$p \leftarrow 0$
repeat
  $p \leftarrow p + 1$
  choose $b \in \{0, 1\}$
until $b = 1$
$n_{\mathrm{poset}} \hookleftarrow p$
loop
  if $M_P^K \vdash_\ell p \notin P$ **then reject**
  else if $M_P^K \vdash_\ell p \in P$ **then**
    for $r \in M_\prec^K(\ell, p)$ **do**
      wait for $r + 3$ **steps**
      $n_{\mathrm{br}} \hookleftarrow r$
      visit state q_{br}

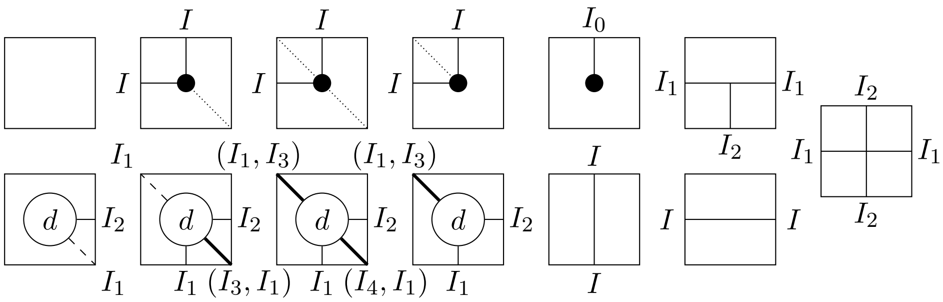

Figure 4.7: The alphabet of the hyperarithmetical layer X_h. The labels I and I_n have the type $Q \times \{0, 1\}^k$, and they represent "slices" of an ID of M. The allowed combinations of these labels, and the forbidden patterns of X_h, are explained in the main text.

The purpose of the tiles with a black circle is to transform a group of horizontal signals to one of vertical signals, while preserving their labels. Conversely, the purpose of the white circles is to transform a group of vertical signals back to horizontal ones, and also compute a transition $d \in \delta$. For this, every 2×2 pattern containing two white circles, connected by a line, with different transitions $d \neq d' \in \delta$, is forbidden. Also, the labels I_1, I_2, I_3, I_4 in the tiles containing white circles are restricted so that a diagonal line of white circles actually implements the transition δ. It is not hard to see that such restrictions can be chosen, since every tile obtains the information about the signals that arrive to its northwest and southeast neighbors, and counter values are either updated in increments or decrements of 1, or copied directly from another counter in case of the special transitions mentioned above.

The labels I_1 and I_2 in the T-junction and crossing tiles have no restrictions, and the purpose of these tiles is to allow one computation to 'merge' into another. The crossing tile must have another crossing tile or a T-junction above it, and the T-junction tile must have an empty tile above it. The label I_0 is the initial ID $(q_0, 0, \ldots, 0)$ of M, and the tile that contains it is called the *seed tile*, since it starts a new computation of M.

Finally, we define a new SFT $Y \subset X_s \times X_h$ that enables the hyperarithmetical layer to interact with the skeleton layer. The idea is that the computational structures of the hyperarithmetical layer will 'attach' themselves to the geometric shapes of the skeleton layer, and that the configurations of Y correspond to M-trees. With this in mind, the additional forbidden patterns of Y are the following. First, a vertex triangle of width more than 4 with no successor is forbidden, and similarly, a vertex triangle of width more than 32 with no child is forbidden. This guarantees that the width of every vertex triangle in Y is a power of 2, and that all leaves of the corresponding skeleton tree have width 4. Second, every tile whose X_h-component contains a black circle, and whose X_s-component is not part of a vertex triangle, is forbidden, and conversely, the southeast corner of every vertex triangle must contain a black circle. Similarly, every tile whose X_h-component is a white circle, and whose X_s-component is neither completely white nor the north corner of a vertex triangle, is forbidden, and the north corner of a vertex triangle must be paired with the fourth tile on the bottom row of Figure 4.7. Finally, we require that in a group of vertical signals, the rightmost one is either a cross or a T-junction, or coincides with the east border of some vertex triangle. Conversely, the east border of every vertex triangle must be paired with a rightmost vertical signal.

The above constraints force the groups of computation signals to travel along the south and east borders of the vertex triangles, so that to each finite vertex triangle Δ in a configuration of Y corresponds an ID $I(\Delta)$ of the counter machine M. In this way, each configuration $y \in Y$ containing

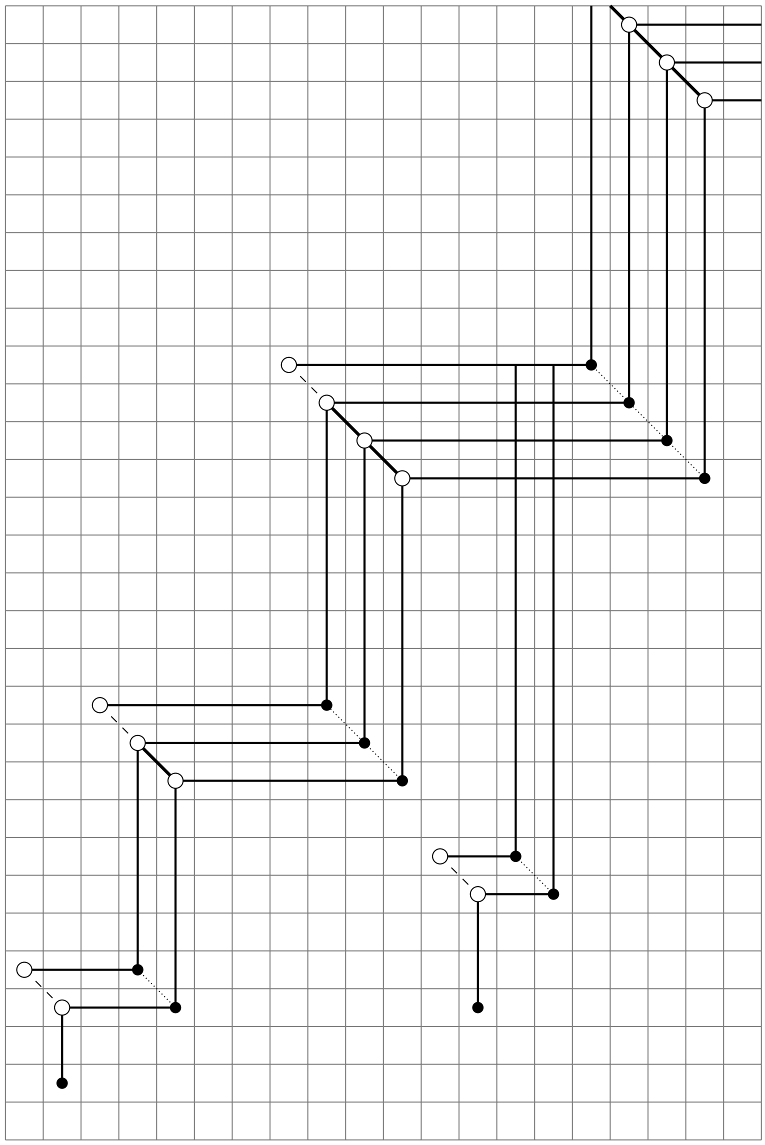

Figure 4.8: A configuration of the hyperarithmetical layer X_h. The white circles have labels from δ (see Figure 4.7). This configuration cannot occur as the second layer of a configuration of Y, since the southernmost black circles cannot be paired with the southeast corners of finite vertex triangles.

a finite vertex triangle corresponds to a unique M-tree $T_M(y)$. As in the previous subsection, the converse also holds.

Lemma 4.30. *Let $T = (V, E, \pi, I)$ be an M-tree. Then there exists a configuration $y \in Y$ such that $T_M(y)$ is isomorphic to T. Furthermore, y is unique up to translation.*

Proof. First, since $T' = (V, E, \pi)$ is a skeleton tree by definition, Lemma 4.28 implies that there exists a configuration $x \in X_s$ of the skeleton layer such that $T_s(x)$ is isomorphic to T', and it is unique up to translation. Thus it suffices to prove that there exists a unique configuration $x' \in X_h$ of the hyperarithmetical layer such that $(x, x') \in Y$ and the tree $T_M(x, x')$ is isomorphic to T. Our construction is visualized in Figure 4.9.

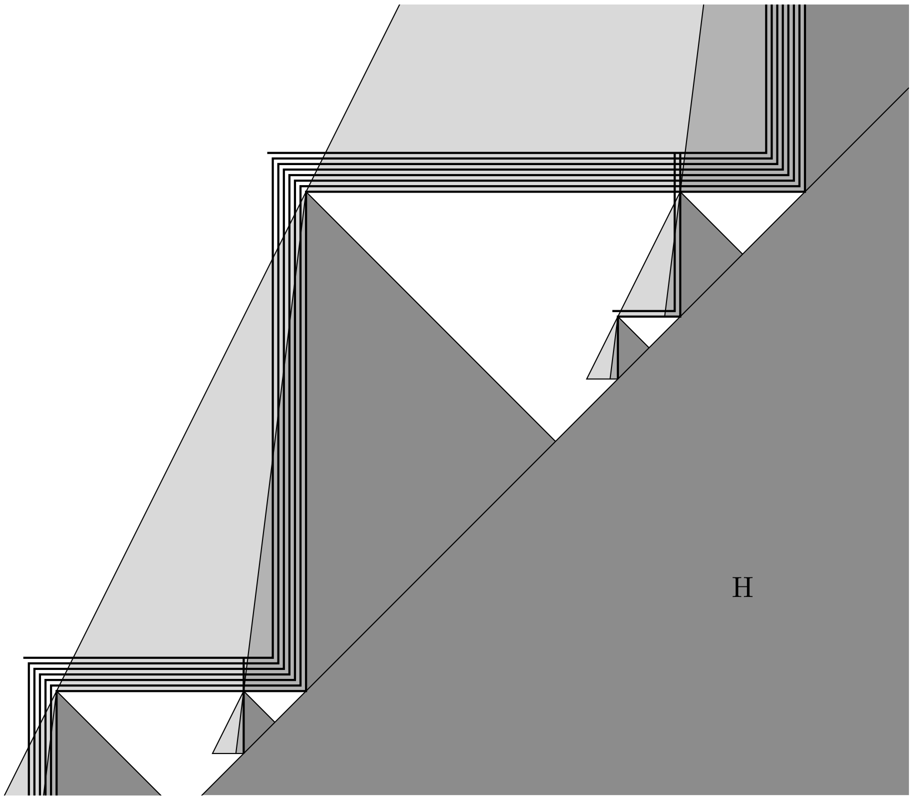

Figure 4.9: Constructing the configuration $x' \in X_h$.

First, take a vertex triangle Δ of x of width 2^n for some $n \geq 2$. We consider Δ as an element of the vertex set V, so that $I(\Delta)$ is an ID of the machine M. We attach to the east border of Δ a group of $n-1$ vertical signals, the eastmost one being superimposed on the border, containing the ID $I(\Delta)$. If $n \geq 3$, we analogously attach a group of $n-1$ horizontal signals to the south border of Δ. In the intersection points of these groups we place $n-1$ black circles.

Now, if Δ has a predecessor Δ', then the north corner of Δ is attached to the west corner of Δ', and since T is a skeleton tree, there exists a transition $d \in \delta$ such that $I(\Delta) \overset{d}{\to} I(\Delta')$ holds. We may assume that the southeast corner of Δ' is located at the origin. It is not hard to see from the geometry of the skeleton layer X_s that the rectangular domain $D = [-2\pi(\Delta), 0] \times [0, 2\pi(\Delta) - 1] \subset \mathbb{Z}^2$ of the configuration x contains only tiles of the vertex triangle Δ and white tiles. This means that in the configuration x', we can extend the vertical signals of Δ and the horizontal signals Δ' into the domain D, and place white circles with label d at their intersections. Finally, if Δ' is the parent of Δ instead of its predecessor, we can simply attach the vertical signals of Δ into the horizontal signals of Δ' using the T-junction and crossing tiles.

The hyperarithmetical tree $T_h(x, x')$ is clearly isomorphic to T, and the configuration x' is completely determined by T and x, since the signals of the hyperarithmetical layer can only be attached to vertex triangles. $\square$

As the skeleton layer X_s, the SFT Y is not countable. In fact, if a configuration $x \in X_s$ contains infinitely many vertex triangles with children, then even the set $\{x' \in X_h \mid (x, x') \in Y\}$ is uncountable, since the infinitely many simulated computations of M can be chosen independently, and the nondeterministic choices of M are visible in the corresponding configuration x'.

4.4.5 The Control Layer

Finally, we define the control layer X_c. The purpose of this layer is to link the different computation paths of M together in a configuration. As with the previous layers, we define a class of trees corresponding to X_c (or X, to be more exact).

Definition 4.31. Let $T = (V, E, \pi, I)$ be an M-tree. We say that T is *controlled*, if the following conditions hold for all vertices $v \in V$.

- v has a child if and only if the M-state of v is q_{br}.
- If v has a child $w \in V$, then the poset element of w is equal to the branch counter value of v, and the initial segments $(\ell_1, K_1), (\ell_2, K_2) \in 2^{\mathbb{N}}_*$ computed at w and v, respectively, satisfy $K_1 = K_2 \cap \{0, 1, \ldots, \ell_1\}$.

The class of controlled M-trees is denoted by $\mathcal{C}_M$.

These constraints have some important consequences. First of all, for an edge $(v, w) \in E$ of a controlled M-tree $T = (V, E, \pi, I)$, the set computed at v is a subset of the set computed at w. This implies that there is a single set $N \subset \mathbb{N}$ such that the set computed at any vertex of T is a subset of N. Because of the checks in Algorithm 3, we necessarily have $N = \mathrm{Tr}(f)$. This means that in any infinite sequence of predecessors, for large enough $\ell \in \mathbb{N}$, the condition $M_P^K \vdash_\ell p \in P$ in the algorithm is actually equivalent to $p \in P$, and similarly for $M_P^K \vdash_\ell p \notin P$ and the set $M_\prec^K(\ell, p)$. Since the branch counter of a vertex $v \in V$ gets its value from the set $M_\prec^K(\ell, p)$, this in turn means that the poset element of the possible child of v is strictly less than that of v in the order $\preceq$. These observations give us the following result.

Lemma 4.32. *Let $T = (V, E, \pi, I)$ be a controlled M-tree, let $v, w \in V$ be two vertices with respective poset elements $p, r \in P \cup \{0\}$, and suppose that $(v, w) \in E$. If $p = 0$, then w is a predecessor of v. If $p \in P$, then either w is a predecessor of v and $r = p$, or w is a parent of v and $r \prec p$.*

Let us define a function $\gamma : \mathcal{C}_M \to P \cup \{0\}$ as follows. Consider a controlled M-tree $T = (V, E, \pi, I) \in \mathcal{C}_M$. If the set

$$R_T = \{p \in P \mid \text{the poset element of some } v \in V \text{ is } p\} \subset P$$

contains a greatest element $p \in P$ with respect to $\preceq$, then $\gamma(T) = p$. Otherwise, $\gamma(T) = 0$. Note that we assumed $0 \notin P$ in the beginning of the construction, so the two cases cannot be confused. Next, we show that this function is actually a complete invariant for the isomorphism of controlled M-trees.

Lemma 4.33. *Two controlled M-trees are isomorphic if and only if they have the same images under γ. In particular, the class of controlled M-trees is countable up to isomorphism.*

The proof of this result is the first and only place where we use the ascending chain condition of the poset $(P, \preceq)$.

Proof. It is clear that isomorphic controlled M-trees have the same γ-value. For the converse, let $T = (V, E, \pi, I) \in \mathcal{C}_M$ be a controlled M-tree. First, suppose that $R_T = \emptyset$, that is, the poset element of every vertex of T is 0. Lemma 4.32 implies that there are no parent relations in T, so it is just an infinite chain of predecessors, where the computation is forever stuck in the first loop of Algorithm 3. Thus the tree T is determined up to isomorphism in this case.

Suppose then that $R_T \subset P$ is nonempty. We show that in this case R_T contains a greatest element. Let $v, w \in V$ be two vertices with respective poset elements $p, r \in P$. We denote $v \longrightarrow w$ if there exists a directed path from v to w in the tree T. By Lemma 4.32, the condition $v \longrightarrow w$ implies

$p \preceq r$, and $w \longrightarrow v$ implies $r \preceq p$. If these conditions do not hold, then there exists a third vertex $u \in V$ such that $v, w \longrightarrow u$. Lemma 4.32 implies that the poset element of u is an upper bound for p and r with respect to $\preceq$. Thus, R_T is a directed subset of P, meaning that any pair of its elements has a upper bound in it. Since P has the ascending chain condition, the set R_T contains a greatest element, which is then $\gamma(T)$.

Now, let $v \in V$ be a leaf of T, let $C = (v_0, v_1, \ldots)$ be the maximal chain of predecessors starting from $v = v_0$, and suppose further that the poset element of some v_n is $\gamma(T)$. Such objects exist, since every vertex of T is either a leaf or a predecessor. By Lemma 4.32, none of the vertices of C have parents, so C is an infinite chain, and the labels of its vertices are completely determined from the facts that the poset element of v_n is $\gamma(T)$, and the set computed at every v_i is a subset of $\mathrm{Tr}(f)$. In particular, the set $C_{\mathrm{br}} \subset C$ of those vertices whose M-state is q_{br} is determined.

Now, every vertex $w \in C_{\mathrm{br}}$ has a child whose poset element $p \in P$ equals the branch counter value of w, and satisfies $p \prec \gamma(T)$. This child is the last vertex in a maximal predecessor chain $C_w = (w_0, \ldots, w_\ell)$ whose length and labels are completely determined by the vertex w. In particular, those vertices of C_w whose M-state is equal to q_{br} are determined, analogously to the set C_{br}. Continuing this argument inductively, we can show that the structure of the tree T, and thus it isomorphism class, is determined by the value $\gamma(T)$. This finishes the proof. $\square$

Lemma 4.34. *The function γ is surjective.*

Proof. Let $p \in P \cup \{0\}$ be arbitrary. We construct a controlled M-tree T such that $\gamma(T) = p$. We already saw in the proof of Lemma 4.33 that $\gamma(T) = 0$ is satisfied by the infinite predecessor chain that never chooses a value for the poset counter, so we assume $p \in P$.

First, let $C = (v_0, v_1, \ldots)$ be an infinite chain of vertices, each a successor of the next one, whose labels give the computation history of M that chooses p as the value of the poset counter, and correctly chooses initial segments of the set $\mathrm{Tr}(f)$ into the set counter. Let $n \in \mathbb{N}$ be such that the M-state of the vertex v_n is q_{br}, and note that the width of v_n is exactly 2^{n+2}. We add to v_n a child, and to that child, a chain of successors of length $n-4$, so that the smallest one has width 4. Denote this chain by $C' = (w_0, \ldots, w_{n-3})$, where w_0 if the leaf of width 4, and w_{n-3} is the child of v_n. Let $r \in P$ be the value of the branch counter at v_n, and note that $r \prec p$. Because the machine M waits for $r+1$ steps on line 3.23 before entering the state q_{br}, we know that the length of the chain C' is at least r. Thus, there is enough time for the simulated machine M to choose r as the value of its poset counter in the chain C', and we choose the labels of the chain so that this is the case. This means that the poset element of w_{n-3} is r. We also choose the set counters in the chain C' to properly contain initial segments of $\mathrm{Tr}(f)$, and then the two conditions of Definition 4.31 hold for the vertex v_n and its child w_{n-3}.

We repeat the above operation of adding a new predecessor chain for each vertex with M-state q_{br}. Of course, this may introduce new such vertices, and we iteratively repeat the procedure for them too. It is easy to see that as a limit of this process, we obtain a controlled M-tree $T \in \mathcal{C}_M$ with $\gamma(T) = p$. $\square$

We now move on to actually defining the control layer X_c. For this, we define another counter machine M_c, called the *control machine*, which is simpler than M. Its behavior is defined in Algorithm 4. The idea is that we superimpose a simulated computation of M_c on every vertex triangle Δ that has a child Δ', which guarantees that the two conditions of Definition 4.31 hold locally. The machine is given four inputs: the set counter $e(\ell_1, K_1)$ of Δ', the set counter $e(\ell_2, K_2)$ of Δ, the poset counter p_1 of Δ', and the branch counter p_2 of Δ. Note that since the computation of M simulated in Δ' has run for 3 fewer steps than the one simulated in Δ, we necessarily have $\ell_1 \leq \ell_2$.

Algorithm 4 The program of the control machine M_c

1: **input** $e(\ell_1, K_1), e(\ell_2, K_2) \in \mathbb{N}$**,** $p_1, p_2 \in \mathbb{N}$
2: **if** $p_1 \neq p_2$ **then reject** ▷ Instantaneous
3: **else if** $\ell_1 > \ell_2$ **then reject**
4: **else if** $K_1 \neq K_2 \cap \{0, 1, \ldots, \ell_1\}$ **then reject** ▷ Total time $t(2\ell_2)$
5: **else accept**

To construct the control layer, let X_{M_c} be the SFT that simulates the control machine M_c, as given by Construction 3.11. The control layer X_c is exactly X_{M_c}, but mirrored along the y-axis, and with the modification that the background letters 0 and 1 are merged into one new letter #. We also allow the letter # to occur as the north or west neighbor of any other letter, and we allow the simulated machine to accept its input without producing a forbidden pattern (rejecting the input still produces one). Finally, we allow the counters to be initialized with arbitrary values. See Figure 4.10 for an example configuration of X_c.

To define the final SFT $X \subset Y \times X_c \subset X_s \times X_h \times X_c$, we introduce some additional forbidden patterns. The idea is that the outer half (the light gray part) of every vertex triangle with a child will contain a computation cone, where a computation of M_c is simulated. First, all tiles of the skeleton layer X_s which are not part of an outer triangle (the light gray part of a vertex triangle) must be paired with the new letter # on the control layer. Second, the east corner of the base of a computation cone must be paired with the southeast corner of an outer triangle with a child, or the fourth tile on the

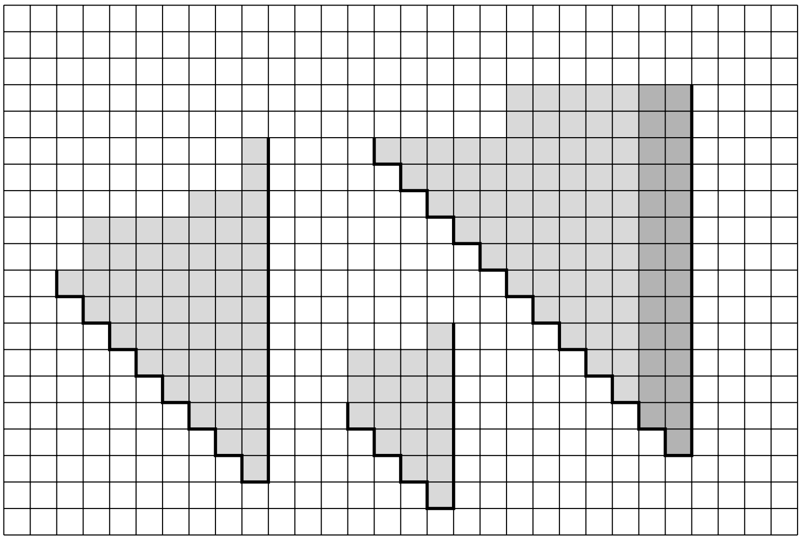

Figure 4.10: A configuration of X_c. The gray areas are truncated computation cones surrounded by #-letters (the white tiles). The rightmost cone has a counter initialized at the value 2. The zig-zag heads are not shown.

bottom row of Figure 4.4, and conversely, this tile can only be paired with said corner of the computation cone. Also, to enforce the first condition of Definition 4.31, we require that the aforementioned tile can only be paired with a signal carrying the branch state q_{br} on the hyperarithmetical layer, and this signal cannot be paired with the third tile on the bottom row of Figure 4.4. This cone must cover as large an area as possible within the outer triangle, which is achieved by requiring that the cone can only be truncated at the border of said triangle. Now, the southwest border of the computation cone intersects the horizontal signals of the vertex triangle Δ that contains it, and the vertical signals of its child. The four inputs of the simulated machine M_c are initialized by forcing a counter symbol to be placed at the intersection of the appropriate signal with the border of the cone. This concludes the definition of X; see Figure 4.11 for a schematic view of the way in which the control layer interacts with Y.

Now, for any configuration $(x, y, z) \in X$, recall that the first two layers $(x, y) \in Y$ define an M-tree $T_M(x, y)$. We show that this tree is necessarily controlled, and that all controlled M-trees arise in this way.

Lemma 4.35. *For all configurations $(x, y, z) \in X$ such that $x \in X_s$ contains a finite vertex triangle, the M-tree $T_M(x, y)$ is controlled.*

Proof. It is enough to show that the two conditions of Definition 4.31 hold for all vertex triangles Δ in x. The first condition, that Δ has a child if

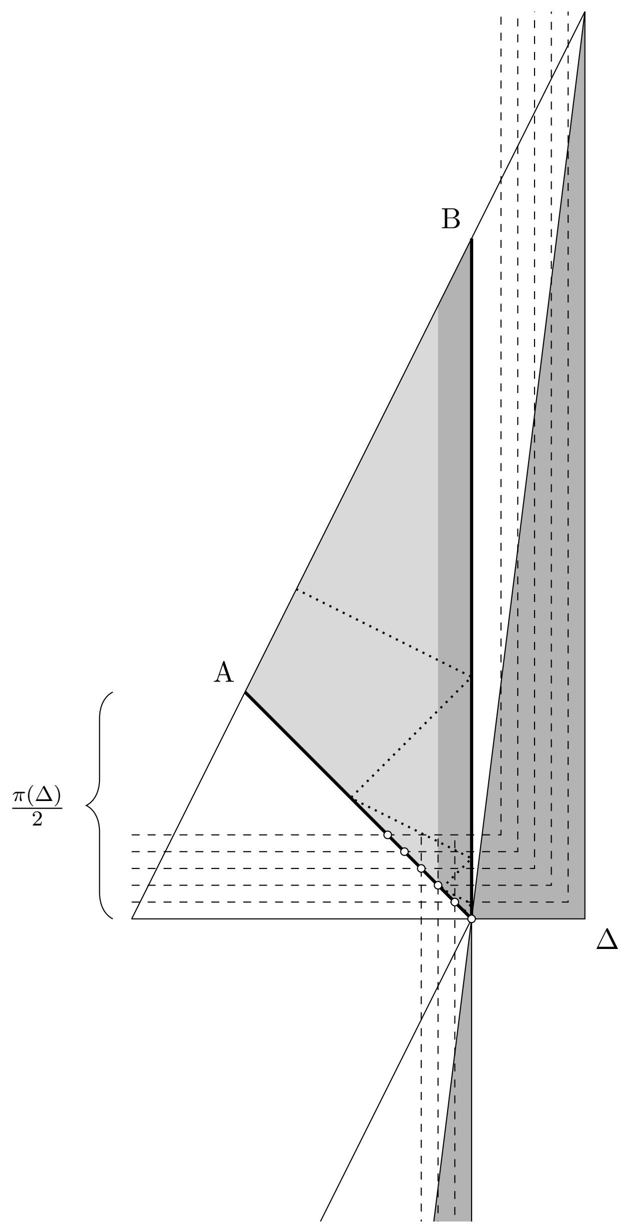

Figure 4.11: The interplay of X_c and Y, drawn out of scale. The light gray area is a truncated computation cone on the control layer, and the dotted line is the zig-zag head. The dashed lines are the signals of the two vertex triangles. The signals intersect the border of the cone at the white circles, and define the initial value of some of the counters (shown in dark gray).

and only if its M-state is q_{br}, follows from the condition that the tile at the southwest corner of the inner triangle of Δ contains the top corner of another vertex triangle Δ' if and only if the signal that it is paired with has state q_{br}.

For the second condition, suppose that the child Δ' exists, and consider the simulated control machine M_c running in the vertex triangle Δ. Let $(\ell_1, K_1), (\ell_2, K_2) \in 2^{\mathbb{N}}_*$ be the initial segments computed at Δ' and Δ, let $p_1 \in P \cup \{0\}$ be the poset counter of Δ', and let $p_2 \in P$ be the branch counter of Δ. These values are given as input to M_c, and in one step it is able to decide whether $p_1 = p_2$. We now show that the computation cone of M_c has enough space for it to complete its computation, which implies that it eventually accepts its inputs, which in turn is equivalent to the second condition of Definition 4.31 holding locally at Δ.

For this, let $\pi(\Delta) = 2^{n+2}$ be the width of Δ, so that its M-state $I(\Delta)$ is the n'th step in the simulated computation of M. An easy calculation shows that the southwest border of the computation cone of M_c meets the hypotenuse of Δ at distance exactly $d = \pi(\Delta)/2 = 2^{n+1}$ from the south border of Δ (point A in Figure 4.11). Because of line 3.10, we know that the machine M has computed for at least $t(2\ell_2)$ steps, or in other words, $n \geq t(2\ell_2)$. By the definition of the function t, this implies that the simulated machine M_c has enough space to complete its computation, which concludes the proof. $\square$

Lemma 4.36. *Let $T = (V, E, \pi, I)$ be a controlled M-tree. Then there exists a configuration $(x, y, z) \in X$, unique up to translation, such that $T_M(x, y)$ is isomorphic to T.*

Proof. The existence and uniqueness of the configuration $(x, y) \in Y$ is given by Lemma 4.30, so it remains to show it for the final component $z \in X_c$ in the control layer.

Consider a vertex triangle Δ of the configuration x. If Δ has no child, we fill its X_c-components in z with the background letter #, which introduces no forbidden patterns. If Δ has a child Δ', we pair the base of a computation cone with the southwest corner of its inner triangle, and extend the cone to the hypotenuse of its outer triangle, as shown in Figure 4.11. If the control machine M_c simulated in this cone has enough time to complete its computation, it accepts its inputs, since the conditions of Definition 4.31 hold for Δ and Δ'. We fill the remaining part of the configuration z with the letter #, and it is easy to see that the resulting configuration (x, y, z) is in X. The uniqueness of z follows from the facts that the bases of the computation cones have to be placed exactly at the aforementioned corners, they can only be truncated at the borders of the triangles, and the algorithm of M_c is deterministic. $\square$

The above lemmas enable us to define a function $\phi : P \to \mathrm{SP}(X)$, where $\mathrm{SP}(X)$ is the subpattern poset of X, as follows. Given an element $p \in P$, let $T \in \mathcal{C}_M$ be a controlled M-tree such that $\gamma(T) = p$. The existence of T is guaranteed by Lemma 4.34, and it is unique up to isomorphism by Lemma 4.33. By Lemma 4.36, there exists a configuration $(x, y, z) \in X$, unique up to translation, such that $T_M(x, y)$ is isomorphic to T. Again by Lemma 4.36, this is equivalent to $\gamma(T_M(x, y)) = p$. We define $\phi(p) \in \mathrm{SP}(X)$ to be the $\approx$-equivalence class of (x, y, z), which is uniquely determined by p. For clarity, we denote $T_M(x, y, z) = T_M(x, y)$ in the following.

Proposition 4.37. *The function ϕ is an order-embedding of P into* $\mathrm{SP}(X)$*.*

Proof. First, let $p, q \in P$ be two elements such that $\phi(p) \leq \phi(q) \in \mathrm{SP}(X)$. Then there exist $x \in \phi(p)$ and $y \in \phi(q)$ such that $\gamma(T_M(x)) = p$ and $\gamma(T_M(y)) = q$. Now, $\gamma(T)$ is defined as the largest element of P that occurs as the poset element of any vertex of $T \in \mathcal{C}_M$. This means that p is the poset element of some vertex triangle Δ in x. Since $x \leq y$ in the subpattern order, the pattern of Δ also occurs in y, which implies $p \preceq \gamma(T_M(y)) = q$. In particular, if we have $\phi(p) = \phi(q)$, then $p = q$, which implies that ϕ is injective.

Now, let $p, q \in P$ be such that $p \prec q$. We show that $\phi(p) \leq \phi(q)$ in the ordering of the subpattern poset $\mathrm{SP}(X)$, and for that, let $x \in \phi(p)$ and $y \in \phi(q)$ be again such that $\gamma(T_M(x)) = p$ and $\gamma(T_M(y)) = q$. Then q is the greatest element of P occurring as the poset element of the vertex triangles of y, so there exists an infinite predecessor chain in the tree $T_M(y)$ where the poset element is eventually chosen as q by the simulated machine M. Since $p \prec q$, the machine enters the branch state q_{br} infinitely many times with the branch counter value p, and attached to each of the vertices where this happens, we find a predecessor chain where the poset element is eventually chosen to be p. The length of these chains grows without bound, and as a limit of the corresponding chains of vertex triangles in y, we find a configuration $z \in X$ with an infinite predecessor chain of vertex triangles with eventual poset element p. It is easy to see that $\gamma(T_M(z)) = p$, and then z is a translate of x by Lemma 4.33 and Lemma 4.36. In particular, we have $x \approx z \leq y$, so that $\phi(p) \leq \phi(q)$.

We have shown that $\phi(p) \leq \phi(q)$ implies $p \preceq q$ and $p \prec q$ implies $\phi(p) \leq \phi(q)$ for all $p, q \in P$, and in particular, ϕ is injective. Furthermore, since ϕ is a function, $p = q$ trivially implies $\phi(p) = \phi(q)$, so that ϕ is indeed an order-embedding. $\square$

Having obtained the desired order-embedding, we still need to analyze the elements of $\mathrm{SP}(X)$ that are not in the range of ϕ.

Proposition 4.38. *Each $\leq$-equivalence class in* $\mathrm{SP}(X)$ *which is not in the image of* ϕ *has height at most* 3 *in* $\mathrm{SP}(X)$*. Moreover, the union of these classes is a countable subset of* X*.*

Proof. Let $\xi = (x, y, z) \in X$ be a configuration whose $\leq$-equivalence class $C = [\xi]$ is in $\mathrm{SP}(X) \setminus \phi(P)$. First suppose that the X_s-component x contains a finite vertex triangle. Then the poset counter of every vertex triangle in ξ is necessarily 0, and in the proof of Lemma 4.33 we saw that the M-tree $T_M(\xi)$ consists of a single infinite predecessor chain, and the simulated computation of M is also completely determined. This implies that the configuration is determined up to translation, so the class C is countable, and the lower segment $\{C' \in \mathrm{SP}(X) \mid C' < C\}$ contains only $\leq$-equivalence classes of configurations without finite vertex triangles. Also, since no triangle in the configuration x has a child, the control layer z does not contain a computation cone.

Suppose then that x contains no finite vertex triangles. We proceed with a case analysis on the structure of the configuration. In the following, let $\eta = (x', y', z') \in X$ be a configuration with $\eta < \xi$.

Claim 4.39*.* The configuration x contains at most two infinite vertex triangles, which have a total of at most one east border and at most one south border.

Proof of claim. An infinite vertex triangle is either an infinite cone that opens in one of three directions, an infinite half plane, or the entire configuration. Thus, if x contained three infinite vertex triangles, they would have to be two cones, and either a cone or a half plane. If one of the infinite cones contains a southeast corner, then that corner is attached to an H-colored half plane, and no other vertex triangles fit into x. Also, if an infinite cone contains a southwest corner, then the north corner of another cone is attached to it. Thus, one of the infinite cones has a north corner, and another then necessarily has a southwest corner attached to it. But then it is easy to see that x has no room for a third infinite vertex triangle. $\square$

In particular, the configuration x' has at most one infinite vertex triangle, which may be either a half plane or the entire configuration. In fact, x' does not contain a corner of any shape: a vertex triangle, an inner triangle or a shadow. From this it follows that any configuration of the skeleton layer X_s that is below x' is uniform.

Claim 4.40*.* The configuration y contains at most 4 groups of signals, of which at most two are horizontal and at most two are vertical. If x contains no south border of a vertex triangle, y has at most one horizontal group, and similarly for east borders and vertical groups.

Proof of claim. It is not hard to see that y can contain at most one vertical group without a rightmost signal. These rightmost signals are attached to the east borders of distinct infinite vertex triangles in x, and the claim for vertical groups follows from Claim 4.39. The case of horizontal groups is completely analogous. □

The above result also implies that y contains a bounded number of 'exceptional' tiles with white or black circles, that is, ones that lie on either end of the diagonal rows of circles, or next to the finitely many signals that encode the counter values of the simulated machine M. From this it follows that y' contains no such exceptional tiles.

Claim 4.41. The configuration z contains at most two (necessarily infinite) computation cones, at most one of which has a base.

Proof of claim. The latter claim follows from Claim 4.39 and the fact that every base of a computation cone must be attached to the south border of a vertex triangle. Also, we claim that z contains at most one west corner of a computation cone (point A in Figure 4.11), and at most one north corner (point B in the figure). Suppose for a contradiction that there are two north corners. They lie on the same line of slope 2, and between them we have a west corner of a computation cone. Borders of the cone extend south from the upper north corner and southeast from the west corner, which meet at the base of the cone. But then we have a finite computation cone inside a finite vertex triangle, a contradiction. Thus there is at most one north corner, and similarly, at most one west corner.

Finally, we show that z cannot contain both the base of a computation cone, and the west corner of one. Namely, if z contains a base, then x contains the south border of a vertex triangle, and the north corner of another vertex triangle. The west corner of a computation cone must be attached to the northwest border of a vertex triangle, but this is impossible in our case, since the vertex triangle extends infinitely to the south. This finishes the proof the the claim. □

As a corollary, the configuration z' does not contain any corner of a computation cone. Now, we focus on the structure of the configuration $\eta = (x', y', z')$.

Claim 4.42. If the configuration x' contains a border of a vertex triangle, then y' does not contain both horizontal and vertical signals, and z' does not contain a border of a computation cone. Also, if y' contains both horizontal and vertical signals, then z' does not contain an east border of a computation cone.

Proof of claim. We know that the configuration y' contains no exceptional tiles with circles, so if it contains both horizontal and vertical signals, then it contains a two-way infinite diagonal line of circles. This line necessarily crosses the border of the vertex triangle in x', which produces a forbidden pattern, since black circles are only allowed inside vertex triangles, and white ones outside them. On the other hand, if z' contains the border of a computation cone, then this border is also two-way infinite and crosses the border of the vertex triangle, producing a forbidden pattern.

For the latter claim, observe that y and z only contain finitely many lines of circles and borders of computation cones, which thus cross in only finitely many coordinates. This implies that no such crossings are present in η. $\square$

Now we enumerate the possibilities for the contents of η.

1. The configuration x' contains an infinite vertex triangle in the shape of a half plane. In this case, y' contains only horizontal or only vertical signals, and z' contains no border of a computation cone. Then z' is in fact the uniform configuration containing only #-letters. Also, η contains no crossings of 'exceptional' signals in y' (ones whose neighbors have different labels, or bordermost ones) with the triangle border in x', since there are only finitely many of those in ξ. Thus η is periodic in one direction and eventually periodic in all others, and its height in $\mathrm{SP}(X)$ is 1.

2. The configuration x' contains an infinite vertex triangle that fills it completely. Then, x' consists of either an inner or an outer triangle. In the former case, the configuration z' is uniform, and in the latter, it may contain a border of a computation cone, and a single sweep of the zig-zag head to the border and back. In both cases, the configuration y' contains no exceptional signals, but may contain an infinite line of black circles. A moment's reflection shows that all configurations below η are then periodic in some direction and eventually periodic in others, so the height of η is at most 2. Furthermore, if the configuration z' contains no computation cone, then η has height at most 1.

3. The configuration x' contains no vertex triangles. In this case, z' is again uniform, and y' may contain an infinite line of white circles, or some exceptional signals, but not both, since there are only finitely many intersections of circle lines and exceptional signals in y. This implies that η is periodic in some direction and eventually periodic in others, and has height at most 1.

We have shown that the height of η in the subpattern poset $\mathrm{SP}(X)$ is at most 2 in all cases. This implies that the height of ξ is at most 3 in the case that it contains no finite vertex triangle. Also, if ξ contains a finite vertex triangle, then it contains no computation cone, so the height of η is at most 1. Thus ξ has height at most 3 also in this case.

Finally, we show that the possibilities for the configurations ξ and η, and those below them, amount to a countable subset of X. Namely, if ξ contains a finite vertex triangle, it is defined up to translation, and if not, then it can be uniquely defined from the shapes and positions of the at most two infinite vertex triangles in x, the positions and labels of the at most 4 groups of signals in y, and the shapes and positions of the at most two computation cones in z and the counters and zig-zag heads in them. Note that the contents of a computation cone with a base are completely determined by the signals in y, since the control machine M_c is deterministic. The configurations below η are even simpler. □

What remains is to collect the above results into a proof of the main theorem of this section.

Proof of Theorem 4.26. We have constructed a two-dimensional SFT X and a function $\phi : P \to \mathrm{SP}(X)$, which we proved to be an order-embedding in Proposition 4.37. Moreover, for each $p \in P$, Lemma 4.33 and Lemma 4.36 imply that the image $\phi(p) \in \mathrm{SP}(X)$ consists of translations of a single configuration of X, and is thus countable. As a countable union of countable sets, the image $\phi(P) \subset X$ is countable. On the other hand, Proposition 4.38 shows that $X \setminus \phi(P)$ is also countable, so that X is a countable SFT. Proposition 4.38 also states that every configuration in the above set has height at most 3 in the subpattern poset. □

4.4.6 Further Results

Because almost all imaginable posets are hyperarithmetical, we obtain the following results as direct corollaries of Theorem 4.26. They were proved in [ST13] using several different constructions. In particular, the first result was proved using a simpler version of the above.

Corollary 4.43. *There exists a countable SFT $X \subset A^{\mathbb{Z}^2}$ such that an infinite descending chain can be order-embedded into* $\mathrm{SP}(X)$*.*

Corollary 4.44. *There exists a countable SFT $X \subset A^{\mathbb{Z}^2}$ such that every finite poset can be order-embedded in* $\mathrm{SP}(X)$*.*

In addition to countable SFTs, it is interesting to study the subpattern posets of countably covered sofic shifts, which constitute a strictly larger class. The following analogue of Theorem 4.26 holds for these subshifts.

Proposition 4.45. *Let $(P, \preceq)$ be a hyperarithmetical poset with the ascending chain condition. Then there exists a two-dimensional countably covered* *sofic shift $Y \subset \{0,1\}^{\mathbb{Z}^2}$ on the binary alphabet, and an order-embedding $\psi : P \to \mathrm{SP}(Y)$, such that $\mathrm{SP}(Y) \setminus \psi(P)$ is a three-element chain.*

Proof. Let $X \subset A^{\mathbb{Z}^2}$ be the countable SFT given by Theorem 4.26 for the poset $(P, \preceq)$, and let $\phi : P \to \mathrm{SP}(X)$ be the associated order-embedding. Define a block map $g : X \to \{0,1\}^{\mathbb{Z}^2}$ as follows. For each configuration $x \in X$ and $\vec{n} \in \mathbb{Z}^2$, if the X_h-layer of the tile $x_{\vec{n}}$ contains a black circle and marks the top of the poset counter of a simulated machine M, then $g(x)_{\vec{n}} = 1$. In all other cases, $g(x)_{\vec{n}} = 0$. In this way, every finite vertex triangle Δ of x is marked with exactly one letter 1, and from its position we can infer the poset element of Δ, if we know the position of its southeast corner. Moreover, the set $S = \{x+y \mid \text{a vertex triangle of } x \text{ has its southeast corner at } (x,y)\}$ can be inferred from the image $g(x)$. Now, the poset element of every leaf of x is 0, so the positions of all the leaves can be determined from S. After that, the entire configuration of x can be inferred, so that g is injective on the set of configurations with finite vertex triangles.

We define $Y = g(X)$, and then Y is a countably covered sofic shift. Since g respects the subpattern order of X and Y, we can lift it to the subpattern posets and define an order-preserving function $\tilde{g} : \mathrm{SP}(X) \to \mathrm{SP}(Y)$. We can then define the desired order-embedding by $\psi = \tilde{g} \circ \phi$.

It remains to analyze the remaining set $\mathrm{SP}(Y) \setminus \psi(P)$. If a configuration $y \in [y] \in \mathrm{SP}(Y) \setminus \psi(P)$ contains two occurrences of 1, then it has a preimage $x \in X$ under g that contains a finite vertex triangle. Then x is necessarily a translate of the configuration with an infinite computation of M that never chooses a nonzero value for its poset counter. Thus y is determined up to translation and has height 2, and below it we find only the configurations with a single 1, and the all-0 configuration. □

4.5 Subpattern Posets and The Bounded Signal Property

We now prove that a countable SFT, or even an uncountable downward deterministic SFT, cannot be used for the construction in the previous section, if it has the downward bounded signal property. More precisely, we show that the subpattern posets of such subshifts have the descending chain property. If both the assumption of countability and the assumption of downward determinism are removed, an infinite downward chain is possible, examplified by the horizontally periodic SFT

$$\{x \in \{0,1\}^{\mathbb{Z}^2} \mid x = \sigma^{(1,0)}(x)\}.$$

Instead of proving the theorem for the bounded signal property directly, we prove the more natural general result that if an SFT is either deterministic with countable projective subdynamics, or is itself countable, then its chains cannot be much longer than the Cantor-Bendixson rank of its projective subdynamics. In the rest of this section, a deterministic tiling system means one where each row actually determines the one below it.

Definition 4.46. Let λ be an ordinal. An SFT $X \subset A^{\mathbb{Z}^2}$ has the *PCB*(λ) property if its horizontal projective subdynamics is ranked and has Cantor-Bendixson rank at most λ.

By Proposition 4.6 and Lemma 3.9, the bounded signal property implies the PCB(r)-property for some finite $r \in \mathbb{N}$. Recall that for a two-dimensional configuration $x \in A^{\mathbb{Z}^2}$ and $i \in \mathbb{Z}$, the notation x_i refers to the i'th row of x, seen as a one-dimensional configuration.

Definition 4.47. Let $X \subset A^{\mathbb{Z}^2}$ be an SFT, and denote its projective subdynamics by $Y = \mathrm{Proj}(X)$. We say X has the *R property* if for all rows $y \in Y$, there exists a number $k \geq 2$ and a configuration $z \in Y^k$ such that $z_1 = y$, $z_k \in \mathcal{O}(y)$ and $z_i \notin \mathcal{O}(y)$ for all $i \in [2, k-1]$ with the following property: For any configuration $x \in X$ and indices $m < n \in \mathbb{Z}$ such that $x_n, x_m \in \mathcal{O}(y)$, but $x_i \notin \mathcal{O}(y)$ for all $i \in [m+1, n-1]$, we must have $n = m + k - 1$ and $x_{[m,n]} \in \mathcal{O}(z)$.

In the above definition, configurations $z = (z_1, \ldots, z_k)$ of the subshift Y^k consist of k configurations of Y stacked on top of each other. Intuitively, an SFT X having the R property means that if two rows of a configuration $x \in X$ have the same content up to a translation, and no other rows between them do, then this translation and all the rows between them are uniquely determined. In the case that X is a tiling system, it follows by induction that if a horizontal row repeats in $x \in X$ up to a translation, the rows in between are taken from a configuration of X with a non-horizontal period.

Lemma 4.48. *Let $X \subset A^{\mathbb{Z}^2}$ be a countable or downward deterministic tiling system. Then, X has the R property.*

Proof. To show this, let $y \in Y = \mathrm{Proj}(X)$ be an arbitrary row of X, and suppose that there exist $i, j, t \in \mathbb{Z}$ and $x \in X$ such that $i < j$ and $x_i = y = \sigma^{-t}(x_j)$. We assume that the difference $j - i$ is minimal with respect to y. Define also the tuple of rows $z = x_{[i,j]} \in Y^{j-i+1}$.

First, suppose that X is downward deterministic, and suppose we have a configuration $x' \in X$ and indices $m < n \in \mathbb{Z}$ such that $x'_n, x'_m \in \mathcal{O}(y)$, but $x'_i \notin \mathcal{O}(y)$ for all $i \in [m+1, n-1]$. This means that $x'_n = \sigma^r(x_i)$ for some $r \in \mathbb{Z}$, and since X is deterministic, we actually have $x'_{n-p} = \sigma^r(x_{i-p})$ for all $p \in \mathbb{N}$. In particular, this implies $n = m + i - j$ and $x'_{[m,n]} = x_{[i,j]} \in \mathcal{O}(z)$.

Now, suppose that X is countable, and let $x' \in X$ and $m < n \in \mathbb{Z}$ be as above. We assume for contradiction that either $n - m \neq j - i$ or $x'_{[m,n]} \notin \mathcal{O}(z)$ holds. In any case we have two distinct configurations $z = x_{[i,j]} \in Y^{j-i+1}$

and $z' = x'_{[m,n]} \in Y^{[n-m+1]}$, which contain no forbidden patterns of X, and whose top and bottom rows are translates of y. Now we can form arbitrarily thick horizontal stripes by stacking suitable translates of these configurations on top of each other, so that their bottom- and topmost rows overlap. Taking the limit, we obtain an uncountable number of configurations of X, which contradicts its countability. □

Lemma 4.49. *If $X \subset A^{\mathbb{Z}^2}$ is an SFT with the R property and countable projective subdynamics and $x \in X$ has a period, then the height of x is at most* 1 *in the subpattern poset* SP(X).

Proof. Suppose $x \in X$ has the period vector $\vec{n} \in \mathbb{Z}^2$. If $\vec{n}$ is horizontal, then its rows are periodic, so there are only finitely many different rows in x. Some row must then repeat infinitely many times upward and some row must repeat infinitely many times downward. Since X has the R property, x is in fact horizontally periodic and vertically eventually periodic, and the claim is proved similarly to Lemma 4.9.

Now, assume that $\vec{n}$ is not horizontal, so it has a nonzero y-coordinate $i \in \mathbb{Z}$, which we may assume to be positive. As the projective sybdynamics Proj(X) is countable,

$$Y = \{x_{[1,i]} \mid x \in X, x = \sigma^{\vec{n}}(x)\}$$

is a countable one-dimensional SFT (since we restrict to the configurations with period $\vec{n}$). If we have $x_{[1,i]} \in Y$, for a configuration $x \in X$, then x is horizontally eventually periodic, and the claim follows as above. □

Using the above lemma, we prove an upper bound for the length of downward chains occurring in an SFT with the R property, in terms of the rank of its projective subdynamics. As a corollary, we obtain the result that infinite downward chains cannot occur in countable SFTs with the bounded signal property. Recall that for a configuration $x \in A^{\mathbb{Z}^2}$, the notation Proj(x) stands for the set of rows in x.

Proposition 4.50. *Let λ be an ordinal, and let $X \subset A^{\mathbb{Z}^2}$ be a countable or downward deterministic tiling system with the horizontal bounded signal property. Then X does not contain a proper decreasing chain of length* rank(Proj$(X)) + 3$.

Proof. Observe first that X has the R property by Lemma 4.48. Denote $\lambda = \text{rank}(\text{Proj}(X)) + 1$, and assume on the contrary that $(x^\alpha)_{\alpha \leq \lambda+2}$ is a decreasing chain in X. Consider an arbitrary row $y = x^\alpha_m$, where $\alpha \leq \lambda$ and $m \in \mathbb{Z}$.

First, consider the case that y is isolated in the subshift $X_\beta \subset S^{\mathbb{Z}}$ generated by the set of rows Proj(x^β), for some $\beta < \alpha$. Then some word $w \sqsubset y$

isolates y in the set $\mathrm{Proj}(x^\beta)$, meaning that every row in x^β that contains w is a translated version of y. Because x^β is strictly above x^α in the subpattern order and the pattern w occurs in x^α, it occurs infinitely many times in x^β.

There are two possibilities: either every occurrence of w in the configuration x^β is on one of finitely many rows, in which case y is periodic, or the word occurs on infinitely many rows. In the first case we have $x^\alpha = \lim_{i\to\infty} \sigma^{(n_i,k)}(x^\beta)$ for some $k \in \mathbb{Z}$ and a sequence $(n_i)_{i\in\mathbb{N}}$ of distinct integers, since every occurrence of w in the configuration x^α is also on one of finitely many rows. Since X has the bounded signal property, every row of x^β is eventually periodic in both directions, and their eventual period is bounded, which implies that x^α has a horizontal period. By Lemma 4.49, the height of x^α is at most 1, contradicting the assumption $\alpha \leq \lambda$.

Thus the word w occurs on infinitely many distinct rows of x^β, and each of these rows is equal to y. Since X has the R property, the contents of the rows between these occurrences are determined up to horizontal translation. In fact, since X is a countable or deterministic tiling system, we can show the contents are completely determined, using the same arguments as in the proof of Lemma 4.48. This implies that x^β is eventually periodic in some direction $\vec{n} \in \mathbb{Z}^2$. Moreover, the long periodic parts of x^β are approximations to x^α, since the pattern w only occurs there, so that x^α has $\vec{n}$ as a period. As above, the height of x^α is at most 1, a contradiction.

We have shown that for every pair of ordinals $\beta < \alpha \leq \lambda$, no row of x^α is isolated in the subshift X_β generated by $\mathrm{Proj}(x^\beta)$. Denote

$$\lambda_\alpha = \min_{m\in\mathbb{Z}} \mathrm{rank}_{\mathrm{Proj}(X)}(x^\alpha_m)$$

for all $\alpha \leq \lambda$; this is the minimal rank of a row of x^α in the projective subdynamics of X. It follows from a straightforward transfinite induction agument that $\alpha \leq \lambda_\alpha \leq \lambda$ for all $\alpha \leq \lambda$. In particular, the Cantor-Bendixon rank of $\mathrm{Proj}(X)$ is at least λ, contradicting the definition of λ. $\square$

Since all (deterministic) SFTs can be recoded into (deterministic) tiling systems, we have the following corollary.

Corollary 4.51. *Let $X \subset A^{\mathbb{Z}^2}$ be a countable or deterministic SFT with the bounded signal property. Then the subpattern poset of X does not contain arbitrarily long chains.*

Corollary 4.52. *Let $X \subset A^{\mathbb{Z}^2}$ be a countable SFT with an infinite downward chain. Then the projective subdynamics of X in any direction has Cantor-Bendixson rank at least ω.*

A similar result was proved in [BDJ08]: if the Cantor-Bendixon rank of a countable SFT $X \subset A^{\mathbb{Z}^2}$ is λ, then there are downward chains of at most length λ in X.

Chapter 5

Two-Dimensional Subshifts Defined by Logical Formulas

5.1 Introduction

As has been established, multidimensional symbolic dynamics is intimately related to computability theory, and thus to mathematical logic. In fact, the initial motivation for defining Wang tiles in [Wan61] was to translate certain problems of first-order logic into the language of discrete geometry, hoping that they would be easier to solve. In this chapter, we explore the connections of multidimensional subshifts and monadic second-order (MSO) logic. More explicitly, we follow the approach of [JT09, JT13] and define two-dimensional subshifts by monadic second-order logical formulas. We show that certain hierarchies obtained by counting quantifier alternations are finite, solving an open problem posed in [JT13].

In mathematical logic and formal language theory, classes of finite structures defined by MSO formulas have been studied extensively. Examples include finite words, trees, grids and graphs; see [LN99, MS08] and references therein for a more complete list. Intuitively, the idea is to find some convenient underlying set for a combinatorial object, like the indices of a finite word or the vertices of a graph, and model the rest of the structure by functions and relations on this set. For example, one would have several unary predicates that express whether a given index of a word holds a certain letter, or in the case of graphs, a binary predicate expressing the existence of an edge between two vertices. For finite words and trees, it is known that MSO formulas define exactly the regular languages, and the quantifier alternation hierarchy collapses to the second level [Büc60]. On the other hand, the analogous hierarchy of picture languages was shown to be infinite in [MT97] and strict in [Sch98]. Although subshifts behave more like sets of words or trees than picture languages in this sense, the reasons are different: MSO-definable languages are regular because the geometry of a finite word is so simple, while the subshift hierarchy collapses since we can simulate arbitrary computation already on the third level. The concept of constructing subshifts by quantifying over infinite configurations is continued in the next chapter with a more restricted model, called *quantifier extension*.

This chapter is based on the conference article [Tör14b].

5.2 Logical Formulas and Structures

In this section, we review the basic idea of classifying finite or infinite structures using logical formulas, and then show how it can be applied to subshifts. As a motivating example, consider a finite directed graph $G = (V, E)$, where V is a finite set of vertices and $E \subset V^2$ the set of edges. We would like define a predicate logic that allows us to quantify over the vertices and edges of the graph, and pose some conditions on its local structure. For this, we define V as the *universe* of the logic, so that first-order variables refer to the vertices of G. The existence of a edge between vertices $v, w \in V$ is modeled by a binary predicate $\mathsf{e}(v, w)$, which holds if and only if $(v, w) \in E$. Using this predicate and the standard logical connectives, we can formulate various properties of G. For example, G is a complete directed graph if and only if the first-order formula $\forall \mathsf{v} \forall \mathsf{w}\, \mathsf{e}(\mathsf{v}, \mathsf{w})$ holds in it, or in other words, G is a *model* for the formula. We may also say that the formula defines the class of complete directed graphs. Similarly, the formula $\forall \mathsf{v} \exists \mathsf{w} (\mathsf{e}(\mathsf{v}, \mathsf{w}) \vee \mathsf{e}(\mathsf{w}, \mathsf{v}))$ defines the class of graphs without isolated vertices. Note that we use a distinct font for the variables of logical formulas in order to distinguish them from the actual values that they hold.

It turns out that some natural properties, such as connectivity, are not definable in this first-order logic [LN99], but can be defined if second-order quantifiers are allowed. For example, connectivity can be defined by the formula

$$\begin{aligned}
\forall \mathsf{v} \forall \mathsf{w} \exists \mathsf{F} (\forall \mathsf{u} (&\mathsf{F}(\mathsf{u}) \neq \mathsf{v}) \wedge \\
&(\mathsf{F}(\mathsf{u}) \neq \mathsf{u} \Rightarrow \\
&\ (\mathsf{e}(\mathsf{u}, \mathsf{F}(\mathsf{u})) \wedge \\
&\ \ ((\exists \mathsf{u}' \neq \mathsf{u} : \mathsf{F}(\mathsf{u}') = \mathsf{F}(\mathsf{u})) \Rightarrow \mathsf{F}(\mathsf{u}) = \mathsf{w}))))
\end{aligned}$$

where the value of the second-order variable F is a function $f : V \to V$. More generally, second-order variables can model functions or predicates on the universe of any number of variables. The above formula guarantees that the iteration $v, f(v), f^2(v), \ldots$ enumerates a path in G that eventually leads to w.

We now introduce the precise model-theoretic terminology used in this section. A *signature* $\tau = (F, R, \alpha)$ consists of a set F or *function labels*, a set R or *relation labels* with $F \cap R = \emptyset$, and an *arity function* $\alpha : F \cup R \to \mathbb{N}$. For example, the signature of graph theory would be $\tau_G = (\emptyset, \{e\}, \alpha)$, where $\alpha(e) = 2$. A *structure* is a tuple $\mathfrak{M} = (U, \tau, I)$, where U is the *universe* or *underlying set*, $\tau = (F, R, \alpha)$ is a signature, and I is an *interpretation function* that assigns to each function label $\mathsf{f} \in F$ an actual function $I(\mathsf{f}) : U^{\alpha(\mathsf{f})} \to U$, and to each relation label $\mathsf{r} \in R$ a subset $I(\mathsf{r}) \subset U^{\alpha(\mathsf{r})}$. The structures of graph theory are vertex sets, and the edge sets are defined by the possible interpretations of the edge relation e.

We continue the line of research of [JT09, JT13], and define subshifts by monadic second-order (MSO) formulas. This means that the second-order variables can only be monadic predicates. For this, we define a signature for two-dimensional configurations over a fixed alphabet A. Every configuration $x \in A^{\mathbb{Z}^2}$ defines a structure $\mathfrak{M}_x = (\mathbb{Z}^2, \tau_A, I_x)$, whose signature τ_A depends only on A and contains the following labels:

- Four unary functions, named North, South, East and West, and called *adjacency functions* in this article. They are interpreted in the structure $\mathfrak{M}_x$ as $I_x(\mathsf{North})((a, b)) = (a, b + 1)$, $I_x(\mathsf{East})((a, b)) = (a + 1, b)$ and so on for $a, b \in \mathbb{Z}$. In particular, this interpretation is independent of both the configuration x and the alphabet A.

- For each symbol $a \in A$, a unary *symbol predicate* P_a. It is interpreted as $I_x(\mathsf{P}_a)(\vec{v})$ for $\vec{v} \in \mathbb{Z}^2$ being true if and only if $x_{\vec{v}} = a$.

The MSO formulas that we use are defined with the signature τ_A as follows.

- A *term (of depth $k \in \mathbb{N}$)* is a chain of k nested applications of the adjacency functions to a first-order variable.

- An *atomic formula* is either $t = t'$ or $P(t)$, where t and t' are terms and P is either a symbol predicate or a second-order variable, which must also be a unary predicate.

- A *formula* is either an atomic formula, or an application of a logical connective $(\wedge, \vee, \neg, \ldots)$ or first- or second-order quantification to other formulas.

The *radius* of a formula is the maximal depth of a term occurring in it. First-order variables (usually denoted $\vec{\mathsf{n}}_1, \ldots, \vec{\mathsf{n}}_\ell$) are interpreted as elements of $\mathbb{Z}^2$, and the monadic second-order variables (usually denoted $\mathsf{X}_1, \ldots, \mathsf{X}_\ell$) as subsets of $\mathbb{Z}^2$. A formula without second-order variables is *first-order*, and a formula without free variables is *closed*.

Let ϕ be a closed MSO formula of radius $r \in \mathbb{N}$, let $D \subset \mathbb{Z}^2$ be an arbitrary domain, and let $P \in \mathcal{P}_2(A)$ be a pattern with $D + [-r, r]^2 \subset D(P)$. We say that P is a *D-model* for ϕ, denoted $P \models_D \phi$, if ϕ is true in the structure $\mathfrak{M}_x$ for any configuration $x \in A^{\mathbb{Z}^2}$ with $x|_{D(P)} = P$, when the quantification of the first-order variables in ϕ is restricted to D. It is clear that the definition of $P \models_D \phi$ does not depend on the choice of the configuration x. If $D = \mathbb{Z}^2$, then $P = x$, and we denote $x \models \phi$ and say that *x models ϕ*. We define a set of configurations $X_\phi = \{x \in A^{\mathbb{Z}^2} \mid x \models \phi\}$, which is always shift-invariant, but may not be a subshift. A subshift is *MSO-definable* if it equals X_ϕ for some MSO formula ϕ.

As we find it more intuitive to quantify over configurations than subsets of $\mathbb{Z}^2$, and we later wish to quantify over the configurations of specific subshifts, we introduce the following definitions.

- The notations $\forall \mathsf{x}[X]$ and $\exists \mathsf{x}[X]$ (read *for all (or exists)* x *in* X) define a new *configuration variable* x, which represents a configuration of a subshift $X \subset B^{\mathbb{Z}^2}$ over a new alphabet B. We say that X is an *auxiliary subshift* of ϕ.
- For $\mathsf{x}[X]$ quantified as above, a symbol $b \in B$ and a term t, the notation $\mathsf{x}_t = b$ defines an atomic formula that is true if and only if the configuration $x \in X$ represented by x satisfies $x_t = b$.

MSO formulas with configuration variables instead of ordinary second-order variables are called *extended MSO formulas*, and the modeling relation $\models$ and the notation X_ϕ are extended to them in the obvious way. Next, we show that if we restrict ourselves to using only MSO-definable auxiliary subshifts, extended MSO formulas have the same expressive power as ordinary ones.

Lemma 5.1. *Let ϕ be an extended MSO formula that defines a subshift $X_\phi \subset A^{\mathbb{Z}^2}$. If the auxiliary subshifts of ϕ are MSO-definable, then so is X_ϕ.*

Proof. We transform the extended MSO formula ϕ into an equivalent MSO formula $\hat{\phi}$. The transformation $\hat{\psi}$ of an extended MSO formula ψ, which need not be closed, is defined inductively as follows.

- If ψ is an atomic MSO formula, then $\hat{\psi} = \psi$.
- The operation $\hat{\cdot}$ commutes with $\neg$ and ordinary MSO quantification, and distributes over $\wedge$ and $\vee$.
- Suppose that $\psi = Q\mathsf{x}[X_\theta]\eta$ for an extended MSO formula η and an MSO formula θ, where $X_\theta \subset B^{\mathbb{Z}^2}$. Denote $B = \{b_1, \ldots, b_k\}$. In the case $Q = \forall$, we define

$$\hat{\psi} = \forall \mathsf{X}^1 \cdots \forall \mathsf{X}^k \Big(\big(\bar{\theta} \wedge \forall \vec{\mathsf{n}} \bigvee_i (\mathsf{X}^i(\vec{\mathsf{n}}) \wedge \bigwedge_{j \neq i} \neg \mathsf{X}^j(\vec{\mathsf{n}}))\big) \Rightarrow \hat{\eta} \Big),$$

and if $Q = \exists$, then

$$\hat{\psi} = \exists \mathsf{X}^1 \cdots \exists \mathsf{X}^k \big(\bar{\theta} \wedge \forall \vec{\mathsf{n}} \bigvee_{i=1}^{k} (\mathsf{X}^i(\vec{\mathsf{n}}) \wedge \bigwedge_{j \neq i} \neg \mathsf{X}^j(\vec{\mathsf{n}})) \wedge \hat{\eta}\big),$$

where $\bar{\theta}$ is obtained from θ by replacing every subformula $\mathsf{P}_{b_i}(t)$ by $\mathsf{X}^i(t)$, where t is a term. In other words, we replace the configuration variable x by the k second-order variables $\mathsf{X}^1, \ldots, \mathsf{X}^k$, each of which represents a symbol of B. In any coordinate, we allow exactly one of the X^i to be true.

- If ψ has the form $\mathsf{x}_t = b_i$, where t is a term, $b_i \in B$, and the configuration variable x is quantified as above, then $\hat{\psi}$ is $\mathsf{X}^i(t)$.

It is easy to see that a configuration $x \in A^{\mathbb{Z}^2}$ models $\hat{\phi}$ if and only if it models ϕ, and thus $X_\phi = X_{\hat{\phi}}$. $\square$

Conversely to the above lemma, we can easily convert an MSO formula to an extended MSO formula by replacing every second-order variable with a configuration variable over the binary full shift. In the rest of this chapter, unless stated otherwise, by second-order variables we mean configuration variables, and by MSO formulas we mean extended MSO formulas.

Example 5.2. The two-dimensional golden mean shift is defined by the formula

$$\forall \vec{\mathsf{n}} \big(\mathsf{P}_1(\vec{\mathsf{n}}) \implies \big(\mathsf{P}_0(\mathsf{North}(\vec{\mathsf{n}})) \wedge \mathsf{P}_0(\mathsf{East}(\vec{\mathsf{n}}))\big)\big).$$

Also, the sunny side up shift is defined by the formula

$$\forall \vec{\mathsf{m}} \forall \vec{\mathsf{n}} \big(\mathsf{P}_1(\vec{\mathsf{n}}) \implies (\mathsf{P}_0(\vec{\mathsf{m}}) \vee \vec{\mathsf{m}} = \vec{\mathsf{n}})\big).$$

Another way to define the sunny side up shift is to use a second-order quantifier:

$$\begin{aligned} &\exists \mathsf{X} \forall \vec{\mathsf{n}} \big(\mathsf{X}(\vec{\mathsf{n}}) \iff \big(\mathsf{X}(\mathsf{North}(\vec{\mathsf{n}})) \wedge \mathsf{X}(\mathsf{West}(\vec{\mathsf{n}}))\big)\big) \\ &\qquad \wedge \big(\mathsf{P}_1(\vec{\mathsf{n}}) \implies \big(\mathsf{X}(\vec{\mathsf{n}}) \wedge \neg \mathsf{X}(\mathsf{South}(\vec{\mathsf{n}})) \wedge \neg \mathsf{X}(\mathsf{East}(\vec{\mathsf{n}}))\big)\big). \end{aligned}$$

We can produce an equivalent extended MSO formula, as per the above remark:

$$\begin{aligned} &\exists \mathsf{x}[\{0,1\}^{\mathbb{Z}^2}] \forall \vec{\mathsf{n}} \big(\mathsf{x}_{\vec{\mathsf{n}}} = 1 \iff (\mathsf{x}_{\mathsf{North}(\vec{\mathsf{n}})} = 1 \wedge \mathsf{x}_{\mathsf{West}(\vec{\mathsf{n}})} = 1)\big) \\ &\qquad \wedge \big(\mathsf{P}_1(\vec{\mathsf{n}}) \implies (\mathsf{x}_{\vec{\mathsf{n}}} = 1 \wedge \mathsf{x}_{\mathsf{South}(\vec{\mathsf{n}})} = 0 \wedge \mathsf{x}_{\mathsf{East}(\vec{\mathsf{n}})} = 0)\big). \end{aligned}$$

5.3 Hierarchies of MSO-Definable Subshifts

In this section, we recall the definition of a hierarchy of subshift classes defined in [JT09, JT13], and then generalize it using extended MSO formulas. We also state some general lemmas about MSO-definable subshifts.

Definition 5.3. Let C be a class of subshifts. An MSO formula ψ is *over C with universal first-order quantifiers*, or C-u-MSO for short, if it is of the form

$$\psi = Q_1 \mathsf{x}_1[X_1] Q_2 \mathsf{x}_2[X_2] \cdots Q_n \mathsf{x}_n[X_n] \forall \vec{\mathsf{n}}_1 \cdots \forall \vec{\mathsf{n}}_\ell \phi,$$

where each Q_i is a quantifier, $X_i \in C$, and ϕ is quantifier-free. If there are k contiguous groups of similar quantifiers (or $k-1$ quantifier alternations in the case $k > 0$) and Q_1 is the existential quantifier $\exists$, then ψ is called $\bar{\Sigma}_k[C]$, and if Q_1 is the universal quantifier $\forall$, then ψ is $\bar{\Pi}_k[C]$. The set X_ψ is given the same classification. If C is the singleton class containing only the binary full shift $\{0,1\}^{\mathbb{Z}^2}$, then ψ is called u-MSO, and we denote $\bar{\Sigma}_k[C] = \bar{\Sigma}_k$ and $\bar{\Pi}_k[C] = \bar{\Pi}_k$. The classes $\bar{\Sigma}_k$ and $\bar{\Pi}_k$ for $k \in \mathbb{N}$ form the *u-MSO hierarchy.*

In [JT13], the u-MSO hierarchy was denoted by the letter $\mathcal{C}$, but we use the longer name for clarity. We note without proof that we have $\bar{\Sigma}_n = \bar{\Sigma}_n[C]$ and $\bar{\Pi}_n = \bar{\Pi}_n[C]$, if C is the class of all full shifts, not just the binary ones. In the rest of this article, C denotes an arbitrary class of subshifts, unless otherwise noted.

Remark 5.4. We use several hierarchies of subshifts obtained by counting quantifier alternations in different kinds of formulas, and the notation for them can be confusing. In general, classes defined by computability conditions (the arithmetical hierarchy) are denoted by Π and Σ, while classes defined by MSO formulas via the modeling relation are denoted by $\bar{\Pi}$ and $\bar{\Sigma}$. As an example, Π^0_1 is the class of subshifts whose language is co-recursively enumerable, while $\bar{\Pi}_1$ is the class of subshifts definable by u-MSO formulas of the form $\forall \mathsf{x}_1 \cdots \forall \mathsf{x}_k \phi$, where ϕ is first order.

We proceed with the following result, stated for u-MSO formulas in [JT13]. The proof is completely analogous, but we present it for completeness.

Theorem 5.5 (Generalization of Theorem 13 of [JT13])**.** *Let ϕ be a closed C-u-MSO formula over an alphabet A. Then for all $x \in A^{\mathbb{Z}^2}$, we have $x \models \phi$ if and only if $x \models_D \phi$ for every finite domain $D \subset \mathbb{Z}^2$.*

Proof. We prove the following stronger claim. Let ψ be the C-u-MSO formula

$$\psi = Q_1 \mathsf{x}_1[X_1] \cdots Q_n \mathsf{x}_n[X_n] \; \forall \vec{\mathsf{n}}_1 \cdots \forall \vec{\mathsf{n}}_\ell$$
$$\eta(\mathsf{y}_1, \ldots, \mathsf{y}_m, \mathsf{x}_1, \ldots, \mathsf{x}_n, \vec{\mathsf{n}}_1, \ldots, \vec{\mathsf{n}}_\ell),$$

where the Q_i are quantifiers, and fix any values $y_i \in B_i^{\mathbb{Z}^2}$ for the free second-order variables y_i. Then the formula ψ is equivalent to

$$\begin{aligned}\psi' = {} & \forall \mathsf{D} \in 2_*^{\mathbb{Z}^2}\ Q_1 \mathsf{x}_1[X_1] \cdots Q_n \mathsf{x}_n[X_n] \\ & \forall \vec{\mathsf{n}}_1 \in \mathsf{D} \cdots \forall \vec{\mathsf{n}}_\ell \in \mathsf{D} \\ & \eta(\mathsf{y}_1, \ldots, \mathsf{y}_m, \mathsf{x}_1, \ldots, \mathsf{x}_n, \vec{\mathsf{n}}_1, \ldots, \vec{\mathsf{n}}_\ell),\end{aligned}$$

where we denote by $2_*^{\mathbb{Z}^2}$ the set of all finite subsets of $\mathbb{Z}^2$. The result follows from this claim by restricting to formulas with no free variables.

First, it is clear that of ψ holds, then so does ψ'. We proceed to prove the converse by induction on n, the number of second-order quantifiers, and thus suppose that ψ' holds. If $n = 0$, the claim is clear, so we assume $n \geq 1$. We first handle the easier case $Q_1 = \forall$. For this, it suffices to note that the order of the universal quantifications $\forall \mathsf{D} \in 2_*^{\mathbb{Z}^2}$ and $\forall \mathsf{x}_1[X_1]$ in ψ' can be freely changed, after which x_1 can be handled as a free variable in a formula with $n-1$ second-order quantifiers, which is equivalent to ψ without the prefix $\forall \mathsf{x}_1[X_1]$ by the induction hypothesis. Considering every possible value for x_1, we obtain the claim.

Consider next the case $Q_1 = \exists$. For each $k \in \mathbb{N}$, if we assign the value $[-k,k]^2 \subset \mathbb{Z}^2$ to the variable D, there exists a valid choice $x_1^d \in X_1$ for the variable x_1 in ψ', and by compactness, the sequence $(x_1^k)_{k \in \mathbb{N}}$ has a limit point $x_1 \in A_1^{\mathbb{Z}^2}$. Let then $D \in 2_*^{\mathbb{Z}^2}$ be arbitrary. Choose $k_0 \in \mathbb{N}$ such that $x_1^{k_0}$ agrees with x_1 on the domain $D + [-r,r]^2$, where $r \in \mathbb{N}$ is the radius of ψ, and choose $[-k_0,k_0]^2$ and $x_1^{k_0}$ as the values of the free variables D and x_1 in the formula ψ' with the prefix $\forall \mathsf{D} \in 2_*^{\mathbb{Z}^2}\ \exists \mathsf{x}_1[X_1]$ removed. This formula is true by the choice of $k_1^{k_0}$, and thus it is also true if we assign the value D to the free variable D; but this means that $x_1^{k_0}$, and then also x_1, is a valid choice for x_1 in the formula ψ' for the value D of the variable D. Thus if we choose x_1 as the value of x_1, the formula ψ' with the order of the quantifications $\forall \mathsf{D} \in 2_*^{\mathbb{Z}^2}$ and $\exists \mathsf{x}_1[X_1]$ swapped is true. By the induction hypothesis, when we consider x_1 as a free variable, the formula ψ is also true. This finishes the proof of the claim, and thus of the theorem. □

We remark here that Theorem 5.5 is the main reason for us to restrict to the class of C-u-MSO formulas, since it no longer holds if we allow arbitrary existential quantification of first-order variables. The following corollaries show the power of this result.

Corollary 5.6. *Every C-u-MSO formula ϕ defines a subshift.*

Proof. Let $r \in \mathbb{N}$ be the radius of ϕ, and A its alphabet. By Theorem 5.5, the set X_ϕ is defined by the set of forbidden patterns

$$\{x|_{D+[-r,r]^2} \mid D \subset \mathbb{Z}^2 \text{ finite}, x \in A^{\mathbb{Z}^2}, x \not\models_D \phi\},$$

and thus is a subshift by definition. □

Corollary 5.7. *For all $k, n \in \mathbb{N}$, we have $\bar{\Pi}_n[\Pi^0_k] \subset \Pi^0_{k+1}$. In particular, the u-MSO hierarchy only contains Π^0_1 subshifts.*

Proof. Let $\phi = \forall \mathsf{x}_1[X_1]\exists \mathsf{x}_2[X_2] \dots Q_n \mathsf{x}_n[X_n]\psi$ be a $\bar{\Pi}_n[\Pi^0_k]$ formula, where each $X_i \subset A_i^{\mathbb{Z}^2}$ is a Π^0_k subshift and ψ is first-order. Then the product subshift $\prod_{i=1}^n X_i$ is also Π^0_k. Let $P \in \mathcal{P}_2(A)$ be a finite pattern, and let $r \in \mathbb{N}$ be the radius of ϕ. Theorem 5.5, together with a basic compactness argument, implies that $P \in \mathcal{B}(X_\phi)$ holds if and only if for all finite domains $D(P) \subset D \subset \mathbb{Z}^2$, there exists a pattern $Q \in A^{D+[-r,r]^2}$ such that $Q|_{D(P)} = P$ and $Q \models_D \phi$. For a fixed D, denote this condition by $C_P(D)$.

We show that deciding $C_P(D)$ for given pattern P and domain D is Δ^0_{k+1}. Denote $E = D + [-r, r]^2$ and $L = \mathcal{B}_E(\prod_{i=1}^n X_i)$. Note that $L \subset A^E$ is a finite set, and computing it from the domain D is Π^0_k. Moreover, given a pattern $Q \in A^E$ and the set L, the condition $Q \models_D \phi$ can be easily checked by a Turing machine. Thus the condition $C_P(D)$ can be decided by an oracle Turing machine that computes the set L from E using a Π^0_k oracle, and then goes through the finite set A^E, searching for such a pattern Q. Thus the condition $C_P(D)$ is Δ^0_{k+1}, which implies that deciding $P \in \mathcal{B}(X_\phi)$ is Π^0_{k+1}. □

Finally, we show that if the final second-order quantifier of a u-MSO formula is universal, it can be dropped. This does not hold for C-u-MSO formulas in general. The proof is exactly that of [JT13, Lemma 7], so we omit it.

Lemma 5.8. *If $k \geq 1$ is odd, then $\bar{\Pi}_k = \bar{\Pi}_{k-1}$, and if it is even, then $\bar{\Sigma}_k = \bar{\Sigma}_{k-1}$.*

Example 5.9. Define the *two-dimensional mirror shift* $X_{\mathrm{mirror}} \subset \{0, 1, \#\}^{\mathbb{Z}^2}$ by the forbidden patterns $\begin{smallmatrix} a \\ \# \end{smallmatrix}$ and $\begin{smallmatrix} \# \\ a \end{smallmatrix}$ for $a \neq \#$, every two-element pattern $\{\vec{0} \mapsto \#, (n, 0) \mapsto \#\}$ for $n > 0$, and every three-element pattern $\{(-n, 0) \mapsto a, \vec{0} \mapsto \#, (n, 0) \mapsto b\}$ for $n > 0$ and $a \neq b$. A 'typical' configuration of X_{mirror} contains one infinite column of $\#$-symbols, whose left and right sides are mirror images of each other. It is well-known that X_{mirror} is not a sofic shift, but we do not have a direct reference for the fact. It can be proved using Lemma 3.15, and the same argument is applied in [KM13, Example 2.4] to a slightly different subshift.

We show that the mirror subshift can be implemented by an SFT-u-MSO formula $\psi = \forall \mathsf{x}[X]\forall \vec{\mathsf{n}}_1 \forall \vec{\mathsf{n}}_2 \forall \vec{\mathsf{n}}_3 \phi$ in the class $\bar{\Pi}_1[\mathrm{SFT}]$. This also shows that Lemma 5.8 fails outside the u-MSO hierarchy.

Let X be the SFT whose alphabet is seen in Figure 5.1, defined by the obvious 2×2 forbidden patterns. Define the quantifier-free part of the

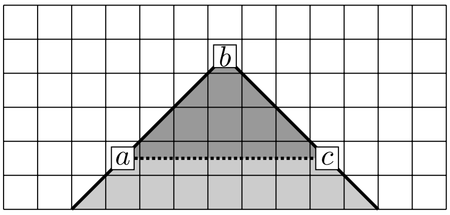

Figure 5.1: A pattern of X in Example 5.9, containing its entire alphabet.

SFT-u-MSO formula ϕ as $\phi = \phi_1 \wedge (\phi_2 \implies \phi_3)$, where

$$\begin{aligned}
\phi_1 &= \mathsf{P}_\#(\vec{\mathsf{n}}_2) \iff \mathsf{P}_\#(\mathsf{North}(\vec{\mathsf{n}}_2)) \\
\phi_2 &= \mathsf{x}_{\vec{\mathsf{n}}_1} = a \wedge \mathsf{x}_{\vec{\mathsf{n}}_2} = b \wedge \mathsf{x}_{\vec{\mathsf{n}}_3} = c \wedge \mathsf{P}_\#(\vec{\mathsf{n}}_2) \\
\phi_3 &= \neg \mathsf{P}_\#(\vec{\mathsf{n}}_1) \wedge \neg \mathsf{P}_\#(\vec{\mathsf{n}}_3) \wedge (\mathsf{P}_0(\vec{\mathsf{n}}_1) \iff \mathsf{P}_0(\vec{\mathsf{n}}_3))
\end{aligned}$$

This formula simply states that the #-symbol only occurs in vertical columns, and if the symbol b of X is on this column, then the symbols a and c are on top of distinct symbols which are not #. It is easy to see that the subshift X_ψ is exactly X_{mirror}, with ψ defined as above.

5.4 The u-MSO Hierarchy

We argue that the u-MSO hierarchy is a quite natural hierarchy of MSO-definable subshifts. Namely, the lack of existential first-order quantification makes it easy to prove that every u-MSO formula actually defines a subshift (Corollary 5.6), and quantifier alternations give rise to interesting hierarchies in many contexts. The following is already known about the hierarchy.

Theorem 5.10 ([JT13]). *The class of subshifts defined by formulas of the form $\forall \vec{n} \phi$, where ϕ is first-order, is exactly the class of SFTs. The class $\bar{\Pi}_0 = \bar{\Sigma}_0$ consists of the* threshold counting shifts*, which are obtained from subshifts of the form $\{x \in A^{\mathbb{Z}^2} \mid P$ occurs in x at most n times$\}$ for patterns $P \in \mathcal{P}_2(A)$ and $n \in \mathbb{N}$ using finite unions and intersections. Finally, the class $\bar{\Sigma}_1$ consists of exactly the sofic shifts.*

The subshifts in the class $\bar{\Pi}_0$ are called threshold counting shifts, since they can 'count' the number of occurrences of finitely many patterns up to a finite threshold. In this section, we show that the u-MSO hierarchy collapses to the third level, which consists of exactly the Π^0_1 subshifts. This gives negative answers to the questions posed in [JT13] of whether the hierarchy is infinite, and whether it only contains sofic shifts.

Theorem 5.11. *For all $n \geq 2$ we have $\Pi^0_1 = \bar{\Pi}_n$.*

Proof. As we have $\bar{\Pi}_n \subset \Pi_1^0$ by Corollary 5.7, and clearly $\bar{\Pi}_n \subset \bar{\Pi}_{n+1}$ also holds for all $n \in \mathbb{N}$, it suffices to prove that $\Pi_1^0 \subset \bar{\Pi}_2$. Let thus $X \subset A^{\mathbb{Z}^2}$ be an arbitrary Π_1^0 subshift. We construct an MSO formula of the form $\phi = \forall \mathsf{y}[B^{\mathbb{Z}^2}]\exists \mathsf{z}[C^{\mathbb{Z}^2}]\forall \vec{\mathsf{n}}\ \psi(\vec{\mathsf{n}}, \mathsf{y}, \mathsf{z})$ such that $X_\phi = X$.

The main idea is the following. We use the universally quantified configuration variable y to specify a finite square $R \subset \mathbb{Z}^2$ and a word $w \in A^*$, which may or may not encode the pattern $x|_R$ of a configuration $x \in A^{\mathbb{Z}^2}$. The existentially quantified variable z enforces that either w does not correctly encode $x|_R$, or that it encodes *some* pattern of $\mathcal{B}(X)$. As R and w are arbitrary and universally quantified, this guarantees $x \in X$. The main difficulty is that the value of the variable y comes from a full shift, so we have no control over it; the configuration may contain infinitely many squares, or none at all.

To overcome this problem, we first define an auxiliary SFT $Y \subset B^{\mathbb{Z}^2}$, whose configurations contain the aforementioned squares. The alphabet B consists of the tiles seen in Figure 5.2, where every label w_i ranges over A, and it is defined by the set F_Y of 2×2 forbidden patterns where some colors or lines of neighboring tiles do not match. A configuration of Y contains at most one maximal pattern colored with the lightest gray in Figure 5.2, and if said pattern is finite, its domain is a square. We call this domain the *input square*, and the word $w \in A^*$ that lies above it is called the *input word*.

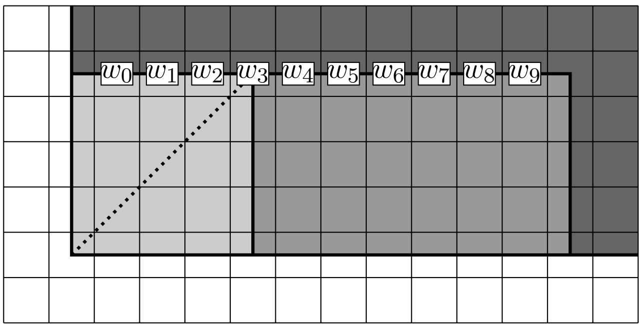

Figure 5.2: A pattern of Y. In this example, the input word $w \in A^*$ is of length 10.

We now define another SFT S, this time on the alphabet $A \times B \times C$. The alphabet C is more complex than B, and we specify it in the course of the construction. The idea is to simulate a computation in the third layer to ensure that if the second layer contains a valid configuration of Y, and the input word encodes the contents of the input square in the first layer, then that square pattern is in $\mathcal{B}(X)$. We also need to ensure that a valid configuration exists even if the encoding is incorrect, or if second layer is not in Y. For this, every locally valid square pattern of Y containing an input square will be covered by another square pattern in the third layer, inside which we perform the computations. We will force this pattern to be infinite if the second layer is a configuration of Y.

Now, we describe the local rules of the SFT S using an example configuration $(x, y, z) \in S$, where y and z are values of the respective variables y and z. The coordinates of every 2×2 rectangle $R \subset \mathbb{Z}^2$ such that $y|_R \in F_Y$ are called *defects*. Also, a non-defect coordinate $\vec{v} \in \mathbb{Z}^2$ such that $y_{\vec{v}} =$ is called a *seed*. Denote $C = C_1 \cup C_2$, where C_1 is the set of tiles depicted in Figure 5.3 (a). Their adjacency rules in S are analogous to those of Y. The rules of S also force the set of seeds to coincide with the set of coordinates $\vec{v} \in \mathbb{Z}^2$ such that $z_{\vec{v}} =$. These coordinates are the southwest corners of *computation squares* in z, whose square shape is again enforced by a diagonal signal. The southwest half of a computation square is colored with letters of C_2. See Figure 5.3 (b) for an example of a computation square.

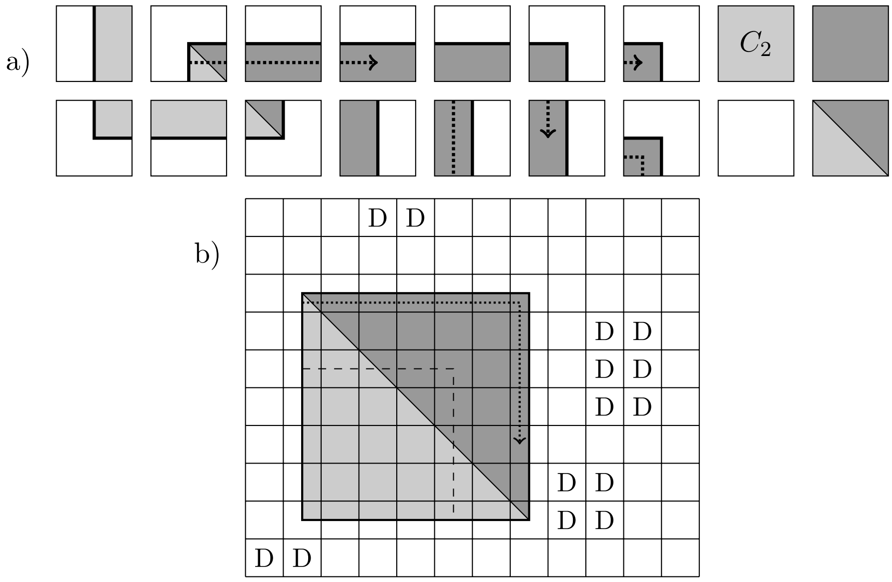

Figure 5.3: The alphabet C (a) and a pattern of the third layer of S (b), with the elements of C_2 represented by the featureless light gray tiles. The dashed line represents the border of an input square on the second layer. Defects are marked with the letter D.

A computation square may not contain defects or coordinates $\vec{v} \in \mathbb{Z}^2$ such that $y_{\vec{v}} =$, and conversely, the north or east border of a computation square contains a tile that is directly or diagonally adjacent to a defect. This constraint is enforced by a signal emitted from the northwest corner of the square (the dotted line in Figure 5.3 (b)), which travels along the north and east borders, and disappears when it first encounters a defect. If the signal reaches the southeast corner of the computation square, a tiling error is produced.

We now describe the set C_2, and for that, let M be a CMS with input alphabet $\Sigma = A \times (A \cup \{0, 1, \#\})$ and two initial states q_1 and q_2. This machine is simulated on the southwest halves of the computation squares using Construction 3.12, and we will fix its functionality later. The alphabet C_2 is the product of S_M, the alphabet of the countable SFT X_M that simulates M, and the alphabet C_3 shown in Figure 5.4, which is used to feed the machine M its input. Note that the colors and lines in C_3 are disjoint from those in C_1, even though the figures suggest otherwise. The idea is to initialize the counter machine M with either the input word (if it correctly encodes the input square), or a proof that the encoding is incorrect, in the form of one incorrectly encoded symbol.

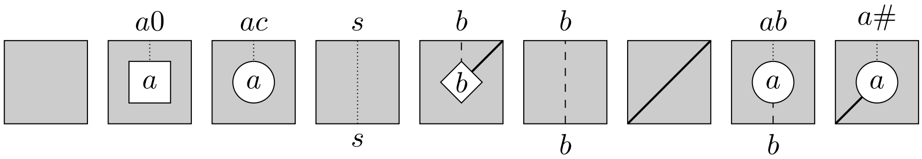

Figure 5.4: The sub-alphabet C_3. The letters a and b range over A, the letter c over $\{0, 1\}$, and the letter s over the input alphabet Σ of M.

The white squares and circles of C_3 must be placed on the letters of the input word $w \in A^*$ of the computation square, the square on the leftmost letter and circles on the rest, and the computation cone of M must be placed on the square. The A-letters of these tiles must match the letters of w, and the second component is 1 if the tile lies on the corner of the input square, 0 if not, $b \in A$ in the presence of a vertical signal, and # in the presence of a diagonal signal. Such signals are sent by a white diamond tile, called a *candidate error*, which can only be placed on an interior tile of the input square, and whose letter must match the letter on the first layer x. The other component of C_2 simulates the machine M, which uses the tiles with the dotted vertical line and label $s \in \Sigma$ (the fourth tile form the left in Figure 5.4) as its input, and can never halt in a valid configuration. See Figure 5.5 for a visualization. We also require that for a pattern $\substack{c_2 \\ c_1}$ to be valid, where $c_i \in C_i$ for $i \in \{1, 2\}$, the tile c_2 should have a gray south border with no lines. Other adjacency rules between tiles of C_1 and C_2 are explained by Figure 5.3 (a).

We now describe the counter machine M. Note first that from an input $u \in \Sigma^*$ of the simulated machine M one can deduce the input word $w \in A^*$, the height $h \in \mathbb{N}$ of the input square, and the positions and contents of all candidate errors. Now, when started in the state q_1, the machine checks that there are no candidate errors at all, that $|w| = h^2$, and that the square pattern $P \in A^{h \times h}$, defined by $P_{(i,j)} = w_{ih+j}$ for all $i, j \in [0, h-1]$, is in $\mathcal{B}(X)$. If all this holds, M runs forever (the check for $P \in \mathcal{B}(X)$ can indeed take infinitely many steps). When started in q_2, the machine checks that there is exactly one candidate error at some position $(i, j) \in [0, h-1]^2$ of the input square containing some letter $b \in A$, and that one of $|w| \neq h^2$ or $w_{ih+j} \neq b$ holds. If this is the case, M enters an infinite loop, and halts otherwise.

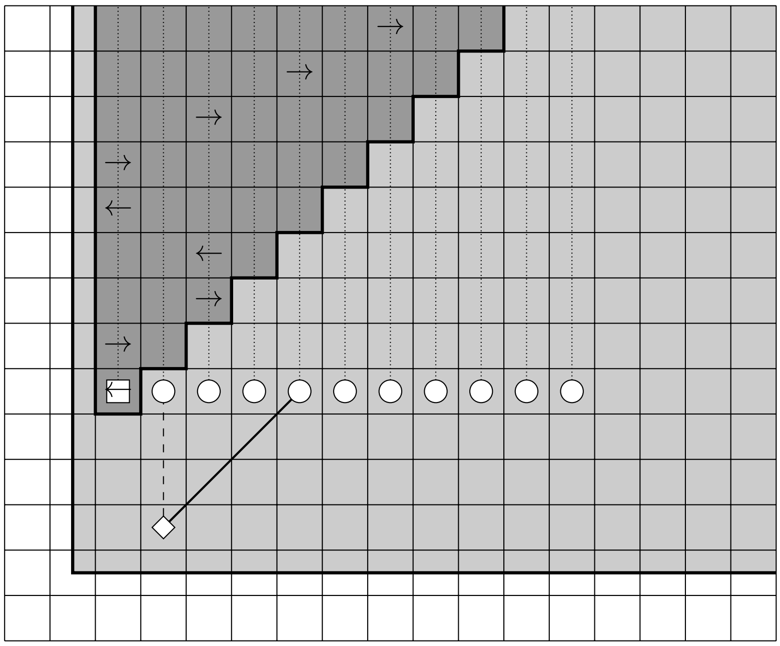

Figure 5.5: An infinite computation square with an input word of length 11 and a single candidate error. The computation cone is shown in dark gray.

The definition of the SFT S is now complete, and it can be realized using a set F of forbidden patterns of size 3×3 over the alphabet $A \times B \times C$. We define the quantifier-free formula $\psi(\vec{\mathsf{n}}, \mathsf{y}, \mathsf{z})$ as $\neg \bigvee_{P \in F} \psi_P$, where ψ_P states that the pattern P occurs at the coordinate $\vec{\mathsf{n}}$. This is easily doable using the adjacency functions, color predicates and the variables y and z. If we fix some values $y \in B^{\mathbb{Z}^2}$ and $z \in C^{\mathbb{Z}^2}$ for the variables y and z, then $x \models \forall \vec{\mathsf{n}} \psi(\vec{\mathsf{n}}, \mathsf{y}, \mathsf{z})$ holds for a given $x \in A^{\mathbb{Z}^2}$ if and only if $(x, y, z) \in S$.

Let $x \in A^{\mathbb{Z}^2}$ be arbitrary. We proceed to show that x is a model for ϕ if and only if $x \in X$. Suppose first that x models ϕ, and let $\vec{v} \in \mathbb{Z}^2$ and $h \geq 1$ be arbitrary. Let $y \in Y$ be a configuration whose input square has interior $D = \vec{v} + [0, h-1]^2$, and whose input word correctly encodes the pattern $x|_D$. By the assumption that $x \models \phi$, there exists a configuration

$z \in C^{\mathbb{Z}^2}$ such that $(x, y, z) \in S$, so that the southwest neighbor of $\vec{v}$ is the southwest corner of a computation square in z, which is necessarily infinite, since no defects occur in y. In this square, M runs forever, and it cannot be initialized in the state q_2 as the encoding of the input square is correct. Thus its computation proves that $x|_D \in \mathcal{B}(X)$. Since $D \subset \mathbb{Z}^2$ was an arbitrary square domain, we have $x \in X$.

Suppose then $x \in X$, and let $y \in B^{\mathbb{Z}^2}$ be arbitrary. We construct a configuration $z \in C^{\mathbb{Z}^2}$ such that $(x, y, z) \in S$, which proves $x \models \phi$. First, let $T \subset \mathbb{Z}^2$ be the set of seeds in y, and for each $\vec{s} \in T$, let $\ell(\vec{s}) \in \mathbb{N} \cup \{\infty\}$ be the height of the maximal square $D(\vec{s}) = \vec{s} + [0, \ell(\vec{s}) - 1]^2$ that contains no defects. We claim that $D(\vec{s}) \cap D(\vec{r}) = \emptyset$ holds for all $\vec{s} \neq \vec{r} \in T$. Suppose the contrary, and let $\vec{v} \in D(\vec{s}) \cap D(\vec{r})$ be lexicographically minimal. Then the coordinate $\vec{v}$ is on the south border of $D(\vec{s})$ and the west border of $D(\vec{r})$ (or vice versa). Since these borders contain no defects, the tile $y_{\vec{v}}$ is a south border tile and a west border tile, a contradiction.

Now, we can define the domain $D(\vec{s})$ to be a computation square in z for every seed $\vec{s} \in T$. If the computation square contains an input square and an associated input word which correctly encodes its contents, we initialize the simulated machine M in the state q_1. Then the computation does not halt, since the input square contains a pattern of $\mathcal{B}(X)$. Otherwise, we initialize M in the state q_2, and choose a single candidate error from the input square such that M does not halt, and thus produces no forbidden patterns. With these choices we have $(x, y, z) \in S$, completing the proof. $\square$

We have now characterized every level of the u-MSO hierarchy. The first level $\bar{\Pi}_0 = \bar{\Sigma}_0$ contains the threshold counting shifts and equals $\bar{\Pi}_1$ by Lemma 5.8, the class $\bar{\Sigma}_1 = \bar{\Sigma}_2$ contains the sofic shifts, and the other levels coincide with the class Π^0_1.

The quantifier alternation hierarchy of MSO-definable picture languages was shown to be strict in [Sch98]. It is slightly different from the u-MSO hierarchy, as existential first-order quantification is allowed. However, in the case of pictures we know the following. Any MSO formula $\mathcal{Q}_L \exists \vec{\mathsf{n}}\, \mathcal{Q}_R\, \phi$, where $\mathcal{Q}_L$ and $\mathcal{Q}_R$ are strings of quantifications and ϕ is quantifier-free, is equivalent to a formula of the form $\mathcal{Q}_L \exists \mathsf{X} \mathcal{Q}_R \forall \vec{\mathsf{n}}\, \psi$, where ψ is also quantifier-free. See [MS08, Section 4.3] for more details. This implies that the analogue of the u-MSO hierarchy for picture languages is infinite. The proof of the result of [Sch98] relies on the fact that one can simulate computations within the pictures, and the maximal time complexity of these computations depends on the number of quantifier alternations. In the case of infinite configurations, this argument naturally falls apart.

Finally, Theorem 5.11 and Lemma 5.1 have the following corollary (which was also proved in [JT13]).

Corollary 5.12. *Every Π^0_1 subshift is MSO-definable.*

5.5 Other C-u-MSO Hierarchies

Next, we generalize Theorem 5.11 to hierarchies of Π^0_k-u-MSO formulas. The construction is similar to the above but easier, since we can restrict the values of the variable y to lie in a geometrically well-behaved subshift.

Theorem 5.13. *For all $k \geq 1$ and $n \geq 2$ we have $\Pi^0_{k+1} = \bar{\Pi}_n[\Pi^0_k]$. Furthermore, $\Pi^0_2 = \bar{\Pi}_n[\mathrm{SFT}]$ for all $n \geq 2$.*

Proof. As in Theorem 5.11, it suffices to show that for a given Π^0_{k+1} subshift $X \subset A^{\mathbb{Z}^2}$, there exists a $\bar{\Pi}_2[\Pi^0_k]$ formula $\phi = \forall \mathsf{y}[Y]\exists \mathsf{z}[Z]\forall \vec{\mathsf{n}}\ \psi$ such that $X_\phi = X$. In our construction, $Y \subset B^{\mathbb{Z}^2}$ is a Π^0_k subshift and $Z = C^{\mathbb{Z}^2}$ is a full shift.

For a square pattern $P \in A^{h \times h}$, define the word $w(P) \in A^{h^2}$ by $w_{ih+j} = P_{(i,j)}$ for all $i, j \in [0, h-1]$. By Lemma 3.16, there is a Π^0_k predicate $R \subset A^* \times \mathbb{N}$ such that the set

$$F = \{P \in A^{h \times h} \mid h \in \mathbb{N}, \exists n \in \mathbb{N} : R(w(P), n)\}$$

of forbidden patterns defines the subshift X. As in Theorem 5.11, configurations of Y may contain one input square with an associated input word. This time, the input word is of the form $w\#^n$ for some $w \in A^*$, $n \in \mathbb{N}$ and a new symbol $\#$. As Y is a Π^0_k subshift, we can enforce that $R(w, n)$ holds, so that w does *not* encode any square pattern of X. This can be enforced by SFT rules if $k = 1$: a simulated counter machine checks $R(w, n)$ by running forever if it holds. As before, the existentially quantified variable z enforces that w does *not* correctly encode the contents of the input square in the first layer.

We now prove that $X = X_\phi$, and for that, let $x \in X$ and $y \in Y$ be arbitrary. If y has a finite input square $D \in \mathbb{Z}^2$ and input word $w\#^n$, then $w \in A^*$ cannot correctly encode the pattern $x|_D \in \mathcal{B}(X)$, and thus a valid choice for the variable z exists. Degenerate cases of y (with, say, an infinite input square) are handled as in Theorem 5.11. Thus we have $x \models \phi$. Next, suppose that $x \notin X$, so there is a square domain $D \subset \mathbb{Z}^2$ with $x|_D \notin \mathcal{B}(X)$. Construct the configuration $y \in Y$ such that the input square has domain D, the word $w \in A^*$ correctly encodes $x|_D$, and the number $n \in \mathbb{N}$ of $\#$-symbols is such that $R(w, n)$ holds. For this value of the variable y, no valid choice for z exists, and thus $x \not\models \phi$. □

Corollary 5.12, Theorem 5.13 and a simple induction argument give the following corollary.

Corollary 5.14. *For every $k \in \mathbb{N}$, every Π^0_k subshift is MSO-definable.*

However, the converse does not hold, since one can construct an MSO formula defining a subshift whose language is not Π^0_k for any $k \in \mathbb{N}$.

5.6 Lower Levels of C-u-MSO Hierarchies

The results of [JT13] and Theorem 5.11 give a complete classification of the different levels of the u-MSO hierarchy. In this section, we study the possible analogues of these results in some more general C-u-MSO hierarchies. Example 5.9 already showed that the class $\bar{\Pi}_1[\mathrm{SFT}]$ is not contained in the class of sofic shifts, so we proceed with a characterization of this class. Before the theorem, we give the following definition.

Definition 5.15. Let $k \in \mathbb{N}$. A subshift $X \subset A^{\mathbb{Z}^2}$ is Σ^0_k-*bounded*, if there exists $K \in \mathbb{N}$ and a Σ^0_k set $F \subset \mathcal{P}_2(A)$ of forbidden patterns for X such that $|D(P)| \leq K$ for all $P \in F$. We denote by $\mathrm{B}\Sigma^0_k$ the class of Σ^0_k-bounded subshifts.

In other words, a subshift is Σ^0_k-bounded, if it has a Σ^0_k set of forbidden patterns whose domains have bounded size, although their diameter may be unbounded. Note that $\mathrm{B}\Sigma^0_k$ is not a subset of Σ^0_k nor Π^0_k, but we do have $\mathrm{B}\Sigma^0_k \subset \Pi^0_{k+1}$.

Example 5.16. The mirror shift $X_{\mathrm{mirror}} \subset \{0, 1, \#\}^{\mathbb{Z}^2}$ of Example 5.9 is Σ^0_1-bounded. In the example, an infinite set of forbidden patterns is given for it, and all of them have domains of at most 3 coordinates. It is also easy to see that the set is recursively enumerable, or Σ^0_1.

Theorem 5.17. *For all $k \geq 1$, we have $\bar{\Pi}_1[\Pi^0_k] = \mathrm{B}\Sigma^0_{k+1}$. We also have $\bar{\Pi}_1[\mathrm{SFT}] = \mathrm{B}\Sigma^0_2$.*

Proof. First, suppose that a subshift $X \subset A^{\mathbb{Z}^2}$ is in the class $\bar{\Pi}_1[\Pi^0_k]$. Since Π^0_k subshifts are closed under direct product, we may assume that X is defined by a formula $\psi = \forall \mathsf{y}[Y] \forall \vec{\mathsf{n}}_1 \cdots \forall \vec{\mathsf{n}}_\ell \phi(\mathsf{y}, \vec{\mathsf{n}}_1, \ldots, \vec{\mathsf{n}}_\ell)$, where $Y \subset B^{\mathbb{Z}^2}$ is a Π^0_k subshift. Let $r \in \mathbb{N}$ be the radius of ψ. For an ℓ-tuple of vectors $\vec{n} \in (\mathbb{Z}^2)^\ell$, denote $S(\vec{n}) = \bigcup_{i=1}^{\ell} \vec{n}_i + [-r, r]^2$. Then X is defined by the set of forbidden patterns

$$F = \{P \in \mathcal{P}_2(A) \mid \vec{n} \in (\mathbb{Z}^2)^\ell, P \in A^{S(\vec{n})}, Q \in \mathcal{B}_{S(\vec{n})}(Y), P \not\models_{S(\vec{n})} \phi(Q, \vec{n})\}.$$

The set F is Σ^0_{k+1}, since $\mathcal{B}(Y)$ is Π^0_k, and we have $|D(P)| \leq \ell(2r+1)^2$ for all $P \in F$. This shows that $\bar{\Pi}_1[\Pi^0_k] \subset \mathrm{B}\Sigma^0_{k+1}$.

We prove the converse direction for $\bar{\Pi}_1[\mathrm{SFT}]$, since the general case is essentially similar. Let $X \subset A^{\mathbb{Z}^2}$ be defined by a Σ^0_2 set of forbidden patterns $F \subset \mathcal{P}_2(A)$, whose domain size is bounded by $K \in \mathbb{N}$. Then there exists a Π^0_1 predicate $R \subset \mathcal{P}_2(A) \times \mathbb{N}$ such that $F = \{P \in \mathcal{P}_2(A) \mid \exists n \in \mathbb{N} : R(P, n)\}$. Define an SFT $Y \subset B^{\mathbb{Z}^2}$ as follows. It has a total of $K + 1$ layers, labeled Y_i for $i \in \{0, \ldots, K\}$, so that $Y \subset \prod_{i=0}^{K} Y_i$. For each $i \in \{0, \ldots, K\}$, let $\pi_i : Y \to Y_i$ be the natural projection map. The last K layers are identical

SFTs defined by the allowed 2×2 patterns of Figure 5.6 for all $a \in A$. The central tile in the figure is called the *point* of the layer, and we denote the tile by $\bar{a}$. Note that in a configuration $y \in Y$ containing all K points at some coordinates $(i_1, j_1), \dots, (i_K, j_K) \in \mathbb{Z}^2$, the coordinate

$$p(y) = \big(\min_{1 \leq \ell \leq K} i_\ell, \min_{1 \leq \ell \leq K} j_\ell\big) \in \mathbb{Z}^2$$

can be recognized by a local condition.

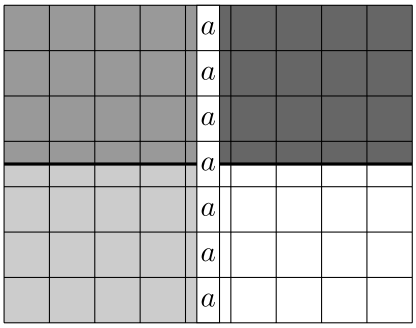

Figure 5.6: A configuration of the layer Y_i of Y in Theorem 5.17, where $i \in \{1, \dots, K\}$ and a ranges over A.

The first layer Y_0 is given by Construction 3.12, and its configurations simulate the computations of a CMS M. The computation cone of the simulated machine is based on the coordinate $p(y)$, so that whenever a configuration of Y contains all K points, it also contains the simulation of a computation. Its input consists of the intersection of the horizontal and vertical signals of the last K layers with the diagonal border of the computation cone, as shown in Figure 5.7, so that M effectively receives a K-tuple $\vec{n} \in (Z^2)^K$ of relative positions in $\mathbb{Z}^2$, together with a K-tuple $\vec{a} \in A^K$ of symbols from A. Let $P = \{\vec{n}_i \mapsto \vec{a}_i \mid i \in \{1, \dots, K\}\} \in \mathcal{P}_2(A)$ be the pattern formed by the coordinates and their respective symbols. Now, the machine M first nondeterministically guesses a number $n \in \mathbb{N}$ by incrementing a counter and stopping at some point, visits a special state q_{guess}, and then checks (with an infinite computation) that $R(P, n)$ holds. If the condition fails, then M halts, producing a tiling error. This means that the configurations of Y where all K points occur and the simulated machine eventually makes a guess enumerate exactly the forbidden patterns $P \in F$.

Of course, it is possible that the simulated machine never makes the guess for n in a configuration of Y, but we can ignore this case by enforcing the guess with a first-order variable. Define the extended MSO formula

$$\phi = \forall \mathsf{y}[Y] \forall \vec{\mathsf{n}}_0 \cdots \forall \vec{\mathsf{n}}_K (\pi_0(\mathsf{y})_{\vec{\mathsf{n}}_0} \neq q_{\text{guess}}) \vee \bigvee_{i=1}^{K} \bigvee_{a \in A} (\mathsf{P}_a(\vec{\mathsf{n}}_i) \wedge \pi_i(\mathsf{y})_{\vec{\mathsf{n}}_i} \neq \bar{a}),$$

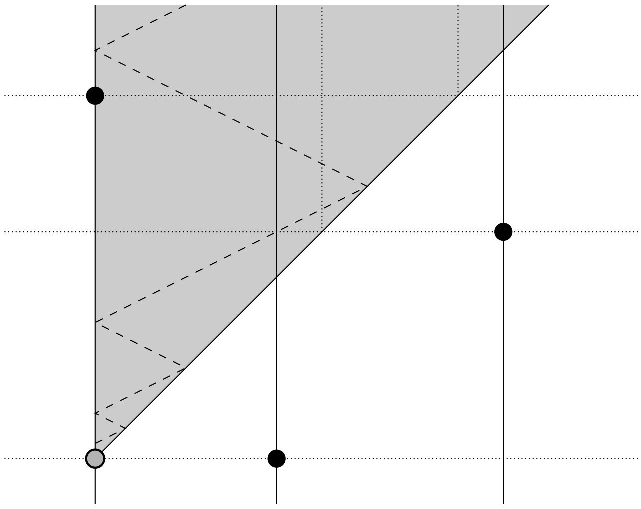

Figure 5.7: A schematic diagram of a simulation of the CMS in a configuration y of the SFT Y. Note how the information about the points (the black dots) is passed to the simulated machine. The computation cone has its base at $p(y)$ (the gray circle).

where we have denoted by a the elements of A, and by $\bar{a}$ the corresponding points of the Y_i. The formula simply states that none of the patterns enumerated by Y occur in the configuration simultaneously with the special state of M. It is clear that a configuration $x \in A^{\mathbb{Z}^2}$ models the formula ϕ if and only if $x \in X$. The general claim follows similarly, except that the extra tiles of the last K layers and the CMS of the first layer can simply be replaced by a Π^0_k rule. □

We now consider the larger classes $\bar{\Sigma}_2[\Pi^0_k]$, and show that their relation to the Σ^0_k-bounded subshifts is the same as that of sofic shifts to SFTs.

Proposition 5.18. *Let $k \in \mathbb{N}$. A subshift $X \subset A^{\mathbb{Z}^2}$ is in the class $\bar{\Sigma}_2[\Pi^0_k]$ if and only if it is the image of a Σ^0_{k+1}-bounded subshift under a block map.*

Proof. First, let $X = f(X_\phi)$, where $\phi = \forall \mathsf{y}[Y]\forall\vec{\mathsf{n}}_1 \cdots \forall\vec{\mathsf{n}}_\ell \psi$ is a $\bar{\Pi}_1[\Pi^0_k]$ formula over an alphabet B, and $f : X_\phi \to X$ is a block map. For each $a \in A$, let $\theta_a(\mathsf{x}, \vec{\mathsf{n}})$ be the quantifier-free formula with free variables x (a configuration variable over B) and $\vec{\mathsf{n}}$ stating that $f(\mathsf{x})_{\vec{\mathsf{n}}} = a$. Such a formula exists, since the block map f has a finite neighborhood. Then we have $X = X_\eta$, where η is the $\bar{\Sigma}_2[\Pi^0_k]$ formula

$$\exists \mathsf{x}[B^{\mathbb{Z}^2}]\forall \mathsf{y}[Y]\forall\vec{\mathsf{n}}_1 \cdots \forall\vec{\mathsf{n}}_\ell \forall n\big(\bar{\psi} \wedge \bigwedge_{a\in A} (\mathsf{P}_a(\vec{\mathsf{n}}) \Rightarrow \theta_a(\mathsf{x}, \vec{\mathsf{n}}))\big),$$

and $\bar{\psi}$ is obtained from the quantifier-free part ψ by replacing each symbol predicate $\mathsf{P}_b(t)$ by $\mathsf{x}_t = b$.

Second, suppose that X is defined by the $\bar{\Sigma}_2[\Pi^0_k]$ formula $\exists \mathsf{y}[Y]\forall \mathsf{z}[Z]\phi$, where $Y \subset B^{\mathbb{Z}^2}$ and Z are Π^0_k subshifts and ϕ is first-order. Then

$$\{(x, y) \mid x \in A^{\mathbb{Z}^2}, y \in Y, \forall z \in Z : x \models \phi(y, z)\} \subset (A \times B)^{\mathbb{Z}^2}$$

is clearly a $\bar{\Pi}_1[\Pi^0_k]$ subshift and thus Σ^0_{k+1}-bounded, and its image in the projection map $(x, y) \mapsto x$ is exactly X. □

Finally, we give some separation results for the classes $\bar{\Pi}_1[\Pi^0_k]$ and $\bar{\Sigma}_2[\Pi^0_k]$. We first show that Σ^0_k-bounded subshifts have certain geometric restrictions, so that the class does not contain some relatively simple sofic shifts.

Example 5.19. We present a binary sofic shift $X \subset \{0,1\}^{\mathbb{Z}^2}$ such that $X \notin \bar{\Pi}_1[\text{subshifts}]$. Define X as the vertically constant subshift whose projective subdynamics is exactly the one-dimensional subshift $\mathcal{B}^{-1}(0^*110^+10^+110^*)$. It is clear that X is sofic (this also follows from Theorem 3.18). Suppose that we have $X \subset X_\psi$ for an extended MSO formula

$$\psi = \forall \mathsf{y}_1[Y_1] \cdots \forall \mathsf{y}_n[Y_n] \forall \vec{\mathsf{n}}_1 \cdots \forall \vec{\mathsf{n}}_\ell \ \phi,$$

where ϕ is quantifier-free and the Y_i are arbitrary subshifts. Let $r \in \mathbb{N}$ be the radius of ψ, and let $x \in \{0,1\}^{\mathbb{Z}^2}$ be the vertically constant configuration defined by

$$\cdots x_{(-2,0)} x_{(-1,0)} . x_{(0,0)} x_{(1,0)} \cdots = {}^\infty 011.0^{r\ell+1}110^\infty.$$

Now, for all values of the variables y_i and the coordinates $\vec{\mathsf{n}}_j$, there exists $m \in [0, r\ell]$ such that the vertical column $V = \{m\} \times \mathbb{Z}$ does not contain the value of any term in ϕ. This means that changing every cell $x_{\vec{v}}$ for $\vec{v} \in V$ into a 1 does not affect the truth value of ϕ. The changed configuration is in X, so it models ϕ with the above choices of the y_i and $\vec{\mathsf{n}}_j$, and thus x also models ϕ. But we clearly have $x \notin X$, which implies $X \neq X_\psi$.

From Example 5.16 and Example 5.19, we have the following corollary.

Corollary 5.20. *The class of sofic shifts is incomparable to* $\bar{\Pi}_1[\text{SFT}]$.

It is easy to see that for all $k \in \mathbb{N}$, the class $\bar{\Sigma}_2[\Pi^0_k]$ contains all subshifts of the form $X \cap Y$, where $X \in \bar{\Pi}_1[\Pi^0_k]$ and Y is a sofic shift. We end this section by presenting an example, due to Ville Salo, which shows this inclusion to be strict.

Example 5.21. Let $A = \{0, 1, \#\}$, and let $X \subset A^{\mathbb{Z}^2}$ be the subshift defined by the following set of forbidden patterns. First, every three-element pattern

$\{\vec{0} \mapsto \#, (0,1) \mapsto b, (0,n) \mapsto \#\}$ for $b \in \{0,1\}$ and $n \geq 2$ is forbidden, as is every pattern containing two #-symbols on different vertical columns. This means that a configuration of X contains at most one vertical segment of #-symbols, and is otherwise binary. Second, every rectangular pattern $P \in A^{(2n+1)\times(n+2)}$ whose central vertical column contains $b\#^n c$ for some $b, c \neq \#$, and whose two sides satisfy

$$|\{(i,j) \in [0,n-1]^2 \mid P_{(i,j+1)} \neq P_{(n+1+i,j+1)}\}| < \lceil \log n \rceil,$$

is forbidden. In other words, the two square patterns on either side of a height-n segment of #-symbols can disagree on at most $\lceil \log n \rceil$ coordinates.

First, we sketch a proof for the fact that X is the image of a Σ^0_0-bounded subshift under a block map. For that, let Y be the following Σ^0_0-bounded subshift defined over an alphabet $A \times B$. We require by the same forbidden patterns as with X that configurations of Y contain at most one vertical segment of #-symbols on the first layer. On the second layer, we simulate a CMS M using Construction 3.12 so that the computation cone is placed on the unique pattern $\substack{b \\ \#}$, where $b \in \{0,1\}$. Its infinite input string is the entire first layer, copied onto the right border of the computation cone by infinitely many two-cell forbidden patterns, using some computable ordering of $\mathbb{Z}^2$. The machine M checks that the first layer is actually a configuration of X by simply checking for larger and larger square patterns around the base of its cone that they do not contain any of the forbidden patterns mentioned in the previous paragraph. The projection to the first layer of Y then gives exactly X, and it is easy to see that the set of forbidden patterns for Y is computable.

Second, we show that X is not an intersection of a $\bar{\Pi}_1[\Pi^0_k]$ subshift and a sofic shift for any $k \in \mathbb{N}$. For that, suppose for contradiction that $Y \cap Z = X$, where $Y \in \bar{\Pi}_1[\Pi^0_k]$ and Z is a sofic shift. First, Theorem 5.17 implies that Y is defined by a set of forbidden patterns whose domain size is bounded by some $K \in \mathbb{N}$. For each $n \geq \exp(K)$, the subshift Y contains every configuration $y \in \{0,1,\#\}^{\mathbb{Z}^2}$ with exactly one height-n vertical segment of #-symbols, since every size-K patterns that occurs in y is in $\mathcal{B}(X)$. Thus, given such a configuration y, we have $y \in X$ if and only if $y \in Z$. Let $\mathcal{P} \subset \{0,1\}^{[0,n-1]^2}$ be a maximal-cardinality set of binary $n \times n$ square patterns, any two elements of which differ in more than $\lceil \log n \rceil$ coordinates. We clearly have $|\mathcal{P}| \geq 2^{\lfloor n^2/(2+\log n) \rfloor}$. Now, for each pattern $P \in \mathcal{P}$, let $x^P \in X$ be a configuration that contains the pattern P at the origin (that is, $x^P|_{[0,1]^2} = P$), a height-n segment of #-symbols on its right border, and on the right-hand side of the segment, a translated copy of P. By the above, we have $x^P \in Z$, and since the size of $\mathcal{P}$ is superexponential in n, Lemma 3.15 implies that for large enough n, there are two distinct patterns $P, Q \in \mathcal{P}$ such that $c(x^P, x^Q, n) \in Z$. But P and Q differ in more than $\lceil \log n \rceil$ coordinates, so that $c(x^P, x^Q, n) \notin X$, a contradiction.

Chapter 6

Quantifier Extensions of Two-Dimensional Subshifts

6.1 Introduction

In this chapter, we continue the theme of defining multidimensional subshifts by logical formulas, although in a more restricted form than in Chapter 5. We define two operations that extend one subshift by another, called *quantifier extensions*. The name comes from the intuition that these operations in some sense correspond to universal and existential quantification in logical formulas. Also, while second-order quantifiers in extended MSO formulas allow us to define a subshift $X \subset A^{\mathbb{Z}^2}$ with the help of another (possibly entirely unrelated) subshift $Y \subset B^{\mathbb{Z}^2}$, the quantifier extensions take an auxiliary subshift Y over the same alphabet A, and extend X into a subshift $Z \supset X$ over a larger alphabet $\hat{A} \supset A$. Intuitively, the extended alphabet contains a 'wildcard' symbol where we can substitute a pattern of Y, and the extended subshift contains those configurations where this substitution can be done. They are inspired by the concept of *multi-choice shift spaces*, as defined in [LMP13] and studied further in [MP11]. The multi-choice shift space of a binary subshift is simply its universal extension by the binary full shift. Multi-choice shift spaces, on the other hand, were inspired by *constrained systems with unconstrained positions*, which are studied in coding theory [CMNW02, PCM06].

The main results of this chapter show that the class of sofic shifts is closed under the quantifier extensions in the one-dimensional case, but not in general, even when the sofic shifts are extended by very simple subshifts. In a way, this is to be expected, since one-dimensional sofic shifts correspond to regular languages, so most natural (and some unnatural) operations respect the property of being sofic, and on the other hand, the class of multidimensional sofic shifts is much less well-behaved, and still badly understood. The topic of this chapter is more concrete than that of Chapter 5, and the results are also sharper, at least in some sense. Our main result also solves an open problem presented in [LMP13].

This chapter is based on the article [Tör14a].

6.2 The Quantifier Extensions

We begin by defining our objects of interest, the quantifier extensions of multidimensional subshifts.

Definition 6.1. Let $X, Y \subset A^{\mathbb{Z}^d}$ be d-dimensional subshifts, let $\diamond \notin A$ be a new symbol and denote $\hat{A} = A \cup \{\diamond\}$. For two possibly infinite patterns $P \in \hat{A}^D$ and $Q \in A^D$ of the same shape $D \subset \mathbb{Z}^d$, denote by $P^{(Q)} \in A^D$ the pattern with

$$P^{(Q)}_{\vec{v}} = \begin{cases} Q_{\vec{v}}, & \text{if } P_{\vec{v}} = \diamond \\ P_{\vec{v}}, & \text{otherwise.} \end{cases}$$

In other words, $P^{(Q)}$ is defined by replacing the $\diamond$-symbols of P with the corresponding contents of Q. We define two *quantifier extension subshifts* for X and Y, the *universal extension* $A(X, Y) \subset \hat{A}^{\mathbb{Z}^d}$ and the *existential extension* $E(X, Y) \subset \hat{A}^{\mathbb{Z}^d}$, with the respective sets of forbidden patterns

$$\{P \in \mathcal{P}_d(A) \mid \exists Q \in \mathcal{B}_{D(P)}(Y) : P^{(Q)} \notin \mathcal{B}(X)\} \text{ and}$$
$$\{P \in \mathcal{P}_d(A) \mid \forall Q \in \mathcal{B}_{D(P)}(Y) : P^{(Q)} \notin \mathcal{B}(X)\}$$

It is easy to see that for a configuration $x \in \hat{A}^{\mathbb{Z}^d}$, we have $x \in A(X, Y)$ if and only if $x^{(y)} \in X$ holds for all $y \in Y$, and $x \in E(X, Y)$ if and only if there exists $y \in Y$ such that $x^{(y)} \in X$. In the formalism of MSO logic, the two extensions are defined by the MSO formulas

$$Q\mathsf{y}[Y]\exists\mathsf{x}[X]\forall\vec{\mathsf{n}} \bigwedge_{a \in A} \big((\mathsf{x}_{\vec{\mathsf{n}}} = a) \Rightarrow (\mathsf{P}_a(\vec{\mathsf{n}}) \vee (\mathsf{P}_\diamond(\vec{\mathsf{n}}) \wedge \mathsf{y}_{\vec{\mathsf{n}}} = a))\big), \tag{6.1}$$

where the quantifier Q is $\forall$ in the case of $A(X, Y)$, and $\exists$ in the case of $E(X, Y)$. If the subshift X is definable by an MSO formula ϕ, we can simplify (6.1) to $Q\mathsf{y}[Y]\phi'$, where ϕ' is obtained from ϕ by replacing every symbol predicate $\mathsf{P}_a(t)$ with the formula $\mathsf{P}_a(t) \vee (\mathsf{P}_\diamond(t) \wedge \mathsf{y}_t = a)$. Thus, the symbol $\diamond$ represents a 'hole' in the configuration x that can be filled with the contents of another configuration y, and the extensions quantify over all such y to decide whether x is valid or not.

Our perspective in this chapter is to study which properties the extensions respect. More explicitly, if one or both of X and Y possess a property, like being an SFT or sofic shift, do the extensions also possess it? We mainly focus on the universal extension, as the question of soficness is particularly interesting for it. First, we show that the universal extension respects the property of being an SFT in the following sense.

Proposition 6.2. *Let $X \subset A^{\mathbb{Z}^d}$ be an SFT and $Y \subset A^{\mathbb{Z}^d}$ any subshift. Then the universal extension $A(X,Y)$ is an SFT.*

Proof. Let $F \subset \mathcal{P}_d(A)$ be a finite set of forbidden patterns for X. Without loss of generality, we may assume that the patterns of F have a common finite domain $D \subset \mathbb{Z}^d$. We claim that

$$\hat{F} = \{P \in \hat{A}^D \mid \exists Q \in \mathcal{B}_D(Y) : P^{(Q)} \in F\}$$

is a set of forbidden patterns for $A(X,Y)$. For that, let $x \in \hat{A}^{\mathbb{Z}^d}$ be arbitrary. If $x \in A(X,Y)$, it is clear that no patterns of $\hat{F}$ occur in x. Conversely, if $x \notin A(X,Y)$, then there exists a configuration $y \in Y$ such that $x^{(y)} \notin X$, or in other words, there exists $\vec{n} \in \mathbb{Z}^d$ such that $x^{(y)}|_{D+\vec{n}} \in F$. This implies that the pattern $P = x|_{D+\vec{n}}$ of x is in $\hat{F}$, since the corresponding pattern $Q = y|_{D+\vec{n}} \in \mathcal{B}_D(Y)$ of y satisfies $P^{(Q)} \in F$. Since $\hat{F}$ is finite, $A(X,Y)$ is an SFT, as claimed. $\square$

Example 6.3. The above result does not hold for the existential extension, even in the simple one-dimensional case where $Y = \{0,1\}^{\mathbb{Z}}$ and $X \subset \{0,1\}^{\mathbb{Z}}$ is a mixing SFT. Namely, let X be defined by the single forbidden pattern 0100, and consider the eventually periodic configurations $x = {}^\infty 01(0\diamond)^\infty$ and $x' = {}^\infty(0\diamond)0^\infty$. We have $x \in E(X,Y)$ by substituting the letter 1 to every $\diamond$, and $x' \in E(X,Y)$ by substituting 0. It is easy to see that these are the only possible substitutions, and thus ${}^\infty 01(0\diamond)^n 0^\infty \notin E(X,Y)$ for all $n \geq 1$. This implies that $E(X,Y)$ is not an SFT.

The following result shows in particular that even though the extension in the above example is not an SFT, it is sofic. The special case $X \subset Y = \{0,1\}^{\mathbb{Z}}$ of the one-dimensional universal extension was first proved in [PCM06].

Proposition 6.4. *Let $X, Y \subset A^{\mathbb{Z}^d}$ be d-dimensional sofic shifts. Then the existential extension $E(X,Y)$ is sofic. If $d = 1$, then the universal extension $A(X,Y)$ is also sofic.*

Proof. Let $\phi : X' \to X$ and $\psi : Y' \to Y$ be surjective block maps, where X' and Y' are SFTs, and define the auxiliary subshift $Z = \{\#, \diamond\}^{\mathbb{Z}^d}$. Then $E(X,Y)$ is obtained from the SFT

$$\{(x,y,z) \mid \forall \vec{v} \in \mathbb{Z}^d : \phi(x)_{\vec{v}} = \psi(y)_{\vec{v}} \vee z_{\vec{v}} \neq \diamond\} \subset X' \times Y' \times Z,$$

by applying the block map ξ defined by

$$\xi(x, y, z)_{\vec{v}} = \begin{cases} \diamond, & \text{if } z_{\vec{v}} = \diamond \\ \phi(x)_{\vec{v}}, & \text{otherwise.} \end{cases}$$

Alternatively, since X and Y are sofic shifts, they are defined by some $\bar{\Sigma}_1$ formulas of the form $\phi_1 = \exists \mathsf{Z}_1^1 \cdots \exists \mathsf{Z}_1^n \forall \vec{\mathsf{n}}_1 \psi_1$ and $\phi_2 = \exists \mathsf{Z}_2^1 \cdots \exists \mathsf{Z}_2^m \forall \vec{\mathsf{n}}_2 \psi_2$, respectively, where the ψ_i are quantifier-free. By the simplified version of (6.1), the extension $E(X, Y)$ is then defined by an extended MSO formula $\exists \mathsf{y}[Y] \exists \mathsf{Z}_1^1 \cdots \exists \mathsf{Z}_1^n \forall \vec{\mathsf{n}}_1 \psi_1'$, where ψ_1' is also quantifier-free. We then apply Lemma 5.1 to this formula, rearrange the terms, and reuse the first-order variables, obtaining the equivalent MSO formula

$$\phi = \exists \mathsf{Y}^1 \cdots \exists \mathsf{Y}^k \exists \mathsf{Z}_1^1 \cdots \exists \mathsf{Z}_1^n \exists \mathsf{Z}_2^1 \cdots \exists \mathsf{Z}_2^m \forall \vec{\mathsf{n}} (\bar{\psi}_2 \wedge \bigvee_{i=1}^{k} (\mathsf{Y}^i(\vec{\mathsf{n}}) \wedge \bigwedge_{j \neq i} \neg \mathsf{Y}^j(\vec{\mathsf{n}})) \wedge \hat{\psi}_1'),$$

where the Y^i are ordinary second-order variables, and $k \in \mathbb{N}$ is the size of the alphabet of Y. Also, the subformulas $\hat{\psi}_1'$ and $\bar{\psi}_2$ are quantifier-free, so ϕ is a $\bar{\Sigma}_1$ formula. Theorem 5.10 implies that ϕ defines a sofic shift.

In the case $d = 1$, we know that a subshift is sofic if and only if its language is regular, if and only if it can be defined by a regular set of forbidden words [LM95, Chapter 3]. The sets $\mathcal{B}(Y)$ and $L = A^* \setminus \mathcal{B}(X)$ are thus regular, so they are recognized by some finite automata F_1 and F_2. Now, $A(X, Y)$ is defined by the set

$$\{w \in \hat{A}^* \mid \exists v \in \mathcal{B}_{|w|}(Y) : w^{(v)} \in L\}$$

of forbidden words. This set is regular, since it can be recognized by a nondeterministic finite automaton that guesses the word v one letter at a time, and checks that $v \in \mathcal{B}(Y)$ and $w^{(v)} \in L$ by simulating F_1 and F_2. Thus the universal extension $A(X, Y)$ is sofic. $\square$

In higher dimensions, we have a very weak analogue of Proposition 6.4 for the universal extension, where automata theory is replaced by computability theory. Recall from Lemma 3.16 that the language of a multidimensional sofic shift is always Π_1^0.

Lemma 6.5. *Let $X, Y \subset A^{\mathbb{Z}^d}$ be Π_1^0 subshifts. Then the universal extension $A(X, Y)$ is Π_2^0. If X is Π_1^0 and Y is Σ_1^0, then $A(X, Y)$ is also Π_1^0.*

Proof. Let $P \in \hat{A}^{[0,n-1]^d}$ be an arbitrary hypercube pattern. We have $P \notin \mathcal{B}(A(X, Y))$ if and only if there exists a pattern $Q \in A^{[0,n-1]^d}$ such that $Q \in \mathcal{B}(Y)$ but $P^{(Q)} \notin \mathcal{B}(X)$. Since the languages of X and Y are Π_1^0, the proposition $Q \in \mathcal{B}(Y)$ is Π_1^0, while $P^{(Q)} \notin \mathcal{B}(X)$ is Σ_1^0. Thus the complement of $\mathcal{B}(A(X,Y))$ is a Σ^0_2 language, implying that $\mathcal{B}(A(X,Y))$ is a Π^0_2 subshift. Furthermore, if $\mathcal{B}(Y)$ is a Σ^0_1 language, then the complement of $\mathcal{B}(A(X,Y))$ is also Σ^0_1, and we obtain the latter claim. $\square$

The first bound in the above result is sharp, even in the case of one-dimensional subshifts.

Proposition 6.6. *There exist two countable one-dimensional Π^0_1 subshifts $X, Y \subset A^{\mathbb{Z}}$ such that the language of the universal extension $A(X,Y)$ is Π^0_2-complete.*

Proof. Define $A = \{0, \ldots, 6\}$, and define X by the set of forbidden patterns

$$\{46\} \cup \{ij \mid i,j \in S, i > j\} \cup \{01^a2^{b+1}3^c4^d5 \mid a,b,c,d \in \mathbb{N}, a = c\}.$$

It is clear that X is a Π^0_0 subshift, and it is easily seen to be countable. Next, let Φ be an arithmetical formula with bounded quantifiers such that the set

$$N = \{k \in \mathbb{N} \mid \forall m \in \mathbb{N} : \exists n \in \mathbb{N} : \Phi(k,m,n)\}$$

is Π^0_2-hard. Define Y by the set

$$\{0, 1, 46\} \cup \{ij \mid i,j \in S, i > j\} \cup \{23^k4^m5 \mid k,m \in \mathbb{N}, \exists n \in \mathbb{N} : \Phi(k,m,n)\}$$

of forbidden patterns. Since this set is Σ^0_1 by form, Y is a countable Π^0_1 subshift.

Define the function $w : \mathbb{N} \to L = \{01^k2\diamond \mid k \in \mathbb{N}\}$ by $w(k) = 01^k2\diamond$. We note that for all $a \in A$, there exists a letter $b \in \mathcal{B}_1(Y)$ such that ba is forbidden in X, which implies that $\diamond a$ is forbidden in $A(X,Y)$. Thus $w(k)$ occurs in $A(X,Y)$ if and only if the infinite tail $w(k)\diamond^\infty$ does. By the definition of X and Y, this is the case if and only if $23^k4^m5 \notin \mathcal{B}(Y)$ for all $m \in \mathbb{N}$. But this is equivalent to $k \in N$, which means that $N = w^{-1}(L \cap \mathcal{B}(A(X,Y)))$, and thus the language of $A(X,Y)$ is Π^0_2-hard. Lemma 6.5 implies that $A(X,Y)$ is a Π^0_2 subshift, so the claim is proved. $\square$

As a corollary of Lemma 6.5, Theorem 3.18, and the above proposition, we obtain the following counterpart of Proposition 6.4.

Corollary 6.7. *There exist countable sofic shifts $X, Y \subset A^{\mathbb{Z}^2}$ such that the universal extension $A(X,Y)$ is not sofic.*

Proof. Let $X', Y' \subset S^{\mathbb{Z}}$ be the Π^0_1 subshifts of Proposition 6.6. By Theorem 3.18, the vertically periodic countable two-dimensional subshifts $X, Y \subset A^{\mathbb{Z}^2}$ whose projective subdynamics are X' and Y', respectively, are sofic. It is easy to see that the extension $A(X,Y)$ is just the vertically periodic two-dimensional version of $A(X',Y')$. Thus $A(X,Y)$ is Π^0_2-hard, and by Lemma 3.16, it is not sofic. $\square$

While this result is interesting in itself, the proof is not very satisfying, since the subshift Y that we extend by is computationally complex, and we use the simpler structure of X only to check a universally quantified property of $\mathcal{B}(Y)$. It can be said that the computational complexity of the extension is entirely due to that of Y. However, if we restrict Y to be a computable subshift, meaning that its language is both Π^0_1 and Σ^0_1, then Lemma 3.16 shows that the language of $A(X,Y)$ is Π^0_1, so a recursion theoretic proof for the nonsoficness of $A(X,Y)$ will no longer work.

In the next section, we concentrate on the problem of finding pairs of computationally simple sofic shifts X and Y such that the extension $A(X,Y)$ is not sofic. The following special case was implicitly left as an open problem in [LMP13]: is the extension $A(X,\{0,1\}^{\mathbb{Z}^2})$ a sofic shift for every two-dimensional binary sofic shift $X \subset \{0,1\}^{\mathbb{Z}^2}$? Theorem 6.9 in particular shows that the answer is negative.

In [LMP13], the following notions were defined.

Definition 6.8. Let $X \subset A^{\mathbb{Z}^d}$ be a d-dimensional subshift, and denote by $\tilde{A}$ the nonempty subsets of A. The *multi-choice shift space* associated to X is defined as the subshift $\tilde{X} \subset \tilde{A}^{\mathbb{Z}^d}$, which contains exactly those configurations $\tilde{x} \in \tilde{A}^{\mathbb{Z}^d}$ for which every configuration $x \in A^{\mathbb{Z}^d}$ that satisfies $x_{\vec{n}} \in \tilde{x}_{\vec{n}}$ for all $\vec{n} \in \mathbb{Z}^d$ is in X. For a pattern $P \in \mathcal{P}_d(\tilde{A})$, denote $\Pi(P) = \prod_{\vec{n} \in D(P)} |P_{\vec{n}}|$. The *independence entropy* of X is the quantity

$$h_{\text{ind}}(X) = \lim_{n \to \infty} \frac{1}{n^d} \max\{\log \Pi(P) \mid P \in \mathcal{B}_{[0,n-1]^d}(\tilde{X})\}$$

Intuitively, the independence entropy of X is a measure for how much of the entropy of X comes from collections of coordinates whose value can be chosen independently. In the binary case $A = \{0,1\}$, the multi-choice shift space $\tilde{X}$ is identical to $A(X,\{0,1\}^{\mathbb{Z}^d})$ up to renaming the symbols. In [LMP13], it was asked whether the multi-choice shift space associated to a two-dimensional sofic shift is necessarily sofic, and the above problem is a restatement of this question in the binary case.

6.3 Universal Extensions of Sofic Shifts

In this section, we characterize those two-dimensional subshifts Y that always yield sofic universal extensions $A(X,Y)$ for sofic shifts X. It turns out that this holds only in the class of finite subshifts, or in other words, every infinite subshift extends some sofic shift to a nonsofic shift, and conversely, a finite subshift does not. The result is likely to hold for any number of dimensions, but we restrict our attention to the two-dimensional case for simplicity.

Theorem 6.9. *Let $Y \subset A^{\mathbb{Z}^2}$ be a subshift. The following conditions are equivalent:*

1. *Y is finite.*
2. *$\mathcal{A}(X, Y)$ is sofic for all sofic shifts $X \subset B^{\mathbb{Z}^2}$ over all alphabets B.*
3. *$\mathcal{A}(X, Y)$ is sofic for all strongly irreducible sofic shifts $X \subset A^{\mathbb{Z}^2}$.*
4. *$\mathcal{A}(X, Y)$ is sofic for all countably covered sofic shifts $X \subset A^{\mathbb{Z}^2}$.*

Before proceeding to the proof, we present a couple of auxiliary results. Most of them are not needed in the special case of binary full shifts, but we use them because Theorem 6.9 is much more general. First, we need the following result from symbolic dynamics. It is slightly stronger than the version in [LM95], but the missing details can be extracted from its proof.

Lemma 6.10 (Marker Lemma)**.** *Let A be a finite alphabet. For all $n \in \mathbb{N}$ there exists a block map $f_n : A^{\mathbb{Z}} \to \{0,1\}^{\mathbb{Z}}$, called the n-*marker map*, such that:*

1. *The radius of f_n is at most $r_n = n^{|A|^{2n+1}}$.*
2. *For $k < n-1$, the word 10^k1 does not occur in any configuration of $f_n(A^{\mathbb{Z}})$.*
3. *If $x \in A^{\mathbb{Z}}$ is such that $f_n(x)_{[-n+1,n-1]} = 0^{2n-1}$, then the word $x_{[-n,n]}$ is periodic with period less than n.*
4. *The function $(n, w) \mapsto F_n(w)$, where $w \in A^{r_n}$ and F_n is the local function of f_n, is computable.*

Proof. Lemma 10.1.8 of [LM95] states the existence of a clopen set $C \subset A^{\mathbb{Z}}$ such that the shifted sets $\sigma^i(C)$ are disjoint for all $i \in [0, n-1]$, and if a configuration $x \in A^{\mathbb{Z}}$ satisfies $\sigma^i(x) \notin C$ for all $i \in [-n+1, n-1]$, then the word $x_{[-n,n]}$ is periodic with period less than n. We define the marker map by $f_n(x)_i = 1$ if and only if $\sigma^i(x) \in C$ for all $i \in \mathbb{Z}$, and then $f_n : A^{\mathbb{Z}} \to \{0,1\}^{\mathbb{Z}}$ is a block map satisfying (2) and (3). The remaining claims follow from the construction of C in [LM95]. $\square$

Finally, we present a general construction of grid-like countably covered sofic shifts.

Lemma 6.11. *For all $m, n \in \mathbb{N} \cup \{\infty\}$, there exists a countably covered sofic shift $X = X_{G(m,n)}$ over the alphabet $\{\#\} \cup \{0,1\}^2$ such that:*

- *For all $x \in X$, the set $D(x) = \{\vec{v} \in \mathbb{Z}^2 \mid x_{\vec{v}} \neq \#\}$ is a (possibly infinite) rectangle.*

- *For all* $k \geq 2$, $1 \leq a \leq \binom{m+k}{k}$ *and* $1 \leq b \leq \binom{n+k}{k}$, *there is a configuration* $x = x^{k,a,b} \in X$ *such that* $D(x) = [0, ak] \times [0, bk]$, *and for all* $\vec{v} = (i, j) \in D(x)$ *we have* $\pi_1(x_{\vec{v}}) = 1$ ($\pi_2(x_{\vec{v}}) = 1$) *if and only if* $i \equiv 0 \bmod k$ ($j \equiv 0 \bmod k$, *respectively), where* $\pi_1, \pi_2 : \{0,1\}^2 \to \{0,1\}$ *are the projections to the first and second components.*

- *If* $m = \infty$ ($n = \infty$), *then every horizontal (vertical) line of* 1*'s in the second (first) layer of* X *is infinite to the right (upwards), and otherwise, every configuration of* X *contains only finitely many vertical (horizontal) lines of* 1*'s in its first (second) layer.*

Proof. We construct a countable SFT $Y \subset A^{\mathbb{Z}^2}$ and a symbol map $\pi : A \to \{0,1\}$ such that $\pi(Y) = X$. The alphabet A is the set of tiles in Figure 6.1, where the labels C_ℓ range over $[0, \ell]$ if $\ell \in \mathbb{N}$, and $\{\infty\}$ if $\ell = \infty$. Note that some tiles are forbidden if m or n is infinite. Every 2×2 pattern where the lines or colors of some tiles do not match (including the diagonal lines) is forbidden in Y. Then the regions colored by L, R, B, and T in a configuration $y \in Y$, if nonempty, form a left half plane, a right half plane, a downward infinite rectangle, and an upward infinite rectangle, respectively. The rectangular area not contained in them is called the *grid* of y. It is divided into rectangles by the *grid lines* (the thick lines in Figure 6.1), which stretch from one end of the grid to the other. These rectangles are actually squares, all of the same size, because of the diagonal lines. We also forbid every 2×2 pattern containing such a square, so that every square contains at least one *interior tile*, shown in the fifth column of the figure. If m (n) is infinite, then so is the width (height) of every grid, as it cannot have a right (top, respectively) border. See Figure 6.2 for an example configuration of Y.

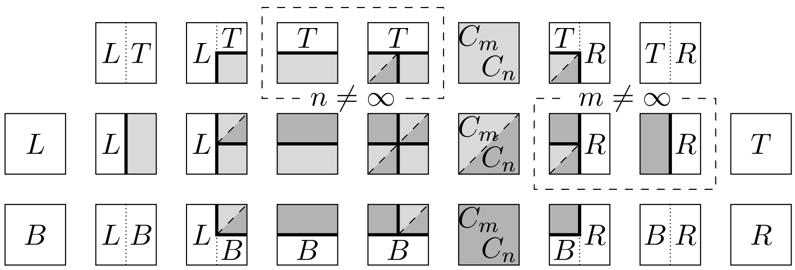

Figure 6.1: The alphabet of the grid SFT Y in the proof of Lemma 6.11.

The labels C_m of the interior tiles have the following rules if $m \neq \infty$. Inside a square of the grid, they must be horizontally constant and downward increasing (all patterns cd for $c \neq d$ and $\begin{smallmatrix} c \\ d \end{smallmatrix}$ for $c > d$ are forbidden). On the

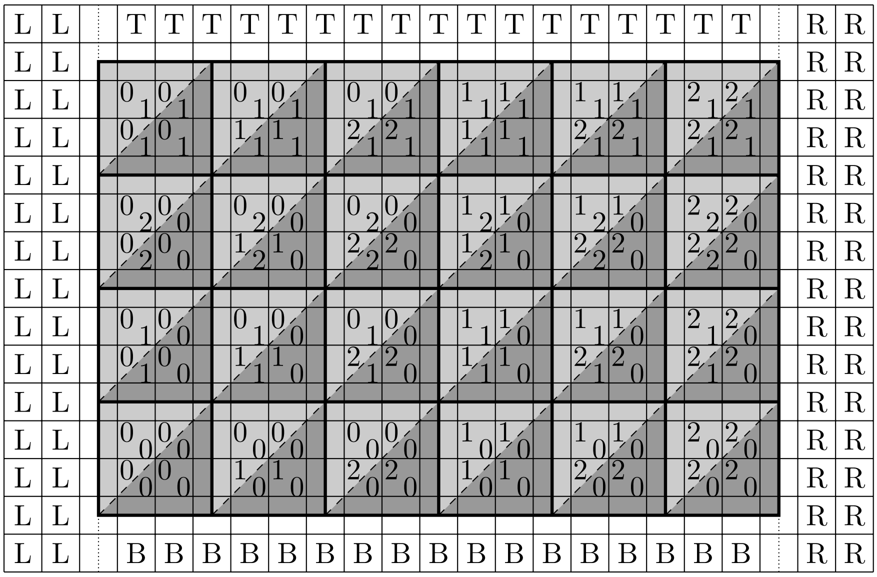

Figure 6.2: A configuration of the grid SFT Y, where $n = m = 2$. The white tiles are mapped to the symbol $\#$, and the horizontal and vertical grid lines to symbols 1 in the two layers of $X_{G(m,n)}$.

border of two horizontally adjacent squares, all 3×2 patterns except

$$\begin{bmatrix} - + - \\ e \mid e \end{bmatrix} \begin{bmatrix} - + - \\ e \mid e{+}1 \end{bmatrix} \begin{bmatrix} e \mid e \\ f \mid f \end{bmatrix} \begin{bmatrix} e \mid e \\ f \mid f{+}1 \end{bmatrix} \begin{bmatrix} e \mid e{+}1 \\ m \mid e{+}1 \end{bmatrix} \begin{bmatrix} m \mid c \\ m \mid c \end{bmatrix} \begin{bmatrix} e \mid e{+}1 \\ - + - \end{bmatrix} \begin{bmatrix} m \mid c \\ - + - \end{bmatrix}$$

for $c, d \in [0, m]$ and $e, f \in [0, m-1]$ are forbidden (the symbols $-$, $|$ and $+$ represent horizontal, vertical and crossing grid lines, including T-junctions). Denote by V_m^k the set of length-k downward increasing column vectors over $[0, m]$. The above rules imply that for all horizontally adjacent grid squares with $k \times k$ interior tiles, the column vector formed by the top-left labels of the interior tiles of the right square is the lexicographical successor of that of the left square with respect to the set V_m^k. For example, the lexicographical successor of $\begin{smallmatrix} 0 \\ 2 \\ 2 \end{smallmatrix}$ in V_2^3 is $\begin{smallmatrix} 1 \\ 1 \\ 1 \end{smallmatrix}$, not $\begin{smallmatrix} 1 \\ 0 \\ 0 \end{smallmatrix}$, since the latter vector is not in the set V_2^3. Because of this, the width of any grid containing such a square is at most $|V_m^k| = \binom{m+k}{k}$ squares. To ensure countability, we also require that the labels next to the left border of a grid are all 0. We introduce analogous rules for the top-left labels, but transposed, so that the height of the grid is at most $\binom{n+k}{k}$ squares. This concludes the definition of Y, and to define X, we specify the symbol map π. We set $\pi(t) \neq \#$ if and only if the tile t contains a gray region, and then $\pi_1(\pi(t)) = 1$ ($\pi_2(\pi(t)) = 1$) if and only if t contains a vertical (horizontal) grid line. The three conditions for X follow easily.

Finally, we show that Y is countable, and for that, let $y \in Y$. If a finite grid square occurs in y, then there are countably many choices for the position of the grid, which uniquely determines its contents (because of the restrictions on the column vectors introduced above) and the rest of y. If y contains no grid tiles, then it consists of the L, R, B and T-tiles, for which we have countably many choices. In the case of infinite squares, since the labels of the interior tiles are decreasing in one direction and constant in the other, our choices are again restricted to a countable set. $\square$

We are now ready to prove Theorem 6.9. In its proof, we will modify the sofic shifts $X_{G(m,n)}$ by superimposing new symbols on top of their configurations with Lemma 3.14. They provide a rigid geometric structure for the construction.

Proof of Theorem 6.9. ($1 \Rightarrow 2$): Let $Y \subset A^{\mathbb{Z}^2}$ be a finite subshift, and denote $Y = \{y^1, \ldots, y^k\}$. The configurations y^i must all be periodic, and we let $p \in \mathbb{N}$ be a common horizontal and vertical period for all of them. Let Z be a two-dimensional SFT and $\phi : Z \to X$ a surjective block map, and define the SFT $Z' \subset Z^k \times Y^k \times \{\diamond, \#\}^{\mathbb{Z}^2}$ as follows. A configuration $z' = (z^1, \ldots, z^k, y^{n_1}, \ldots, y^{n_k}, t) \in Z^k \times Y^k \times \{\diamond, \#\}^{\mathbb{Z}^2}$ is in Z' if and only if

- $\{y^{n_1}, \ldots, y^{n_k}\} = \{y^1, \ldots, y^k\}$, which can be checked by $p \times p$ patterns,
- if $t_{\vec{v}} = \diamond$ for some $\vec{v} \in \mathbb{Z}^2$, then $\phi(z^i)_{\vec{v}} = y^{n_i}_{\vec{v}}$ for all $i \in [1, k]$, and
- if $t_{\vec{v}} = \#$ for some $\vec{v} \in \mathbb{Z}^2$, then $\phi(z^i)_{\vec{v}} = \phi(z^j)_{\vec{v}}$ for all $i, j \in [1, k]$.

We then define the block map $\psi : Z' \to \hat{A}^{\mathbb{Z}^2}$ by

$$\psi(z')_{\vec{v}} = \begin{cases} \diamond, & \text{if } t_{\vec{v}} = \diamond, \\ \phi(z^1)_{\vec{v}}, & \text{otherwise.} \end{cases}$$

It is easily verified that $\psi(Z') = A(X, Y)$, and thus the extension is sofic.

($2 \Rightarrow 3$): Trivial.

($3 \Rightarrow 4$): Let $Z \subset B^{\mathbb{Z}^2}$ be a countable SFT defined by forbidden patterns of size 2×2, and let $\phi : B^{\mathbb{Z}^2} \to X$ be a symbol map with $\phi(Z) = X$, so that $X \subset A^{\mathbb{Z}^2}$ is a countably covered sofic shift. We may assume $|A| \geq 2$, which implies $X \neq A^{\mathbb{Z}^2}$, so let $P \in A^{n \times n}$ be a forbidden pattern of X for some $n \geq 2$, and let $P' \in B^{n \times n}$ be a pattern such that $\phi(P') = P$. Note that P' does not occur in the subshift Z. Define a new SFT $Z' \subset B^{\mathbb{Z}^2}$ by the set of forbidden patterns

$$F = \{Q \in B^{[-3n, 4n-1]^2} \mid Q|_{[0,n-1]^2} \notin \mathcal{B}(Z) \text{ and } P' \not\sqsubset Q\},$$

so that Z' intuitively contains configurations that locally look like configurations of Z, except in the vicinity of occurrences of P', where everything is allowed. Our intermediate goal is to prove that Z' is strongly irreducible, and high-level the idea is that given any two correct patterns of Z' whose domains are far apart, we fill the remaining part of $\mathbb{Z}^2$ with copies of P' and obtain a valid configuration. However, the since two patterns only have to be correct in Z', but not necessarily in Z, it may be that we have to carefully place some additional copies of P' near their borders, which makes the proof quite technical.

Claim 6.12. The SFT Z' is strongly irreducible with constant $12n$.

Proof of Claim. For this, let $Q^1, Q^2 \in \mathcal{B}(Z')$ be two patterns whose domains have minimum distance at least $12n$. We construct a configuration $y \in Z'$ such that $y|_{D(Q_i)} = Q_i$ for $i \in \{1, 2\}$. For this, let $z^i \in Z'$ be a configuration such that $z^i|_{D(Q^i)} = Q^i$. Denote

$$D^i = \{\vec{v} \in \mathbb{Z}^2 \mid \forall \vec{w} \in [0, 2n-1]^2 : (\vec{v} - \vec{w} + [0, 2n-1]^2) \cap D(Q^i) \neq \emptyset\}$$

and

$$E^i = [0, n-1]^2 + \{\vec{v} \in \mathbb{Z}^2 \mid z^i|_{[0,n-1]^2+\vec{v}} = P', D^i \cap ([-2n-1, 3n]^2 + \vec{v}) \neq \emptyset\},$$

and set $G^i = D^i \cup E^i$. The set D^i is a 'thickened' version of $D(Q^i)$, which has the property that every $\vec{v} \notin D^i$ is an element of some $2n \times 2n$ square that does not intersect D^i. The set E^i, on the other hand, contains the domain of every occurrence of P' in z^i not too far from the thickened set D^i.

Let $x \in B^{\mathbb{Z}^2}$ be a periodic configuration defined by $x|_{[0,n-1]^2+(in,jn)} = P'$ for all $i, j \in \mathbb{Z}$, and define the configuration y by

$$y_{\vec{v}} = \begin{cases} z^1_{\vec{v}}, & \text{if } \vec{v} \in G^1, \\ z^2_{\vec{v}}, & \text{if } \vec{v} \in G^2, \\ x_{\vec{v}}, & \text{otherwise.} \end{cases}$$

Note that $G^1 \cap G^2 = \emptyset$, since the minimum distance between these sets is at least $2n + 2$. Since $D(Q^i) \subset G^i$, we have $y|_{D(Q_i)} = Q_i$ for $i \in \{1, 2\}$, so it remains to show that $y \in Z'$. For this, let $\vec{v} \in \mathbb{Z}^2$ be arbitrary, and denote $Q = y|_{[-3n,4n-1]+\vec{v}}$. We prove that $Q \notin F$.

Suppose first that $\vec{v} + \vec{w} \notin D^1 \cup D^2$ for some $\vec{w} \in [0, n-1]^2$. Because of the properties of D^1 and D^2, and since their minimum distance is at least $4n + 2$, there exists a $2n \times 2n$ square domain $D \subset \mathbb{Z}^2$ which is disjoint from $D^1 \cup D^2$ and contains $\vec{w}$. If D intersects $G^1 \cup G^2$, then it intersects $E^1 \cup E^2$, which implies that $P' \sqsubset Q$. If D does not intersect $G^1 \cup G^2$, then we have $y|_D = x|_D$, and it again follows that $P' \sqsubset y|_D \sqsubset Q$. In either case, we have $Q \notin F$.

Suppose then that $\vec{v}+[0, n-1]^2 \subset D^1 \cup D^2$. Since the sets D^i have positive minimum distance, we may further assume $\vec{v}+[0, n-1]^2 \subset D^1$ by symmetry.

Consider the pattern $Q' = z^1|_{D(Q)} \in \mathcal{B}(Z')$. If we have $Q'|_{[0,n-1]^2+\vec{v}} \in \mathcal{B}(Z)$, then $Q \notin F$, since $Q|_{[0,n-1]^2+\vec{v}} = Q'|_{[0,n-1]^2+\vec{v}}$. Otherwise we must have $P' \sqsubset Q'$ by the definition of Z', so let $\vec{w} \in [-3n, 3n]^2$ be such that $Q'|_{[0,n-1]^2+\vec{w}+\vec{v}} = P'$. A simple calculation shows that

$$([0, n-1]^2 + \vec{v}) \cap ([-2n-1, 3n]^2 + \vec{w} + \vec{v}) \neq \emptyset,$$

and since $[0, n-1]^2 + \vec{v} \subset D^1$, we have $[0, n-1]^2 + \vec{v} + \vec{w} \subset E^1 \subset G^1$ by the definition of the set E^1. Thus we have $Q|_{[0,n-1]^2+\vec{v}+\vec{w}} = P'$ by the definition of y, implying $Q \notin F$. This shows that $y \in Z'$, and finishes the proof. $\square$

Now, the image $X' = \phi(Z')$ is a strongly irreducible sofic shift, and then the extension $A(X', Y)$ is sofic by assumption. We define an SFT $\tilde{X} \subset \hat{A}^{\mathbb{Z}^2}$ by the forbidden patterns

$$\{Q \in \hat{A}^{[0,n-1]^2} \mid \exists T \in \mathcal{B}_{[0,n-1]^2}(Y) : Q^{(T)} = P\},$$

and claim that $A(X, Y) = A(X', Y) \cap \tilde{X}$, so that $A(X, Y)$ is also sofic. From $X \subset X'$ and $P \not\sqsubset X$ it follows that $A(X, Y) \subset A(X', Y) \cap \tilde{X}$. Let then $x \in A(X', Y) \cap \tilde{X}$, and let $y \in Y$ be arbitrary. Since $x \in A(X', Y)$, we have $x^{(y)} \in X'$, and since $x \in \tilde{X}$, we also have $P \not\sqsubset x^{(y)}$. This implies that $x^{(y)}$ has a ϕ-preimage in Z' where P' does not occur, and thus $x^{(y)} \in X$. Since y was arbitrary, we have $x \in A(X, Y)$, and the claim is proved.

($4 \Rightarrow 1$): Let $Y \subset A^{\mathbb{Z}^2}$ be infinite. Then for all $p \in \mathbb{N}$ there exists $y \in Y$ which is not p-periodic either in the horizontal or the vertical direction. If either condition cannot be satisfied for some p, then the other can be satisfied for all p. Thus we may assume that Y has no common horizontal period. We also have $|A| \geq 2$.

Our goal is now to construct a countably covered sofic shift $X \subset A^{\mathbb{Z}^2}$ such that $A(X, Y)$ is not sofic. We proceed by constructing a countably covered sofic shift Z and a block map $\phi : Z \to A^{\mathbb{Z}^2}$ whose image we define as X. After this, it will be easy to show that the extension is not sofic, using Lemma 3.15. The subshift Z is a subset of $Z_1 \times Z_2 \times Z_3$, where Z_1 is the *input layer*, Z_2 the *computation layer*, and Z_3 the *output layer*. We define the layers sequentially:

1. Define Z_1, a countably covered sofic shift, using Lemma 6.11.

2. Define the SFT Z_2 using Construction 3.12, and a countable sofic shift $Z' \subset Z_1 \times Z_2$.

3. Define Z_3, a countably covered sofic shift, using Lemma 6.11 a second time, and construct the sofic shift $Z \subset Z' \times Z_3$.

The purpose of the three layers is the following. First, the input layer encodes a number $m \in \mathbb{N}$, an arbitrary square pattern $S \in \{0,1\}^{[0,m-1]^2}$, and an arbitrary coordinate $\vec{n} \in [0, m-1]^2$. The index $\vec{n}$ is further encoded into a word $w \in A^*$, but it can be decoded using the marker map of Lemma 6.10. This data is given as input to a CMS M simulated on the computation layer. Finally, the output layer encodes another square pattern $R \in \{0,1\}^{[0,m-1]^2}$, which is required to satisfy $S_{\vec{n}} = R_{\vec{n}}$. The information about the number n and the coordinate $\vec{n}$ is passed to the output layer by the output counters of M. We construct Z and the block map ϕ so that the word w can be an arbitrary pattern of Y in the universal extension $A(X,Y)$, which implies that we must have $S_{\vec{n}} = R_{\vec{n}}$ for all $\vec{n} \in [0, m-1]^2$, implying $S = R$. An application of Lemma 3.15 then gives us a contradiction.

We begin with Z_1, which is defined by superimposing a label from $A \times \{0,1\}$ on each vertical column of the grid shift $X_{G(1,\infty)}$ given by Lemma 6.11. More formally, let us define an auxiliary label alphabet $L_A = \{\square\} \cup (A \times \{0,1\})$, where $\square$ is a new symbol meaning 'no label,' and denote by $G = \{\#\} \cup \{0,1\}^2$ the alphabet of $X_{G(1,\infty)}$. We define $Z_1 = Z'_1 \cap (X_{G(1,\infty)} \times L_A^{\mathbb{Z}^2})$, where Z'_1 is the SFT with the following forbidden patterns:

- every symbol $(t, \ell) \in G \times L_A$ where exactly one of $t \in \{(1,0),(1,1)\}$ and $\ell = \square$ holds, and
- every 1×2 pattern $\begin{smallmatrix} (t_1,\ell_1) \\ (t_2,\ell_2) \end{smallmatrix} \in (G \times L_A)^{1\times 2}$ where ℓ_1 and ℓ_2 differ from $\square$ and each other.

It is easy to see that the cells of every vertical column of 1's in $X_{G(1,\infty)}$ are given the same label, and no other cells are labeled. As all configurations of $X_{G(1,\infty)}$ contain only finitely many columns, the subshift Z_1 is countable, and thus a countably covered sofic shift by Lemma 3.14.

We move on to the computation layer Z_2, where we once again simulate a counter machine. Let $M = (k, k', L_A, Q, \delta, q_0, q_f)$ be a CMS whose functionality we define later; for now, we only require that its input alphabet is L_A, the label alphabet of the input layer Z_1. Let $X_M \subset A_M^{\mathbb{Z}^2}$ be the SFT given by Construction 3.12 that simulates M, and define Z_2 from it by the following modification. There is a new symbol H in the alphabet of Z_2 that is treated as part of the computation cone. When the simulated machine M reaches the final state q_f, the next horizontal row of the computation cone is filled with the H-symbols, and we guarantee by 2×2 forbidden patterns that all the subsequent rows are filled by H as well. The row containing the final state q_f is called the *output row* of the computation cone.

The subshift $Z' \subset Z_1 \times Z_2$ is defined by finitely many additional forbidden patterns. First, the base of the computation cone in Z_2 can only be paired with the bottom left corner of the grid in Z_1, that is, the upper right corner of a pattern $\begin{smallmatrix} a & b \\ & c \end{smallmatrix}$ with $a = c = \# \neq b$, and conversely, the bottom left corner of the grid can only be paired with the base of the computation cone. Second, the input word $w \in L_A^*$ of the simulated machine M is given by the labels of the input grid, which are vertically constant. In particular, if the input word is infinite, then so are the squares of the input grid, and then the input word is of the form $\ell\square^\infty$ for some $\ell \in A \times \{0,1\}$. This means that there are only countably many possible input words, and if the CMS M is deterministic, then Z' is also countable, thus a countably covered sofic shift by Lemma 3.14.

We now specify the CMS M, and for that, let $g : \mathbb{N} \to \mathbb{N}$ be any computable function with $m \geq \binom{g(m)+2}{g(m)}$ for all $m \geq 3$, $g(m) \to \infty$ and $g(m)/m \to 0$, for example $m \mapsto \lfloor \sqrt{m} \rfloor - 1$. The machine M has four output counters, and on an input word $w \in L_A^*$, it behaves as follows. First, it checks that $w = u_0\square^{n-1}u_1\square^{n-1} \cdots \square^{n-1}u_{n-1}$ for some $n \in \mathbb{N}$ and $u =\in (A \times \{0,1\})^n$, rejecting if not. It then checks that $n = m^2 + 2r_{m^2}$ for some $m \in \mathbb{N}$, again rejecting if not, where $r_{m^2} = m^{2|A|^{2m^2+1}}$ is the radius of the m^2-marker map $f_{m^2} : A^{\mathbb{Z}} \to \{0,1\}^{\mathbb{Z}}$ from Lemma 6.10. Decompose the word $u = (u^{(1)}, u^{(2)})$ into $u^{(1)} \in A^n$ and $u^{(2)} \in \{0,1\}^n$. Using the radius r_{m^2} for f_{m^2}, the machine M next computes the image $v = F_{m^2}(u^{(1)}) \in \{0,1\}^{m^2}$ of $u^{(1)}$ under the local function $F_{m^2} : A^* \to \{0,1\}^*$. By the properties of f_{m^2}, the resulting word v contains at most one 1-symbol.

Next, M checks whether $v_i = 1$ for some (then necessarily unique) index $i \in [0, m^2-1]$. If this is the case, then denoting $c = u^{(2)}_{r_{m^2}+i} \in \{0,1\}$, the machine M outputs the four numbers $mg(m)+1$, $g(m)$, $ag(m)+c$ and $bg(m)$, where $a, b \in [0, m-1]$ are such that $i = a + mb$. Otherwise, M outputs $mg(m)+1$, $g(m)$, 0 and 0.

We then define the output layer Z_3, which is similar in structure to the input layer Z_1. It is defined by superimposing new symbols on the grid shift $X_{G(2,2)}$ as follows:

1. Each vertical column of 1's gets a label from $\{\square, 0, 1\}$ and each horizontal row one from $\{\square, \$\}$. A row or column whose label is not $\square$ is called *special*.

2. Every intersection of a row and a column (that is, every symbol $(1,1)$) gets a label from $\{0,1\}$. These labels are called the *elements* of the grid that contains them. The intersection of a special row and a special column has the same label as the column, and such elements are called *marked*.

3. Every coordinate in a grid (every symbol except #) gets a label from $\{ , , \}$, with the obvious 2×2 forbidden patterns. The leftmost coordinate of a special horizontal row must have label , and conversely, a label on the left edge of the grid must lie on a special horizontal row.

Because of the diagonal signal of the third item, if a configuration of Z_3 contains a finite grid, then at most one of its rows can be special. Since every configuration of $X_{G(2,2)}$ contains only finitely many rows and columns of 1's, Z_3 is countable, and thus a countably covered sofic shift by Lemma 3.14.

Finally, we define the subshift $Z \subset Z' \times Z_3$. The idea is to place the finite grid of Z_3 on the southwest corner of the H-region of Z_2, and using the outputs of M to control its size and labels. For this, every symbol (a, b) where the Z_2-layer of a is not H (part of an output row of M) and b is not # (the outside of the grid in Z_3), is forbidden in Z, so the grid of Z_3 lies on the H-region of Z_2.

Next, recall that the four output values of the simulated CMS M are of the form $mg(m)+1$, $g(m)$, $ag(m)+c$ and $bg(m)$, where $m \in \mathbb{N}$, $a, b \in [0, m-1]$ and $c \in \{0, 1\}$. Recall also that the counters of the machine M simulated in Z_2 are represented as words $\mathrm{P}_i \cdots \mathrm{P}_i \mathrm{Z}_i \cdots \mathrm{Z}_i$. For a configuration $z \in Z$ and $(i, j) \in \mathbb{Z}^2$, we require that $z_{(i,j)}$ is on the bottom row of the Z_3-grid if and only if $z_{(i,j-1)}$ contains a P_1 on the sublayer of the first counter of M. This means that the output $mg(m) + 1$ of the first counter is exactly the width of the Z_3-grid, if either (and thus both) exists. Similarly, we force the second output value $g(m)$ to be exactly the width of a grid square, by stating that if $z_{(i,j)}$ contains the bottom-most 1 of a column and $z_{(i-1,j)}$ is also in the grid, then $z_{(i-1,j-1)} z_{(i,j-1)}$ contains $\mathrm{P}_2\mathrm{Z}_2$ or $\mathrm{Z}_2\mathrm{Z}_2$ on the sublayer of the second counter. A column whose bottom coordinate is at (i, j) is special if and only if $z_{(i-1,j-1)} z_{(i,j-1)} z_{(i+1,j-1)}$ contains either $\mathrm{P}_3\mathrm{Z}_3\mathrm{Z}_3$ or $\mathrm{P}_3\mathrm{P}_3\mathrm{Z}_3$ on the sublayer of the third counter, and the label of the special column is 0 in the former case and 1 in the latter. With the above inputs, this means that the a'th column from the left is special, with label c. Finally, the position of the diagonal tile $\boxbslash$ on the bottom row of the grid must coincide with the fourth output counter, so the b'th row from the bottom is special.

See Figure 6.3 for a visualization of a configuration of Z. Since the position of the output grid is determined by the computation layer, Z is countable, and thus a countably covered sofic shift by Lemma 3.14 and the fact that countably covered sofic shifts are closed under direct product. As we explained earlier in the proof, the finite-width Z_1-grid (which contains $m^2 + 2r_{m^2}$ columns for some $m \in \mathbb{N}$) and the finite Z_3-grid (which contains m rows and columns with distance $g(m)$) encode two square patterns $S, R \in \{0, 1\}^{[0,m-1]^2}$ in every configuration that contains both of them. The pattern S is encoded in the binary word $v_{[r_{m^2}, r_{m^2}+m^2-1]}$ of length m^2, where the long word $v \in \{0, 1\}^{m^2+2r_{m^2}}$ consists of the binary labels of the Z_1-columns, while the pattern R is encoded in the elements of the Z_3-grid. In the configuration, we may also specify one coordinate $\vec{n} \in [0, m-1]^2$ by choosing the A-labels of the Z_1-columns so that the marker map computed by M places a 1 at that exact coordinate. The output counters of M force the corresponding element of the Z_3-grid to be marked, and then S and R must agree on that coordinate.

Finally, we define the block map $\phi : Z \to A^{\mathbb{Z}^2}$ with neighborhood $\{0\} \times \{0,1,2,3\} \subset \mathbb{Z}^2$ and local function Φ as follows. Without loss of generality, we assume $0, 1 \in A$ by renaming the elements of A if necessary. Let c_1, c_2, c_3, c_4 be symbols of the alphabet of Z. First, if the Z_1-layer of the symbol c_4 is $\#$, then

$$\Phi\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} = \begin{cases} b, & \text{if the } Z_1\text{-layer of } c_3 \text{ has label } (a,b) \in A \times \{0,1\}, \\ a, & \text{if the } Z_1\text{-layer of } c_3 \text{ is } \# \text{ and that of } c_2 \text{ has label } (a,b), \\ 1, & \text{if the } Z_1\text{-layer of } c_2 \text{ is } \# \text{ and that of } c_1 \text{ is not}, \\ 0, & \text{otherwise.} \end{cases}$$

Second, if c_4 is an element of the Z_3-grid, then the image of Φ at that coordinate is its label (in $\{0,1\}$). Finally, if the Z_2-label of c_4 is not H and the Z_3-label of c_3 is not $\#$, then the Φ-image is 1. In all cases not mentioned here, the Φ-image is 0.

Now, we describe the sofic shift $X = \phi(Z)$. Choose such values for the number $m \geq 3$, the words $u \in A^{m^2+2r_{m^2}}$ and $v \in \{0,1\}^{m^2+2r_{m^2}}$, and the pattern $R \in \{0,1\}^{[0,m-1]^2}$, that if the image $F_{m^2}(u) \in \{0,1\}^{m^2}$ of the marker map contains a 1 at some (necessarily unique) coordinate $i = a + bm$, then $v_{r_{m^2}+i} = R_{(a,b)}$. Then there exists a configuration $x = x^{m,u,v,R} \in X$ as follows. For every $(i,j) \in [0,n-1]^2$, we have $x_{(g(m)i,g(m)j)} = R_{(i,j)}$ (given by the elements of the Z_3-grid, whose southwest corner lies at the origin). We also have $x_{(i,-1)} = 1$ for all $i \in [0, mg(m)]$. For some $n < 0$, the configuration x has a horizontal row of 1's of length $m(m^2+2r_{m^2})+1$ to the right of $(0,n) \in \mathbb{Z}^2$, above which are the two rows $u_0 0^{m-1} u_1 0^{m-1} \cdots 0^{m-1} u_{m^2+2r_{m^2}-1}$ and $v_0 0^{m-1} v_1 0^{m-1} \cdots 0^{m-1} v_{m^2+2r_{m^2}-1}$ of the same length (given by the bottom rows of the Z_1-grid). For all other coordinates $\vec{v} \in \mathbb{Z}^2$, we have $x_{\vec{v}} = 0$. Configurations of X which are not translates of some $x^{m,u,v,R}$ do not contain such finite rows of 1's.

Finally, we show that the universal extension $A(X,Y) \in \hat{A}^{\mathbb{Z}^2}$ is not sofic. Let the configuration $x = x^{m,u,v,R} \in X$ and $n < 0$ be as above, and denote by $S \in \{0,1\}^{[0,m-1]^2}$ the square pattern encoded by the binary word $v_{[r_{m^2}, r_{m^2}+m^2-1]} \in \{0,1\}^{m^2}$. Construct a new configuration $\hat{x} \in \hat{A}^{\mathbb{Z}^2}$ by replacing those symbols in x that encode u with $\diamond$-symbols.

Claim 6.13. With the above definitions, we have $\hat{x} \in A(X,Y)$ if and only if $S = R$.

Proof of Claim. Suppose first that $S = R$, and let $y \in Y$ be any configuration. If the F_{m^2}-image of the word $y_{(0,1-n)} y_{(m,1-n)} \cdots y_{(m(m^2+2r_{m^2}-1),1-n)} \in A^{m^2+2r_{m^2}}$ substituted to the $\diamond$-symbols of $\hat{x}$ contains a 1, then the patterns

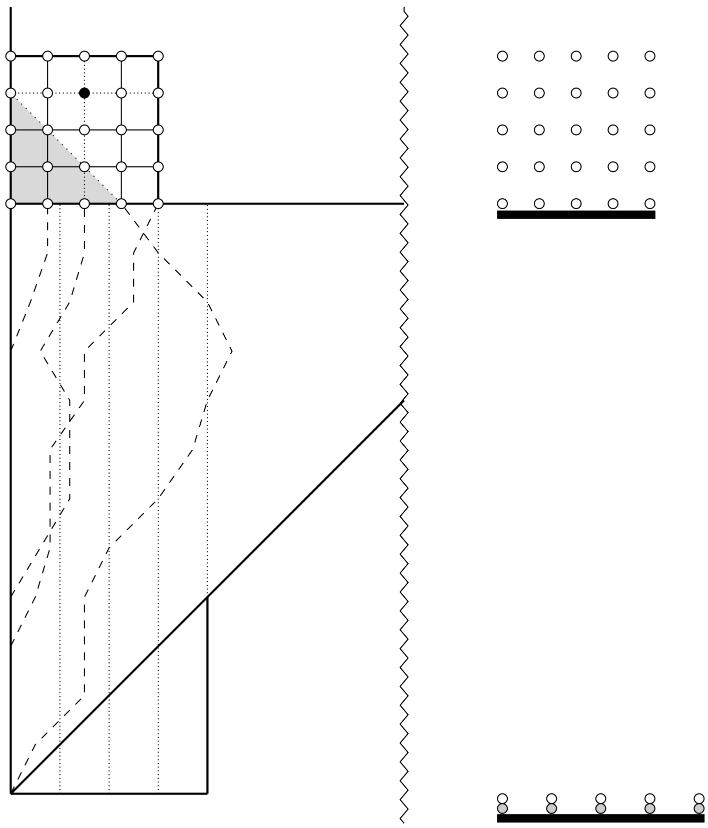

Figure 6.3: Left: a schematic diagram of a configuration of Z with a finite Z_3-grid, not drawn to scale. The dotted lines represent the columns of the Z_1-grid, whose labels are read by the CMS. The dashed lines represent the output counters. The computation ends at the base of the Z_3-grid, whose elements are denoted by small circles, of which the filled one is marked. Right: its Φ-image. The white and gray circles represent symbols picked from $\{0, 1\}$ and A, respectively. The horizontal bars represent rows of symbols 1.

S and R agree on the respective coordinate since they are equal, and we have $\hat{x}^{(y)} \in X$. If the image does not contain a 1, we automatically have $\hat{x}^{(y)} \in X$.

Conversely, suppose we have $S \neq R$, so there exists $i = a + mb \in [0, m^2 - 1]$ such that $v_{r_{m^2}+i} \neq R_{(a,b)}$. Denote $k = m^2 + 2r_{m^2}$ and $D = \{(0, 1-n), (m, 1-n), \ldots, (m(k-1), 1-n)\} \subset \mathbb{Z}^2$. Since Y is not horizontally periodic, there exists a pattern $P \in \mathcal{B}_D(Y)$ that, when interpreted as a word $u \in A^k$, satisfies $F_{m^2}(u)_i = 1$ for the marker map F_{m^2}. We now have $\hat{x}^{(y)} \notin X$, as otherwise the machine M would mark the column a and row b in the ϕ-preimage of this configuration, and we would have $S_{(a,b)} = R_{(a,b)}$, a contradiction. Thus $\hat{x} \notin A(X, Y)$, and we have shown that $\hat{x} \in A(X, Y)$ if and only if $S = R$. $\square$

Suppose now for a contradiction that $A(X, Y)$ is sofic, and let $C > 0$ be given for it by Lemma 3.15. Let $R \in \{0,1\}^{[0,m-1]^2}$ be a square pattern, and let $\hat{x}^{m,R} = \hat{x}^{m,u,v,R}$, where $u \in A^{m^2+2r_{m^2}}$ is arbitrary, and $v \in \{0,1\}^{m^2+2r_{m^2}}$ encodes the pattern R. There are 2^{m^2} such configurations for a given $m \in \mathbb{N}$, and $C^{mg(m)} < 2^{m^2}$ holds when m is large enough, by the definition of the function g. Then there are two patterns $R \neq R' \in \{0,1\}^{[0,m-1]^2}$ such that, with the notation of Lemma 3.15, we have $c(\hat{x}^{m,R}, \hat{x}^{m,R'}, mg(m) + 1) \in A(X, Y)$. But this configuration is exactly $\hat{x}^{m,u,v',R}$ where $v' \in \{0,1\}^{m^2+r_{m^2}}$ encodes the pattern R', contradicting Claim 6.13. $\square$

6.4 Quantifier Extensions of Deterministic Subshifts

In this section, we briefly discuss a possible strengthening of Theorem 6.9. Namely, we note that the subshift X constructed in the proof of Theorem 6.9 is not downward deterministic, which leads us to studying the extensions of downward deterministic subshifts. The next result may seem surprising, but the proof is elementary.

Proposition 6.14. *Let $X \subset A^{\mathbb{Z}^2}$ be a downward deterministic subshift, and let $Y \subset A^{\mathbb{Z}^2}$ be any subshift. If $A(X, Y) \neq X$, then Y is downward deterministic, and if Y contains at least two configurations, then $A(X, Y)$ is downward deterministic.*

In other words, when extending a deterministic subshift X by another subshift Y, if one of Y or $A(X, Y)$ is nontrivial, then the other must be deterministic.

Proof. Let $n \in \mathbb{N}$ be such that the rectangle $x_{[-n,n]\times[1,n]}$ determines the cell $x_{\vec{0}}$ for all configurations $x \in X$, and denote $D = [-n, n] \times [1, n] \cup \{\vec{0}\}$.

Suppose first that $A(X,Y) \neq X$, and let $P \in \mathcal{B}_D(A(X,Y))$ be such a pattern that $P_{\vec{0}} = \diamond$. Let $Q, R \in \mathcal{B}_D(Y)$ be two patterns such that $Q_{[-n,n]\times[1,n]} = R_{[-n,n]\times[1,n]}$, and consider the substitutions $P^{(Q)}, P^{(R)} \in \mathcal{B}_D(X)$. We have $P^{(Q)}_{[-n,n]\times[1,n]} = P^{(R)}_{[-n,n]\times[1,n]}$, which implies $P^{(Q)}_{\vec{0}} = P^{(R)}_{\vec{0}}$ by the determinism of X, and thus $Q_{\vec{0}} = R_{\vec{0}}$. But this means that Y is downward deterministic.

Suppose then that Y is a nontrivial subshift, and let $P, Q \in \mathcal{B}_D(A(X,Y))$ be such patterns that $P_{[-n,n]\times[1,n]} = Q_{[-n,n]\times[1,n]}$. Now, if $P_{\vec{0}} \in A$, let $R \in \mathcal{B}_D(Y)$ be such that $R_{\vec{0}} \neq P_{\vec{0}}$. Such a pattern R exists, since at least two distinct letters occur in Y. Since $P^{(R)}, Q^{(R)} \in \mathcal{B}(X)$ agree on the domain $[-n,n] \times [1,n]$, we have $Q_{\vec{0}} = P_{\vec{0}}$ by the determinism of X. Symmetrically, if we have $Q_{\vec{0}} \in A$, then $P_{\vec{0}} = Q_{\vec{0}}$, and thus $P_{\vec{0}} = \diamond$ if and only if $Q_{\vec{0}} = \diamond$. This means that the extension $A(X,Y)$ is downward deterministic. $\square$

The subshifts constructed in Corollary 6.7 are vertically constant, thus downward deterministic, so even for downward deterministic sofic shifts $X, Y \subset S^{\mathbb{Z}^2}$, the universal extension $A(X,Y)$ need not be sofic. However, the construction relies on Y being computationally difficult, and not much can be said if Y is computable. Namely, by Proposition 6.14 and Lemma 6.5, if there exist downward deterministic sofic shifts X and Y, with Y recursive, such that $A(X,Y)$ is not sofic, then $A(X,Y)$ is a downward deterministic Π^0_1 subshift which is not sofic, and it is currently unknown whether such an object exists. In particular, Lemma 3.15 cannot be applied, since all downward deterministic subshifts also satisfy its conclusion. Recall Proposition 3.17, which states that every multidimensional sofic shift X has a collection of sofic subsystems whose entropies are dense in the interval $[0, h(X)]$. But since downward deterministic subshifts have zero entropy, they always satisfy this condition, too.

Finally, we remark that if the existential extension $E(X,Y)$ of two subshifts $X, Y \subset A^{\mathbb{Z}^2}$ is not equal to X, then it cannot be deterministic, since any subset of $\diamond$-symbols in a configuration of $E(X,Y)$ can always be replaced by some symbols of A.

Chapter 7

Multidimensional Subshifts Defined by Finite Automata

7.1 Introduction

In this section, we discuss multihead finite automata on infinite multidimensional configurations, which we call plane-walking automata, and use them to define classes of subshifts. Our model is based on the general idea of a graph-walking automaton. In this model, the automaton is placed on one of the nodes of a graph with colored nodes, and it repeatedly reads the color of the current node, updates its internal state, and steps to an adjacent node. The automaton eventually enters an accepting or rejecting state, or runs forever without making a decision. Usually, we collect the graphs that it accepts, or the ones that it does not reject, and call this collection the language of the automaton. We restrict our attention to machines that are deterministic, although an interesting continuation of our research would be to consider nondeterministic or alternating machines.

Well-known such models include the two-way deterministic finite automata (2DFA) walking back-and-forth on a finite word, and tree-walking automata traversing a tree. See [HKM09] for a survey on multihead automata on words, and the references in [Boj08] for information on tree-walking automata. In multiple dimensions, our automata are based on the concept of picture-walking (or 4-way) automata for accepting picture languages, defined in [BH67] and surveyed in [IT91, KS11]. In the article [DM02], a study on pebble automata was conducted that is very similar to the one we present here, and we rely on some of their results in the simplest case where our automata have only one head. However, their approach is different from ours in several other respects too; for example, their automata accept topologically open sets of configurations, not subshifts. Also, in the recent preprint [ABS14], the authors define a class of G-subshifts[1] for a finitely generated group G by allowing a Turing machine to walk on the Cayley graph of G. Their approach is very similar to ours in the case $G = \mathbb{Z}^d$, with the exception that we use multiple heads that cannot modify the contents of the input configuraion.

The first question about subshifts accepted by plane-walking automata is how this class relates to existing classes of subshifts. We compare the class of subshifts accepted by a one-head deterministic automaton to SFTs and sofic shifts, which correspond to local and regular languages. It is well-known that in the one-dimensional finite case, graph-walking automata with a single head (2DFA) define precisely the regular languages. However, for more complicated graphs, deterministic graph-walking automata often define a smaller class than the one obtained by letter-to-letter projections from local languages (which is often considered the natural generalization of regularity): deterministic tree-walking automata do not define all regular tree languages [BC08] and deterministic picture-walking automata do not accept all recognizable picture languages [GVR97]. We show in Theorem 7.11 that this is also the case for a one-head deterministic plane-walking automaton in the multidimensional case: the class of subshifts defined is strictly between SFTs and sofic shifts.

Already in [BH67], the basic model of picture-walking automata was augmented by pebbles, and we similarly consider classes of subshifts defined by multihead plane-walking automata. In [BH67, Theorem 3], it was shown that the hierarchy obtained as the number of pebbles grows is infinite in the case of pictures (by a diagonalization argument). Similar results are known for one-dimensional words [HY75] and trees [BSSS06]. In [DM02], it was shown that for a specific model of accepting infinite configurations by automata with finitely many pebbles, the hierarchy collapses to the third level, which is characterized by a computability condition. We show that the same result holds for multihead automata: three heads are enough to recognize any subshift definable in our model, and this class coincides with Π^0_1 subshifts. In particular, it properly contains the class of sofic shifts. However, we are not able to separate the second and third levels in the case of one or two dimensions, although we find it very likely that they are distinct, as is the case in three or more dimensions. We discuss why this problem appears hard to us, suggest a possible separating language, and state a related open problem for two-counter machines.

This chapter is based on the conference article [ST14].

[1]For a group G, a *G-subshift over an alphabet A* is a subset of A^G which is closed in the product topology, and invariant under the translation action of G.

7.2 Choosing the Machines

The basic idea in this article is to define subshifts by deterministic and multihead finite automata as follows: Given a configuration $x \in S^{\mathbb{Z}^d}$, we initialize the heads of the automaton on some of its cells, and let them run indefinitely, moving around and reading the contents of x. If the automaton halts in a rejecting state, then we consider x to be rejected, and otherwise it is accepted.

After this high-level idea has been established, there are multiple a priori inequivalent ways of formalizing it, and we begin with a discussion of such choices. Much of this freedom is due to the fact that many different definitions and variants of multihead finite automata exist in the literature, both in the case of finite or infinite pictures and one-dimensional words (see [HKM09] and references therein).

Heads or pebbles? A multihead automaton can be defined as having multiple heads capable of moving around the input, or as having one mobile head and several immobile pebbles that the head can move around. In the latter case, one must also decide whether the markers are indistinguishable or distinct, and whether they can store information or not. In this article, we choose the former approach of having multiple mobile heads. The opposite choice was made in [DM02], where the authors consider finite automata with one mobile head and several stationary pebbles, which are distinguishable from each other but cannot store any other information.

Global control or independent heads? Next, we must choose how the heads of our machines interact. The traditional approach is to have a single global state that controls each head, but in our model, this could be considered 'physically infeasible,' as the heads may travel arbitrarily far from each other. For this reason, and in order not to have too strong a model, the heads of our automata are independent, and can interact only when they lie in the same cell. Note that in the case of only one mobile head, this choice is moot.

Synchronous or asynchronous motion? Now that the heads have no common memory, we must decide whether they still have a common perception of time and can synchronize their motion. In the synchronous updating scheme, all heads update their states and positions simultaneously, so that the distance between two heads moving in the same direction in steps of the same length stays constant. The other option is asynchronous updating, where the heads may update at different paces, possibly nondeterministically. We choose the synchronous scheme, as it is easier to formalize and allows us to shoot carefully synchronized signals, which we feel are the most interesting aspect of multihead plane-walking automata.

Next, we need to decide how exactly a plane-walking automaton defines a subshift. Recall that a subshift is defined by a possibly infinite set of finite forbidden patterns in a translation-invariant way. In our model, the forbidden patterns should be exactly those that support a rejecting run of the automaton. In some sense, this choice is dual to that made in [DM02], where an infinite configuration is accepted if and only if the automaton eventually enters an accepting state on it.

How do we start? There are at least three different methods of initializing the automaton. First, we could always initialize our automata at the origin $\vec{0} \in \mathbb{Z}^d$, decide the acceptance of a configuration based on this single run, and restrict to automata that define translation-invariant sets. Second, we may quantify over all coordinates of $\mathbb{Z}^d$, initialize all the heads at the same coordinate, and reject if some choice leads to rejection. In the third option, we quantify over all k-tuples of coordinates, and place the k heads in them independently. The first definition is not very satisfying, since most one-head automata would have to be discarded, and of the remaining two, we choose the former, as it is more restrictive. Note, however, that the first definition of always initializing the automaton at the origin was chosen in [DM02], without restricting to translation-invariant sets. We also quantify over a set of initial states, so that our subshift classes are closed under finite intersection, and accordingly seem more natural.

How do we end? Finally, we have a choice of what constitutes as a rejecting state. Can a single head cause the whole computation to reject, or does every head have to reject at the same time, and if that is the case, are they further required to be at the same position? We again choose the most restrictive option.

All of the above models are similar, in that by adding a few more heads or counters, one can usually simulate an alternative definition. Sometimes, one can even show that two models are equivalent. For example, [BH67, Theorem 2.3] states that being able to distinguish markers is not useful in the case of finite pictures; however, the argument seems impossible to apply to plane-walking automata.

To recap, our definition of choice is the *deterministic k-head plane-walking finite automaton with local information sharing, synchronous updating, quantification over single initial coordinate and initial state, and rejection with all heads at a single coordinate*, with the (necessarily ambiguous) shorthand kPWDFA.

7.3 Definitions

We now formally define our machines, runs, acceptance conditions and the subshifts they define. For this section, let the dimension $d \geq 1$ be fixed.

Definition 7.1. Let $k \geq 1$. A *k-headed deterministic plane-walking automaton*, kPWDFA for short, is a 5-tuple $M = (Q, A, \delta, I, R)$, where $Q =$

$Q_1 \times \cdots \times Q_k$ is the finite set of *global states*, the Q_i are the *local states*, A is the finite *alphabet*, and $\delta = (\delta_1, \ldots, \delta_k)$ is the list of *transition functions*

$$\delta_j : \tilde{Q}_j \times A \to Q_j \times \mathbb{Z}^d,$$

where $\tilde{Q}_j = Q'_1 \times \cdots \times Q'_{j-1} \times Q_j \times Q'_{j+1} \times \cdots \times Q'_k$, and $Q'_i = Q_i \cup \{?\}$ for a new symbol $? \notin Q_i$. We call $I \subset Q$ the set of *initial states*, and $R \subset Q$ the set of *rejecting states*.

Note that all functions above are total, so a plane-walking automaton always has a next state to jump to. The intuition behind this definition is that δ_j is the transition function of head j, and the state of another head is given as the symbol ? if it does not lie in the same coordinate as head j.

Definition 7.2. Let $M = (Q, A, \delta, I, R)$ be a kPWDFA. An *instantaneous description* or *ID* of M is an element of $\mathrm{ID}_M = (\mathbb{Z}^d)^k \times Q$. Given a configuration $x \in A^{\mathbb{Z}^d}$, we define the *update function* $M_x : \mathrm{ID}_M \to \mathrm{ID}_M$. Namely, given $c = (\vec{v}^1, \ldots, \vec{v}^k, q_1, \ldots, q_k) \in \mathrm{ID}_M$, we define $M_x(c)$ as follows. If $(q_1, \ldots, q_k) \in R$ and $\vec{v}^1 = \cdots = \vec{v}^k$, then we say c is *rejecting*, and $M_x(c) = c$. Otherwise, $M_x(c) = (\vec{w}^1, \ldots, \vec{w}^k, p_1, \ldots, p_k)$, where $\vec{w}^j = \vec{v}^j + \vec{u}^j$ and

$$\delta_j(q'_1, \ldots, q'_{j-1}, q_j, q'_{j+1}, \ldots, q'_k, x_{\vec{v}^j}) = (p_j, \vec{u}^j),$$

where we write $q'_i = q_i$ if $\vec{v}^i = \vec{v}^j$, and $q'_i = ?$ otherwise. The *run of M on $x \in A^{\mathbb{Z}^d}$ from $c \in \mathrm{ID}_M$* is the infinite sequence $M_x^\infty(c) = (M_x^n(c))_{n \in \mathbb{N}}$. We say the run is *accepting* if no $M_x^n(c)$ is rejecting. We define the *subshift of M* by

$$\mathrm{Acc}(M) = \{x \in A^{\mathbb{Z}^d} \mid \forall q \in I, \vec{v} \in \mathbb{Z}^d : M_x^\infty(\vec{v}, \ldots, \vec{v}, q) \text{ is accepting.}\}$$

Intuitively, the set $\mathrm{Acc}(M)$ consists of those configurations where one can choose any any initial coordinate of $\mathbb{Z}^d$, place all the heads of M on that coordinate in any initial state, and be sure that the automaton runs forever. It could also be defined by forbidding those finite patterns that support a rejecting run of M starting from an initial state, which perhaps makes it clearer that $\mathrm{Acc}(M)$ is actually a subshift. We now define our hierarchy of interest:

Definition 7.3. For $k > 0$, define

$$\mathcal{PW}_k^d = \{\mathrm{Acc}(M) \mid M \text{ is a } d\text{-dimensional } k\text{PWDFA.}\}$$

It is easy to see that $\mathcal{PW}_k^d \subset \mathcal{PW}_{k+1}^d$ holds for all $k > 0$, and that every $\mathcal{PW}_k^d$ only contains Π_1^0 subshifts. Since a deterministic finite state automaton can clearly check any local property, every d-dimensional SFT is in the class $\mathcal{PW}_1^d$. To make it easier to express these kinds of relations between the $\mathcal{PW}_k^d$ and other classes of multidimensional subshifts, we introduce the following notation for the rest of this chapter.

Definition 7.4. For $d \geq 1$, we denote by SFT^d the class of d-dimensional SFTs, and by sofic^d the class of d-dimensional sofic shifts.

With this notation, the above remark states that $\mathrm{SFT}^d \subset \mathcal{PW}_1^d$.

Remark 7.5. We note some robustness properties of our definition of a kPWDFA and the classes $\mathcal{PW}_k^d$. While the definition only allows information sharing when several heads lie in the same cell, we may assume that heads can communicate if they are at most t cells away from each other. Namely, if we had a stronger k-head automaton where such behavior is allowed, then we could simulate its computation step by $\Theta(kt^d)$ steps of a kPWDFA whose heads visit, one by one, the $\Theta(t^d)$ cells at most t steps away from them, and remember which other heads they saw in which states. Also, while we allow the machines to move by any finite vector, we may assume these vectors all have length 0 or 1 by simulating a step of length r by r steps of length 1.

Proposition 7.6. *The classes $\mathcal{PW}_k^d$ are closed under conjugacy, finite intersection, and the action of any transformation in $SL_d(\mathbb{Z})$.*

Proof. For the first claim, let $X \in \mathcal{PW}_k^d$ be a d-dimensional subshift over an alphabet A, and $\phi : Y \to X$ a conjugacy between X and another subshift $Y \subset B^{\mathbb{Z}^d}$ defined by the local rule $\Phi : B^N \to A$ on the finite neighborhood $N \subset \mathbb{Z}^d$. Since $X \in \mathcal{PW}_k^d$, there is a kPWDFA $M = (Q, A, \delta, I, R)$, where $Q = Q_1 \times \cdots \times Q_k$, such that $X = \mathrm{Acc}(M)$. We define a new kPWDFA $\hat{M} = (\hat{Q}, B, \hat{\delta}, \hat{I}, R)$ as follows. The local states of new the state set, that is, the components of $\hat{Q} = \prod_{i=1}^k \left(Q_i \times B^{\leq |N|}\right)$, consist of the local states of M, together with all words of length at most $|N|$ over the alphabet B. The new transition function $\hat{\delta}$ works in the following way. If the local state of a head is (q, ϵ) at a coordinate $\vec{n} \in \mathbb{Z}^d$ of a configuration $y \in B^{\mathbb{Z}^d}$, then it enters the *neighborhood loop*, in which it visits the coordinates $N + \vec{n}$ in a fixed order, and records the contents of the pattern $y|_{N+\vec{n}}$ one by one into the second component of its state, keeping the first component constant. When the loop is complete, the head returns to the coordinate $\vec{n}$, and performs the transition determined by δ in the case that the symbol under the head is $\Phi(y|_{N+\vec{n}})$. The initial states of $\hat{M}$ are exactly $\hat{I} = \{((q_1, \epsilon), \ldots, (q_k, \epsilon)) \mid (q_1, \ldots, q_k) \in I\}$, and then it is easy to see that $\hat{M}$ simulates the computation of M on the configuration $\phi(y)$, with each step of M corresponding to exactly $|N|$ steps of $\hat{M}$.

Next, to prove the case of finite intersections, it suffices to prove that the intersection of two subshifts $X, Y \in \mathcal{PW}_k^d$ is also in $\mathcal{PW}_k^d$. For this, it suffices to take the kPWDFA whose state set, transition function, initial state set and rejecting state set are all disjoint unions of those of the kPWDFA M and M' that accept the subshifts X and Y. Namely, when started on the initial states of M, the automaton accepts exactly the configurations of X, and when started on an initial state of M', it accepts exactly those of Y, and thus the subshift is accepts is exactly $X \cap Y$.

Finally, let $M = (Q, A, \delta, I, R)$ be a kPWDFA, and let $L \in SL_d(\mathbb{Z})$ be a linear transformation. Consider the kPWDFA $\bar{M} = (Q, S, \bar{\delta}, I, R)$, where the new transition function $\bar{\delta} = (\bar{\delta}_1, \ldots, \bar{\delta}_k)$ is defined by $\bar{\delta}_i(q_1, \ldots, q_k, a) = (p, L(\vec{v}))$, where $(p, \vec{v}) = \delta_i(q_1, \ldots, q_k, a)$, for all $q_j \in \tilde{Q}_j$ and $a \in A$. If the automaton $\bar{M}$ is initialized at some coordinate $\vec{n}$ of a configuration $x \in A^{\mathbb{Z}^d}$, then it follows from the definition of $L^{-1}(x)$ and the linearity of L that it essentially simulates the computation of M on the configuration $L^{-1}(x)$. Thus we have $\mathrm{Acc}(\bar{M}) = L(\mathrm{Acc}(M))$. □

In particular, the two-dimensional classes $\mathcal{PW}_k^2$ are closed under horizontal and vertical mirroring, and rotation by $\frac{\pi}{2}$.

As stated, the main goal of this chapter is to compare the classes $\mathcal{PW}_k^d$ to existing classes of subshifts and each other, and for this, we recall and define a few subshifts and classes thereof. In most of our examples, we use the binary alphabet $A = \{0, 1\}$, the configurations contain the symbol 0 in all but a bounded number of coordinates.

Definition 7.7. Recall the *two-dimensional sunny side up subshift* from Example 2.2, which is the subshift of $\{0, 1\}^{\mathbb{Z}^2}$ where at most one symbol 1 can occur in a configuration. We generalize this subshift by the *d-dimensional m-sunny side up subshift*, which is the d-dimensional subshift $X_m^d \subset \{0, 1\}^{\mathbb{Z}^d}$ with forbidden patterns $\{P \mid |P|_1 > m\}$, that is, every finite pattern where the letter 1 occurs more than m times. A d-dimensional subshift is *m-sparse* if it is a subshift of X_m^d, and *sparse* if it is m-sparse for some $m \in \mathbb{N}$.

If $X \subset A^{\mathbb{Z}^d}$ is a d_1-dimensional subshift and $d_2 > d_1$, we define $X^{\mathbb{Z}^{d_2 - d_1}}$ as the d_2-dimensional subshift where the contents of every d_1-dimensional hyperplane $\vec{m} + \{\sum_{i=1}^{d_1} n_i \vec{e}_i \mid \vec{n} \in \mathbb{Z}^{d_1}\} \subset \mathbb{Z}^{d_2}$ for $\vec{m} \in \{0\}^{d_1} \times \mathbb{Z}^{d_2 - d_1}$ are independently taken from X.

Put simply, an n-sparse subshift is one where at most n symbols 1 may occur, and the sunny side up subshifts are the ones with no additional constraints. Recall also the definition of the two-dimensional mirror subshift $X_{\mathrm{mirror}} \subset \{0, 1, \#\}^{\mathbb{Z}^2}$ from Example 5.9 in Chapter 5. We generalize this example to multiple dimensions as follows.

Definition 7.8. Let $d \geq 2$. The *d-dimensional mirror subshift* $X_{\mathrm{mirror}}^d \subset \{0, 1, \#\}^{\mathbb{Z}^d}$ is defined by the following forbidden patterns.

- All patterns P of domain $\{\vec{0}, \vec{e}_i\}$ for $i \in \{2, \ldots, d\}$ with $|P|_\# = 1$.
- All patterns $\{\vec{0} \mapsto \#, n \cdot \vec{e}_i \mapsto \#\}$ for $i \in \{2, \ldots, d\}$ and $n \geq 1$.
- All patterns $\{-n \cdot \vec{e}_1 \mapsto a, \vec{0} \mapsto \#, n \cdot \vec{e}_1 \mapsto b\}$ for $n \geq 1$ and $a \neq b$.

Intuitively, the rules of X^d_{mirror} are that if two symbols # are adjacent on some $(d-1)$-dimensional hyperplane perpendicular to the first axis $\vec{e}_1$, then that hyperplane must be filled with the symbols #, and there is at most one such hyperplane, whose two sides are mirror images of each other. In two dimensions, the hyperplane is just a vertical line. None of the multidimensional mirror subshifts is sofic.

7.4 One Head

Now, let us begin analyzing the classes $\mathcal{PW}^d_k$, starting with the case of a single head. Of course, plane-walking automata with one head are trivially equivalent to pebble automata with no pebbles, which means that there is a definite overlap with [DM02] in the two-dimensional case. Thus we are able to directly use, or at least easily generalize, some results from the aforementioned article, including the following.

Lemma 7.9 (proved as Proposition 6 of [DM02])**.** *Let $M = (Q, A, \delta, I, R)$ be a $1PWDFA$ of dimension d, and let $x \in A^{\mathbb{Z}^d}$ be a uniform configuration. Let $(\vec{v}_n, q_n)_{n\in\mathbb{N}}$ be the run of M on x from the origin. Then the sequence of states $(q_n)_{n\in\mathbb{N}}$ and the sequence of displacements $(\vec{v}_{n+1} - \vec{v}_n)$ are eventually periodic, with the periods and transient lengths being bounded by $|Q|$.*

In [DM02], it was also proved that automata with only one pebble exhibit similar behavior, only with slightly higher bounds.

Our first results place the class $\mathcal{PW}^d_1$ between SFT^d and sofic^d.

Lemma 7.10. *In all dimensions $d \geq 1$, we have $(X^1_1)^{\mathbb{Z}^{d-1}} \in \mathcal{PW}^d_1 \setminus \mathrm{SFT}^d$. In particular, the class $\mathcal{PW}^d_1$ properly contains SFT^d.*

Proof. Note that $X = (X^1_1)^{\mathbb{Z}^{d-1}}$ is the d-dimensional subshift where no 'row' (a translate of the set $\mathbb{Z} \cdot \vec{e}_1$) may contain two symbols 1. First, we claim X is not an SFT. Suppose on the contrary that it is defined by a finite set of forbidden patterns with domain $[0, n-1]^d$ for some $n \in \mathbb{N}$. Consider the configurations $x^0, x^1 \in \{0,1\}^{\mathbb{Z}^d}$ where $x^i_{\vec{0}} = x^i_{n\cdot\vec{e}_1+i\cdot\vec{e}_2} = 1$ and $x^i_{\vec{v}} = 0$ for all other $\vec{v} \in \mathbb{Z}^d$. Since any pattern with domain $[0, n-1]^d$ occurs in x^0 if and only if it occurs in x^1, we have $x^0 \in X$ if and only if $x^1 \in X$, a contradiction since clearly $x^0 \notin X$ and $x^1 \in X$.

To show that $X \in \mathcal{PW}^d_1$, we construct a one-head automaton M_1 such that $X = \mathrm{Acc}(M_1)$. The idea is that the head will walk in the direction of the first axis, and increment a counter when it sees a symbol 1. If the counter reaches 2, the automaton rejects. More precisely, the automaton is $M_1 = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, \{q_0\}, \{q_2\})$, where $\delta(q_0, a) = (q_a, \vec{e}_1)$, $\delta(q_1, a) = (q_{1+a}, \vec{e}_1)$ and $\delta(q_2, a) = (q_2, \vec{0})$ for $a \in \{0,1\}$. If there are two symbols 1 on any of the rows of a configuration $x \in \{0,1\}^{\mathbb{Z}^d}$, say $x_{\vec{v}} = x_{\vec{w}} = 1$ where

$\vec{w} = \vec{v} + n\vec{e}_1$ for some $n \geq 1$, then the run of M_1 on x from the initial ID $(q_0, \vec{v})$ is not accepting, since the rejecting ID $(q_2, \vec{w} + \vec{e}_1)$ is entered after $n+1$ steps. Thus, $x \notin \mathrm{Acc}(M_1)$. On the other hand, it is easy to see the if no row of $x \in \{0,1\}^{\mathbb{Z}^2}$ contains two symbols 1, then $x \in \mathrm{Acc}(M_1)$. □

Theorem 7.11. *In all dimensions $d \geq 1$, we have $\mathcal{PW}_1^d \subset \mathrm{sofic}^d$, with equality if $d = 1$.*

Proof. We first show $\mathcal{PW}_1^d \subset \mathrm{sofic}^d$. The proof of this is quite standard, see for example [KM01]. Suppose that $X \in \mathcal{PW}_1^d$ is a subshift over the alphabet A, and let $M = (Q, A, \delta, I, R)$ be a 1PWDFA accepting X. We construct an SFT Y over the alphabet $2^Q \times A$, such that the second component of Y contains exactly X. The forbidden patterns of Y are

- every symbol $(Q', a) \in 2^Q \times A$ such that $I \not\subset Q'$ or $R \cap Q' \neq \emptyset$, and
- every two-element pattern $\{\vec{0} \mapsto (Q_1, a_1), \vec{v} \mapsto (Q_2, a_2)\}$ such that $\delta(q_1, a_1) = (q_2, \vec{v})$ for some $q_1 \in Q_1$ and $q_2 \notin Q_2$.

Now, if we initialize M on the second component of some $y \in Y$ in any initial state $q_0 \in I$, it is easy to see by induction that if it lies at $\vec{v}$ in state $q \in Q$ after some n steps, then the first component of $y_{\vec{v}}$ contains q, which implies that $q \notin R$. Thus, every run of M on the second component of y is accepting. Conversely, suppose that M accepts a configuration $x \in A^{\mathbb{Z}^d}$. For each $\vec{n} \in \mathbb{Z}^d$, denote

$$z_{\vec{n}} = \bigcup \{q \in Q \mid \vec{m} \in \mathbb{Z}^d, q_0 \in I, n \in \mathbb{N}, M_x^n(\vec{m}, q_0) = (q, \vec{n})\}.$$

These are exactly the states in which the automaton M may step on the coordinate $\vec{n}$ in one of its infinite runs. We have defined a configuration $z \in (2^Q)^{\mathbb{Z}^d}$, and it is easy to see that $(z, x) \in Y$. Thus $X = \mathrm{Acc}(M)$.

It is well-known that a one-dimensional subshift is sofic if and only if it can be defined by a regular language of forbidden words [LM95]. Since 2-way deterministic finite automata only recognize regular languages, we have $\mathrm{sofic}^1 \subset \mathcal{PW}_1^1$, and from the first part of the proof it follows that the classes coincide. □

Remark 7.12. For all dimensions $d_1 < d_2$, all $k \geq 1$, and all subshifts $X \in \mathcal{PW}_k^{d_1}$, we have $X^{\mathbb{Z}^{d_2 - d_1}} \in \mathcal{PW}_k^{d_2}$, since a d_2-dimensional kPWDFA can simply simulate a d_1-dimensional one on any d_1-dimensional hyperplane. In particular, if a subshift $X \subset S^{\mathbb{Z}}$ is sofic, then $X^{\mathbb{Z}^{d-1}} \in \mathcal{PW}_1^d$ for any dimension $d \geq 2$.

Of course, since multidimensional SFTs may contain very complicated configurations, the same is true for the classes $\mathcal{PW}_1^d$. In particular, for all $d \geq 2$ there are subshifts in $\mathcal{PW}_1^d$ whose languages are Π_1^0-complete.

However, just like in the case of SFTs, the sparse parts of subshifts in $\mathcal{PW}_1^d$ are simpler.

Theorem 7.13. *Let the dimension $d \geq 1$ be arbitrary, and let $X \in \mathcal{PW}_1^d$. For all $m \in \mathbb{N}$, the intersection $X \cap X_m^d$ of X with the m-sunny side up shift is recursive.*

Proof. Let $X = \mathrm{Acc}(M)$ for a 1PWDFA $M = (Q, \{0,1\}, \delta, I, R)$ that only takes steps of length 0 and 1 (this assumption is valid by Remark 7.5), and denote the intersection $X \cap X_m^d$ by $Y \subset \{0,1\}^{\mathbb{Z}^d}$. We assume without loss of generality that Y contains a configuration with at least two symbols 1.

Claim 7.14. It is decidable whether a given configuration $y \in X_m^d$ is in Y.

In other words, it is decidable whether there exists $\vec{v} \in \mathbb{Z}^d$ such that started from $\vec{v}$ in one of the initial states $q_0 \in I$, the automaton M eventually rejects y.

Proof of Claim. To decide this, note first that if M does not see any symbols 1, then it does not reject – otherwise, the all-0 configuration would not be in Y. By Lemma 7.9, the automaton enters a cycle of length at most $|Q|$ after at most $|Q|$ steps, and by our assumption, the net displacements in the cycle and the transient part are both in the set $W = \{\vec{u} \in \mathbb{Z}^d \mid \|\vec{u}\| \leq |Q|\}$. If we denote $\mathbb{Z}W = \{n \cdot \vec{w} \mid n \in \mathbb{Z}, w \in W\}$, then the automaton stays within the domain $\vec{v} + \mathbb{Z}W + W + W$ until it finds a 1. Let $E \subset \mathbb{Z}^d$ be the convex hull of $D = \{\vec{n} \in \mathbb{Z}^d \mid y_{\vec{n}} = 1\}$, and let $F = E + W + W$.

Now, since the automaton eventually enters the domain F, we have $F \cap \vec{v} + \mathbb{Z}W + W + W \neq \emptyset$, or in other words, $v \in F + W + W + \mathbb{Z}W$. Denote $G = F + W + W + W$, and suppose that we have $v \notin G$, so that $\min\{\|\vec{v} - \vec{u}\| \mid u \in F\} \geq 3|Q|$. Since the automaton enters a loop of length at most $|Q|$ after at most $|Q|$ steps, it will cycle through that loop at least twice before entering F. If we denote by $\vec{d} \in \mathbb{Z}^d$ the net displacement of M in the course of the loop, then the starting position $\vec{v} + \vec{d}$ results in the head of M entering the domain F in exactly the same state and position as with $\vec{v}$ (note that it will not enter F during the transient part of its run). This argument is visualized in Figure 7.1; if we translate the starting position by the vector $\vec{d}$, the number of loops that M makes before entering the set F just decreases by one. Thus it suffices to analyze the initial positions in the finite domain G.

From each starting position in the domain G, we now simulate the machine until it first enters F or exits $G + W$ (in which case it never enters F). Now, we note that if the machine exits F after the first time it is entered, then it does not reject y. Namely, the domain $F = E + W + W$ is convex and contains a 0-filled border thick enough that A must be in a loop, heading off to infinity. Thus, if M ever rejects y, it must do so by entering F from G

without exiting $W + G$, then staying inside F, and rejecting before entering a loop, which we can easily detect. This finishes the proof of decidability of $y \in Y$. □

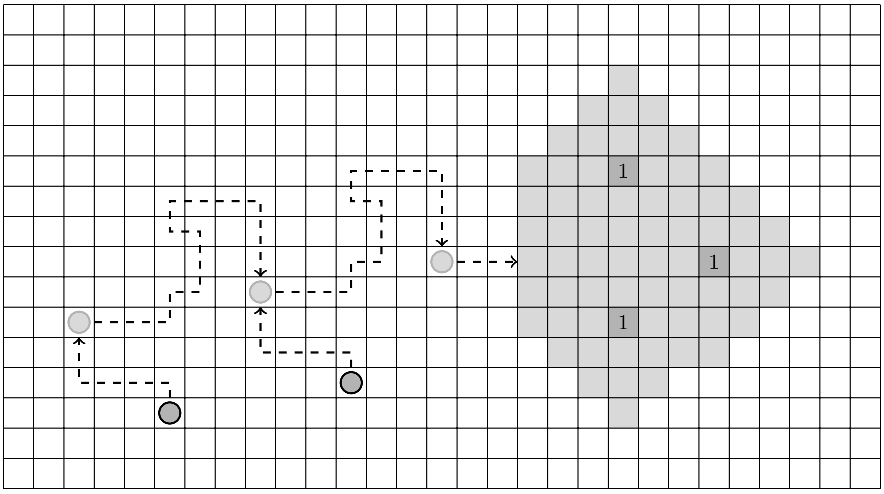

Figure 7.1: An illustration of the proof of Theorem 7.13, not drawn to scale. The gray zone represents the set F, and the circles represent the head of M. The leftmost dark circle is at the coordinate $\vec{v}$, and the rightmost dark circle is at $\vec{v} + \vec{d}$.

Now, given a binary pattern $P \in \mathcal{P}_d(\{0, 1\})$ with domain $D \subset \mathbb{Z}^d$, we claim that it is decidable whether P occurs in a configuration of Y. If $|P|_1 > m$, the answer is of course 'no' since Y is m-sparse, so suppose $|P|_1 \leq m$. Construct the configuration y with $y|_D = P$ and $y_{\vec{v}} = 0$ for all $\vec{v} \in \mathbb{Z}^d \setminus D$. If $y \in Y$, which is decidable by the above argument, then we answer 'yes'. If $y \notin Y$ and $|P|_1 = m$, then we can safely answer 'no'.

In the remaining case $y \notin Y$ and $|P|_1 < m$, we have found a rejecting run of M that only visits some finite set of coordinates $C \subset \mathbb{Z}^d$. If there exists a configuration $x \in Y$ such that $x|_D = P$, then necessarily $x_{\vec{v}} = 1$ for some coordinate $\vec{v} \in C \setminus D$, for otherwise M would reject x too. For all these finitely many $\vec{v}$, we construct a new pattern by adding the singleton pattern $\{\vec{v} \mapsto 1\}$ into P, and call this algorithm recursively on it. If one of the recursive calls returns 'yes', then we answer 'yes' as well. Otherwise, we answer 'no'. The correctness of this algorithm follows easily by induction. □

For the previous result to be nontrivial, it is important to explicitly take the intersection with a sparse subshift instead of assuming that the subshift X is sparse, for the following reason.

Proposition 7.15. *For all dimensions $d \geq 2$, the class $\mathcal{PW}_1^d$ contains no nontrivial sparse subshifts.*

Proof. Let M be a 1PWDFA such that the subshift $\mathrm{Acc}(M)$ is sparse and contains at least two configurations. We may assume that $X_1^d \subset \mathrm{Acc}(M)$ by recoding if necessary. Recall the notation of the proof of Theorem 7.13. If M can reach a position $\vec{w} \in \mathbb{Z}^d$ from an initial position $\vec{v} \in \mathbb{Z}^d$ the origin without encountering a 1, then $\vec{v} - \vec{w} \in W + W + \mathbb{Z}W$. Let $V \subset \mathbb{Z}^d$ be an infinite set such that $\vec{v} - \vec{w} \notin \mathbb{Z}W + W + W$ for all $\vec{v} \neq \vec{w} \in V$. Such a set exists since $d \geq 2$. Define the configuration $x \in \{0,1\}^{\mathbb{Z}^d}$ by $x_{\vec{v}} = 1$ for all $\vec{v} \in V$, and $x_{\vec{n}} = 0$ for all $\vec{n} \in \mathbb{Z}^d \setminus V$. Then $x \in \mathrm{Acc}(M)$, since the automaton M encounters at most one symbol 1 on every run on x. This contradicts the sparsity of $\mathrm{Acc}(M)$. $\square$

7.5 Two Heads

In this section, we show that two heads are already quite powerful in the one- and two-dimensional settings, and results such as the above do not hold for them. This results from the fact that the heads can carefully synchronize their movement even when moving in different directions at different speeds, so that they meet again at a precisely defined coordinate. In two dimensions, this allows us to perform exhaustive search on a configuration, in the following sense.

Proposition 7.16. *For all $m \geq 0$, the m-sunny side up shift X_m^2 is in $\mathcal{PW}_2^2$.*

Proof. We assume $m \geq 1$, since the case $m = 0$ is trivial. For all $a, b, c, d \in \mathbb{N}$ with $a+b+c+d = m+1$, we construct a two-head automaton $M_{a,b,c,d}$ with the following property. When started on top of a symbol 1 at the coordinate $\vec{0}$, the automaton rejects a configuration if and only if

- the top right quarter-plane $\mathbb{N} \times \mathbb{N}$ contains at least a symbols 1,
- the top left quarter-plane $(-\infty, -1] \times \mathbb{N}$ contains at least b symbols 1,
- the bottom left quarter-plane $(-\infty, -1] \times (-\infty, -1]$ contains at least c symbols 1, and
- the bottom right quarter-plane $\mathbb{N} \times (-\infty, -1]$ contains at least d symbols 1.

Clearly, the intersection of the subshifts accepted by the finitely many automata $M_{a,b,c,d}$ is precisely X_m^2.

Since the four cases are essentially symmetric, it is enough to construct an automaton M_a that checks that there are at least a symbols 1 in the top right quarter-plane, and then returns to its starting position. First, the automaton checks that it is indeed on top of a symbol 1, and enters an infinite loop if not.

The two heads of M_a are called the *L-head* and the *diagonal head*. Both heads remember a number $j \in [0, a]$, the number of the diagonal head being called the *count*, and the other the *height*. In the initial state, the count is 1 and the height is 0. We inductively preserve the following invariant. If both heads are at the coordinate $(0, n)$ for some $n \in \mathbb{N}$ and the count is $\ell < a$, then there are exactly ℓ symbols 1 in the coordinates $D_n = \{(i, j) \in \mathbb{N} \times \mathbb{N} \mid i+j \leq n\}$, and if $\ell = a$, then the domain D_n contains at least a symbols 1. Also, the height is precisely the number of 1s on the column between $(0, n)$ and $(0, 0)$. We now explain how, if both heads of the automaton are at the coordinate $(0, n)$ with count ℓ and height h so that the invariant holds, the heads can move to the coordinate $(0, n+1)$, preserving the invariant.

The automaton sends its L-head to south at speed 1, and the diagonal head southeast at speed 1/2 (that is, the diagonal head is translated by $(1, -1)$ every other step). Every time the L-head encounters a symbol 1, it decrements the height counter, which reaches the value 0 precisely at the origin. From the origin, the L-head turns to east, again using the height counter to remember the number of 1's it has seen on the row. The two heads meet at the coordinate $(n, 0)$. Now, the heads move one step to the right, possibly incrementing the width and height counters. The heads then repeat the procedure in reverse, with the difference that the diagonal head increments the count value for every 1 it encounters on its way northwest, up to the value of a. The heads meet again at $(0, n+1)$, and the invariant is preserved. See Figure 7.2 for a visualization of this process.

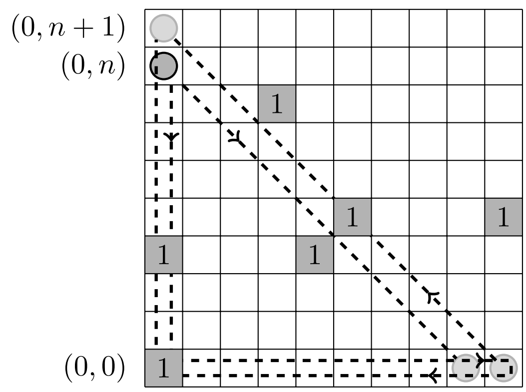

Figure 7.2: The movements of the L-head and the diagonal head.

Finally, if the count is a and the heads are at position $(0, n)$, they can return to the origin together with the aid of the height counter. $\square$

In [DM02], it was shown that on a uniform configuration, an automaton with two pebbles can perform a 'sweeping search' on a half-plane similarly to the above, and that such a search is essentially its only possible nontrivial behavior. Using this construction, it would not be very hard to show that, with the appropriate formalism, two-pebble automata can also recognize every two-dimensional m-sunny side up subshift.

The following proposition gives the separation of the classes $\mathcal{PW}_1^d$ and $\mathcal{PW}_2^d$ for $d \leq 2$. It can be thought of as an analogue of the well-known result that two counters are enough for arbitrarily complicated (though not arbitrary) computation. The result is of a computational nature, and before proving it, we recall Construction 3.12, where we embed the computations of an arithmetical program in a countable SFT, and especially the remark after it. In practice, the construction gives us a way of simulating an arithmetical program with a one-dimensional 2PWDFA; the rows of the associated SFT correspond to the steps of a run of the automaton. Note that in order to use the construction as such, we allow the heads to communicate with each other over a distance of one cell; see Remark 7.5.

Proposition 7.17. *For $d \leq 2$, there is a 2-sparse Π_1^0-complete subshift $X \in \mathcal{PW}_2^d$.*

Proof. We only prove the case $d = 2$, as the one-dimensional case is even easier. Let $X \subset \{0,1\}^{\mathbb{Z}^2}$ be the subshift of X_2^2 where either the two symbols 1 are on different rows, or their distance is not 2^n for any $n \in L$, for a fixed Σ_1^0-complete set $L \subset \mathbb{N}$. It is clear that X is a Π_1^0-complete subshift. Let P be an arithmetical program that recognizes the set $L' = \{2^n \mid n \in L\}$.

To prove that $X \in \mathcal{PW}_2^2$, we construct a 2PWDFA M for it. Since the class $\mathcal{PW}_2^2$ is closed under finite intersection, Proposition 7.16 implies that we may restrict our attention to configurations of the 2-sunny side up shift X_2^2. First, our machine checks whether it is started on a symbol 1 and that another symbol 1 occurs on the same row to the left, by doing a left-and-right sweep with one of the heads. If the other 1 is never found, then M runs forever without halting. Otherwise, the two heads position themselves on top of the rightmost 1 and to its left, and start simulating the arithmetical program P exactly as in Construction 3.12. The leftmost 1 is used as the border of the computation cone, so that the distance $N \geq 1$ of the two symbols 1 is the input of P. The heads also remember on which side of the rightmost 1 they are, in order to recognize the border; apart from this, the rightmost 1 is ignored after the beginning of the simulation. The automaton rejects the configuration as soon as the simulated program P halts, and otherwise continues the computation infinitely.

Now, let $x \in X_2^2$ be arbitrary. If M is not started on the rightmost 1 of a row of x that contains two 1's, then it does not reject x. Suppose then that this holds, and let $N \geq 1$ be the distance between the two 1's. If $N \in L'$, then the simulated program P eventually halts and the automaton rejects, and we have $x \notin X$. Otherwise, the program and thus the automaton run forever, and $x \in X$ since M does not reject x from any starting position. This shows that $\mathrm{Acc}(M) = X$. $\square$

We do not believe that *all* 2-sparse Π^0_1-complete subshifts are in $\mathcal{PW}^d_2$ for $d \leq 2$, but we cannot currently prove this. In three or more dimensions, however, we obtain the following analogue of Proposition 7.15, which is proved similarly.

Theorem 7.18. *For all dimensions $d \geq 3$, the class $\mathcal{PW}^d_2$ contains no nontrivial sparse subshifts.*

Proof. Let M be a 2PWDFA taking only steps of length 0 or 1 such that $\mathrm{Acc}(M)$ is sparse and contains at least two configurations. We may again assume that $X^d_1 \subset \mathrm{Acc}(M)$. As in the proof of Theorem 7.13, we can use Lemma 7.9 to show that there exists some $p \in \mathbb{N}$ such that, denoting again $W = \{\vec{v} \in \mathbb{Z}^d \mid \|\vec{v}\| \leq p\}$ and $\mathbb{Z}W = \{n \cdot \vec{w} \mid n \in \mathbb{Z}, \vec{w} \in W\}$, we have the following. Let the two heads of M be initialized on some coordinates $\vec{v} = \vec{v}_0 \in \mathbb{Z}^2$ and $\vec{w} = \vec{w}_0 \in \mathbb{Z}^2$ in any states, and denote by $(\vec{v}_n)_{n \leq N}$ and $(\vec{w}_n)_{n \leq N}$ their itineraries up to some timestep $N \in \mathbb{N}$. If we have $\|\vec{v} - \vec{w}\| \leq p$, then $\vec{v}_n \in \vec{v} + \mathbb{Z}W + W$ and $\vec{w}_n \in \vec{w} + \mathbb{Z}W + W$ until either head sees a symbol 1. If $\|\vec{v} - \vec{w}\| > p$, then this holds until either head sees a symbol 1 or the heads meet each other. Note that in the former case, the heads may travel together, but in that case we can view them as a single head with a larger but finite 'combined' state space.

Analogously to the proof of Proposition 7.15, let $V \subset \mathbb{Z}^d$ be an infinite set such that $\vec{v} - \vec{w} \notin \mathbb{Z}W + \mathbb{Z}W + W + W$ for all $\vec{v}, \vec{w} \in V$, and $\vec{0} \in V$. Define a configuration $x \in \{0,1\}^{\mathbb{Z}^d}$ by $x_{\vec{v}} = 1$ for all $\vec{v} \in V$, and $x_{\vec{v}} = 0$ for all $\vec{v} \in \mathbb{Z}^d \setminus V$. We prove that x is accepted by M, contradicting the sparsity of $\mathrm{Acc}(M)$, and for that, let $M^\infty_x(c)$ be a run of M in x started from some initial ID $c = (\vec{w}, \vec{w}, q_0)$ for $\vec{w} \in \mathbb{Z}^d$.

Claim 7.19. The heads of M visit at most one letter 1 during the run $M^\infty_x(c)$.

Proof of claim. We may assume that M encounters a 1 at the origin after some number of steps. By the first paragraph, both heads stay in the region $\vec{w} + \mathbb{Z}W + W$ until the origin is found, say by the first head. This implies that $\vec{w} \in \mathbb{Z}W + W$, so the second head stays in the domain $\mathbb{Z}W + \mathbb{Z}W + W + W$ until it encounters the origin or the first head. The first head, on the other hand, is restricted to the domain $\mathbb{Z}W + W$ until it meets the second head again. If the heads meet, they must do so in a coordinate of $\mathbb{Z}W + W$, and after this, they are confined to the domain $\mathbb{Z}W + \mathbb{Z}W + W + W$ until one of them reaches the origin again. All in all, the heads never leave the domain $\mathbb{Z}W + \mathbb{Z}W + W + W$, and thus never reach a symbol 1 outside the origin. $\square$

Define a configuration $y \in X_1^d$ that has a single letter 1 at the origin, and 0 everywhere else. By the above claim, we have $M_x^\infty(c) = M_y^\infty(c)$, that is, the run of M on y from the initial ID c is exactly the same as that on x, since $x_{\vec{n}} = y_{\vec{n}}$ for all coordinates $\vec{n} \in \mathbb{Z}^d$ visited during the run. Since $X_1^d \subset \mathrm{Acc}(M)$, the run is accepting, and thus $x \in \mathrm{Acc}(M)$, contradicting the sparsity of $\mathrm{Acc}(M)$. $\square$

The following result does not concern plane-walking automata as such, but combined with Theorem 7.18, it provides examples of sofic shifts outside the class $\mathcal{PW}_2^d$. Its proof uses a construction very similar to that of Theorem 5.17. However, since we cannot control the structure of our auxiliary subshifts with first-order variables, the proofs differ in many details. Also, this construction is one of the few one in this dissertation where we use a Turing machine for computation, instead of some variant of a counter machine.

Theorem 7.20. *In all dimensions $d \geq 2$, every sparse Π_1^0 subshift is sofic.*

Proof. We prove the result in two dimensions, and then discuss how to generalize it to $d \geq 3$.

We will show that every two-dimensional Π_1^0 subshift X over the alphabet $\{0, \ldots, k\}$ containing all symbols except 0 at most once is sofic. This proves the original claim, since the class of sofic shifts is closed under renaming the symbols. Since X is a Π_1^0, there is a computable sequence $(P_i)_{i \in \mathbb{N}}$ of forbidden patterns for it. We will construct an SFT $Y \subset \{0, \ldots, k\}^{\mathbb{Z}^2} \times Z$, where Z is also an SFT, such that the projection of Y to the first layer is exactly X. The SFT $Z \subset \prod_{i=1}^k Z_i$ consists of k identical layers Z_i, and in each of them, we simulate a Turing machine T with state set Q, initial state $q_0 \in Q$, and tape alphabet Γ, whose functionality we describe later.

The k signal layers are identical to each other, and their tiles are shown in Figure 7.3. The labels $q_i \in Q$ in the gray circles are the states of the Turing machine T, and $a_i, b_i \in \Gamma$ are symbols of the tape alphabet. The upper half plane of the configuration is the *computation region*, and the background colors form the *signals* of the layer. On the computation region, an infinite computation of T is simulated in a standard way, with time increasing upwards. The initial state q_0 is assigned to the coordinate of intersection of the signals, called the *point* of the configuration, and the other tiles of the bottom row of the computation region are given arbitrary labels from the tape alphabet Γ. One computation step of T is simulated on each subsequent row using suitable 2×2 forbidden patterns, and if T ever halts, a tiling error is introduced.

Now, the point of each layer corresponds exactly to one of the nonzero symbols of a configuration of X, similarly to Theorem 5.17. More precisely, for all $i \in \{0, \ldots, k\}$, we forbid from the subshift Y all symbols $(j, s) \in$

$\{0,\dots,k\}^{\mathbb{Z}^2} \times \mathcal{A}(Z)$ where $j = i$ but the Z_i-component of s is not the point of Z_i, or $j \neq i$ but the Z_i-component of s is the point of Z_i. This already implies that every configuration on the first layer of Y contains at most one occurrence of i.

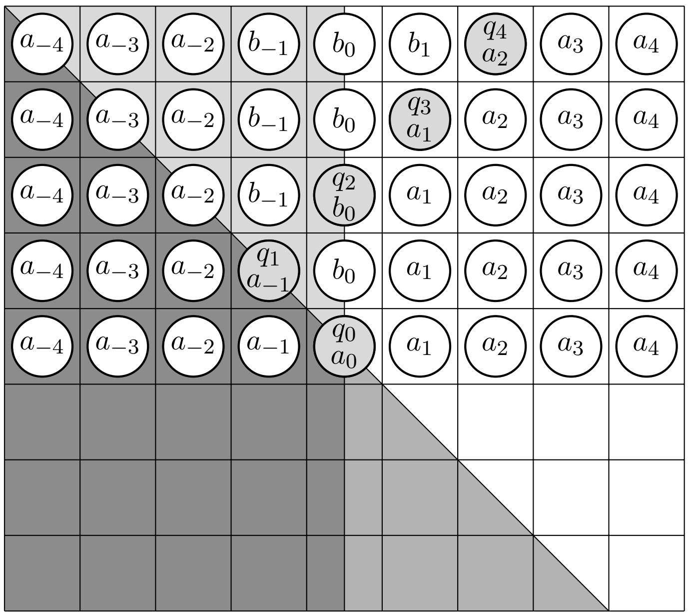

Figure 7.3: A configuration of the signal layer Z_i. The gray circles contain the head of the simulated Turing machine T, and the white circles contain elements of its tape alphabet Γ.

Next, we relate the different layers Z_i to one another by introducing restrictions on the initial tape contents of the simulated machines. We require the tape alphabet Γ to contain the $(k-1)$-fold Cartesian product $\{\boxbslash, \boxbar, \boxbslash, \square\}^{k-1}$ as a subset; these are the discolored and label-less versions of the tiles of the other layers Z_j. We require that on the bottom of the computation region of Z_i, the tape symbol of each tile consists exactly of the discolored versions of the tiles of the other layers at that coordinate. The reason for this is that now the relative positions of all points in the configuration can be deduced from the tape of the Turing machine T simulated by the layer Z_i, in the sense that the simulated machine T can enumerate the rectangular patterns $R_j = x|_{[-j,j]^2+\vec{n}}$ for all $j \in \mathbb{N}$, where $\vec{n} \in \mathbb{Z}^2$ is the coordinate of the i'th point of x. See Figure 7.4 for a visualization of this argument. Finally, we construct the machine T so that it enumerates in parallel the forbidden patterns $(P_i)_{i\in\mathbb{N}}$ and the rectangular patterns $(R_j)_{j\in\mathbb{N}}$, and halts as soon as any element of the former sequence occurs in an element of the latter.

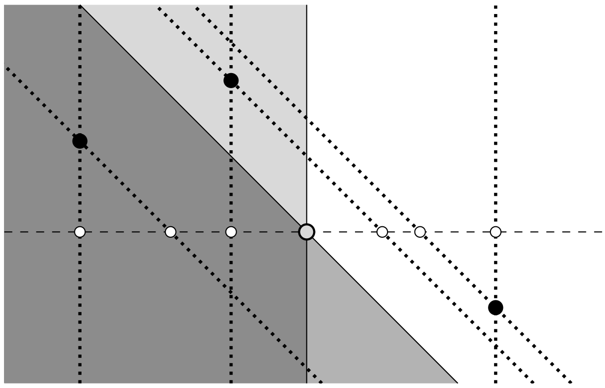

Figure 7.4: The signal regions of the other layers transmit the information about the positions of their points (the black circles) to the layer Z_i, whose point is shown in gray.

Now, a given configuration $x \in \{0, \ldots, k\}^{\mathbb{Z}^d}$ is a projection of a configuration of Y if and only if every symbol $i \in \{1, \ldots, k\}$ occurs in x at most once, and for the one occurring at the coordinate $\vec{n} \in \mathbb{Z}^2$ as defined above, no forbidden pattern P_i occurs in the rectangle $x|_{[-j,j]^2+\vec{n}}$ for any $i, j \in \mathbb{N}$. This is equivalent to $x \in X$.

This construction can be generalized to $d \geq 3$ dimensions in a relatively straightforward way. Namely, instead of the signal and computation regions being half-planes, they are d-dimensional half-spaces. Likewise, the signal regions of the layers Z_i contain d-dimensional pyramids instead of two-dimensional cones, and the simulated Turing machine is run on a $(d-1)$-dimensional lattice instead of a one-dimensional tape. The rest of the construction is unchanged. □

Combining Theorem 7.20, Theorem 7.18 and Proposition 7.15, we obtain the following.

Corollary 7.21. *For all dimensions $d \geq 2$, we have $\mathcal{PW}_1^d \subsetneq \mathrm{sofic}^d$, and for all dimensions $d \geq 3$, we have $\mathcal{PW}_2^d \not\subset \mathrm{sofic}^d$.*

While Theorem 7.20 in particular implies that all sparse $\mathcal{PW}_2^d$ subshifts are sofic, we can show that this is no longer true if we drop the requirement of sparsity. In particular, the next result shows that the class $\mathcal{PW}_1^d$ is properly contained in $\mathcal{PW}_2^d$ for all $d \geq 2$.

Proposition 7.22. *In all dimensions $d \geq 2$, we have $X^d_{\mathrm{mirror}} \in \mathcal{PW}_2^d \backslash \mathrm{sofic}^d$.*

Proof. First, as stated earlier in this chapter, the mirror subshift $X^d_{\mathrm{mirror}} \subset \{0, 1\#\}^{\mathbb{Z}^d}$ is not sofic for any dimension $d \geq 2$. To show that $X^d_{\mathrm{mirror}} \in \mathcal{PW}_2^d$, we informally describe a 2PWDFA for it. Using the fact that $\mathcal{PW}_2^d$ is closed under intersection, we restrict to the SFT defined by the forbidden patterns of the first point of Definition 7.8: all patterns P of domain $\{\vec{0}, \vec{e}_i\}$ with $|P|_\# = 1$. We can also assume there is at most one hyperplane of symbols #, as this can be checked by a 1PWDFA that walks in the direction of the first axis from its initial position, and halts if it sees the symbol # twice.

Under these assumptions, the mirror property is easy to check. When initialized on a coordinate containing a letter $b \in \{0, 1\}$, one of the heads of the 2PWDFA memorizes the letter b in its finite memory. Then, one of the heads starts traveling to the direction $\vec{e}_1$, and the other to $\vec{e}_1 + \vec{e}_2$. If the latter sees a hyperplane of symbols #, it turns to the direction $\vec{e}_1 - \vec{e}_2$. If the heads meet, they check that the letter b in the initial position matches the letter under the current position, and if not, the configuration is rejected. $\square$

7.6 Three Heads

In this section, we collapse the hierarchy of plane-walking automata to its third level. This can be thought of as an analogue of the well-known result that three counters and a finite memory are enough for arbitrary computation, as opposed to two counters, which can perform arbitrarily complex but not arbitrary computation. We also note that it was proved in [DM02] that pebble automata with three pebbles are equally powerful to pebble automata with an arbitrary number of counters, and indeed to Turing machines, when it comes to recognizing topologically open sets of configurations. The next result shows that in this sense, three-headed plane-walking automata are similar to three-pebble automata. It should also be compared to the results of Chapter 5, where several quantifier alternation hierarchies were shown to collapse into finitely many different levels.

Theorem 7.23. *In all dimensions $d \geq 1$, the classes $\mathcal{PW}_k^d$ for $k \geq 3$ coincide with the class of Π_1^0 subshifts.*

Proof. We only need to show that $\mathcal{PW}_3^d$ contains all Π_1^0 subshifts. Namely, $\mathcal{PW}_k^d \subset \mathcal{PW}_{k+1}^d$ holds for all $k \geq 1$, and since a counter machine can easily enumerate patterns supporting a rejecting computation of a multihead finite automaton, every $\mathcal{PW}_k^d$ subshift is also Π_1^0.

Let C be a counter machine that, when started from the initial ID c_0, outputs a sequence $(P_i)_{i \in \mathbb{N}}$ of patterns by writing an encoding of each of them in turn to a special counter, and visiting a special state q_{out}. We construct a 3PWDFA M_C accepting exactly those configurations where no P_i occurs. The heads of M_C are called the *pointer head*, the *zig-zag head*, and the *counter head*. The machine has a single initial state, and when started from any position $\vec{v} \in \mathbb{Z}^d$ of a configuration x, it checks that no P_i

occurs in x at $\vec{v}$. Since M_C is started from every position, it will then forbid all translates of the P_i.

The machine simulates an arithmetical program as in the proof of Proposition 7.17, but in place of the 'leftmost symbol 1', we use the pointer head. The crucial difference here is that unlike a symbol 1, the pointer head can be moved freely. This allows us to walk around the configuration, and extract any information we want from it. The arithmetical program simulates Algorithm 5, which finally simulates the counter machine M_C.

Algorithm 5 The algorithm that the three-head automaton M_C simulates.

$c \leftarrow c_0$ ▷ An ID of C, set to the initial ID
$\vec{u} \leftarrow \vec{0} \in \mathbb{Z}^d$ ▷ Position of the pointer head relative to initial position
$P : \emptyset \to \{0,1\}$ ▷ A finite pattern at the initial position
loop
 repeat
  $c \leftarrow \textsc{NextID}_C(c)$ ▷ Simulate one step of C
 until $\textsc{State}(c) = q_{\text{out}}$ ▷ C outputs something
 $P' \leftarrow \textsc{OutputOf}(c)$ ▷ A forbidden pattern
 while $D(P') \not\subset D(P)$ **do**
  $\vec{w} \leftarrow \textsc{LexMin}(D(P) \setminus D(P'))$ ▷ Lexicographic minimum
  while $\vec{u} \neq \vec{w}$ **do**
   $\vec{d} \leftarrow \textsc{NearestUnitVector}(\vec{w} - \vec{u})$ ▷ Nearest vector $\pm\vec{e}_i$
   $\textsc{MoveBy}(\vec{d})$ ▷ Move the heads of M_C to the given direction
   $\vec{u} \leftarrow \vec{u} + \vec{d}$
  $b \leftarrow \textsc{ReadSymbol}$ ▷ Read symbol of x under the pointer head
  $P \leftarrow P \cup \{\vec{u} \mapsto b\}$ ▷ Expand P by one coordinate
 if $P|_{D(P')} = P'$ **then halt** ▷ The forbidden pattern P' was found

The algorithm keeps in its memory a finite pattern $P = x|_{D(P)+\vec{v}}$, where $\vec{v} \in \mathbb{Z}^d$ is the initial position of the heads, and a vector $\vec{u} \in \mathbb{Z}^d$ containing $\vec{w}-\vec{v}$, where $\vec{w} \in \mathbb{Z}^d$ is the current position of the pointer head. The machine C is simulated step by step, and whenever it outputs a forbidden pattern P', the algorithm checks whether $D(P)$ contains its domain. If so, it then checks whether $x|_{D(P')+\vec{v}} = P'$. If this holds, then the algorithm halts, the arithmetical program simulating it halts, and the automaton M_C moves all of its heads to the pointer and rejects. If P' does not occur, the simulation of C continues.

If the domain $D(P')$ is not contained in $D(P)$, then the algorithm expands P, which is done in the outer **while**-loop of Algorithm 5. To find out the contents of x at some coordinate $\vec{w} + \vec{v}$ for $\vec{w} \in D(P')$, the algorithm chooses a unit direction (one of $\pm\vec{e}_i$ for $i \in \{1, \dots, d\}$) that would take the pointer head closer to $\vec{w} + \vec{v}$, and signals it to M_C via the arithmetical program. In a single sweep of the zig-zag head to the pointer and back, M_C can easily move all of its heads one step in any unit direction. Then the simulation continues, and the algorithm updates $\vec{u}$ accordingly. When $\vec{u} = \vec{w}$ finally holds, the algorithm orders M_C to read the symbol $x_{\vec{v}+\vec{u}}$ under the pointer, which is again doable in a single sweep. The bit $b = x_{\vec{v}+\vec{u}}$ is given to the algorithm, which expands P by defining $P_{\vec{u}} = x_{\vec{v}+\vec{u}}$.

For a configuration x and initial coordinate $\vec{v} \in \mathbb{Z}^d$, the automaton M_C thus computes the sequence of patterns $(P_i)_{i \in \mathbb{N}}$ and checks for each $i \in \mathbb{N}$ whether $x|_{D(P_i)+\vec{v}} = P_i$ holds, rejecting if it does. Since $\vec{v}$ is arbitrary, we have $x \in \mathrm{Acc}(M_C)$ if and only if no P_i occurs in x. Thus the class $\mathcal{PW}_3^d$ contains an arbitrary Π_1^0 subshift. $\square$

Chapter 8

Conclusions

8.1 The Structure of Subshifts

In Chapter 4, we investigated the topological and dynamical structure of multidimensional subshifts, in particular countable SFTs, formalized as their Cantor-Bendixson ranks and subpattern posets. In Theorem 4.13, we presented a countable SFT whose iterated derivatives are maximally complex from the computational point of view. This construction has been published already in [ST13]. Following that, we proved in Theorem 4.14 that any rank of the form $\lambda + 3$, where λ is a computable ordinal, can be attained by a countable SFT, and Theorem 4.15 from the preprint [JV11] shows that this is actually the characterization of the ranks of countable SFTs. Using the same construction, we presented an example of a countable SFT of rank 5 that contains an uncomputable configuration. It is known from the results of the preprint [JV11] that all configurations of countable STFs of rank at most 4 are computable, so the example proves the strictness of this bound.

In Section 4.4, we concentrated on the subpattern posets, and proved Theorem 4.26 with the most complex construction in this dissertation. Together with Corollary 4.24, it gives a complete characterization for the 'higher structure' of the subpattern posets of countable Π^0_1 subshifts and SFTs, in the sense that only the elements of height at most 3 cannot be precisely controlled. Finally, we showed that the bounded signal property severely restricts the structure of a countable SFT, forcing its subpattern poset to have the descending chain property.

While the Cantor-Bendixson ranks of one-dimensional sofic shifts and two-dimensional countable SFTs have now been completely characterized (as the finite ordinals in the first case, and those of the form $\lambda+3$ for computable ordinals λ in the second), the exact structures of their derivatives have not. In particular, it would be interesting to understand which SFTs and sofic shifts are 'integrable' within the classes of SFTs or sofic shifts, in the sense of arising as the Cantor-Bendixson derivative of another subshift in the class, and which subshifts have SFT derivatives. For example, we claim that the one-dimensional countable SFT $X = \mathcal{B}^{-1}(0^*1^*2^*)$ is not integrable within the class of sofic shifts. For this, suppose on the contrary that $Y \subset A^{\mathbb{Z}}$ is a sofic shift with $Y^{(1)} = X$. For all $m \in \mathbb{N}$, the subshift Y necessarily contains a pattern $a0^\ell 1^m 2^n b$ for some $(a, b) \neq (0, 2)$ and arbitrarily large $\ell, n \in \mathbb{N}$. But since Y is sofic, for some $(a, b) \neq (0, 2)$ and $\ell, n \in \mathbb{N}$ there exist infinitely many m as above, and then both $a0^\ell 1$ and $12^n b$ are patterns of the derivative $Y^{(1)} = X$, a contradiction (in fact, X is not integrable at all, by a slightly extended argument). An analogous argument shows that the sofic shift $\mathcal{B}^{-1}(0^*10^*20^*)$ is not integrable within the class of sofic shifts either, even though it has the integral $\mathcal{B}^{-1}(\{0^k 10^\ell 10^m 20^n \mid k, \ell, m, n \in \mathbb{N}, \ell \geq m\})$. Of course, it would also be interesting to know what happens in the case of an uncountable sofic shift.

The bounded signal property and its relatives are interesting also in the two-dimensional uncountable case: We can construct an uncountable SFT X which has the bounded sofic signal property (the projective sybdynamics is contained in a countable sofic shift) in all rational directions except the horizontal one, and even the horizontal projective subdynamics is countable. This SFT is like a transposed version of Example 3.4, but one edge of the infinite cone is not a straight line, but instead a discrete version of a parabola. Furthermore, the signal inside the cone always bounces in the same way, but every bounce is independently given an label from $\{0, 1\}$. Figure 8.1 shows an example configuration of X.

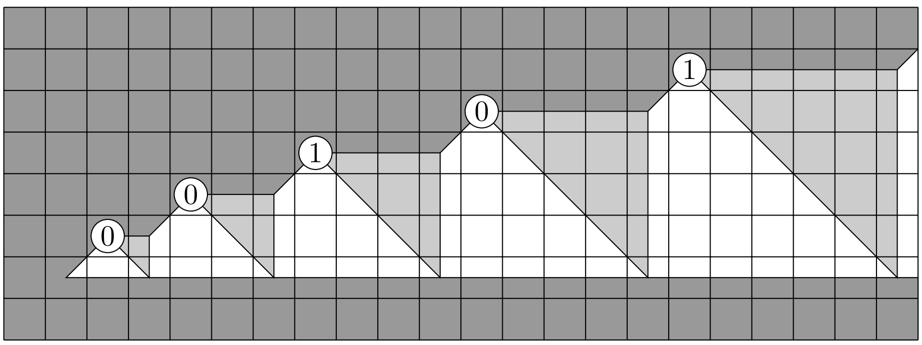

Figure 8.1: An example configuration of the uncountable X which has the bounded sofic signal property in all rational directions except one.

For any direction $\vec{d} \in \mathbb{Z}^2$ except the horizontal one, there is a bound on the number of gray triangles and labeled tiles that a line in direction $\vec{d}$ can pass in a configuration of X, and thus the $\vec{d}$-projective subdynamics of X is contained in a countable sofic shift. It is also easy to see that the rows of X form a countable set, since all configurations of X are translates of each other except for the labels, and each row contains at most one labeled tile. Furthermore, the image of X under the symbol map $f : X \to \{0,1\}^{\mathbb{Z}^2}$ that sends each tile with label 1 to 1 and everything else to 0 is an uncountable sofic shift with the bounded sofic signal property in every direction. However, we do not see how to construct an uncountable SFT which has the bounded signal property or bounded sofic signal property in every direction. In fact we conjecture that this cannot be done.

Conjecture 8.1. An uncountable SFT cannot have the bounded signal property, or even the bounded sofic signal property, in every rational direction.

With Theorem 4.26, we have obtained a pretty complete understanding of the 'high-level structure' of the subpattern posets of countable SFTs. Conversely, it would be interesting to study the elements of height at most 3 in these posets, and in particular, obtain a characterization for the finite posets that are *exactly* realizable as subpattern posets. For example, we know that the two-element poset $\{0, 1\}$ with $0 < 1$ cannot be realized as the subpattern poset of a countable SFT, but it is of course order-isomorphic to the subpattern poset of the sunny side up subshift, which is a countably covered sofic shift. This direction of research has already been initiated in the preprint [BJ13], which studies the geometric structure of configurations of low rank in countable SFTs and countable subshifts in general. In addition, even though Theorem 4.26 shows that infinite descending chains can be found in the subpattern posets of countable SFTs, [BJ13] provides some restrictions on the structure of these chains in relation to the whole SFT; namely, the set of configurations that lie below all elements of the chain in the subpattern order must contain a configuration of height at least 2. Intuitively, this means that complicated structures in subpattern posets need some additional 'support' in the form of auxiliary configurations of bounded height.

8.2 Subshifts and Logical Quantification

In Chapter 5, we studied classes of two-dimensional subshifts defined by monadic second-order logic, focusing on the natural class of u-MSO formulas. We provided a complete characterization of the class of subshifts definable by u-MSO formulas, and for all the (finitely many) levels of the associated quantifier alternation hierarchy, which correspond to well-known classes of subshifts. More explicitly, it was already known from the results of [JT13] that the class $\bar{\Sigma}_0 = \bar{\Pi}_0 = \bar{\Pi}_1$ corresponds exactly to the threshold counting shifts, while $\bar{\Sigma}_1$ coincides with the class of sofic shifts. We showed that for all $n \geq 2$, the class $\bar{\Pi}_n$ contains exactly the Π^0_1 subshifts, and in particular, the whole hierarchy collapses to $\bar{\Pi}_2$. The proof of this fact was an intricate construction of a $\bar{\Pi}_2$ formula implementing an arbitrary Π^0_1 subshift. We also extended this construction to prove the analogous result for the larger classes $\bar{\Pi}_n[\Pi^0_k]$ of generalized u-MSO formulas: the class $\bar{\Pi}_n[\Pi^0_k]$ coincides with Π^0_{k+1} for all $k \geq 1$ and $n \geq 2$. Since every extended MSO formula can be mechanically translated into an equivalent standard MSO formula, provided that its auxiliary subshifts are MSO-definable, we obtained as a corollary that all arithmetical (Π^0_k for some $k \geq 0$) subshifts are MSO-definable.

Having proved that the extended u-MSO hierarchies are finite, we moved on to study their lower levels. Even though the equality $\bar{\Pi}_1 = \bar{\Pi}_0$ holds for ordinary u-MSO formulas, it is no longer true in the extended case: when quantifying universally over the configurations of a nontrivial subshift, the contents of arbitrarily distant coordinates may be strongly correlated. We introduced the classes $\mathrm{B}\Sigma^0_k$ of Σ^0_k-bounded subshifts for $k \geq 0$, which are defined by a Σ^0_k set of forbidden patterns with bounded domain size (but not necessarily bounded diameter). The geometric properties of Σ^0_k-bounded subshifts are interesting: each of the classes $\mathrm{B}\Sigma^0_k$ properly contains the SFTs, but is incomparable with the class of sofic shifts. It turned out that universal quantification over arithmetical subshifts gives rise to the Σ^0_k-bounded classes. More precisely, we showed in Theorem 5.17 that $\bar{\Pi}_1[\Pi^0_k] = \mathrm{B}\Sigma^0_{k+1}$ holds for all $k \geq 1$. As for the class $\bar{\Sigma}_2[\Pi^0_k]$, its relation to $\bar{\Pi}_1[\Pi^0_k]$ is exactly the same as that of $\bar{\Sigma}_2$ to $\bar{\Pi}_1$, that is, its closure under images of block maps.

The class of Σ^0_k-bounded subshifts, and more generally, the class of subshifts defined by forbidden patterns of bounded-cardinality domains, is an interesting object of study in its own right. In Example 5.19, we presented a simple binary sofic shift which does not lie in the latter class, while in Example 5.16, we proved that the nonsofic mirror subshift X_{mirror} does. This shows that the geometry of the class is, in some sense, very limited, even though it contains uncountably many subshifts, and the computational complexity of the subshifts it contains is unbounded. It is also easy to see that the class contain every SFT, and every sparse subshift. We showed that the classes $\bar{\Sigma}_2[\Pi^0_k]$ on the next level of the hierarchy are obtained by taking the closures of the corresponding classes $\bar{\Pi}_1[\Pi^0_k]$ under block map images. The sofic shifts are contained in all these classes, and Example 5.21 shows that the relationship between sofic shifts and Σ^0_k-bounded subshifts is anything but simple.

Moreover, in this dissertation we have only considered classes of u-MSO formulas. The reason for this is mainly that general MSO formulas are less well-behaved from the viewpoint of multidimensional symbolic dynamics, in the sense that they do not always define subshifts. For example, if we take any closed formula ϕ over an alphabet A such that $X_\phi \subset A^{\mathbb{Z}^2}$ is a nontrivial subshift, then $\neg\phi$ defines the set $A^{\mathbb{Z}^2} \setminus X_\phi$, or the complement of X_ϕ, which is not topologically closed. In particular, we have not presented a characterization for MSO-definable subshifts in this dissertation. It is very likely that MSO-definable are exactly those whose language satisfies some suitable (hyper)computability condition, such as being hyperarithmetical or analytical (a subset of $\mathbb{N}$ is analytical, if it can be defined by a second-order logical formula), but we leave this problem open for now.

Question 8.2. Is there a simple characterization for the class of MSO-definable subshifts?

The following chapter continued the theme of constructing subshifts using logical operations, but with a more concrete and combinatorial approach. We studied the quantifier extension operations on two-dimensional subshifts, which can be seen as 'simplified' universal and existential quantifiers, and proved some of their basic properties. We showed that the classes of SFTs and one-dimensional sofic shifts are closed under the universal extension, and that the class of general sofic shifts is closed under the existential extension. However, as may have been be expected from the results of Chapter 5, the universal extension of a multidimensional sofic shift may not be sofic itself. We proved this first by a computation-theoretical argument using a Π^0_1-complete extension, and then in Theorem 6.9 by a geometric construction using an arbitrary infinite subshift. As with most of the proofs of non-soficness in this dissertation, the crucial counting argument was given by Lemma 3.15.

In the final section of the latter Chapter 6, we discussed the effect of determinism on the universal extension. It turned out that if two subshifts $X, Y \subset A^{\mathbb{Z}^2}$ are both deterministic, then so is their extension $A(X, Y) \subset \hat{A}^{\mathbb{Z}^2}$; however, we do not know whether Theorem 6.9 can be extended to the case of deterministic sofic shifts, since the existence of deterministic Π^0_1 subshifts which are not sofic is still an open problem.

Question 8.3. Let $X, Y \subset A^{\mathbb{Z}^2}$ be downward deterministic sofic shifts, and suppose that the language $\mathcal{B}(Y)$ is computable. Is the universal extension $A(X, Y)$ necessarily a sofic shift?

In Theorem 6.9, we fixed the infinite subshift Y, and then meticulously constructed a sofic shift X such that the extension $A(X, Y)$ is not sofic. This raises the natural dual problem in which, given a fixed sofic shift X, we try to find another subshift Y that yields a nonsofic extension.

Question 8.4. For a given sofic shift $X \subset A^{\mathbb{Z}^2}$, does there exist a (sofic or recursive) subshift $Y \subset A^{\mathbb{Z}^d}$ such that the universal extension $A(X, Y)$ is not sofic?

By Proposition 6.2, the answer is negative in both cases if X is an SFT. Corollary 6.7 and Theorem 6.9 show that there exist some particular and quite intricate sofic shifts X for which the answer is positive in both cases.

Finally, recall from Proposition 6.4 that the existential extension of every one-dimensional SFT by another SFT is a sofic shift, but Example 6.3 showed that it may not be an SFT, even when extending by a full shift. This raises the following question.

Question 8.5. Let $X, Y \subset A^{\mathbb{Z}}$ be SFTs. When is the existential extension $E(X, Y)$ an SFT?

8.3 Plane-Walking Automata and the Use of Arithmetical Programs

In Chapter 7, we introduced another way of defining multidimensional subshifts: plane-walking finite automata with multiple heads. As with MSO logic, this model also gives rise to a natural hierarchy of subshift classes, obtained by increasing the number of heads. We showed that the classes $\mathcal{PW}_1^d$ of subshifts recognized by one-headed automata lie between the d-dimensional SFTs and sofic shifts, and in the one-dimensional case we even proved $\mathcal{PW}_1^1 = \text{sofic}^1$. In the case of one and two heads, we also showed that there exists a 'critical dimension' (2 in the case of one head, and 3 in the case of two) in which no sparse subshifts can be recognized. Using the terminology of [DM02], this is due to the inability of these automata to *search the world*, that is, visit every coordinate of a uniform or sparse configuration from a single initial position. In [DM02] it was proved that an automaton with at most 2 immobile pebbles cannot search the world on a uniform configuration, although it can search an infinite quarter plane, using a technique similar to that in the proof of Proposition 7.16. However, even if we restrict our attention to sparse subshifts, two heads are still vastly more powerful than one, especially in the computational sense: the sparse part of every subshift in $\mathcal{PW}_1^d$ has a decidable language, while $\mathcal{PW}_2^1$ contains a Π_1^0-complete sparse subshift. We also showed that, like each Σ_k^0-bounded class $\mathrm{B}\Sigma_k^0$, the classes $\mathcal{PW}_2^d$ are incomparable with sofic shifts. Finally, we proved the analogue of Theorem 5.11 for plane-walking automata: in every dimension d, the hierarchy $(\mathcal{PW}_k^d)_{k \geq 1}$ collapses to the third level $\mathcal{PW}_3^d$, since three-headed automata can recognize any Π_1^0 subshift.

The basic comparisons obtained in this chapter are summarized in Figure 8.2. A major missing link in our classification is the separation of $\mathcal{PW}_2^d$ and $\mathcal{PW}_3^d$ in dimensions $d \leq 2$. We leave this problem unsolved, but state the following conjecture.

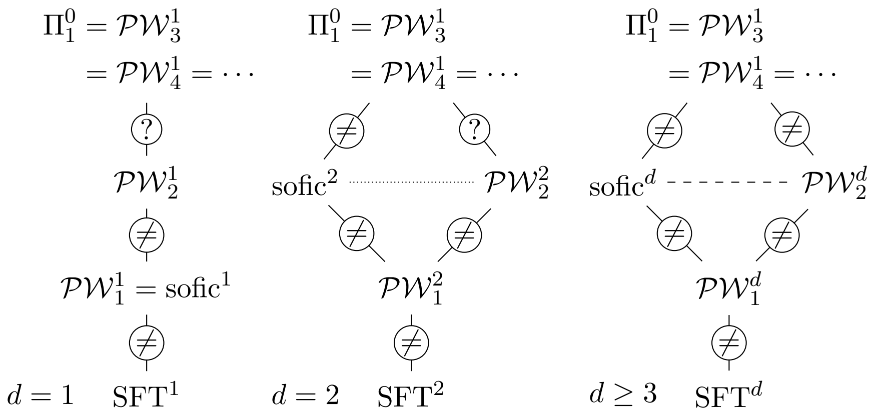

Figure 8.2: A comparison of the classes of subshifts studied in Chapter 7 in dimensions $d = 1$, $d = 2$ and $d \geq 3$. The solid, dashed and dotted lines denote inclusion, incomparability and an unknown relation, respectively, as we only know $\mathcal{PW}^d_2 \not\subset \text{sofic}^d$ for $d = 2$.

Conjecture 8.6. For $d \leq 2$, there exists a sparse Π^0_1 subshift which is not in $\mathcal{PW}^d_2$. In particular we have $\mathcal{PW}^d_2 \subsetneq \mathcal{PW}^d_3$, and the classes sofic^2 and $\mathcal{PW}^2_2$ are incomparable.

Recall from Construction 3.12, and its application to the proof of Proposition 7.17, that two heads are enough for a plane-walking automaton to simulate any arithmetical program on a sparse subshift. It is known that two-counter machines (which are basically equivalent to arithmetical programs by [Sch72]) cannot compute all recursive functions, and in particular cannot recognize the set of prime numbers [IT93]. A natural candidate for realizing Conjecture 8.6 in the one-dimensional case would thus be the subshift $X \subset X^1_2$ where the distance of the two 1's cannot be a prime number.

However, instead of simply simulating an arithmetical program, the automaton may use the position of the rightmost 1 in the middle of the computation, and a priori compute something an ordinary arithmetical program cannot. In some sense it thus simulates an arithmetical program that remembers its input. Conversely, we also believe that a run of a 2PWDFA on a 2-sparse subshift can be simulated by such a machine. All currently known proof techniques for limitations of two-counter machines break down if one is allowed to remember the input value, which raises the following question.

Question 8.7. Can arithmetical programs (or two-counter machines) that remember their input (for example, in the sense that they can check whether the current counter value is greater than the input) recognize all recursively enumerable sets? In particular, can they recognize the set of prime numbers?

Other tools for separating classes of multihead automata are diagonalization, where an automaton with much more than k heads can analyze the behavior of one with k heads, and choose to act differently from it on some inputs, and computability arguments, where algorithms of certain complexity can only be computed by machines with enough heads. These techniques have been succesfully applied in the past to classes of finite structures defined by different types of automata, but they do not extend to the infinite case, or at least d-dimensional lattices. In particular, they cannot separate $\mathcal{PW}_2^d$ from $\mathcal{PW}_3^d$, since both are capable of universal computation.

Finally, we remark that the method of simulating arithmetical programs with finite automata, Construction 3.12, was also used in Chapter 4 to prove Theorem 4.14 and Corollary 4.18, both of which give strict upper bounds for the possible Cantor-Bendixson ranks of certain countable SFTs; the former characterizes the set of ranks of countable SFTs, while the latter gives the minimal rank of a countable SFT with an uncomputable configuration. The corresponding lower bounds have been proved in the preprint [BJ13], and close upper bounds had previously been obtained in [BDJ08] and [ST13], first using Turing machines and then counter machines. The use of arithmetical programs in Construction 3.12 seems to be close to optimal in the sense of minimizing the Cantor-Bendixson rank and the size of the subpattern poset of the resulting countable SFT, while being able to simulate arbitrarily complex computation. We believe that it can be applied to other problems of a similar nature, that is, to construct in some precise sense the simplest possible SFT that satisfies a given condition, like containing an uncomputable configuration.

8.4 Collapsing Hierarchies

The main results of Chapter 5 and Chapter 7 prove the collapse of certain hierarchies of multidimensional subshifts: one definable by u-MSO formulas in the former case, and those recognized by multi-headed finite automata in the latter. The results show that in this respect, subshifts are not at all similar to languages of finite pictures, since in the latter context, all analogous hierarchies tend to be infinite, and usually even strict. The crucial difference here is infinite space: As mentioned above, the main general tools for proving the infinity of hierarchies of multi-headed automata are diagonalization and computability arguments, and they can be used also in the context of logical formulas. However, the reason that diagonalization works is that it is possible for an automaton to exhaustively simulate another, which is no longer true in the presence of infinite resources; this is of course the intuitive message of the Halting Problem. Computability arguments also break down when all sufficiently high classes of the hierarchy are capable of universal computation. However, they can still be used to separate the lower levels of the hierarchies, as we did with $\mathcal{PW}_1^d$ and $\mathcal{PW}_2^d$.

On the other hand, the MSO hierarchy is known to collapse to the second level, the analogue of $\bar{\Sigma}_2$, in the case of languages of finite words [Büc60]. However, the aforementioned class is exactly that of regular languages, which is the prototypical example of a nontrivial but computationally simple family of formal languages. In view of this fact, it could be said that the collapse of the hierarchies in the case of finite words and infinite multidimensional configurations happens for very different reasons: in the former case, the geometry of the space is so simple that MSO logic is unable to make much use of it, whereas infinite configurations provide so great a freedom that all possible logical structures can be simulated 'in software' already with three quantifier alternations.

8.5 Beyond Integer Lattices

In this dissertation, we have studied subshifts defined on the d-dimensional lattices $\mathbb{Z}^d$, and mostly the case $d = 2$. However, as mentioned briefly in the beginning of Chapter 7, the theory of multidimensional symbolic dynamics can be generalized to cover more general classes of groups. We repeat the definition for convenience: for a group G, a *G-subshift over an alphabet A* is a subset of A^G which is closed in the product topology, and invariant under the translation action of G. The elements of A^G can be visualized as labelings of the Cayley graph of G with labels drawn from the alphabet A. This generalization is not new: after Robinson presented his simplified proof for the undecidability of the tiling problem on the plane in the 1971 paper [Rob71], he proved a weaker version of the result on the hyperbolic plane in [Rob78]. The undecidability of the unrestricted tiling problem on the hyperbolic plane was finally proved independently in [Mar08] and [Kar08]. The tiling problem has also been considered in the context of more exotic groups; see for example [AK13]. A very fruitful branch of symbolic dynamics on arbitrary groups considers cellular automata on the full shifts A^G. See [CSC09] for a comprehensive presentation of the topic. For example, the important class of *amenable groups* has been characterized in terms of the surjectivity and injectivety properties of cellular automata on the corresponding full group shifts [CSMS99, Bar10] (presented as Corollary 5.12.2 in [CSC09]).

In the recent preprint [ABS14], a class of G-subshifts called *G-effective subshifts* was defined for a finitely generated group G. Intuitively, a subshift $X \subset A^G$ is G-effective if there exists a Turing machine that walks on the Cayley graph of G along the finitely many generators, and recognizes a set of forbidden patterns for X. The similarity with the concept of plane-walking automata is immediate. An article on the generalization of plane-walking automata to subshifts on arbitrary finitely generated groups, tentatively named *group-walking automata*, is currently in the making by the author and Ville Salo. Some of the results from Chapter 7 carry over to group-walking automata as such, but there are also some major differencies. First, if G is a *torsion group* (meaning that every element generates a finite subgroup), then no number of heads is enough to recognize a nontrivial sparse subshift. This surprising result is due to the fact that the automaton will eventually walk in one of finitely many 'directions', and every direction in G leads to a finite loop, so it is impossible for the automaton to search the world, to use the terminology of [DM02]. Second, if the group G is not torsion, but has an undecidable word problem, then we can only prove that the hierarcy of automata collapses to the fourth level, instead of the third. The proof is similar to that of Theorem 7.23, except that the fourth head is used for probing the geometry of the group, which cannot be deduced by the simulated arithmetical program. More exotic groups may give rise to even stranger phenomena; for example, we do not currently know whether the hierarchy given by the number of heads collapses on any infinite torsion group.

Bibliography

[ABS14] Nathalie Aubrun, Sebastián Barbieri, and Mathieu Sablik. A notion of effectiveness for subshifts on finitely generated groups. *ArXiv e-prints*, December 2014.

[AK13] Nathalie Aubrun and Jarkko Kari. Tiling problems on Baumslag-Solitar groups. In Turlough Neary and Matthew Cook, editors, *Proceedings Machines, Computations and Universality 2013*, volume 128 of *Electronic Proceedings of Theoretical Computer Science*, page 35–46. Springer, 2013.

[AS13] Nathalie Aubrun and Mathieu Sablik. Simulation of effective subshifts by two-dimensional subshifts of finite type. *Acta Appl. Math.*, 126(1):35–63, August 2013.

[Bar10] Laurent Bartholdi. Gardens of Eden and amenability on cellular automata. *J. Eur. Math. Soc. (JEMS)*, 12(1):241–248, 2010.

[Bax99] R. J. Baxter. Planar lattice gases with nearest-neighbor exclusion. *Ann. Comb.*, 3(2-4):191–203, 1999. On combinatorics and statistical mechanics.

[BC08] Mikołaj Bojańczyk and Thomas Colcombet. Tree-walking automata do not recognize all regular languages. *SIAM J. Comput.*, 38(2):658–701, 2008.

[BDJ08] Alexis Ballier, Bruno Durand, and Emmanuel Jeandel. Structural aspects of tilings. In Pascal Weil Susanne Albers, editor, *Proceedings of the 25th Annual Symposium on the Theoretical Aspects of Computer Science*, pages 61–72, Bordeaux, France, February 2008. IBFI Schloss Dagstuhl. 11 pages.

[Ber66] Robert Berger. The undecidability of the domino problem. *Mem. Amer. Math. Soc. No.*, 66, 1966. 72 pages.

[BH67] Manuel Blum and Carl Hewitt. Automata on a 2-dimensional tape. In *Switching and Automata Theory, 1967. SWAT 1967.*

IEEE Conference Record of the Eighth Annual Symposium on, pages 155 –160, oct. 1967.

[BJ13] Alexis Ballier and Emmanuel Jeandel. Structuring multi-dimensional subshifts. *ArXiv e-prints*, September 2013.

[BL97] Mike Boyle and Douglas Lind. Expansive subdynamics, 1997.

[Boj08] Mikołaj Bojańczyk. Tree-walking automata. In Carlos Martín-Vide, Friedrich Otto, and Henning Fernau, editors, *LATA*, volume 5196 of *Lecture Notes in Computer Science*, pages 1–2. Springer, 2008.

[BPS10] Mike Boyle, Ronnie Pavlov, and Michael Schraudner. Multi-dimensional sofic shifts without separation and their factors. *Trans. Amer. Math. Soc.*, 362(9):4617–4653, 2010.

[BSSS06] Mikołaj Bojańczyk, Mathias Samuelides, Thomas Schwentick, and Luc Segoufin. Expressive power of pebble automata. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, volume 4051 of *Lecture Notes in Computer Science*, pages 157–168. Springer Berlin Heidelberg, 2006.

[Büc60] J Richard Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960.

[CCS+86] Douglas Cenzer, Peter Clote, Rick L. Smith, Robert I. Soare, and Stanley S. Wainer. Members of countable Π^0_1 classes. *Ann. Pure Appl. Logic*, 31(2-3):145–163, 1986. Special issue: second Southeast Asian logic conference (Bangkok, 1984).

[CDJS93] Douglas Cenzer, Rodney Downey, Carl Jockusch, and Richard A. Shore. Countable thin Π^0_1 classes. *Ann. Pure Appl. Logic*, 59(2):79–139, 1993.

[CDTW12] Douglas Cenzer, Ali Dashti, Ferit Toska, and Sebastian Wyman. Computability of countable subshifts in one dimension. *Theory of Computing Systems*, 51(3):352–371, 2012.

[CMNW02] Jorge Campello de Souza, Brian H. Marcus, Richard New, and Bruce A. Wilson. Constrained systems with unconstrained positions. *IEEE Trans. Inform. Theory*, 48(4):866–879, 2002.

[CR98] D. Cenzer and J. B. Remmel. Π^0_1 classes in mathematics. In *Handbook of recursive mathematics, Vol. 2*, volume 139 of *Stud. Logic Found. Math.*, pages 623–821. North-Holland, Amsterdam, 1998.

[CSC09] Tullio Ceccherini-Silberstein and Michel Coornaert. *Cellular Automata and Groups*. Springer, 2009.

[CSMS99] Tullio G. Ceccherini-Silberstein, Antonio Machi, and Fabio Scarabotti. Amenable groups and cellular automata. *Annales de l'institut Fourier*, 49(2):673–685, 1999.

[Cul96] Karel Culik, II. An aperiodic set of 13 wang tiles. *Discrete Mathematics*, 160(1–3):245 – 251, 1996.

[Des06] Angela Desai. Subsystem entropy for $\mathbb{Z}^d$ sofic shifts. *Indag. Math. (N.S.)*, 17(3):353–359, 2006.

[DFW13] A. Dennunzio, E. Formenti, and M. Weiss. Multidimensional Cellular Automata: closing property, quasi-expansivity and (un)decidability issues. submitted to Theoretical Computer Science, 2013.

[DM02] Marianne Delorme and Jacques Mazoyer. Pebble automata. Figures families recognition and universality. *Fund. Inform.*, 52(1-3):81–132, 2002. Special issue on cellular automata.

[DRS10] Bruno Durand, Andrei Romashchenko, and Alexander Shen. Effective closed subshifts in 1D can be implemented in 2D. In *Fields of logic and computation*, volume 6300 of *Lecture Notes in Comput. Sci.*, pages 208–226. Springer, Berlin, 2010.

[DRS12] Bruno Durand, Andrei Romashchenko, and Alexander Shen. Fixed-point tile sets and their applications. *J. Comput. System Sci.*, 78(3):731–764, 2012.

[Dur99] Bruno Durand. Tilings and quasiperiodicity. *Theoret. Comput. Sci.*, 221(1-2):61–75, 1999. ICALP '97 (Bologna).

[FW65] N. J. Fine and H. S. Wilf. Uniqueness theorems for periodic functions. *Proc. Amer. Math. Soc.*, 16:109–114, 1965.

[Gui12] Pierre Guillon. Projective subdynamics and universal shifts. In *Automata 2011—17th International Workshop on Cellular Automata and Discrete Complex Systems*, Discrete Math. Theor. Comput. Sci. Proc., AP, pages 123–133. Assoc. Discrete Math. Theor. Comput. Sci., Nancy, 2012.

[GVR97] Dora Giammarresi, Foscari Venezia, and Antonio Restivo. Two-dimensional languages, 1997.

[Had98] Jacques Hadamard. Les surfaces à courbures opposées et leurs lignes géodésiques. *Journal de Mathématiques Pures et appliquées*, 4:27–74, 1898.

[Hed69] Gustav A. Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Math. Systems Theory*, 3:320–375, 1969.

[HKC92] Lyman P. Hurd, Jarkko Kari, and Karel Culik. The topological entropy of cellular automata is uncomputable. *Ergodic Theory Dynam. Systems*, 12(2):255–265, 1992.

[HKM09] Markus Holzer, Martin Kutrib, and Andreas Malcher. Multi-Head Finite Automata: Characterizations, Concepts and Open Problems. *ArXiv e-prints*, June 2009.

[HM10] Michael Hochman and Tom Meyerovitch. A characterization of the entropies of multidimensional shifts of finite type. *Ann. of Math. (2)*, 171(3):2011–2038, 2010.

[Hoc09] Michael Hochman. On the dynamics and recursive properties of multidimensional symbolic systems. *Invent. Math.*, 176(1):131–167, 2009.

[HY75] Pei Hsia and Raymond T. Yeh. Marker automata. *Information Sciences*, 8(1):71 – 88, 1975.

[IT91] Katsushi Inoue and Itsuo Takanami. A survey of two-dimensional automata theory. *Information Sciences*, 55(1–3):99 – 121, 1991.

[IT93] Oscar H. Ibarra and Nicholas Q. Trân. A note on simple programs with two variables. *Theor. Comput. Sci.*, 112(2):391–397, May 1993.

[JM69] C. G. Jockusch, Jr. and T. G. McLaughlin. Countable retracing functions and Π^0_2 predicates. *Pacific J. Math.*, 30:67–93, 1969.

[JR15] Emmanuel Jeandel and Michael Rao. Personal communication, 2015.

[JT09] Emmanuel Jeandel and Guillaume Theyssier. Subshifts, languages and logic. In *Developments in language theory*, volume 5583 of *Lecture Notes in Comput. Sci.*, pages 288–299. Springer, Berlin, 2009.

[JT13] Emmanuel Jeandel and Guillaume Theyssier. Subshifts as models for MSO logic. *Inform. and Comput.*, 225:1–15, 2013.

[JV11] Emmanuel Jeandel and Pascal Vanier. Π_1^0 sets and tilings. In *Theory and Applications of Models of Computation (TAMC)*, volume 6648 of *Lecture Notes in Computer Science*, pages 230–239, 2011.

[Kar96] Jarkko Kari. A small aperiodic set of Wang tiles. *Discrete Math.*, 160(1-3):259–264, 1996.

[Kar08] Jarkko Kari. On the undecidability of the tiling problem. In Viliam Geffert, Juhani Karhumäki, Alberto Bertoni, Bart Preneel, Pavol Návrat, and Mária Bieliková, editors, *SOFSEM 2008: Theory and Practice of Computer Science*, volume 4910 of *Lecture Notes in Computer Science*, pages 74–82. Springer Berlin Heidelberg, 2008.

[Kit98] Bruce P. Kitchens. *Symbolic dynamics – One-sided, two-sided and countable state Markov shifts.* Universitext. Springer-Verlag, Berlin, 1998.

[KM01] Jarkko Kari and Cristopher Moore. New results on alternating and non-deterministic two-dimensional finite-state automata. In *STACS 2001 (Dresden)*, volume 2010 of *Lecture Notes in Comput. Sci.*, pages 396–406. Springer, Berlin, 2001.

[KM13] Steve Kass and Kathleen Madden. A sufficient condition for non-soficness of higher-dimensional subshifts. *Proc. Amer. Math. Soc.*, 141(11):3803–3816, 2013.

[KS11] Jarkko Kari and Ville Salo. A survey on picture-walking automata. In Werner Kuich and George Rahonis, editors, *Algebraic Foundations in Computer Science*, pages 183–213. Springer-Verlag, Berlin, Heidelberg, 2011.

[KSW60] G. Kreisel, J. Shoenfield, and Hao Wang. Number theoretic concepts and recursive well-orderings. *Arch. Math. Logik Grundlagenforsch.*, 5:42–64, 1960.

[Kůr03] Petr Kůrka. *Topological and symbolic dynamics*, volume 11 of *Cours Spécialisés [Specialized Courses].* Société Mathématique de France, Paris, 2003.

[Lin84] Douglas A. Lind. The entropies of topological Markov shifts and a related class of algebraic integers. *Ergodic Theory Dynam. Systems*, 4(2):283–300, 1984.

[LM95] Douglas Lind and Brian Marcus. *An introduction to symbolic dynamics and coding.* Cambridge University Press, Cambridge, 1995.

[LMP13] Erez Louidor, Brian Marcus, and Ronnie Pavlov. Independence entropy of $\mathbb{Z}^d$-shift spaces. *Acta Applicandae Mathematicae*, pages 1–21, 2013.

[LN99] Leonid Libkin and Juha Nurmonen. Counting and locality over finite structures: a survey. In *Generalized quantifiers and computation (Aix-en-Provence, 1997)*, volume 1754 of *Lecture Notes in Comput. Sci.*, pages 18–50. Springer, Berlin, 1999.

[Mar08] Maurice Margenstern. The domino problem of the hyperbolic plane is undecidable. *Theoretical Computer Science*, 407(1–3):29 – 84, 2008.

[MH38] Marston Morse and Gustav A. Hedlund. Symbolic dynamics. *American Journal of Mathematics*, 60(4):pp. 815–866, 1938.

[MH40] Marston Morse and Gustav A. Hedlund. Symbolic dynamics ii. sturmian trajectories. *American Journal of Mathematics*, 62(1):pp. 1–42, 1940.

[Min67] Marvin L. Minsky. *Computation: finite and infinite machines.* Prentice-Hall Inc., Englewood Cliffs, N.J., 1967. Prentice-Hall Series in Automatic Computation.

[Mor96] Kenichi Morita. Universality of a reversible two-counter machine. *Theoretical Computer Science*, 1996.

[Moz89] Shahar Mozes. Tilings, substitution systems and dynamical systems generated by them. *J. Analyse Math.*, 53:139–186, 1989.

[MP11] Tom Meyerovitch and Ronnie Pavlov. On independence and entropy for high-dimensional isotropic subshifts. *ArXiv e-prints*, December 2011.

[MS08] Oliver Matz and Nicole Schweikardt. Expressive power of monadic logics on words, trees, pictures, and graphs. In Jörg Flum, Erich Grädel, and Thomas Wilke, editors, *Logic and Automata*, volume 2 of *Texts in Logic and Games*, pages 531–552. Amsterdam University Press, 2008.

[MT97] Oliver Matz and Wolfgang Thomas. The monadic quantifier alternation hierarchy over graphs is infinite. In *In Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 236–244. IEEE, 1997.

[Odi89] Piergiorgio Odifreddi. *Classical recursion theory*, volume 125 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1989. The theory of functions and sets of natural numbers, With a foreword by G. E. Sacks.

[OP15] Nic Ormes and Ronnie Pavlov. Extender sets and multidimensional subshifts. *Ergodic Theory and Dynamical Systems*, FirstView:1–16, 1 2015.

[Pav12] Ronnie Pavlov. Approximating the hard square entropy constant with probabilistic methods. *Ann. Probab.*, 40(6):2362–2399, 2012.

[Pav13] Ronnie Pavlov. A class of nonsofic multidimensional shift spaces. *Proc. Amer. Math. Soc.*, 141(3):987–996, 2013.

[PCM06] T. Lei Poo, Panu Chaichanavong, and Brian H. Marcus. B.h.: Trade-off functions for constrained systems with unconstrained positions. *IEEE Trans. Inf. Theory*, pages 1425–1449, 2006.

[PS14] Ronnie Pavlov and Michael Schraudner. Classification of sofic projective subdynamics of multidimensional shifts of finite type. To appear in Transactions of the American Mathematical Society. Published online November 4, 2014, 2014.

[Rob71] Raphael M. Robinson. Undecidability and nonperiodicity for tilings of the plane. *Invent. Math.*, 12:177–209, 1971.

[Rob78] Raphael M. Robinson. Undecidable tiling problems in the hyperbolic plane. *Inventiones mathematicae*, 44(3):259–264, 1978.

[Ros87] A. Rosenthal. Strictly ergodic models and topological mixing for $\mathbb{Z}^2$-action. *Israel Journal of Mathematics*, 60(1):31–38, 1987.

[Sac90] G. E. Sacks. *Higher recursion theory.* Perspectives in mathematical logic. Springer-Verlag, 1990.

[Sch72] Rich Schroeppel. A two counter machine cannot calculate 2^N, 1972. Massachusetts Institute of Technology A.I. Memo 257.

[Sch98] Nicole Schweikardt. The monadic quantifier alternation hierarchy over grids and pictures. In *Computer science logic (Aarhus, 1997)*, volume 1414 of *Lecture Notes in Comput. Sci.*, pages 441–460. Springer, Berlin, 1998.

[ST12a] Ville Salo and Ilkka Törmä. Computational aspects of cellular automata on countable sofic shifts. *Mathematical Foundations of Computer Science 2012*, pages 777–788, 2012.

[ST12b] Ville Salo and Ilkka Törmä. On Derivatives and Subpattern Orders of Countable Subshifts. *ArXiv e-prints*, August 2012.

[ST13] Ville Salo and Ilkka Törmä. Constructions with countable subshifts of finite type. *Fundamenta Informaticae*, 126(2-3):263–300, 2013.

[ST14] Ville Salo and Ilkka Törmä. Plane-Walking Automata. *ArXiv e-prints*, August 2014. Presented in a shortened form at AUTOMATA 2014.

[Tör14a] Ilkka Törmä. Quantifier Extensions of Multidimensional Sofic Shifts. *ArXiv e-prints*, January 2014. To appear in Proceedings of the AMS.

[Tör14b] Ilkka Törmä. Subshifts, MSO logic, and collapsing hierarchies. In Josep Diaz, Ivan Lanese, and Davide Sangiorgi, editors, *Theoretical Computer Science*, volume 8705 of *Lecture Notes in Computer Science*, pages 111–122. Springer Berlin Heidelberg, 2014.

[Wan61] Hao Wang. Proving theorems by pattern recognition II. *Bell System Technical Journal*, 40:1–42, 1961.

[Wei85] Benjamin Weiss. Strictly ergodic models for dynamical systems. *Bulletin (New Series) of the American Mathematical Society*, 13(2):143–146, 10 1985.

Turku Centre for Computer Science
TUCS Dissertations

1. **Marjo Lipponen**, On Primitive Solutions of the Post Correspondence Problem
2. **Timo Käkölä**, Dual Information Systems in Hyperknowledge Organizations
3. **Ville Leppänen**, Studies on the Realization of PRAM
4. **Cunsheng Ding**, Cryptographic Counter Generators
5. **Sami Viitanen**, Some New Global Optimization Algorithms
6. **Tapio Salakoski**, Representative Classification of Protein Structures
7. **Thomas Långbacka**, An Interactive Environment Supporting the Development of Formally Correct Programs
8. **Thomas Finne**, A Decision Support System for Improving Information Security
9. **Valeria Mihalache**, Cooperation, Communication, Control. Investigations on Grammar Systems.
10. **Marina Waldén**, Formal Reasoning About Distributed Algorithms
11. **Tero Laihonen**, Estimates on the Covering Radius When the Dual Distance is Known
12. **Lucian Ilie**, Decision Problems on Orders of Words
13. **Jukkapekka Hekanaho**, An Evolutionary Approach to Concept Learning
14. **Jouni Järvinen**, Knowledge Representation and Rough Sets
15. **Tomi Pasanen**, In-Place Algorithms for Sorting Problems
16. **Mika Johnsson**, Operational and Tactical Level Optimization in Printed Circuit Board Assembly
17. **Mats Aspnäs**, Multiprocessor Architecture and Programming: The Hathi-2 System
18. **Anna Mikhajlova**, Ensuring Correctness of Object and Component Systems
19. **Vesa Torvinen**, Construction and Evaluation of the Labour Game Method
20. **Jorma Boberg**, Cluster Analysis. A Mathematical Approach with Applications to Protein Structures
21. **Leonid Mikhajlov**, Software Reuse Mechanisms and Techniques: Safety Versus Flexibility
22. **Timo Kaukoranta**, Iterative and Hierarchical Methods for Codebook Generation in Vector Quantization
23. **Gábor Magyar**, On Solution Approaches for Some Industrially Motivated Combinatorial Optimization Problems
24. **Linas Laibinis**, Mechanised Formal Reasoning About Modular Programs
25. **Shuhua Liu**, Improving Executive Support in Strategic Scanning with Software Agent Systems
26. **Jaakko Järvi**, New Techniques in Generic Programming – C++ is more Intentional than Intended
27. **Jan-Christian Lehtinen**, Reproducing Kernel Splines in the Analysis of Medical Data
28. **Martin Büchi**, Safe Language Mechanisms for Modularization and Concurrency
29. **Elena Troubitsyna**, Stepwise Development of Dependable Systems
30. **Janne Näppi**, Computer-Assisted Diagnosis of Breast Calcifications
31. **Jianming Liang**, Dynamic Chest Images Analysis
32. **Tiberiu Seceleanu**, Systematic Design of Synchronous Digital Circuits
33. **Tero Aittokallio**, Characterization and Modelling of the Cardiorespiratory System in Sleep-Disordered Breathing
34. **Ivan Porres**, Modeling and Analyzing Software Behavior in UML
35. **Mauno Rönkkö**, Stepwise Development of Hybrid Systems
36. **Jouni Smed**, Production Planning in Printed Circuit Board Assembly
37. **Vesa Halava**, The Post Correspondence Problem for Market Morphisms
38. **Ion Petre**, Commutation Problems on Sets of Words and Formal Power Series
39. **Vladimir Kvassov**, Information Technology and the Productivity of Managerial Work
40. **Frank Tétard**, Managers, Fragmentation of Working Time, and Information Systems

41. **Jan Manuch**, Defect Theorems and Infinite Words
42. **Kalle Ranto**, Z_4-Goethals Codes, Decoding and Designs
43. **Arto Lepistö**, On Relations Between Local and Global Periodicity
44. **Mika Hirvensalo**, Studies on Boolean Functions Related to Quantum Computing
45. **Pentti Virtanen**, Measuring and Improving Component-Based Software Development
46. **Adekunle Okunoye**, Knowledge Management and Global Diversity – A Framework to Support Organisations in Developing Countries
47. **Antonina Kloptchenko**, Text Mining Based on the Prototype Matching Method
48. **Juha Kivijärvi**, Optimization Methods for Clustering
49. **Rimvydas Rukšėnas**, Formal Development of Concurrent Components
50. **Dirk Nowotka**, Periodicity and Unbordered Factors of Words
51. **Attila Gyenesei**, Discovering Frequent Fuzzy Patterns in Relations of Quantitative Attributes
52. **Petteri Kaitovaara**, Packaging of IT Services – Conceptual and Empirical Studies
53. **Petri Rosendahl**, Niho Type Cross-Correlation Functions and Related Equations
54. **Péter Majlender**, A Normative Approach to Possibility Theory and Soft Decision Support
55. **Seppo Virtanen**, A Framework for Rapid Design and Evaluation of Protocol Processors
56. **Tomas Eklund**, The Self-Organizing Map in Financial Benchmarking
57. **Mikael Collan**, Giga-Investments: Modelling the Valuation of Very Large Industrial Real Investments
58. **Dag Björklund**, A Kernel Language for Unified Code Synthesis
59. **Shengnan Han**, Understanding User Adoption of Mobile Technology: Focusing on Physicians in Finland
60. **Irina Georgescu**, Rational Choice and Revealed Preference: A Fuzzy Approach
61. **Ping Yan**, Limit Cycles for Generalized Liénard-Type and Lotka-Volterra Systems
62. **Joonas Lehtinen**, Coding of Wavelet-Transformed Images
63. **Tommi Meskanen,** On the NTRU Cryptosystem
64. **Saeed Salehi**, Varieties of Tree Languages
65. **Jukka Arvo**, Efficient Algorithms for Hardware-Accelerated Shadow Computation
66. **Mika Hirvikorpi**, On the Tactical Level Production Planning in Flexible Manufacturing Systems
67. **Adrian Costea**, Computational Intelligence Methods for Quantitative Data Mining
68. **Cristina Seceleanu**, A Methodology for Constructing Correct Reactive Systems
69. **Luigia Petre**, Modeling with Action Systems
70. **Lu Yan**, Systematic Design of Ubiquitous Systems
71. **Mehran Gomari**, On the Generalization Ability of Bayesian Neural Networks
72. **Ville Harkke**, Knowledge Freedom for Medical Professionals – An Evaluation Study of a Mobile Information System for Physicians in Finland
73. **Marius Cosmin Codrea**, Pattern Analysis of Chlorophyll Fluorescence Signals
74. **Aiying Rong**, Cogeneration Planning Under the Deregulated Power Market and Emissions Trading Scheme
75. **Chihab BenMoussa**, Supporting the Sales Force through Mobile Information and Communication Technologies: Focusing on the Pharmaceutical Sales Force
76. **Jussi Salmi**, Improving Data Analysis in Proteomics
77. **Orieta Celiku**, Mechanized Reasoning for Dually-Nondeterministic and Probabilistic Programs
78. **Kaj-Mikael Björk**, Supply Chain Efficiency with Some Forest Industry Improvements
79. **Viorel Preoteasa**, Program Variables – The Core of Mechanical Reasoning about Imperative Programs
80. **Jonne Poikonen**, Absolute Value Extraction and Order Statistic Filtering for a Mixed-Mode Array Image Processor
81. **Luka Milovanov**, Agile Software Development in an Academic Environment
82. **Francisco Augusto Alcaraz Garcia**, Real Options, Default Risk and Soft Applications
83. **Kai K. Kimppa**, Problems with the Justification of Intellectual Property Rights in Relation to Software and Other Digitally Distributable Media
84. **Dragoş Truşcan**, Model Driven Development of Programmable Architectures
85. **Eugen Czeizler**, The Inverse Neighborhood Problem and Applications of Welch Sets in Automata Theory

86. **Sanna Ranto**, Identifying and Locating-Dominating Codes in Binary Hamming Spaces
87. **Tuomas Hakkarainen**, On the Computation of the Class Numbers of Real Abelian Fields
88. **Elena Czeizler**, Intricacies of Word Equations
89. **Marcus Alanen**, A Metamodeling Framework for Software Engineering
90. **Filip Ginter**, Towards Information Extraction in the Biomedical Domain: Methods and Resources
91. **Jarkko Paavola**, Signature Ensembles and Receiver Structures for Oversaturated Synchronous DS-CDMA Systems
92. **Arho Virkki**, The Human Respiratory System: Modelling, Analysis and Control
93. **Olli Luoma**, Efficient Methods for Storing and Querying XML Data with Relational Databases
94. **Dubravka Ilić**, Formal Reasoning about Dependability in Model-Driven Development
95. **Kim Solin**, Abstract Algebra of Program Refinement
96. **Tomi Westerlund**, Time Aware Modelling and Analysis of Systems-on-Chip
97. **Kalle Saari**, On the Frequency and Periodicity of Infinite Words
98. **Tomi Kärki**, Similarity Relations on Words: Relational Codes and Periods
99. **Markus M. Mäkelä**, Essays on Software Product Development: A Strategic Management Viewpoint
100. **Roope Vehkalahti**, Class Field Theoretic Methods in the Design of Lattice Signal Constellations
101. **Anne-Maria Ernvall-Hytönen**, On Short Exponential Sums Involving Fourier Coefficients of Holomorphic Cusp Forms
102. **Chang Li**, Parallelism and Complexity in Gene Assembly
103. **Tapio Pahikkala**, New Kernel Functions and Learning Methods for Text and Data Mining
104. **Denis Shestakov**, Search Interfaces on the Web: Querying and Characterizing
105. **Sampo Pyysalo**, A Dependency Parsing Approach to Biomedical Text Mining
106. **Anna Sell**, Mobile Digital Calendars in Knowledge Work
107. **Dorina Marghescu**, Evaluating Multidimensional Visualization Techniques in Data Mining Tasks
108. **Tero Säntti**, A Co-Processor Approach for Efficient Java Execution in Embedded Systems
109. **Kari Salonen**, Setup Optimization in High-Mix Surface Mount PCB Assembly
110. **Pontus Boström**, Formal Design and Verification of Systems Using Domain-Specific Languages
111. **Camilla J. Hollanti**, Order-Theoretic Mehtods for Space-Time Coding: Symmetric and Asymmetric Designs
112. **Heidi Himmanen**, On Transmission System Design for Wireless Broadcasting
113. **Sébastien Lafond**, Simulation of Embedded Systems for Energy Consumption Estimation
114. **Evgeni Tsivtsivadze**, Learning Preferences with Kernel-Based Methods
115. **Petri Salmela**, On Commutation and Conjugacy of Rational Languages and the Fixed Point Method
116. **Siamak Taati**, Conservation Laws in Cellular Automata
117. **Vladimir Rogojin**, Gene Assembly in Stichotrichous Ciliates: Elementary Operations, Parallelism and Computation
118. **Alexey Dudkov**, Chip and Signature Interleaving in DS CDMA Systems
119. **Janne Savela**, Role of Selected Spectral Attributes in the Perception of Synthetic Vowels
120. **Kristian Nybom**, Low-Density Parity-Check Codes for Wireless Datacast Networks
121. **Johanna Tuominen**, Formal Power Analysis of Systems-on-Chip
122. **Teijo Lehtonen**, On Fault Tolerance Methods for Networks-on-Chip
123. **Eeva Suvitie**, On Inner Products Involving Holomorphic Cusp Forms and Maass Forms
124. **Linda Mannila**, Teaching Mathematics and Programming – New Approaches with Empirical Evaluation
125. **Hanna Suominen**, Machine Learning and Clinical Text: Supporting Health Information Flow
126. **Tuomo Saarni**, Segmental Durations of Speech
127. **Johannes Eriksson**, Tool-Supported Invariant-Based Programming

128. **Tero Jokela**, Design and Analysis of Forward Error Control Coding and Signaling for Guaranteeing QoS in Wireless Broadcast Systems
129. **Ville Lukkarila**, On Undecidable Dynamical Properties of Reversible One-Dimensional Cellular Automata
130. **Qaisar Ahmad Malik**, Combining Model-Based Testing and Stepwise Formal Development
131. **Mikko-Jussi Laakso**, Promoting Programming Learning: Engagement, Automatic Assessment with Immediate Feedback in Visualizations
132. **Riikka Vuokko**, A Practice Perspective on Organizational Implementation of Information Technology
133. **Jeanette Heidenberg**, Towards Increased Productivity and Quality in Software Development Using Agile, Lean and Collaborative Approaches
134. **Yong Liu**, Solving the Puzzle of Mobile Learning Adoption
135. **Stina Ojala**, Towards an Integrative Information Society: Studies on Individuality in Speech and Sign
136. **Matteo Brunelli**, Some Advances in Mathematical Models for Preference Relations
137. **Ville Junnila**, On Identifying and Locating-Dominating Codes
138. **Andrzej Mizera**, Methods for Construction and Analysis of Computational Models in Systems Biology. Applications to the Modelling of the Heat Shock Response and the Self-Assembly of Intermediate Filaments.
139. **Csaba Ráduly-Baka**, Algorithmic Solutions for Combinatorial Problems in Resource Management of Manufacturing Environments
140. **Jari Kyngäs**, Solving Challenging Real-World Scheduling Problems
141. **Arho Suominen**, Notes on Emerging Technologies
142. **József Mezei**, A Quantitative View on Fuzzy Numbers
143. **Marta Olszewska**, On the Impact of Rigorous Approaches on the Quality of Development
144. **Antti Airola**, Kernel-Based Ranking: Methods for Learning and Performace Estimation
145. **Aleksi Saarela**, Word Equations and Related Topics: Independence, Decidability and Characterizations
146. **Lasse Bergroth**, Kahden merkkijonon pisimmän yhteisen alijonon ongelma ja sen ratkaiseminen
147. **Thomas Canhao Xu**, Hardware/Software Co-Design for Multicore Architectures
148. **Tuomas Mäkilä**, Software Development Process Modeling – Developers Perspective to Contemporary Modeling Techniques
149. **Shahrokh Nikou**, Opening the Black-Box of IT Artifacts: Looking into Mobile Service Characteristics and Individual Perception
150. **Alessandro Buoni**, Fraud Detection in the Banking Sector: A Multi-Agent Approach
151. **Mats Neovius**, Trustworthy Context Dependency in Ubiquitous Systems
152. **Fredrik Degerlund**, Scheduling of Guarded Command Based Models
153. **Amir-Mohammad Rahmani-Sane**, Exploration and Design of Power-Efficient Networked Many-Core Systems
154. **Ville Rantala**, On Dynamic Monitoring Methods for Networks-on-Chip
155. **Mikko Pelto**, On Identifying and Locating-Dominating Codes in the Infinite King Grid
156. **Anton Tarasyuk**, Formal Development and Quantitative Verification of Dependable Systems
157. **Muhammad Mohsin Saleemi**, Towards Combining Interactive Mobile TV and Smart Spaces: Architectures, Tools and Application Development
158. **Tommi J. M. Lehtinen**, Numbers and Languages
159. **Peter Sarlin**, Mapping Financial Stability
160. **Alexander Wei Yin**, On Energy Efficient Computing Platforms
161. **Mikołaj Olszewski**, Scaling Up Stepwise Feature Introduction to Construction of Large Software Systems
162. **Maryam Kamali**, Reusable Formal Architectures for Networked Systems
163. **Zhiyuan Yao**, Visual Customer Segmentation and Behavior Analysis – A SOM-Based Approach
164. **Timo Jolivet**, Combinatorics of Pisot Substitutions
165. **Rajeev Kumar Kanth**, Analysis and Life Cycle Assessment of Printed Antennas for Sustainable Wireless Systems
166. **Khalid Latif**, Design Space Exploration for MPSoC Architectures

167. **Bo Yang**, Towards Optimal Application Mapping for Energy-Efficient Many-Core Platforms
168. **Ali Hanzala Khan**, Consistency of UML Based Designs Using Ontology Reasoners
169. **Sonja Leskinen**, m-Equine: IS Support for the Horse Industry
170. **Fareed Ahmed Jokhio**, Video Transcoding in a Distributed Cloud Computing Environment
171. **Moazzam Fareed Niazi**, A Model-Based Development and Verification Framework for Distributed System-on-Chip Architecture
172. **Mari Huova**, Combinatorics on Words: New Aspects on Avoidability, Defect Effect, Equations and Palindromes
173. **Ville Timonen**, Scalable Algorithms for Height Field Illumination
174. **Henri Korvela**, Virtual Communities – A Virtual Treasure Trove for End-User Developers
175. **Kameswar Rao Vaddina**, Thermal-Aware Networked Many-Core Systems
176. **Janne Lahtiranta**, New and Emerging Challenges of the ICT-Mediated Health and Well-Being Services
177. **Irum Rauf**, Design and Validation of Stateful Composite RESTful Web Services
178. **Jari Björne**, Biomedical Event Extraction with Machine Learning
179. **Katri Haverinen**, Natural Language Processing Resources for Finnish: Corpus Development in the General and Clinical Domains
180. **Ville Salo**, Subshifts with Simple Cellular Automata
181. **Johan Ersfolk**, Scheduling Dynamic Dataflow Graphs
182. **Hongyan Liu**, On Advancing Business Intelligence in the Electricity Retail Market
183. **Adnan Ashraf**, Cost-Efficient Virtual Machine Management: Provisioning, Admission Control, and Consolidation
184. **Muhammad Nazrul Islam**, Design and Evaluation of Web Interface Signs to Improve Web Usability: A Semiotic Framework
185. **Johannes Tuikkala**, Algorithmic Techniques in Gene Expression Processing: From Imputation to Visualization
186. **Natalia Díaz Rodríguez**, Semantic and Fuzzy Modelling for Human Behaviour Recognition in Smart Spaces. A Case Study on Ambient Assisted Living
187. **Mikko Pänkäälä**, Potential and Challenges of Analog Reconfigurable Computation in Modern and Future CMOS
188. **Sami Hyrynsalmi**, Letters from the War of Ecosystems – An Analysis of Independent Software Vendors in Mobile Application Marketplaces
189. **Seppo Pulkkinen**, Efficient Optimization Algorithms for Nonlinear Data Analysis
190. **Sami Pyöttiälä**, Optimization and Measuring Techniques for Collect-and-Place Machines in Printed Circuit Board Industry
191. **Syed Mohammad Asad Hassan Jafri**, Virtual Runtime Application Partitions for Resource Management in Massively Parallel Architectures
192. **Toni Ernvall**, On Distributed Storage Codes
193. **Yuliya Prokhorova**, Rigorous Development of Safety-Critical Systems
194. **Olli Lahdenoja**, Local Binary Patterns in Focal-Plane Processing – Analysis and Applications
195. **Annika H. Holmbom**, Visual Analytics for Behavioral and Niche Market Segmentation
196. **Sergey Ostroumov**, Agent-Based Management System for Many-Core Platforms: Rigorous Design and Efficient Implementation
197. **Espen Suenson**, How Computer Programmers Work – Understanding Software Development in Practise
198. **Tuomas Poikela**, Readout Architectures for Hybrid Pixel Detector Readout Chips
199. **Bogdan Iancu**, Quantitative Refinement of Reaction-Based Biomodels
200. **Ilkka Törmä**, Structural and Computational Existence Results for Multidimensional Subshifts

Turku Centre *for* Computer Science

Joukahaisenkatu 3-5 B, 20520 Turku, Finland | www. tucs.fi

University of Turku
Faculty of Mathematics and Natural Sciences
- Department of Information Technology
- Department of Mathematics and Statistics

Turku School of Economics
- Institute of Information Systems Science

Åbo Akademi University
Faculty of Science and Engineering
- Computer Engineering
- Computer Science

Faculty of Social Sciences, Business and Economics
- Information Systems

ISBN 978-952-12-3237-4
ISSN 1239-1883

Ilkka Törmä

Structural and Computational Existence Results for Multidimensional Subshifts

Ilkka Törmä

Structural and Computational Existence Results for Multidimensional Subshifts