

2006

Recognizable languages defined by two-dimensional shift spaces

Joni Burnette Pirnot
University of South Florida

Follow this and additional works at: <http://scholarcommons.usf.edu/etd>

 Part of the [American Studies Commons](#)

Scholar Commons Citation

Pirnot, Joni Burnette, "Recognizable languages defined by two-dimensional shift spaces" (2006). *Graduate Theses and Dissertations*.
<http://scholarcommons.usf.edu/etd/2660>

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.

Recognizable Languages Defined by Two-dimensional Shift Spaces

by

Joni Burnette Pirnot

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Mathematics
College of Arts and Sciences
University of South Florida

Major Professor: Nataša Jonoska, Ph.D.
Committee Chair: Lilia Woods, Ph.D.
Gregory McColm, Ph.D.
Stephen Suen, Ph.D.
Masahico Saito, Ph.D.
Xiang-Dong Hou, Ph.D.

Date of approval:
October 24, 2006

Keywords: Recognizable picture languages, Dot systems,
Two-dimensional finite automata, Transitivity, Doubly periodic

© Copyright 2006, Joni Burnette Pirnot

DEDICATION

For Holly, who once told me to never waste talent.

TABLE OF CONTENTS

List of Figures	iii
Abstract	v
Preface	vi
1 Introduction	1
1.1 History and Overview	1
1.2 Notation and Terminology	5
1.3 Dot Systems	12
2 Finite Automata Recognizing Two-dimensional Shift Spaces	17
2.1 Recognition of Shifts of Finite Type by $\mathcal{M}_{F(X)}$	19
2.2 Recognition of Sofic Shifts by $\mathcal{M}_{F(X)}^\Phi$	28
2.3 Forbidden and Forced Structures in $\mathcal{M}_{F(X)}$ and $\mathcal{M}_{F(X)}^\Phi$	34
3 Transitivity	39
3.1 Factor Languages of Dot Systems Defined over $\mathbb{Z}/2\mathbb{Z}$	39
3.2 Graph Representations	54
4 Periodicity	64
4.1 Doubly Periodic Points in Two-dimensional Shift Spaces	65
4.2 Examples	69
5 Monoids and Follower Sets	85
5.1 Monoids for Dot Systems	86
5.2 Follower Sets and Predecessor Sets	100

Conclusion	106
References	108
About the Author	End Page

LIST OF FIGURES

1.1	Enclosure of two blocks	9
1.2	Different types of transitivity	11
1.3	Shapes within minimal rectangles	13
1.4	Extending a block upward	14
1.5	Extending an allowed block	16
2.1	Sample set of allowed blocks for an REC language	18
2.2	Horizontal and vertical k -concatenations	20
2.3	Full shift as vertex shift (a) and edge shift (b)	24
2.4	Graph representing the Three-dot System	24
2.5	Comparison of $q \rightarrow r \uparrow s$ and $q \uparrow r' \rightarrow s$	26
2.6	An example of a block path where $k = 2$	27
2.7	Graph $\mathcal{M}_{F(X)}$ representing the Diagonal-shift System	28
2.8	Set of five allowed blocks for subshift of Example 2.2.2	30
2.9	Shift space where any b is surrounded entirely by a 's	31
2.10	In $\mathcal{M}_{F(X)}^h$, graph diamonds of length 2 are forbidden.	35
2.11	Horizontal graph triangle of Corollary 2.3.6	38
3.1	The Wallpaper Pattern	41
3.2	The inductive step	46
3.3	The process of filling up the block B	49
3.4	Extension of a block into a triangle	51
3.5	Possible positions for intersection of right isosceles triangles	52
3.6	Full-square System	55
3.7	One-column extension having even sum	56
3.8	One-column extension having odd sum	57

3.9	A 4×3 block of 1's meets 4×3 block of 0's	57
3.10	Uniform transitivity in direction (u, v)	60
3.11	Blocks and block paths over 2×2 states	61
4.1	Point with least double period $(6, 4)$	65
4.2	Vertex shifts in one-dimensional case	70
4.3	Points of double period $(2, 1)$ and $(1, 2)$	71
4.4	Subgraphs representing point of double period $(2, 2)$	72
4.5	State amalgamation	75
4.6	Non-isomorphic graphs representing points of period $(3, 3)$	78
4.7	Amalgamation of two pairs of states	78
4.8	Subgraph showing forced labels	79
4.9	Amalgamation of three pairs of states	80
4.10	Amalgamation of states 1, 5, and 9 dictates other labels	80
4.11	Multiple amalgamations suggested by 1/5/9 amalgamation	81
4.12	Represented points have period $(1, n)$ for all $n \geq 3$	82
4.13	Represented points have period (n, n) for all $n \geq 4$	83
5.1	Horizontal translations of S	94
5.2	Strictly sofic shift with follower-separated graph	101
5.3	Graph of one-dimensional strictly sofic shift	103
5.4	Strictly sofic shift; graph not follower separated	104
5.5	Graph of reduced size recognizing Y	105

RECOGNIZABLE LANGUAGES DEFINED BY TWO-DIMENSIONAL SHIFT SPACES

JONI BURNETTE PIRNOT

ABSTRACT

There are numerous connections between the theory of formal languages and that of symbolic dynamics. In each, the transition from one dimension to two dimensions is accompanied by much difficulty due in large part to the emptiness problem, which is related to the presence (or lack thereof) of periodic points and is known to be undecidable. Here, we focus on two-dimensional languages that have the property that all blocks allowed by the language can be extended to a configuration of the plane satisfying the structure of the language; for such languages the emptiness problem is not an issue. We first show that dot systems may be associated with two-dimensional languages having this property, so that we might employ these languages as varied examples. We next define a new type of finite automaton and with it, a tool for recognizing two-dimensional “strings” of data. It is then shown that these automata correctly represent the sofic shift spaces that result from the application of block maps to shifts of finite type. Thereafter, these automata are utilized to investigate properties of transitivity in the two-dimensional languages that they represent. More specifically, new definitions for different types of two-dimensional transitivity are adapted from topological dynamics and then illustrated through the use of dot systems. The appearance of periodic points in the languages represented by these automata is also explored, with a main result being that the existence of a periodic point is guaranteed under certain conditions. Finally, issues of equivalence are introduced in the two-dimensional setting with regards to formal languages (syntactic monoids) and symbolic dynamics (the follower sets of a graph representing a sofic shift space).

PREFACE

It has been said that the sum of the parts is greater than the whole. Albeit difficult for a mathematician to condone such illogical rhetoric, I must acknowledge that in my case I find truth in this statement. The development of my cognitive skills, this dissertation, and my outlook on mathematics and life in general have been motivated, encouraged, and sustained by many people over the course of my life. I am grateful for the opportunity to thank some of those people here, although a few will never have the opportunity to read here what a lasting impression they have left on me.

From an early age, my mother, father, and sisters were kind enough to tolerate my bookishness and to overlook my lack of common sense. My mother planted a love of mathematics in my young heart when she explained percents, decimals, and fractions to me in a way that illustrated the beautiful intercomplexity of mathematical truths, and my father taught me that quiet moments deep in thought could be some of the most valuable times in a person's life. Dee and Leah have always had words of encouragement for me, Holly started me on an academic journey with words I will never forget, and Amy accompanied me on part of that journey as she studied our culture's most memorable words.

When I was a young student, Mike Frame first taught me how to do serious mathematics, and when I thought I had lost my way, Soo Bong Chae became my guide. It was he who propelled me on to graduate school. Only one other person was as adamant about my attending graduate school as Soo Bong, and that was John. I look forward to sharing the title of Dr. J with him.

When my graduate-school life came to a halt in order to make way for new life, my 103-year-old friend Muggie told me that "Life is what happens while you are busy making plans". Zena and Mona were born in 1990 and 1992, respectively, and they quickly became my *raison d'être*. I thank them both for the many important lessons they have taught me.

When I finally made my way back to USF after a five-year hiatus, I met the woman who would change my life. Dr. Nataša Jonoska assisted me academically and emotionally, she inspired me mathematically and professionally, and she supported me personally and financially. (I am grateful for funding I received through the grants CCF # 0432009 and EIA # 0086015.) I would never have attempted entering the Ph.D. program without her guidance. There are not enough words to thank her properly.

I would also like to thank the other members of my committee who devoted their time to the reading and editing of this dissertation. In particular, Dr. Gregory McColm has been an integral part of my education, and I thank him for his input and insight. Additionally, I thank Dr. Jarkko Kari, who on several occasions offered thoughtful dialogue with respect to symbolic dynamics and automata theory.

My work could not have been completed without the support of Manatee Community College. Dr. John Rosen, Dr. Mike Mears, Dr. Dennis Runde, and Altay Özgener have been especially supportive and helpful these past several years.

Finally, I wish to thank my psychiatrist, physician, bartender, banker, chauffeur, masseuse, chef, and best friend. The sum of these parts is one whole person - my husband, Steve Pirnot - and I thank him for the many, many parts he has played over the years.

1 INTRODUCTION

In this chapter, the required concepts from the fields of symbolic dynamics and formal language theory are introduced. First we outline a history of the research done in these fields in the two-dimensional case, and we give an overview of the main ideas and results found within this paper. Next we provide notation and terminology required for discussion of the two-dimensional case; in particular, we define several types of two-dimensional transitivity by modifying similar notions found in the study of topological dynamical systems. Finally we explain dot systems as they exist in the literature, and we then prove that any finite block allowed by the structure of a dot system also appears as a subblock of some configuration of the plane found within that dot system.

1.1 History and Overview

One way to study a changing (dynamical) system is to make the time discrete so as to study the iterates of a single action on the system. The field of symbolic dynamics takes this idea a step further by also making the space that represents the system discrete. The general idea is to use a finite set of states to record the action on the space by first assigning a symbol to each state and then keeping track of these states in discrete time steps by representation via an infinite sequence of symbols [25]. So a symbolic dynamical system studies a space of infinite sequences being acted upon by a translation map that shifts observation from one part of a sequence to another part of the sequence. Such a system is referred to as a shift space, and the infinite sequences are referred to as points of the shift space. One way to keep track of the action on certain shift spaces is to use a finite automaton, which is a directed graph with transitions between the states that represent the discrete time steps. This links the study of symbolic dynamics with that of formal languages, since recognizable languages are precisely those that can be represented by a finite automaton.

While many well-developed theories exist in one dimension for the study of symbolic dynamics [25], automata theory [21, 28], and formal languages [10, 22, 29], many of these results do not hold true in two dimensions. The transition from one to two dimensions is complicated by several factors, one of which is related to the possible non-existence of periodic points in a two-dimensional shift space. As an illustration, consider the following question posed by Hao Wang [31] in 1961: Is it decidable whether a finite set of equal-sized square tiles with colors on each edge can tile the plane in such a way that contiguous edges will always have the same color? Wang answered his own question in the affirmative (see, for example, [32]) by constructing an algorithm that hinged on the assumption that any set of tiles capable of tiling the plane would admit a periodic tiling. At the time, it seemed a reasonable assumption: the set of all possible tilings of the plane using a given set of Wang tiles defines a two-dimensional shift of finite type, and in the language of one-dimensional symbolic dynamics, a shift of finite type is nonempty if and only if it contains a periodic point [25]. However, in a 1966 publication, Robert Berger shows that Wang's solution is incorrect [2] by demonstrating the existence of a set of Wang tiles that can only tile the plane aperiodically. Wang's question has come to be known as the emptiness problem. It is now known [7] that the emptiness problem is equivalent to the halting problem for Turing machines and is therefore undecidable.

Efforts to investigate two-dimensional recognizable languages do so in the context of finite rectangular pictures (arrays of symbols) and non-rectangular shapes [1]. In [7], Giammarresi and Restivo introduce the class REC of recognizable picture languages as those that can be obtained by projection of a local picture language. (A local picture language over the alphabet Σ is defined as one that can be completely described by a set of allowed 2×2 tiles over $\Sigma \cup \{\#\}$, with $\#$ being a non-alphabet symbol placed around the border of each picture.) It is known that the class REC is not closed under complementation [8], which motivates the discussion of hierarchy within the family of two-dimensional languages. For example, Kari and Moore [18] show that languages recognized by 4-way alternating finite automata are incomparable to REC. It is demonstrated in [23] that every recognizable picture language can be obtained as the projection of an hv-local picture language. (In an hv-local picture language, the 2×2 tiles that describe the language are replaced by horizontal and vertical dominoes - 1×2 tiles and 2×1 tiles, respectively - so that horizontal and vertical reading of the pictures can be accomplished

separately.) This suggests representation of two-dimensional languages through the use of two separate graphs or matrices [14, 26]. However, the main drawback to having separate graphs for horizontal and vertical movement is that when a block map is applied to the graph representing the local language, the newly-labeled graph fails to correspond to the sofic language intended [5]: the inherent properties of the symbols that result from interlacing horizontal and vertical movement cannot be described. Other attempts to represent picture languages focus on the use of a particular kind of cellular automaton [11, 13]. A comprehensive survey of the research done in two-dimensional finite automata during the time period beginning with Blum and Hewitt in 1967 [4] and up to 1991 can be found in [12], while an excellent survey of more recent results can be found in [18].

The focus of this dissertation is on two-dimensional recognizable languages having the property that any finite picture allowed by the structure of the language may be infinitely extended to some configuration of the plane that also satisfies the structure of the language; that is, where the emptiness problem is solvable since the language is *prolongable*. Furthermore, we shall be interested in two-dimensional languages that are *factorial*; that is, languages having the property that for any block found in the language, all of its subblocks are also found in the language. In one dimension, there are well-developed theories concerning languages that are factorial, prolongable, and recognizable (FPR-languages). Here we develop a theory for two-dimensional factorial, prolongable, and recognizable languages (2DFPR-languages) which are the factor languages of certain two-dimensional shift spaces.

Of further interest in one dimension is the set of factorial, transitive, and recognizable languages (FTR-languages) that are a subset of the FPR-languages. For one-dimensional languages, there is only one notion of *transitivity*: given any two blocks found in the language, there exists a third block, also in the language, which contains the given blocks as subblocks. In Section 1.2 it is demonstrated that several different notions of transitivity exist for two-dimensional languages and that each defines an invariant property for two-dimensional shift spaces. “Dot systems”, which are initiated in [19], are described in Section 1.3 and it is then shown that these shift spaces belong to the class of 2DFPR-languages. Dot systems can therefore provide rich examples for the theory found in subsequent chapters.

In Section 2.1, we define a new type of automaton that is capable of recognizing

2DFPR-languages. By dispensing with the boundary symbol that forms the border of the pictures found in the two-dimensional languages belonging to the class REC, this new type of construction allows the automaton to generate two-dimensional shift spaces (and therefore, two-dimensional factor languages) in a way quite similar to the way in which one-dimensional shift spaces are generated. It is explained that the crucial component of the automaton's construction is a definition of acceptance that gives the automaton specific instructions regarding the dimensions and structure of blocks that are deemed to be recognizable. In Section 2.2, it is verified that this new type of automaton correctly represents the image of a shift space (of finite type) under a block code; no such graph representation with this capability exists in the literature. Throughout Chapter 2, the class of dot systems is employed to generate some manageable examples of two-dimensional shift spaces and their graph representations. Section 2.3 closes the chapter with several Propositions and Corollaries that will be of use in the discussion of periodicity that is found in Chapter 4.

Chapter 3 revisits the notion of different types of transitivity existing in two-dimensional languages. Most of the results on transitivity found within this chapter have already been published in [16]. In Section 3.1, as an illustration of the various types of two-dimensional transitivity, dot systems are partially categorized based on the shapes that define them. In Section 3.2, it is shown how the type of graph representation defined in Chapter 2 can reveal information regarding transitivity in the related factor languages. In particular, a main result of the chapter and of the dissertation is that for a 2DFPR-language, there is an algorithm that decides whether the given language exhibits a particular type of transitivity.

In Section 4.1, the main result from Chapter 3 is linked to the existence of periodic points under certain conditions. Section 4.2 then offers detailed examples of periodic points found in two-dimensional shift spaces with respect to the appearance of such points in corresponding graph representations.

Finally, issues of equivalence for the blocks of a two-dimensional language are discussed in Chapter 5. In Section 5.1 a monoid is defined based on equivalence classes for the blocks of a picture language, and then some partial results are achieved with respect to dot systems. A different approach is suggested in Section 5.2 by investigating the follower sets of blocks that act as the states of a graph representing a two-dimensional sofic shift

space. These final topics serve as an introduction to the many open questions that exist in the field of two-dimensional formal language theory with regards to symbolic dynamics and finite automata.

1.2 Notation and Terminology

For notation, terminology, and basic results of one-dimensional symbolic dynamical systems, see [25]. For notation, terminology, and basic results of one-dimensional formal language theory, see [10]. Some additional notation and terminology will be required for the discussion of the two-dimensional case.

Given a finite alphabet Σ , define the *two-dimensional full Σ -shift* to be $\Sigma^{\mathbb{Z}^2}$. A *point* $x \in \Sigma^{\mathbb{Z}^2}$ is a function $x : \mathbb{Z}^2 \rightarrow \Sigma$, that is, a configuration of the plane where the integer lattice \mathbb{Z}^2 has been populated with choices from the alphabet Σ . For $x \in \Sigma^{\mathbb{Z}^2}$ and $w \in \mathbb{Z}^2$, we will sometimes denote $x(w)$ as x_w and may refer to the coordinate point $w \in \mathbb{Z}^2$ as a cell. Similarly, for $x \in \Sigma^{\mathbb{Z}^2}$ and $R \subseteq \mathbb{Z}^2$, let x_R denote the restriction of x to R . We call R a region, and we call a finite region $S \subseteq \mathbb{Z}^2$ a *shape*. In particular, $[-j, j]^2$ is the square shape of size $2j + 1$ centered at the origin.

The set $\Sigma^{\mathbb{Z}^2}$ is a compact metric space under the metric $\rho(x, y) = 2^{-j}$, where for $x, y \in \Sigma^{\mathbb{Z}^2}$, j is the largest integer such that $x_{[-j, j]^2} = y_{[-j, j]^2}$. (When $x = y$, define $\rho(x, y) = 0$.) Informally, the closer two points are to each other, the larger the centered square shape on which they agree. For $v \in \mathbb{Z}^2$, define the *two-dimensional translation in direction v* as σ^v where σ^v is defined by $(\sigma^v(x))_w = x_{w+v}$. A subset $X \subseteq \Sigma^{\mathbb{Z}^2}$ is said to be translation invariant if for all $v \in \mathbb{Z}^2$, $\sigma^v(X) \subseteq X$. If $X \subseteq \Sigma^{\mathbb{Z}^2}$ is translation invariant and closed with respect to the metric ρ , we say that X is a *two-dimensional shift space* (or a *subshift* of the full shift).

We define a *design* γ on a shape S to be a function $\gamma : S \rightarrow \Sigma$, where the given shape S has been *normalized* so that $\min\{i : (i, j) \in S\} = 0$ and $\min\{j : (i, j) \in S\} = 0$. In other words, the shape has only non-negative integer coordinates with boundaries lying on the coordinate axes. The number of occurrences of the symbol $a \in \Sigma$ in a design γ

shall be denoted $|\gamma|_a$. If Γ is a set of designs on a fixed shape S , then the set

$$X := \{x \in \Sigma^{\mathbb{Z}^2} : \forall v \in \mathbb{Z}^2, (\sigma^v(x))_S \in \Gamma\} \quad (1.2.1)$$

is a two-dimensional shift space that is called a *two-dimensional shift of finite type*. For shifts of finite type defined through a finite set of several different shapes, there is no loss of generality in assuming that X is defined through a single rectangular shape having size sufficient to contain all other shapes.

Given a design γ on a rectangular shape $T \subset \mathbb{Z}^2$ having m rows and n columns, we call γ an $m \times n$ *block* and denote such designs by $B_{m,n}$. For ease of notation, we may sometimes drop the subscripts when the number of rows and columns is irrelevant, and we may refer to blocks as β_i when the index of the block does not refer to its dimension. We shall say that an $m \times n$ block has *height* m , *length* n , and *thickness* $k = \max\{m, n\}$. If $m = 0$ or $n = 0$, then $B_{m,n}$ is the *empty block* and is denoted by ε . For a design $B : T \rightarrow \Sigma$, a *subblock* B' of B is the restriction of the design to a rectangular subset $T' \subseteq T \subset \mathbb{Z}^2$. In such cases, we sometimes say that B *encloses* B' and denote this by $B' \subseteq B$. For fixed r and c , the set of all $r \times c$ subblocks of B is denoted as $F_{r,c}(B)$, and the set of all rectangular subblocks of B is denoted with $F(B)$. The set of all blocks of a fixed size $m \times n$ over Σ is denoted $\Sigma^{m,n}$, and the set of all blocks of any size over Σ is denoted by Σ^{**} .

A language L is any subset of a free monoid. (A monoid is a set with a binary associative operation and an identity.) A *picture language over* the alphabet Σ is defined to be a subset of Σ^{**} . In particular, a *local picture language* L is one where $B \in L$ if and only if $F_{k,k}(B) \subseteq Q$, where Q is a finite set of allowed $k \times k$ blocks. Also of interest will be those languages that are *factorial*: a language L is said to be factorial iff $L = F(L) := \{F(B) | B \in L\}$. All languages under discussion in this paper are *recognizable* languages: that is, languages that can be represented by a finite automaton. In particular, Giammarresi and Restivo [7] introduce the class REC of *recognizable picture languages* as those that can be obtained by the projection of a local picture language, where the local language is taken over the alphabet Σ and the set Q of allowed 2×2 tiles is taken over $\Sigma \cup \{\#\}$, with $\#$ being a non-alphabet symbol placed around the border of each rectangular picture.

There is a language associated to each shift space. We say a block $B : T \rightarrow \Sigma$ *occurs* in $X \subseteq \Sigma^{\mathbb{Z}^2}$ if there exists an $x \in X$ such that $x_T = B$. The *factor language* of a shift space X is

$$F(X) := \{F_{m,n}(x) : m, n \geq 0, x \in X\}, \quad (1.2.2)$$

i.e. the collection of all subblocks that occur in points of X . For shift spaces, the factor language of the shift space uniquely determines the shift space; that is, for two shift spaces X and Y , $X = Y$ if and only if $F(X) = F(Y)$ [25]. For this reason, when a graph represents the factor language of a shift space, we shall more generally refer to the graph as a representation of X .

Let X be a two-dimensional shift of finite type defined by a set of designs Γ on a normalized shape S . For the shape S , define the number of rows in S to be $r = 1 + \max\{j : (i, j) \in S\}$ and define the number of columns in S to be $c = 1 + \max\{i : (i, j) \in S\}$. We shall refer to a shape S as having dimension $r \times c$ although S may be a proper subset of the cells that comprise the normalized $r \times c$ rectangle T . Cells that appear in T but not in S will be of particular interest.

Definition 1.2.1 Given an $r \times c$ shape S and the $r \times c$ rectangle T that contains it, $w \in T$ is called a free cell if $w \notin S$.

For thickness $k = \max\{r, c\}$, set $Q = F_{k,k}(X)$ and let ψ be a normalized $k \times k$ square shape. There is no loss of generality in assuming that the shift of finite type X is defined by Q rather than by Γ , that is,

$$X := \{x \in \Sigma^{\mathbb{Z}^2} : \forall v \in \mathbb{Z}^2, \sigma^v(x)_\psi \in Q\}. \quad (1.2.3)$$

Note also that if X is a two-dimensional shift of finite type defined through a set of blocks $F_{k,k}(X)$, then for all $K \geq k$, X may also be defined through $F_{K,K}(X)$. In some cases, however, it will be preferable to employ the set of blocks $F_{r,c}(X) = \Psi$ whose dimensions minimally contain the shape S . In particular, the *allowed blocks* of a shift of finite type X is defined here as the local picture language $A(X)$ of all blocks $B \in \Sigma^{**}$ that satisfy the condition $F_{r,c}(B) \subseteq \Psi$. (For $B = B_{m,n}$ with $m < r$ or $n < c$, we say $B_{m,n}$ is in $A(X)$ if there exists $B'_{m',n'}$ with $m' \geq r$ and $n' \geq c$ such that $B'_{m',n'}$ is in $A(X)$ and $B_{m,n} \subset B'_{m',n'}$.) More specifically, when discussing the blocks of a language defined by a two-dimensional

shift space, we shall often use the set $F_{r,c}(X) = \Psi$ in order to simplify the proofs, but when discussing the symbolic dynamics of a two-dimensional shift space, we shall find it preferable to employ the set $F_{k,k}(X) = Q$. We point out that for the set of blocks $F(X)$, $F_{r,c}(B) \subseteq \Psi$ is necessary for $B \in F(X)$ but is not sufficient: B must also occur in some point of X . On the other hand, for the set of blocks $A(X)$, $F_{r,c}(B) \subseteq \Psi$ is both necessary and sufficient for $B \in A(X)$.

For a one-dimensional shift of finite type, the factor language of the shift space is always a local language; that is, $A(X) = F(X)$ for all one-dimensional shifts of finite type [25]. In a two-dimensional shift of finite type, however, a block in $A(X)$ need not appear as a block in $F(X)$ since the emptiness problem raises the question of whether the local picture language $A(X)$ is prolongable. For example, Kari [17] provides a small aperiodic set of Wang tiles describing a shift of finite type X having the property that $F(X) \subsetneq A(X)$: using the given set of Wang tiles, one can construct blocks that conform to the structure of the language yet can not be extended any farther in certain directions and therefore can not appear in the set $F(X)$.

Just as REC refers to the class of languages that can be obtained through the projection of local picture languages, two-dimensional shifts of finite type may be “projected” to form another class of shift spaces. More specifically, for a given two-dimensional shift of finite type X , we can transform $x \in X$ into a new point $y \in Y$ where $Y \subseteq \Delta^{\mathbb{Z}^2}$ employs some new alphabet Δ . For the $k \times k$ square shape T , a function $\Phi : F_{k,k}(X) \rightarrow \Delta$ that maps $k \times k$ blocks in X to symbols in Δ by $\Phi(x_{T+w}) = y_w$ is called a $k \times k$ -block map. The map $\phi : X \rightarrow \Delta^{\mathbb{Z}^2}$ defined by $y = \phi(x)$ with y_w induced by Φ is called a $k \times k$ -block code, and its image $Y = \phi(X)$ is called a *two-dimensional sofic shift*. When the block code ϕ is invertible, we refer to ϕ as a conjugacy and say that the spaces X and Y are *conjugate*. A key feature of block codes is that for a block code $\phi : X \rightarrow Y$ and a point $x \in X$, computing ϕ at the shifted point $\sigma^{(i,j)}(x)$ gives the same result as shifting the image $\phi(x)$ using $\sigma^{(i,j)}$ in the space Y . That is, the diagram in (1.2.4) commutes ([25].)

In symbolic dynamics, we are often interested in properties that are *invariant*; that is, properties that hold true for all shifts that are conjugate to a given shift. For two-dimensional languages, we will be interested in whether a given pair of blocks might coexist within a single point of the shift space. In one dimension, such a question is one of transitivity. Unlike the one-dimensional case, however, there are several types of

transitivity that appear in two-dimensional languages, each of which defines an invariant property for conjugate shift spaces (see Proposition 1.2.4 for a proof).

$$\begin{array}{ccc}
 & \sigma^{(i,j)} & \\
 X & \rightarrow & X \\
 \phi \downarrow & & \downarrow \phi \\
 Y & \rightarrow & Y \\
 & \sigma^{(i,j)} &
 \end{array} \tag{1.2.4}$$

To discuss transitivity in the two-dimensional case, we first need to define distance and direction between a pair of blocks in a two-dimensional space.

Definition 1.2.2 A block B encloses the pair of blocks B' and B'' if $B', B'' \in F(B)$. Furthermore, a block B minimally encloses B' and B'' if B encloses B' and B'' in such a way that both the bottom and top rows of B as well as the left and right columns of B all intersect at least one of the blocks B', B'' .

If a block $B_{m,n}$ minimally encloses the pair of blocks $B'_{p,q}, B''_{s,t}$ we say that $d(B', B'') = \max\{0, m - p - s, n - q - t\}$ is the distance at which B' meets B'' .

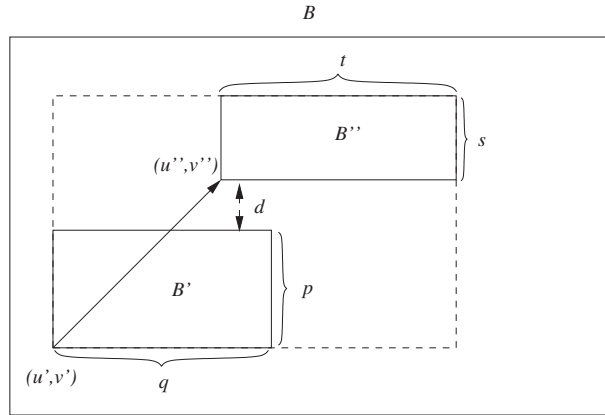


Figure 1.1: Enclosure of two blocks

Let L be a two-dimensional language containing the blocks B' and B'' . If there is a block B in L that minimally encloses B' and B'' without allowing them to overlap, where the bottom-left corners of B' and B'' appear at vertices (u', v') and (u'', v'') respectively, we say that B' meets B'' within L in direction (u, v) for any (u, v) having integer coordinates that is a non-zero multiple of (i.e., parallel to) the vector $(u'' - u', v'' - v')$.

Informally, we say that B encloses B' and B'' if both B' and B'' appear as subblocks of B . This is depicted in Figure 1.1, where the block that minimally encloses B' and B'' is indicated with dotted lines. The direction (u, v) in which B' and B'' meet is determined by the bottom-left corners of B' and B'' , and therefore $u, v \in \mathbb{Z}$. Note that it might be the case that the two blocks touch, in which case the distance at which they meet would be 0.

Definition 1.2.3 We say that a two-dimensional language L is

- transitive in direction (u, v) if for every pair of blocks $B', B'' \in L$ the block B' meets B'' in direction (u, v) within L .
- transitive if for every pair of blocks $B', B'' \in L$ there is a block $B \in L$ that encloses B' and B'' .
- uniformly transitive if there is a positive integer K such that for every pair of blocks $B', B'' \in L$ there is a block $B \in L$ that minimally encloses B' and B'' in a way that $d(B', B'') < K$.
- mixing if for every pair of blocks $B', B'' \in L$, there is a positive integer K such that the block B' meets B'' in every direction within L provided $d(B', B'') > K$.
- uniformly mixing if there is a positive integer K such that for every pair of blocks $B', B'' \in L$, the block B' meets B'' in every direction within L provided $d(B', B'') > K$.

Different types of transitivity are presented in Figure 1.2. Directional transitivity is worth naming in two particular cases: If L is transitive in direction $(1, 0)$, we shall say that L is *horizontally transitive* (see diagram (a) in Figure 1.2), and similarly we say that L is *vertically transitive* if L is transitive in direction $(0, 1)$. Transitivity is more general than uniform transitivity as there is no bound on how far apart the blocks B' and B'' might be: that is, uniform transitivity ensures that blocks need not extend too far in order to meet each other (see diagram (b) in Figure 1.2). A notion of *uniform directional transitivity* could be defined in the same manner. Mixing allows for two blocks to meet everywhere outside of a certain “neighborhood”, whereas uniform mixing guarantees that there is a bound on the “radius” of this neighborhood regardless of the size of the blocks. For

example in diagram (c) of Figure 1.2, the shaded region in (c) indicates the “neighborhood” defined by the constant K ; here, the block B'' can appear in any direction from B' provided it is outside this neighborhood. From the definitions proposed here, one can see that we have the following implications:

$$\text{mixing} \Rightarrow \forall(u, v), \text{ transitive in direction}(u, v) \Rightarrow \text{transitive} \quad (1.2.5)$$

Furthermore, the implications in (1.2.5) still hold true when we insert the word “uniform” in front of each property.

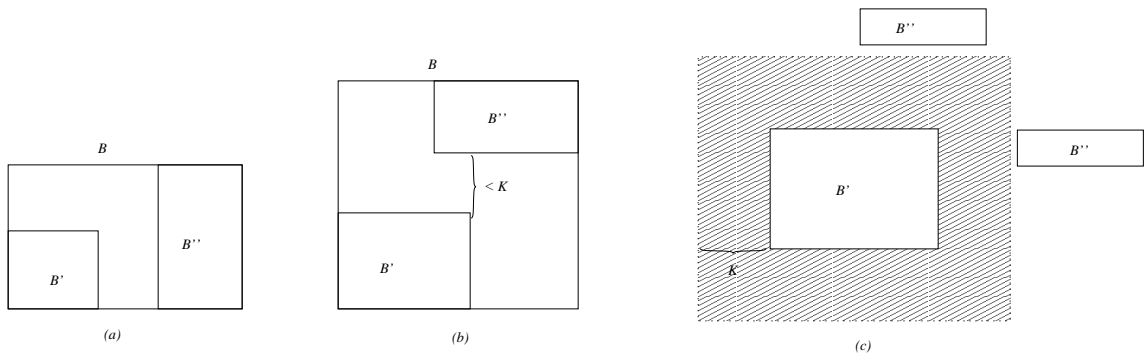


Figure 1.2: Different types of transitivity

In the case of one-dimensional recognizable languages, all notions of (uniform) transitivity and mixing coincide. Directional transitivity most closely resembles the notion of one-dimensional transitivity, as one can examine a bi-infinite sequence of “blocks” in the specified direction. The definitions presented here for transitivity and mixing in two-dimensional languages are similar to those defined for topological transformation groups [27, 9] in the study of topological dynamics. Proposition 1.2.4 describes transitivity in the two-dimensional case as an invariant property for conjugate shift spaces. Proofs for the invariance of mixing and/or uniformity properties can be accomplished in a similar fashion.

Proposition 1.2.4 *Let X and Y be two-dimensional shift spaces, and let $\phi : X \rightarrow Y$ be a conjugacy from X to Y . If $L = F(X)$ is transitive, then $L' = F(Y)$ is also transitive.*

Proof. Let $B' : T' \rightarrow \Sigma$ and $B'' : T'' \rightarrow \Sigma$ be blocks in the language $F(Y)$ over the alphabet Σ . We seek a block $B \in F(Y)$ that encloses B' and B'' . Since ϕ is invertible, B' and B'' have unique preimages, say $\phi^{-1}(B') = \beta'$ and $\phi^{-1}(B'') = \beta''$. Furthermore,

since $F(X)$ is transitive, there exists a block $\beta \in F(X)$ that encloses β' and β'' . That is, there must exist $w', w'' \in \mathbb{Z}^2$ such that $\beta(T' + w') = \beta'$ and $\beta(T'' + w'') = \beta''$. Therefore, $\phi(\beta) = B \in F(Y)$ is the desired block since $B \supset \phi(\beta(T' + w')) = \phi(\beta') = \phi(\phi^{-1}(B')) = B'$ and $B \supset \phi(\beta(T'' + w'')) = \phi(\beta'') = \phi(\phi^{-1}(B'')) = B''$. ■

1.3 Dot Systems

When G is a finite group, $G^{\mathbb{Z}^2}$ is a group via coordinate-wise product. Let X be a two-dimensional shift space which is also a subgroup of $G^{\mathbb{Z}^2}$: Such a subshift is called a *two-dimensional group shift*. It has been shown [19] that all two-dimensional group shifts are shifts of finite type. In a publication dated 1992, Kitchens and Schmidt define a type of group shift that they call a dot system and then show that every two-dimensional group shift is a finite intersection of these dot systems [20, 30]. For the work herein, we shall employ dot systems that are defined via some shape S over the group $G = \mathbb{Z}/2\mathbb{Z}$. That is,

$$X := \left\{ x \in \{0, 1\}^{\mathbb{Z}^2} : \forall v \in \mathbb{Z}^2, \sum_{w \in S+v} (x_w) = 0 \right\}. \quad (1.3.6)$$

Informally, a point x belongs to this type of dot system X if and only if all translates of the shape S in x contain an even number of 1 's. Dot systems may also be defined over finite abelian groups other than $\mathbb{Z}/2\mathbb{Z}$ in the obvious way: that is, the product of the symbols that appear within translates of S should equate to the identity element for that group. (See Lemma 1.3.1 and Proposition 1.3.4 to follow.) It is useful to think of S as a shape of dots within a defining rectangle T as illustrated in Figure 1.3. In particular, the subshift defined through shape (a) is known in the literature as the Three-dot System. For convenience, we shall also occasionally refer to the elements of S simply as dots. Unless stated otherwise, we disregard the singleton case when $S = (0, 0)$, as the shift space resulting from this shape is the trivial one comprised of the single point of all zeros.

Dot systems share a particularly nice property that allows us to employ the set of allowed blocks when investigating properties of the factor language of the shift space: that is, any block allowed by the structure of the dot system also appears as a subblock of some

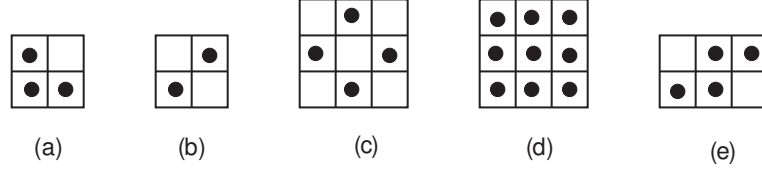


Figure 1.3: Shapes within minimal rectangles

point in the shift space. The proof of such is facilitated by the observation that in dot systems, any block that is too narrow to contain the shape S may be “extended” into a larger (allowed) block through the assignment of additional rows and/or columns.

Lemma 1.3.1 (Block Extension) *For a dot system X defined by the finite abelian group G over an $r \times c$ shape S , $\{G^{m,n} : m < r \text{ or } n < c\} \subset A(X)$.*

Proof. Denote the identity element of G as e , use \prod to denote repeated use of the group operation, and denote the inverse of $g \in G$ as g^{-1} . Suppose $B'_{m,n} : T' \rightarrow G$ is such that $m < r$ and $n < c$. Fix a dot from S with coordinates (i', j') such that $i' \geq n$ or $j' \geq m$. We may then define $B \in G^{r,c} \cap A(X)$ in the following way:

$$B(i, j) = \begin{cases} B'_{m,n}(i, j) & \text{for } 0 \leq i \leq n - 1, 0 \leq j \leq m - 1 \\ e & \text{for } i \geq n \text{ or } j \geq m, (i, j) \neq (i', j') \\ (\prod_{(s,t) \in S \cap T'} B'_{m,n}(s, t))^{-1} & \text{for } (i, j) = (i', j') \end{cases} \quad (1.3.7)$$

(The assignment of the identity element in (1.3.7) is chosen only to ease notation: one should note that with the exception of the fixed dot, all assignments are arbitrary.)

The case when only one of m or n is smaller than r or c , respectively, is treated similarly; however, care should be taken in the choice of which dot from S to fix initially. Without loss of generality, suppose that $B'_{m,n} : T' \rightarrow G$ is such that $m < r$ and $n \geq c$: refer to Figure 1.4 as needed, where the shaded area represents S , the shape T' is comprised of the cells having a solid border, and the dotted lines show the cells that we intend to define as we “extend” block $B'_{m,n}$ upward. First, let $i' = \max\{i : (i, r - 1) \in S\}$ and fix the dot $(i', r - 1) \in S$ (indicated by a darkly-shaded dot in Figure 1.4). Next, begin to define $B \in G^{r,n} \cap A(X)$ as detailed in (1.3.8). (Here again, the assignment of the identity

element is chosen merely to ease notation.)

$$B(i, j) = \begin{cases} B'_{m,n}(i, j) & \text{for } 0 \leq i \leq n-1, 0 \leq j \leq m-1 \\ e & \text{for } 0 \leq i \leq n-1, m \leq j < r-1 \\ e & \text{for } 0 \leq i < i', j = r-1 \\ (\prod_{(s,t) \in S \cap T'} B'_{m,n}(s, t))^{-1} & \text{for } (i, j) = (i', r-1) \end{cases} \quad (1.3.8)$$

Note that if $i' < c-1$, that is, if there exist free cells in the top right corner of the $r \times c$ block T containing S , then there will be $V = (c-1) - i'$ cells not yet defined in B_T ; in addition, there exists an undefined cell in B for each of $U = n - c$ translates of S . Therefore, in the definition of B , one should sequentially assign the cells on the top row of B such that for $0 \leq u \leq U$, $B(i' + u, r-1) = (\prod_{(i,j) \in \{S+(u,0)\} \setminus \{(i'+u, r-1)\}} B(i, j))^{-1}$. Following U translates, any undefined cells of B that remain in the top row may then be populated with the identity element, that is: for $1 \leq v \leq V$, let $B(i' + U + v, r-1) = e$.

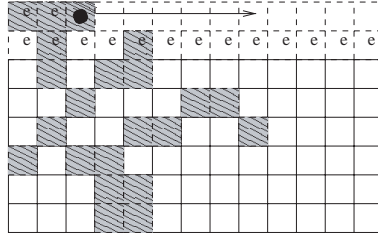


Figure 1.4: Extending a block upward

When $B'_{m,n} : T' \rightarrow G$ is such that $m \geq r$ and $n < c$, one can analogously extend $B'_{m,n}$ to the right by fixing the point $(c-1, j') \in S$ for $j' = \max\{j : (c-1, j) \in S\}$. Alternatively, we could extend blocks to the left (and/or downward) by varying the choice of which dot to fix initially before applying a symmetric argument. ■

We refer to the type of construction found in the proof of Lemma 1.3.1 as *block extension*. Later, when we employ dot systems for the discussion of various types of two-dimensional transitivity, we shall elaborate on the importance of fixing an initial dot before defining additional cells in a particular direction. We shall also make use of the following results regarding cardinality when the group $G = \mathbb{Z}/2\mathbb{Z}$.

Corollary 1.3.2 For a dot system X defined by the group $G = \mathbb{Z}/2\mathbb{Z}$ over some $r \times c$ shape S , $|F_{m,n}(X)| = 2^{mn}$ whenever $m < r$ or $n < c$.

Corollary 1.3.3 If X is a dot system defined through some $r \times c$ shape S , and $k = \max\{r, c\}$, then $|F_{k,k}(X)| = 2^{k^2-1-|r-c|}$.

Proof. If $r = c = k$, then the shape S is minimally enclosed by the $k \times k$ square shape T . Therefore, within any block $B \in F_{k,k}(X)$ fix one dot of the shape S , randomly populate all other cells of B , and then define the fixed dot as needed for the appropriate sum. So when $r = c$, $|F_{k,k}(X)| = |F_{r,c}(X)| = 2^{rc-1} = 2^{k^2-1-|r-c|}$.

Without loss of generality then, assume that $r > c$. Similar to the proof of Lemma 1.3.1, one can begin to define a block $B \in F_{k,k}(X)$ by fixing a dot in the top row of the normalized shape S before arbitrarily populating all other cells of the normalized $r \times c$ rectangular shape T' . Then for each of $r - c$ horizontal translates of S , the shape S remains enclosed by the $k \times k$ square shape T , so that each cell in the empty column intersecting the newly-translated shape S may be arbitrarily populated - with the exception of the one cell containing the translate of the fixed dot whose value is dictated by the required sum. After $r - c$ horizontal translates of the shape S , the shape is no longer enclosed by T , so that any remaining cells of T may be arbitrarily populated. Therefore, for the k^2 cells located in shape T , all but $1 + |r - c|$ may be arbitrarily populated with either 0 or 1, and the result follows. When $c > r$, the proof is analogous. ■

Proposition 1.3.4 is true for dot systems defined over any finite abelian group G .

Proposition 1.3.4 For the dot system X defined over the finite abelian group G , $A(X) = F(X)$.

Proof. Since all dot systems are shifts of finite type, X a dot system $\Rightarrow X$ is shift of finite type $\Rightarrow F(X) \subseteq A(X)$.

For the reverse inclusion, suppose $B'_{m,n} : T' \rightarrow G$ is such that $B'_{m,n} \in A(X)$. We will demonstrate that there exists a block $B_{m+2,n+2} \in A(X)$ such that $B_{m+2,n+2}(T' + (1, 1)) = B'_{m,n}$ (see Figure 1.5). Then by König's Lemma, $B'_{m,n}$ must occur in a point of the shift space X , since there exists an infinite process by which one can extend the block via

selection from a finite set of acceptable cell assignments. Without loss of generality, we may assume that $B'_{m,n}$ is such that $m \geq r$ and $n \geq c$. (Otherwise, use the appropriate block extension from Lemma 1.3.1 to construct some larger block in $A(X)$ that encloses $B'_{m,n}$ as a subblock.) Consider the subblock $\beta_1 = \{B'_{m,n}(i, j) : 0 \leq i \leq n-1, m-(r-1) \leq j \leq m-1\}$. Then β_1 has dimension $(r-1) \times n$, and by Lemma 1.3.1, there exists an upward extension of β_1 . Note that translates of S that lie entirely in $B'_{m,n}$ are not affected by the one-row extension of β_1 : therefore, this extension of β_1 also serves as an extension of $B'_{m,n}$. Denote the extended block as $B''_{m+1,n} \in A(X)$, and next consider the subblock $\beta_2 = \{B''_{m+1,n}(i, j) : n-(c-1) \leq i \leq n-1, 0 \leq j \leq m\}$ having dimension $(c-1) \times (m+1)$. By Lemma 1.3.1, there exists an extension to the right of β_2 resulting in the construction

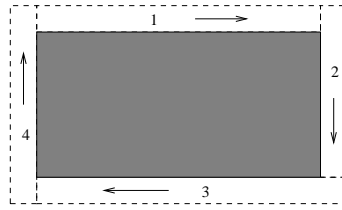


Figure 1.5: Extending an allowed block

of a new block $B_{m+1,n+1}^{(3)} \in A(X)$. However, to maintain construction in a clockwise direction, we use $j' = \min\{j : (c-1, j) \in S\}$ to fix the dot $(c-1, (m+1) - r + j')$ for the translate $S + (0, (m+1) - r)$. We then assign cell values for β_2 , beginning at the top of the new column and working our way downward. To continue the process, consider the subblock $\beta_3 = \{B_{m+1,n+1}^{(3)}(i, j) : 0 \leq i \leq n, 0 \leq j \leq r-2\}$ having dimension $(r-1) \times (n+1)$. We may extend β_3 below by first taking $i'' = \min\{i : (i, 0) \in S\}$ and fixing the dot $((n+1) - c + i'', 0)$ in the translate $S + ((n+1) - c, 0)$. We then begin to assign a larger block $B^{(4)}$ in such a way that $B_{m+2,n+1}^{(4)}(i, j) = B_{m+1,n+1}^{(3)}(i, j-1)$ for $0 \leq i \leq n, 1 \leq j \leq m+1$, and complete the assignment by extending β_3 (and hence $B_{m+2,n+1}^{(4)}$) by sequentially inspecting translates of S to the left of the initial fixed dot. Finally, using the subblock $\beta_4 = \{B_{m+2,n+1}^{(4)}(i, j) : 0 \leq i \leq c-2, 0 \leq j \leq m+1\}$, $B_{m+2,n+1}^{(4)}$ may be extended to the left by fixing a dot $(0, j'') \in S$ for $j'' = \max\{j : (0, j) \in S\}$, inspecting translates of S upward from this dot, and so on, to form $B_{m+2,n+2}$. ■

2 FINITE AUTOMATA RECOGNIZING TWO-DIMENSIONAL SHIFT SPACES

The graph construction defined in this chapter is based on interlacing horizontal and vertical movement in the graphs' transitions. In this way, the graph representation takes full advantage of the rich complexity of those languages that are the factor languages of two-dimensional shifts of finite type having the property that $A(X) = F(X)$. The main component of the construction is a tool for recognizing rectangular blocks that conform to the structure of the language. In Section 2.2 it is verified that for a two-dimensional shift of finite type having property $A(X) = F(X)$, the constructed graphs accurately represent the shift spaces that result from applying a block code to the shift space X . In the proof of Proposition 2.3.3, an explicit example demonstrates how to apply a higher block code to the states $F_{k,k}(X) = Q$, which changes the alphabet but not the dynamics, so that the resulting shift space is conjugate to the original but has states of size 2×2 . This in turn allows us to make several observations about certain structures within the graph and about the localized recognizable pictures that they force.

When Giammarresi and Restivo define the class REC of recognizable picture languages, they do so in the context of finite rectangular pictures that can be surrounded by a non-alphabet border symbol. By going to a higher block code as needed, one may assume that any local language in REC can be defined through a finite set of allowed 2×2 blocks that have been populated by symbols from the (new) alphabet Σ' and the non-alphabet symbol $\#$. For two-dimensional shift spaces, there is a similar notion of the language of a shift of finite type being contained within a local language defined by a finite set of 2×2 blocks. That is, for any shift space X , there is a local language $A(X)$ such that $F(X) \subseteq A(X)$, with the factor languages of the shift space being local in the case when $A(X) = F(X)$. If a language L is in REC but is not local, then L must be the projection of some local language that is in REC. In the same manner, if a sofic shift space Y is the image under a block code of a shift of finite type X having the property that $A(X) = F(X)$, then $F(Y)$

is in REC even if Y is not a shift of finite type. (When Y is a sofic shift space that is not a shift of finite type, $F(Y)$ is not a local language).

The following illustrates subtle distinctions that must be made for a two-dimensional picture language where a boundary symbol is inherent in the definition of the language.

Example 2.0.5 Define $L \subseteq \{a, b\}^{**}$ to be a two-dimensional picture language with $L \in REC$ such that for every block $B \in L$, any appearance of b is completely surrounded by a 's. The language L can be defined by the set of allowed blocks depicted in Figure 2.1. For example, the block

$$B = \begin{array}{ccccc} \# & \# & \# & \# & \# \\ \# & a & a & b & \# \\ \# & a & a & a & \# \\ \# & b & a & a & \# \\ \# & \# & \# & \# & \# \end{array} \quad (2.0.1)$$

would not be in the language, although

$$B' = \begin{array}{ccccccc} \# & \# & \# & \# & \# & \# & \# \\ \# & a & a & a & a & a & \# \\ \# & a & a & a & b & a & \# \\ \# & a & a & a & a & a & \# \\ \# & a & b & a & a & a & \# \\ \# & a & a & a & a & a & \# \\ \# & \# & \# & \# & \# & \# & \# \end{array} \quad (2.0.2)$$

would be in the language.

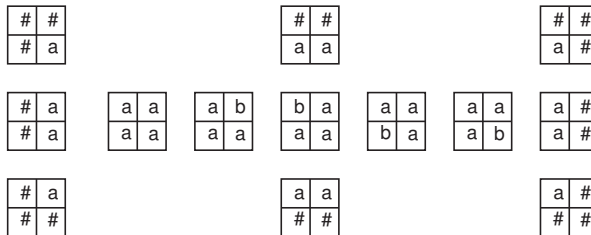


Figure 2.1: Sample set of allowed blocks for an REC language

Blocks B and B' in (2.0.1) and (2.0.2), respectively, indicate that $F(L) \neq L$. Therefore, the language L of Example 2.0.5 is not a factorial language. As we consider the factor

languages of shifts spaces, all languages in the sequel are factorial. In Section 2.2, we shall revisit Example 2.0.5 in the context of two-dimensional shift spaces.

2.1 Recognition of Shifts of Finite Type by $\mathcal{M}_{F(X)}$

Our goal is to construct a finite automaton that will allow the input data to consist of $m \times n$ blocks that can be scanned locally (and intermittently) by both horizontal and vertical transitions. To do so would require distinct sets of edges for horizontal and vertical transitions, say E_h and E_v , respectively, and would require that we define what we mean by acceptance. The general idea is that given an input block, we will consider sequences of symbols that appear in a window of fixed size as we scan the input block from the lower-left corner to the upper-right corner by traveling in two directions - up and/or to the right - within the constraints of the block's dimensions. If the automaton accepts all such sequences of symbols appearing as a result of such moves and it is determined that these sequences overlap progressively in some sense (to be made clear later), then we shall say that the block itself is accepted by the automaton.

To ease the formal discussion of constructing finite automata capable of recognizing two-dimensional subshifts, we shall refer to the extension of a design (not necessarily a block) by one row (column) of length (height) k as a k -concatenation. Informally, beginning with a $k \times k$ block, we allow a sequence of concatenations consisting of $k \times 1$ blocks concatenated horizontally to the right of the upper-most symbols in the existing design and/or $1 \times k$ blocks concatenated vertically above the right-most symbols in the existing design. More formally, suppose we begin with a $k \times k$ block and proceed to extend this block by the vertical k -concatenation of m' -many $1 \times k$ blocks. Let $m = k + m'$ and let $B_{m,k}$ be the result of these concatenations. We next allow a $k \times 1$ block $B'_{k,1}$ to be horizontally k -concatenated to the existing block. The binary operation is denoted by \rightarrow , and the resulting design $B_{m,k} \rightarrow B'_{k,1}$ (see the first diagram in Figure 2.2) is defined by

$$\gamma(i, j) = \begin{cases} B_{m,k}(i, j) & \text{for } 0 \leq i \leq k - 1, 0 \leq j \leq m - 1 \\ B'_{k,1}(0, j - (m - k)) & \text{for } m - k \leq j \leq m - 1 \end{cases} . \quad (2.1.3)$$

In the same way, a $k \times n$ block $B_{k,n}$ with $n \geq k$, may be extended by vertical k -concatenation with a $1 \times k$ block $B''_{1,k}$. The binary operation is denoted by \uparrow , and the

resulting design $B_{k,n} \upharpoonright B''_{1,k}$ is defined by

$$\gamma(i, j) = \begin{cases} B_{k,n}(i, j) & \text{for } 0 \leq i \leq n-1, 0 \leq j \leq k-1 \\ B''_{1,k}(i - (n-k), 0) & \text{for } n-k \leq i \leq n-1 \end{cases}. \quad (2.1.4)$$

Finally, k -concatenation onto a shape other than a rectangle is allowed provided that the shape is geometrically congruent to a shape that resulted from a finite sequence of k -concatenations. We call such a sequence of allowed k -concatenations a k -phrase, resulting in a set of progressively overlapping blocks of size $k \times k$. We shall denote a $k \times k$ block B that occurs in the k -phrase P by $B_{k,k} \subseteq P$. With each k -phrase P , we may also associate two functions s and t : each k -phrase starts with $s(P) = \beta_\alpha$, where β_α is a $k \times k$ block; and each k -phrase terminates in $t(P) = \beta_\omega$, where β_ω is also a $k \times k$ block. An example of an underlying shape for a k -phrase is depicted in Figure 2.2 to the right.

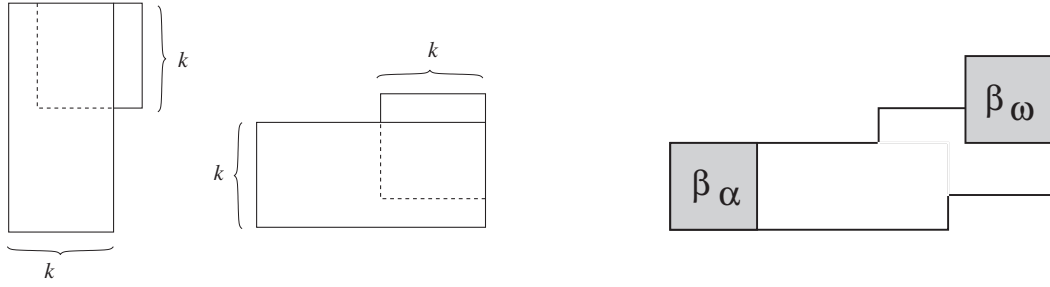


Figure 2.2: Horizontal and vertical k -concatenations

Now suppose a two-dimensional shift of finite type X is defined by $Q = F_{k,k}(X)$ for some $k \geq 2$. If $F(X) = A(X)$ so that the factor language of the shift space is local, then the *finite automaton* $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$ defined through Q is a finite directed graph obtained as follows. (Note that if a shift of finite type is completely described by a set of 1×1 blocks that define the local language $A(X) = F(X)$, then the shift space must be the full shift, since this would imply that any two alphabet symbols may be placed next to each other. In such cases, the full shift X over the alphabet Σ may also be defined through the set $F_{2,2}(X) = Q$ having $|Q| = |\Sigma|^4$.) First, define the vertex set of $\mathcal{M}_{F(X)}$ to be Q . For example, say

$$q = \begin{array}{ccc} q_{(0,k-1)} & \cdots & q_{(k-1,k-1)} \\ \vdots & \ddots & \vdots \\ q_{(0,0)} & \cdots & q_{(k-1,0)} \end{array} \quad \text{and} \quad r = \begin{array}{ccc} r_{(0,k-1)} & \cdots & r_{(k-1,k-1)} \\ \vdots & \ddots & \vdots \\ r_{(0,0)} & \cdots & r_{(k-1,0)} \end{array}$$

are two vertices in $\mathcal{M}_{F(X)}$. Next, the edge set representing the transitions between the states of $\mathcal{M}_{F(X)}$ is defined to consist of horizontal and vertical transitions, E_h and E_v respectively, such that $E = E_h \cup E_v$ and $E_h \cap E_v = \emptyset$. For horizontal transitions, an edge from state q to state r is defined as

$$\begin{array}{cccccccc}
& q_{(1,k-1)} & \cdots & q_{(k-1,k-1)} & & r_{(0,k-1)} & \cdots & r_{(k-2,k-1)} \\
e_h \in E_h \text{ if and only if} & \vdots & \ddots & \vdots & = & \vdots & \ddots & \vdots & \text{and} \\
& q_{(1,0)} & \cdots & q_{(k-1,0)} & & r_{(0,0)} & \cdots & r_{(k-2,0)} \\
\\
q_{(0,k-1)} & \cdots & q_{(k-1,k-1)} & r_{(k-1,k-1)} & & q_{(0,k-1)} & r_{(0,k-1)} & \cdots & r_{(k-1,k-1)} \\
\vdots & \ddots & \vdots & \vdots & = & \vdots & \vdots & \ddots & \vdots & \in F(X). \\
q_{(0,0)} & \cdots & q_{(k-1,0)} & r_{(k-1,0)} & & q_{(0,0)} & r_{(0,0)} & \cdots & r_{(k-1,0)}
\end{array}$$

In this case, the horizontal edge is denoted $e_h = q \rightarrow r$ and is given the label of the $k \times (k + 1)$ block that is the result of the horizontal k -concatenation of q with $\bar{r} := \{r(i, j) : i = k - 1, 0 \leq j \leq k - 1\}$. The $k \times (k + 1)$ block $q \rightarrow \bar{r}$ shall be denoted $\lambda(e_h)$. Similarly for vertical transitions, an edge from state q to state r is defined as

$$\begin{array}{cccccccc}
& q_{(0,k-1)} & \cdots & q_{(k-1,k-1)} & & r_{(0,k-2)} & \cdots & r_{(k-1,k-2)} \\
e_v \in E_v \text{ if and only if} & \vdots & \ddots & \vdots & = & \vdots & \ddots & \vdots & \text{and} \\
& q_{(0,1)} & \cdots & q_{(k-1,1)} & & r_{(0,0)} & \cdots & r_{(k-1,0)} \\
\\
r_{(0,k-1)} & \cdots & r_{(k-1,k-1)} & & r_{(0,k-1)} & \cdots & r_{(k-1,k-1)} \\
q_{(0,k-1)} & \cdots & q_{(k-1,k-1)} & = & \vdots & \ddots & \vdots & \in F(X). \\
\vdots & \ddots & \vdots & & r_{(0,0)} & \cdots & r_{(k-1,0)} \\
q_{(0,0)} & \cdots & q_{(k-1,0)} & & q_{(0,0)} & \cdots & q_{(k-1,0)}
\end{array}$$

Here the vertical edge is denoted $e_v = q \uparrow r$ and is given the label of the $(k+1) \times k$ block that is the result of the vertical k -concatenation of q with $\hat{r} := \{r(i, j) : 0 \leq i \leq k-1, j = k-1\}$, while the $(k + 1) \times k$ block $q \uparrow \hat{r}$ is denoted $\lambda(e_v)$. In addition to the labeling function λ already described, to each graph we may associate two other functions s and t : Each edge $e \in E$ has the *source* at a vertex denoted by $s(e) \in Q$, and each edge has *target* at a vertex denoted $t(e) \in Q$. (The case where $t(e) = s(e)$ is permissible.)

To begin the discussion of the language recognized by $\mathcal{M}_{F(X)}$, we must make more precise the meaning of a path and its label. Define a path in $\mathcal{M}_{F(X)}$ to be a sequence

$\Lambda = q_0x_1q_1x_2\dots q_p$ of vertices and transitions, where for $0 \leq i \leq p$, $q_i \in Q$ and $x_i \in \{\rightarrow, \uparrow\}$ are such that $s(\Lambda) = q_0$, $t(\Lambda) = q_p$, and $\lambda(q_{i-1}x_iq_i) \in F(X)$ for all $i \in \{1, 2, \dots, p\}$. The length of a path is denoted by $|\Lambda|$, the number of vertices visited after leaving q_0 . When Λ is comprised solely of edges from E_h (i.e. $x_i = \rightarrow$ for all $i \in \{1, 2, \dots, p\}$), we shall sometimes refer to Λ as an *h-path*, and extend this nomenclature in the natural way to *v-paths*, *h-cycles*, *v-cycles*, and in particular, *h-loops* and *v-loops*. The label of a path shall be defined inductively: For $|\Lambda| = 1$, $\lambda(\Lambda) = \lambda(q_0x_1q_1)$, i.e. the label of the edge from q_0 to q_1 ; If $\lambda(q_0x_1q_1x_2\dots q_{p-1})$ is given, then $\lambda(q_0x_1q_1x_2\dots q_{p-1}x_px_p) = \lambda(q_0x_1q_1x_2\dots q_{p-1})x_pq_p^+$, where for $x_p = \rightarrow$, $x_pq_p^+$ denotes horizontal k -concatenation of $\lambda(q_0x_1q_1x_2\dots q_{p-1})$ with $\bar{q}_p := \{q_p(i, j) : i = k - 1, 0 \leq j \leq k - 1\}$, and similarly when $x_p = \uparrow$, $x_pq_p^+$ denotes the vertical k -concatenation of $\lambda(q_0x_1q_1x_2\dots q_{p-1})$ with $\hat{q}_p := \{q_p(i, j) : 0 \leq i \leq k - 1, j = k - 1\}$.

The input data for $\mathcal{M}_{F(X)}$ is in the form of rectangular arrays of data - in other words, blocks. Let k be a positive integer and suppose $B_{m,n}$ is given for $m, n \geq k$. A k -phrase $P \subseteq B_{m,n}$ is said to be *accepted by* $\mathcal{M}_{F(X)}$ if and only if there is a path Λ in $\mathcal{M}_{F(X)}$ such that $\lambda(\Lambda) = P$. (Note that if Λ is such that $\lambda(\Lambda) = P$, then $|\Lambda| = m + n - 2k$.) Set

$$\beta_\alpha = \{B_{m,n}(i, j) : 0 \leq i \leq k - 1, 0 \leq j \leq k - 1\}, \quad (2.1.5)$$

i.e. the lower-left $k \times k$ subblock of $B_{m,n}$, and set

$$\beta_\omega = \{B_{m,n}(i, j) : n - k \leq i \leq n - 1, m - k \leq j \leq m - 1\}, \quad (2.1.6)$$

i.e. the upper-right $k \times k$ subblock of $B_{m,n}$.

Definition 2.1.1 Block $B_{m,n}$ is accepted by automaton $\mathcal{M}_{F(X)} = \{Q, E, s, t, \lambda\}$ if and only if $\mathcal{M}_{F(X)}$ accepts all k -phrases of $B_{m,n}$ that start with state $q_\alpha = \beta_\alpha$ and terminate in state $q_\omega = \beta_\omega$ after a sequence of k -concatenations $\chi = x_1, x_2, \dots, x_p \in \{\rightarrow, \uparrow\}^*$ satisfying $|\chi|_\rightarrow = n - k$ and $|\chi|_\uparrow = m - k$. (For $B_{m,n}$ with $m < k$ or $n < k$, we say $B_{m,n}$ is accepted by $\mathcal{M}_{F(X)}$ if there exists $B'_{m',n'}$ with $m', n' \geq k$ such that $B'_{m',n'}$ is accepted by $\mathcal{M}_{F(X)}$ and $B_{m,n} \subset B'_{m',n'}$.)

Definition 2.1.2 The language recognized by $\mathcal{M}_{F(X)}$ is the set

$$L(\mathcal{M}_{F(X)}) = \{B : B \in \Sigma^{**}, B \text{ is accepted by } \mathcal{M}_{F(X)}\}.$$

Proposition 2.1.3 *Let X be a two-dimensional shift of finite type having the property $F(X) = A(X)$. Let the automaton $\mathcal{M}_{F(X)}$ be as constructed above. Then $F(X) = L(\mathcal{M}_{F(X)})$.*

Proof. To show that $L(\mathcal{M}_{F(X)}) = F(X)$, first suppose $B_{m,n} \in F(X)$ is given. If $m < k$ or $n < k$, then there exists $B'_{m',n'} \in F(X)$ such that $B_{m,n} \subset B'_{m',n'}$, and by the definition of acceptance $B_{m,n} \in L(\mathcal{M}_{F(X)})$ if $B'_{m',n'} \in L(\mathcal{M}_{F(X)})$. So suppose that $m, n \geq k$. Since $B_{m,n} \in F(X)$, β_α and $\beta_\omega \in Q$. By construction of the $k \times k$ states Q , β_α represents a unique initial state and β_ω represents a unique terminal state for all k -phrases in $B_{m,n}$. (The case where $\beta_\alpha = \beta_\omega$ is not precluded.) By the definition of the existence of edges in $\mathcal{M}_{F(X)}$, all k -phrases of $B_{m,n}$ are accepted by $\mathcal{M}_{F(X)}$. Specifically, $\mathcal{M}_{F(X)}$ accepts all k -phrases of $B_{m,n}$ that begin with β_α and end with β_ω after a concatenation sequence $\chi = x_1, x_2, \dots, x_{m+n-2k} \in \{\rightarrow, 1\}^*$ satisfying $|\chi|_\rightarrow = n - k$ and $|\chi|_1 = m - k$. Therefore, $B_{m,n} \in L(\mathcal{M}_{F(X)})$ and $F(X) \subseteq L(\mathcal{M}_{F(X)})$.

For the other containment, suppose there exists some block $B_{m,n} \in L(\mathcal{M}_{F(X)})$ such that $B_{m,n} \notin F(X)$. Since $A(X) = F(X)$, it must be the case that there exists some $k \times k$ subblock $B_{k,k} \subseteq B_{m,n}$ such that $B_{k,k} \notin F_{k,k}(X) = Q$. (As before, we may assume that $m, n \geq k$.) Since $B_{m,n} \in L(\mathcal{M}_{F(X)})$, $\mathcal{M}_{F(X)}$ must accept a k -phrase $P \subseteq B_{m,n}$ that passes through $B_{k,k}$. That is, there must exist P such that $s(P) = \beta_\alpha$, $t(P) = \beta_\omega$, and $B_{k,k} \subseteq P$. Furthermore, there must be a path $\Lambda = q_0 x_1 q_1 x_2 \dots q_p$ in $\mathcal{M}_{F(X)}$ such that $\lambda(\Lambda) = P$. But this would imply $B_{k,k} \in Q$, a contradiction. Therefore, $B_{m,n} \in F(X)$ and $L(\mathcal{M}_{F(X)}) \subseteq F(X)$. ■

In one-dimensional symbolic dynamics, a *vertex shift* is a shift of finite type having $\Sigma = Q$ (the set of states is the alphabet) with at most one edge connecting a pair of states, where points in the shift space are precisely the labels of the bi-infinite sequences that result from following paths in the graph. A related idea is that of an *edge shift*, where $\Sigma = E$ (the set of edges is the alphabet) with points in the shift space being the labels of the bi-infinite sequences that result from following paths in the graph. In an edge shift, however, multiple edges are allowed between a pair of states. See Figure 2.3 for examples of the representation of the one-dimensional full shift over $\Sigma = \{0, 1\}$. While the $\mathcal{M}_{F(X)}$ construction is based on the idea of a vertex shift, it does permit two edges to connect a

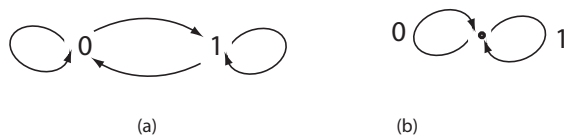


Figure 2.3: Full shift as vertex shift (a) and edge shift (b)

single pair of states if one edge is a horizontal transition and the other edge is a vertical transition. Note that $\mathcal{M}_{F(X)}$ also fulfills the requirements of an edge shift.

We now appeal to dot systems in order to generate some examples of two-dimensional finite state automata of reasonable size. We will be interested in dot systems defined by the group $G = \mathbb{Z}/2\mathbb{Z}$. For such dot systems, the size of the graph representing the shift space is given by Corollary 1.3.3.

Example 2.1.4 Consider the dot system X known in the literature as the Three-dot System, which is defined via the shape $S = \{(0, 0), (0, 1), (1, 0)\}$. (Refer to diagram (a) in Figure 1.3.) For this shift space X , the set $Q = F_{2,2}(X)$ consists of 8 states as provided for by Corollary 1.3.3. We use solid lines to represent horizontal transitions and dashed lines to represent vertical transitions. The directed graph representing $F(X)$ (and therefore X) is depicted in Figure 2.4.

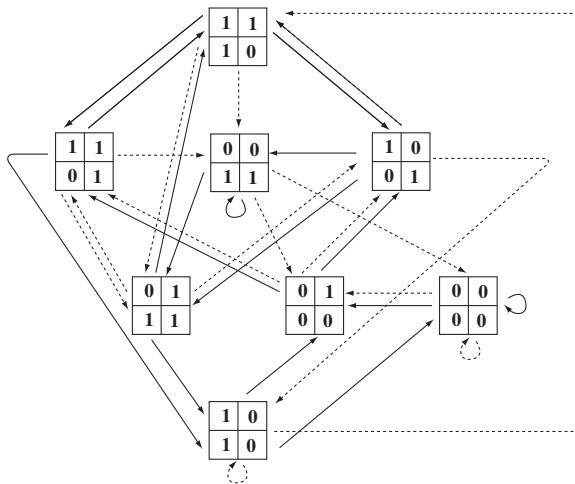


Figure 2.4: Graph representing the Three-dot System

As discussed in relation to vertex and edge shifts in the one-dimensional case, the labels of bi-infinite paths of the graph are precisely the points of the represented shift

space. In one dimension, it is often of interest whether different bi-infinite paths may be a presentation of the same point in the shift space; in other words, whether different bi-infinite paths may have the same label. An analogous representation of bi-infinite paths in two dimensions will require the following definition.

Definition 2.1.5 A grid-infinite path Π in $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$ is defined by a pair of maps, $\Pi_h : \mathbb{Z}^2 \rightarrow Q$ and $\Pi_v : \mathbb{Z}^2 \rightarrow Q$, and is denoted as a collection of states $\{q_{[i,j]}\}$ (square brackets are used to avoid the confusion of the symbol $q_{(i,j)} = a \in \Sigma$ with the state associated to the ordered pair $(i, j) \in \mathbb{Z}^2$) and the transitions that accompany these states.

$$\begin{array}{ccccccc}
& & \vdots & & \vdots & & \vdots \\
& & \uparrow & & \uparrow & & \uparrow \\
\cdots & q_{[-1,1]} & \rightarrow & q_{[0,1]} & \rightarrow & q_{[1,1]} & \rightarrow \cdots \\
& & \uparrow & & \uparrow & & \uparrow \\
\cdots & q_{[-1,0]} & \rightarrow & q_{[0,0]} & \rightarrow & q_{[1,0]} & \rightarrow \cdots \\
& & \uparrow & & \uparrow & & \uparrow \\
\cdots & q_{[-1,-1]} & \rightarrow & q_{[0,-1]} & \rightarrow & q_{[1,-1]} & \rightarrow \cdots \\
& & \vdots & & \vdots & & \vdots
\end{array}$$

This collection of states and transitions is such that $\forall (i, j) \in \mathbb{Z}^2$, the following hold:

- i) $q_{[i,j]} \in Q$,
- ii) $q_{[i,j]} \rightarrow q_{[i+1,j]} = e_h$ is an edge in E_h , and
- iii) $q_{[i,j]} \uparrow q_{[i,j+1]} = e_v$ is an edge in E_v .

The diagram in Definition 2.1.5 commutes in the sense that any block that is described by some portion of a grid-infinite path is independent of the order in which the edges are traversed. Proposition 2.1.6 uses this fact to establish a one-to-one correspondence between blocks in the local language and factors of grid-infinite paths.

Proposition 2.1.6 *Given the two-dimensional shift of finite type X , let $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$ be its graph representation. For a grid-infinite path Π of $\mathcal{M}_{f(X)}$, let Π' be some 2×2 factor of Π comprised of four adjacent states $(q_{[i,j]}, q_{[i+1,j]}, q_{[i,j+1]}, q_{[i+1,j+1]})$ and the transitions that connect them. Then Π' represents a unique $(k+1) \times (k+1)$ block that is recognized by $\mathcal{M}_{F(X)}$.*

Proof. By Definition 2.1.5, condition i), $q_{[i,j]} \in Q$ and is therefore a $k \times k$ block. Denote by $e_{h_1}, e_{h_2}, e_{v_1}$, and e_{v_2} the edges $q_{[i,j]} \rightarrow q_{[i+1,j]}, q_{[i,j+1]} \rightarrow q_{[i+1,j+1]}, q_{[i,j]} \uparrow q_{[i,j+1]}$, and $q_{[i+1,j]} \uparrow q_{[i+1,j+1]}$, respectively, whose labels are provided by Definition 2.1.5, conditions ii) and iii). That is, Π' may be represented by the following diagram.

$$\begin{array}{ccccc}
 & & e_{h_2} & & \\
 & q_{[i,j+1]} & \rightarrow & q_{[i+1,j+1]} & \\
 e_{v_1} & \uparrow & & \uparrow & e_{v_2} \\
 & q_{[i,j]} & \rightarrow & q_{[i+1,j]} & \\
 & & e_{h_1} & &
 \end{array}$$

The normalized k -phrase P_1 described by $\lambda(q_{[i,j]} \rightarrow q_{[i+1,j]} \uparrow q_{[i+1,j+1]})$ has $s(P_1) = q_{[i,j]}$ and $t(P_1) = q_{[i+1,j+1]}$ and is almost completely described by these two blocks with the exception of the symbol assigned to $P_1(k, 0)$ by $q_{[i+1,j]}$. In comparison, the normalized k -phrase P_2 described by $\lambda(q_{[i,j]} \uparrow q_{[i,j+1]} \rightarrow q_{[i+1,j+1]})$ also has $s(P_2) = q_{[i,j]}$ and $t(P_2) = q_{[i+1,j+1]}$ but has a symbol mapped to $P_2(0, k)$ by $q_{[i,j+1]}$. (See the shaded lattice points in Figure 2.5.) Therefore, the $(k + 1) \times (k + 1)$ block represented by Π' is independent of the order of k -concatenation. ■

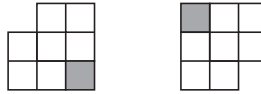


Figure 2.5: Comparison of $q \rightarrow r \uparrow s$ and $q \uparrow r' \rightarrow s$

Corollary 2.1.7 *Let X be a two-dimensional shift of finite type represented by the automaton $\mathcal{M}_{F(X)}$. Then there exists a one-to-one correspondence between points in X and grid-infinite paths in $\mathcal{M}_{F(X)}$.*

Proof. Apply Propositions 2.1.3 and 2.1.6. ■

In the sequel, we shall sometimes refer to an $m \times n$ factor Π' of a grid-infinite path Π as a *block path*. Note that an $m \times n$ block path is comprised of mn states and describes a block $B \in F(X)$ of size $(m+k-1) \times (n+k-1)$. The labeling of a block path shall be denoted with

$\lambda(\Pi') = B$ and the states shall be denoted by $\Pi'(i, j)$ for $0 \leq i \leq n - 1, 0 \leq j \leq m - 1$. See Figure 2.6: The shading in (a) suggests that the 3×7 block of symbols might be represented by a 2×6 block path as in diagram (b) if $\mathcal{M}_{F(X)}$ is defined via 2×2 states. Alternately, the same block of symbols might be described by a 1×5 block path if $\mathcal{M}_{F(X)}$ were defined via 3×3 states.

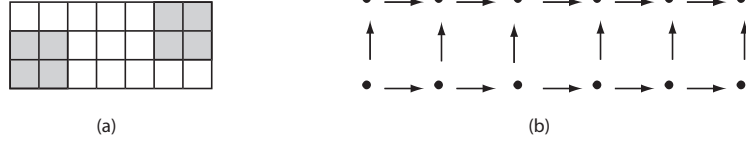


Figure 2.6: An example of a block path where $k = 2$

Example 2.1.8 illustrates the importance of the definition of the accepted language $L(\mathcal{M}_{F(X)})$ being based on blocks rather than on k -phrases alone. In particular, all k -phrases of an accepted block must be represented by some path Λ having source $s(\Lambda) = \beta_\alpha$ and target $t(\Lambda) = \beta_\omega$.

Example 2.1.8 Consider the dot system X defined via the shape $S = \{(0, 0), (1, 1)\}$ and referred to in this paper as the Diagonal-shift System. (See shape (c) in Figure 1.3.) Using solid lines to represent horizontal transitions and dashed lines to represent vertical transitions, a directed graph representing X is depicted in Figure 2.7. For ease of reference, each state $q \in Q = F_{2,2}(X)$ has been numbered.

Define a normalized 3×3 block β such that

$$\beta(i, j) = \begin{cases} 1 & \text{for } (i, j) = (1, 0) \\ 0 & \text{otherwise} \end{cases}.$$

Then $\beta \notin F(X)$, where X is the Diagonal-shift System. By inspection of the graph $\mathcal{M}_{F(X)}$, we see that the 2-phrase of β that is described by the labels of the path $q_1 \uparrow q_0 \rightarrow q_0$ is accepted by $\mathcal{M}_{F(X)}$, although such a design is not a subset of any point found in X . However, the block β itself will not be accepted, as there is no path in $\mathcal{M}_{F(X)}$ that describes the subblock $\beta' := \{\beta(i, j) : 0 \leq i \leq 2, 0 \leq j \leq 1\}$ and hence β can not be described by a factor of any grid-infinite path of $\mathcal{M}_{F(X)}$.

So we have seen that the $\mathcal{M}_{F(X)}$ construction is a true vertex (edge) shift in the sense that points in the two-dimensional shift space X are precisely the labels of the grid-infinite

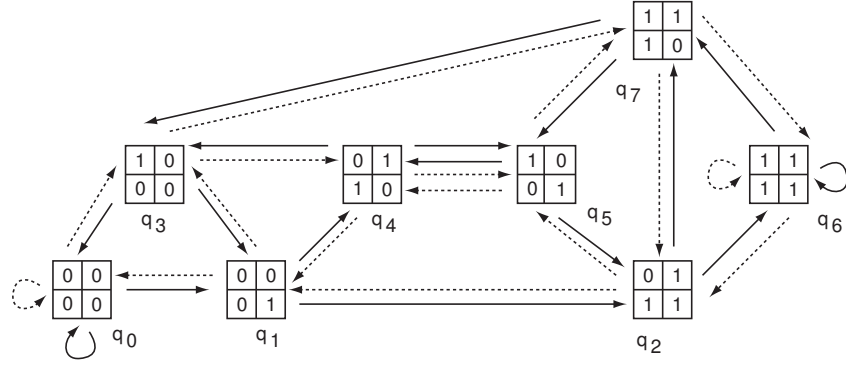


Figure 2.7: Graph $\mathcal{M}_{F(X)}$ representing the Diagonal-shift System

paths found in the graph. Therefore, Definition 2.1.9 can be reworded as follows.

Definition 2.1.9 Block $B_{m,n}$ is accepted by the finite automaton $\mathcal{M}_{F(X)}$ if and only if there exists a block path Π' in $\mathcal{M}_{F(X)}$ having $\lambda(\Pi') = B_{m,n}$.

2.2 Recognition of Sofic Shifts by $\mathcal{M}_{F(X)}^\Phi$

Before exploring properties of the graphs representing two-dimensional languages, we point out the potential usefulness of this new type of finite automaton by proving that the graphs constructed here accurately represent certain two-dimensional *strictly sofic shifts*, i.e. sofic shifts that are not shifts of finite type. Specifically, if X is a two-dimensional shift of finite type over Σ having the property that $A(X) = F(X)$, then the image of the block code $\phi : X \rightarrow \Delta^{\mathbb{Z}^2}$ induced by the $d \times d$ block map $\Phi : B_{d,d}(X) \rightarrow \Delta$ can be represented by the underlying graph of $\mathcal{M}_{F(X)}$ with states and labels adjusted accordingly.

In one-dimensional language theory, a graph is said to be *deterministic* if given a label and a vertex, there is at most one path starting at the given vertex with the specified label. Note that the graph $\mathcal{M}_{F(X)}$ is a deterministic graph in this sense, since given a state and a $k \times (k+1)$ label, at most one horizontal transition is specified. (Analogously, given a state and a $(k+1) \times k$ label, at most one vertical transition is specified.) However, when a block code is applied to the states of $\mathcal{M}_{F(X)}$, the states need no longer have distinct labels and the graph need not be deterministic. However, while there need no longer exist a one-to-one correspondence between grid-infinite paths and their labels, it will still be the

case that points in the represented two-dimensional shift space Y are precisely the labels of the grid-infinite paths found in the graph.

Proposition 2.2.1 *Let X be a two-dimensional shift of finite type represented by $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$, and let $Y = \phi(X)$ be the two-dimensional shift space that is the image of X under the block code ϕ induced by the block map Φ . If $\mathcal{M}_{F(X)}^\Phi$ is the finite automaton having underlying graph $\mathcal{M}_{F(X)}$ with state set Q' relabeled according to $Q' = \phi(Q)$ and edge set E' relabeled according to $\phi(\lambda(e))$, then $L(\mathcal{M}_{F(X)}^\Phi) = F(Y)$.*

Proof. Suppose $B' \in F(Y)$ is given. Since $F(Y) = \phi(F(X)) = \phi(L(\mathcal{M}_{F(X)}))$, there exists some $\beta \in F(X)$ such that $\phi(\beta) = B'$. So for the graph $\mathcal{M}_{F(X)}$, $\beta \in L(\mathcal{M}_{F(X)}) = F(X)$. However, $\mathcal{M}_{F(X)}$ and $\mathcal{M}_{F(X)}^\Phi$ have the same underlying edge set so that $\beta \in L(\mathcal{M}_{F(X)}) \Rightarrow B' \in L(\mathcal{M}_{F(X)}^\Phi)$. Therefore $F(Y) \subseteq L(\mathcal{M}_{F(X)}^\Phi)$.

For the reverse inclusion, say $B' \in L(\mathcal{M}_{F(X)}^\Phi)$ is given. In this case, there must exist some block path Π' in $\mathcal{M}_{F(X)}^\Phi$ such that $\lambda(\Pi') = B'$. Using the underlying graph of Π' , we can find a block path π' with labels from $\mathcal{M}_{F(X)}$ such that $\lambda(\pi') = \beta \in L(\mathcal{M}_{F(X)}) = F(X)$. In other words, $\phi(\beta) = B' \in F(Y)$. Therefore $L(\mathcal{M}_{F(X)}^\Phi) \subseteq F(Y)$. ■

In the proof of Proposition 2.2.1, although the block paths π' and Π' are of the same size, the blocks β and B' that they represent need not be. This is due to the fact that under the block code ϕ , the image $\phi(q) = q'$ of each state $q \in Q$ will have thickness $k' = 1 + k - d$. (We may always assume without loss of generality that $k \geq d$ since whenever $F_{k,k}(X)$ defines X , then $F_{K,K}(X)$ also defines X for all $K \geq k$.) We now revisit Example 2.1 that was introduced in the chapter-opener. That example was derived from the following example, which was introduced in [5] as an illustration of the difficulty in presenting two-dimensional sofic shifts with the type of graphs known at the time. In this example, a 2×2 invertible block code is applied to a shift of finite type X defined through a set $F_{2,2}(X)$ in order to create a conjugate shift of finite type X' defined through a set $F_{1,1}(X')$; thereafter, a (non-invertible) 1×1 block code is applied to X' in order to create a sofic shift space.

Example 2.2.2 Define $X \subseteq \{a, b\}^{\mathbb{Z}^2}$ to be a two-dimensional shift of finite type such that for every point $x \in X$, any appearance of b is surrounded by a 's. This subshift has the

property that $A(X) = F(X)$: given any block $B \in A(X)$, one can simply surround B with a configuration of the plane populated entirely with a 's. The language of the shift space X is defined through the set of allowed 2×2 blocks $Q = F_{2,2}(X)$ depicted in Figure 2.8. By

$$p := \begin{array}{|c|c|} \hline a & a \\ \hline a & a \\ \hline \end{array} \quad q_1 := \begin{array}{|c|c|} \hline b & a \\ \hline a & a \\ \hline \end{array} \quad q_2 := \begin{array}{|c|c|} \hline a & b \\ \hline a & a \\ \hline \end{array} \quad q_3 := \begin{array}{|c|c|} \hline a & a \\ \hline b & a \\ \hline \end{array} \quad q_4 := \begin{array}{|c|c|} \hline a & a \\ \hline a & b \\ \hline \end{array}$$

Figure 2.8: Set of five allowed blocks for subshift of Example 2.2.2

applying a 2×2 invertible block code, X is seen to be conjugate to $X' \subseteq \{p, q_1, q_2, q_3, q_4\}^{\mathbb{Z}^2}$. Consider next the sofic system Y obtained as the image of X' under the 1×1 block code ϕ defined by $\phi(p) = p$ and $\phi(q_i) = q$ for $i = 1, \dots, 4$. By inspection, any occurrence of q_1 in a point $x \in X$ must be preceded horizontally by q_2 and followed vertically by q_3 , whereas q_3 must be preceded horizontally by q_4 . That is, in points $y \in Y$, the q symbol always appears in 2×2 blocks comprised of q 's. One can picture the shift space Y to be the collection of all points that result from concatenations of 1×1 blocks labeled p and 2×2 blocks labeled entirely with q 's.

The sofic system Y described in Example 2.2.2 is strictly sofic. To see this, suppose towards a contradiction that Y is a shift of finite type. Then there would exist some $N \geq 1$ such that for the normalized $N \times N$ square shape ψ , a sufficient condition for a point to be in the shift space Y would be that all translates of ψ had a design belonging to the set $Q = F_{N,N}(Y)$. That is,

$$Y := \{y \in \{p, q\}^{\mathbb{Z}^2} : \forall v \in \mathbb{Z}^2, \sigma^v(y)_\psi \in Q\}. \quad (2.2.7)$$

Now consider a configuration of the plane populated entirely by p 's with the exception of the normalized $(2N + 1) \times 2$ rectangular shape T that is populated with q 's. More specifically, consider the point

$$y(i, j) = \begin{cases} q & \text{for } 0 \leq i \leq 1, 0 \leq j \leq 2N \\ p & \text{otherwise} \end{cases}. \quad (2.2.8)$$

Then y is such that $\forall v \in \mathbb{Z}^2, \sigma^v(y)_\psi \in Q$, which would be sufficient for $y \in Y$; however, this would contradict the definition of the shift space Y since no concatenation of 2×2 blocks of q 's could produce an odd number of rows ($2N + 1$) populated with q 's. So Y can

not be a shift of finite type and therefore must be a strictly sofic shift space.

There was no satisfactory graph representation of Y at the time [5] was written. For example if we let $N = 1$ in 2.2.8, then the use of independent horizontal and vertical scanning admits the point $y \in Y$, which is erroneous. However, if we use the $\mathcal{M}_{F(X)}$ construction to represent the shift of finite type X defined in Example 2.2.2, then the sofic system Y that is the image of the 1×1 block code ϕ can be accurately represented by the underlying graph of Figure 2.9.

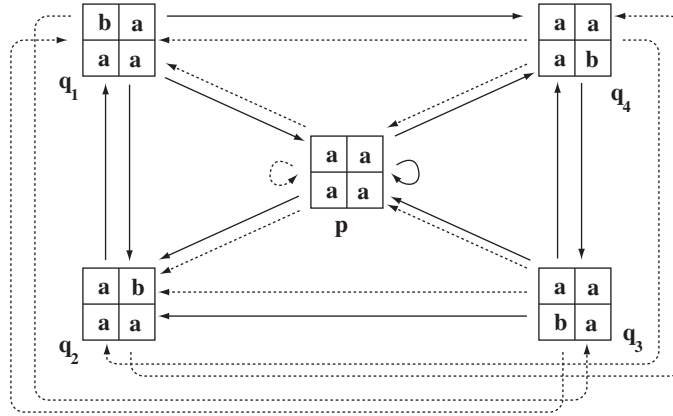


Figure 2.9: Shift space where any b is surrounded entirely by a 's

Caution must be observed to verify the acceptance of blocks in the sofic factor space Y based on block paths rather than merely as a collection of paths accepted by $\mathcal{M}_{F(X)}^\Phi$.

Example 2.2.3 Let the normalized 3×3 block B be given by

$$B(i, j) = \begin{cases} p & \text{for } (i, j) \in \{(0, 0), (1, 1)\} \\ q & \text{otherwise} \end{cases}$$

and let Y be the shift space of Example 2.2.2. Informally,

$$B = \begin{array}{ccc} q & q & q \\ q & p & q \\ p & q & q \end{array}$$

It is apparent that block B can not belong to the factor language of Y since its design has a perimeter comprised of an odd number of q 's which can not be formed by the concatenation of 2×2 blocks of q 's. Now let $\mathcal{M}_{F(X)}^\Phi$ be the underlying graph of Figure 2.9 with states $Q' = \phi(Q)$. The reader can verify that all 1- phrases of block B are accepted

by $\mathcal{M}_{F(X)}^\Phi$. However, there is no 3×3 block path Π' of $\mathcal{M}_{F(X)}^\Phi$ having the property that $\lambda(\Pi') = B$. To see this, denote the states of $\mathcal{M}_{F(X)}^\Phi$ with p and q_i for $i = 1, 2, 3, 4$ as in Figure 2.9 and let Π' be a 3×3 block path of $\mathcal{M}_{F(X)}^\Phi$ that is a candidate for $\lambda(\Pi') = B$. For reference, use subscripts to distinguish the states that comprise the block path Π' .

$$\Pi' = \begin{array}{ccc} q_e & q_f & q_g \\ q_c & p & q_d \\ p & q_a & q_b \end{array}$$

Then the following conditions specify certain states in Π' .

$$p \uparrow q_c \rightarrow p \Rightarrow q_c = q_1 \text{ and } q_1 \uparrow q_e \Rightarrow q_e = q_3$$

$$p \rightarrow q_a \uparrow p \Rightarrow q_a = q_4 \text{ and } q_4 \rightarrow q_b \Rightarrow q_b = q_3$$

As established thus far, six of the nine states in Π' are dictated by the design of B .

$$\Pi' = \begin{array}{ccc} q_3 & q_f & q_g \\ q_1 & p & q_d \\ p & q_4 & q_3 \end{array}$$

Paths in the graph of $\mathcal{M}_{F(X)}^\Phi$ restrict the possibilities for the remaining three states of the block path Π' . There are two options for the top row of Π' .

$$(i) q_3 \rightarrow q_4 \rightarrow q_3 \text{ or } (ii) q_3 \rightarrow q_2 \rightarrow q_1$$

For the far right column of Π' there are also two options.

$$(iii) q_3 \uparrow q_1 \uparrow q_3 \text{ or } (iv) q_3 \uparrow q_2 \uparrow q_4$$

By definition, all paths comprising a block path must end in the same terminal state. So by inspection, only paths (i) and (iii) might co-exist within the block path Π' . This process of elimination reveals that the only possibility for the states of Π' are those given in (2.2.9). However, such a block path does not exist in $\mathcal{M}_{F(X)}^\Phi$ since neither $p \uparrow q_4$ nor $p \rightarrow q_1$ is an edge in $\mathcal{M}_{F(X)}^\Phi$.

$$\Pi' = \begin{array}{ccc} & q_3 & q_4 & q_3 \\ q_1 & p & q_1 & \\ p & q_4 & q_3 & \end{array} \quad (2.2.9)$$

Suppose that $\mathcal{M}_{F(X)}$ represents a two-dimensional shift space X and let \mathcal{A}_h and \mathcal{A}_v denote the adjacency matrices for $\mathcal{M}_{F(X)}^h = (Q, E_h, s, t, \lambda)$ and $\mathcal{M}_{F(X)}^v = (Q, E_v, s, t, \lambda)$, respectively. (An adjacency matrix of a graph is a matrix where the (i, j) entry denotes the number of edges in the graph having initial state i and terminal state j .) In the literature, the representation of two-dimensional shift spaces through the use of two separate graphs/matrices for horizontal and vertical transitions is done in a setting that requires \mathcal{A}_h to commute with \mathcal{A}_v [26]. This property is not necessary for the underlying graph of a shift space represented via the $\mathcal{M}_{F(X)}$ construction.

Example 2.2.4 Let X be the shift space of Example 2.2.2 having the underlying graph of Figure 2.9. With the states ordered $\{p, q_1, q_2, q_3, q_4\}$, the adjacency matrices for $\mathcal{M}_{F(X)}^h = (Q, E_h, s, t, \lambda)$ and $\mathcal{M}_{F(X)}^v = (Q, E_v, s, t, \lambda)$ do not commute. That is,

$$A_h = \begin{array}{ccccc} 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{array} \quad \text{and} \quad A_v = \begin{array}{ccccc} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{array}$$

in which case

$$A_h A_v = \begin{array}{ccccc} 2 & 2 & 2 & 0 & 1 \\ 2 & 2 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 2 & 2 & 2 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{array} \neq A_v A_h = \begin{array}{ccccc} 2 & 1 & 2 & 0 & 2 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 2 & 1 & 2 & 0 & 2 \\ 2 & 1 & 2 & 0 & 2 \end{array}$$

For future reference, we comment here that the labeled states in the graph of Figure 2.9 could alternately be described as the set of 2×2 blocks that are populated from the alphabet $\Sigma = \{a, b\}$ according to the restriction that each block contain “at least three

a 's". In a similar fashion, other two-dimensional shift spaces can be constructed by first defining the set of states according to restrictions placed on a set of localized pictures and then describing the transitions that exist between these states. For example, in Chapter 4 we shall make reference to the shift space represented by a graph comprised of 15 states defined by restricting the set of 2×2 blocks to those having "at least one a ". In the next section, we examine the relationship between graphs and localized pictures from a different angle: that is, given a collection of vertices and a set of horizontal and vertical edges connecting them, what localized pictures are either forbidden or forced as a result of such a (sub)graph?

2.3 Forbidden and Forced Structures in $\mathcal{M}_{F(X)}$ and $\mathcal{M}_{F(X)}^\Phi$

In one-dimensional algebraic automata theory, it is known that a language L is a factorial and prolongable recognizable (FPR) language iff it is recognized by a finite automaton $\mathcal{M} = \{Q, E, s, t, \lambda\}$ where every state $q \in Q$ is both initial and terminal and both source s and target t are surjective functions [3]. Take a (one-dimensional) finite automaton $\mathcal{M} = (Q, E, s, t, \lambda)$ and partition the edges into horizontal and vertical transitions, denoted by $M^h = (Q, E_h, s, t, \lambda)$ and $M^v = (Q, E_v, s, t, \lambda)$ respectively. We say that a vertex q is

- a *two-dimensional sink* if q is a sink with respect to either M^h or M^v ,
- a *two-dimensional source* if q is a source with respect to either M^h or M^v , and
- is *block isolated* if q is isolated with respect to either \mathcal{M}^h or \mathcal{M}^v .

By construction, the finite automata $\mathcal{M}_{F(X)}$ and $\mathcal{M}_{F(X)}^\Phi$ representing two-dimensional sofic shift spaces will contain no block-isolated vertices, no two-dimensional sinks, and no two-dimensional sources. Likewise, any subgraph representing a point in the shift space must contain both incoming and outgoing horizontal and vertical edges for each state. However, since it is undecidable whether $A(X) = F(X)$ for two-dimensional shift spaces, sufficient conditions for a graph representation to be that of a two-dimensional factorial and prolongable recognizable (2DFPR) language must also be undecidable.

When it is known that $A(X) = F(X)$ for a two-dimensional shift space X , the $\mathcal{M}_{F(X)}$ construction makes it possible to document certain subgraphs as either forbidden or forced due to the nature of incoming and outgoing edges that connect nearby states. These observations regarding (sub)graphs are based on the fact that the graph's transitions are defined for a set of states Q having distinct labels. To begin, Proposition 2.3.1 states that certain combinations of paths of length 2 will never appear in a graph underlying the $\mathcal{M}_{F(X)}$ construction.

Proposition 2.3.1 *Let $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$ be the representation of the two-dimensional shift of finite type X . For $q, q', q'', r \in Q$, suppose there exists two h -paths in $\mathcal{M}_{F(X)}$ having length 2 that connect states q and r , that is, $\Lambda_1 = q \rightarrow q' \rightarrow r$ and $\Lambda_2 = q \rightarrow q'' \rightarrow r$. Then $\Lambda_1 = \Lambda_2$; that is, $q' = q''$.*

Proof. Given $q \rightarrow q'$ and $q \rightarrow q''$, we have that

$$\begin{array}{ccccccccc} q'_{(0,k-1)} & \cdots & q'_{(k-2,k-1)} & & q_{(1,k-1)} & \cdots & q_{(k-1,k-1)} & & q''_{(0,k-1)} & \cdots & q''_{(k-2,k-1)} \\ \vdots & \ddots & \vdots & = & \vdots & \ddots & \vdots & = & \vdots & \ddots & \vdots \\ q'_{(0,0)} & \cdots & q'_{(k-2,0)} & & q_{(1,0)} & \cdots & q_{(k-1,0)} & & q''_{(0,0)} & \cdots & q''_{(k-2,0)} \end{array}$$

Given that $q' \rightarrow r$ and $q'' \rightarrow r$, we have that

$$\begin{array}{ccccccccc} q'_{(1,k-1)} & \cdots & q'_{(k-1,k-1)} & & r_{(0,k-1)} & \cdots & r_{(k-2,k-1)} & & q''_{(1,k-1)} & \cdots & q''_{(k-1,k-1)} \\ \vdots & \ddots & \vdots & = & \vdots & \ddots & \vdots & = & \vdots & \ddots & \vdots \\ q'_{(1,0)} & \cdots & q'_{(k-1,0)} & & r_{(0,0)} & \cdots & r_{(k-2,0)} & & q''_{(1,0)} & \cdots & q''_{(k-1,0)} \end{array}$$

Since the labels are the same, it must be the case that $q' = q''$. ■

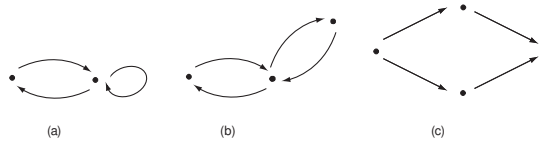


Figure 2.10: In $\mathcal{M}_{F(X)}^h$, graph diamonds of length 2 are forbidden.

So with respect to the horizontal transitions E_h of $\mathcal{M}_{F(X)}$, *graph diamonds* of size 2 are forbidden. Figure 2.10 shows some forbidden (sub)graphs for $\mathcal{M}_{F(X)}$ and $\mathcal{M}_{F(X)}^\Phi$.

Proposition 2.3.1 can naturally be restated in terms of vertical transitions. Furthermore, the result holds true for any graph $\mathcal{M}_{F(X)}^\Phi$ representing the image of X under a block code: although the labels on the states of the new graph need not be distinct, the edges are defined based on the transitions that exist between the distinct states in the preimage and are not altered under any block code that may be applied. (Only the labels on the edges are altered.)

Corollary 2.3.2 *For $\mathcal{M}_{F(X)}$ and $\mathcal{M}_{F(X)}^\Phi$, all h -cycles and v -cycles of length 2 must be disjoint.*

Taken separately, $\mathcal{M}_{F(X)}^h$ and $\mathcal{M}_{F(X)}^v$ are true vertex shifts, but when viewed as one graph, we see that the edge set of $\mathcal{M}_{F(X)}$ does not preclude the existence of both $q \uparrow r$ and $q \rightarrow r$. This results in certain labels being forced upon the states based on the definition of transitions that occur in $\mathcal{M}_{F(X)}$, which thereby forces certain localized pictures. Proposition 2.3.3 and its corollaries hold true for both $\mathcal{M}_{F(X)}$ and $\mathcal{M}_{F(X)}^\Phi$. (The proof is provided for the case $\mathcal{M}_{F(X)}$ only.)

Proposition 2.3.3 *Suppose that $(Q, E, s, t, \lambda) = \mathcal{M}_{F(X)}$, where the states $Q = F_{k,k}(X)$ are populated by choices from the alphabet Σ . If $q, r \in Q$ are such that $q \uparrow r$ and $q \rightarrow r$, then it must be the case that*

$$q = \begin{array}{cc} B & C \\ A & B \end{array} \quad \text{and} \quad r = \begin{array}{cc} C & D \\ B & C \end{array}$$

for some $A, B, C, D \in \Sigma^{(k-1), (k-1)}$.

Proof. Suppose the states of $\mathcal{M}_{F(X)}$ are of size $k \times k$; we shall employ a higher block code to change the alphabet and thereby relabel states q and r . Specifically, rename subblocks of state q as follows.

$$A := \begin{array}{ccc} q_{(0,k-2)} & \cdots & q_{(k-2,k-2)} \\ \vdots & \ddots & \vdots \\ q_{(0,0)} & \cdots & q_{(k-2,0)} \end{array} \quad B := \begin{array}{ccc} q_{(1,k-2)} & \cdots & q_{(k-1,k-2)} \\ \vdots & \ddots & \vdots \\ q_{(1,0)} & \cdots & q_{(k-1,0)} \end{array}$$

$$\begin{array}{ccc}
q_{(0,k-1)} & \cdots & q_{(k-2,k-1)} \\
H := & \vdots & \ddots \\
q_{(0,1)} & \cdots & q_{(k-2,1)}
\end{array}
\quad
\begin{array}{ccc}
q_{(1,k-1)} & \cdots & q_{(k-1,k-1)} \\
C := & \vdots & \ddots \\
q_{(1,1)} & \cdots & q_{(k-1,1)}
\end{array}$$

In the same way, rename subblocks of state r as follows.

$$\begin{array}{ccc}
r_{(0,k-2)} & \cdots & r_{(k-2,k-2)} \\
E := & \vdots & \ddots \\
r_{(0,0)} & \cdots & r_{(k-2,0)}
\end{array}
\quad
\begin{array}{ccc}
r_{(1,k-2)} & \cdots & r_{(k-1,k-2)} \\
F := & \vdots & \ddots \\
r_{(1,0)} & \cdots & r_{(k-1,0)}
\end{array}$$

$$\begin{array}{ccc}
r_{(0,k-1)} & \cdots & r_{(k-2,k-1)} \\
G := & \vdots & \ddots \\
r_{(0,1)} & \cdots & r_{(k-2,1)}
\end{array}
\quad
\begin{array}{ccc}
r_{(1,k-1)} & \cdots & r_{(k-1,k-1)} \\
D := & \vdots & \ddots \\
r_{(1,1)} & \cdots & r_{(k-1,1)}
\end{array}$$

In effect then, q and r may be viewed as 2×2 states under this new (sliding) block code so that

$$q = \begin{array}{cc} H & C \\ A & B \end{array} \quad \text{and} \quad r = \begin{array}{cc} G & D \\ E & F \end{array}.$$

The condition $q \rightarrow r$ forces $E = B$ and $C = G$, while the condition that $q \upharpoonright r$ forces $C = G = F$ and $H = E = B$. The two conditions together imply that

$$q = \begin{array}{cc} B & C \\ A & B \end{array} \quad \text{and} \quad r = \begin{array}{cc} C & D \\ B & C \end{array}.$$

■

As demonstrated in the proof of Proposition 2.3.3, we may always assume a higher block code as needed. We do so in the sequel without any loss of generality. In particular, unless stated otherwise, all results to follow assume 2×2 states are used in the construction of finite automata.

The following Corollaries will be useful in the discussion of periodic points.

Corollary 2.3.4 *Suppose that $(Q, E, s, t, \lambda) = \mathcal{M}_{F(X)}$. If $q, r \in Q$ are such that $q \upharpoonright r$ and*

$r \rightarrow q$, then it must be the case that

$$q = \begin{array}{cc} C & A \\ A & B \end{array} \quad \text{and} \quad r = \begin{array}{cc} A & B \\ B & D \end{array} .$$

Corollary 2.3.5 *Suppose that $(Q, E, s, t, \lambda) = \mathcal{M}_{F(X)}$. If $q, r \in Q$ are such that $q \upharpoonright r, q \rightarrow r$, and $r \upharpoonright q$, then it must be the case that*

$$q = \begin{array}{cc} B & A \\ A & B \end{array} \quad \text{and} \quad r = \begin{array}{cc} A & B \\ B & A \end{array} .$$

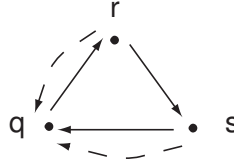


Figure 2.11: Horizontal graph triangle of Corollary 2.3.6

Corollary 2.3.6 *Suppose that $(Q, E, s, t, \lambda) = \mathcal{M}_{F(X)}$. If $q, r, s \in Q$ are such that $s \upharpoonright q, s \rightarrow q, r \upharpoonright q, q \rightarrow r$, and $r \rightarrow s$, then*

$$q = \begin{array}{cc} A & A \\ A & A \end{array} \quad r = \begin{array}{cc} A & A \\ A & B \end{array} \quad \text{and} \quad s = \begin{array}{cc} A & A \\ B & A \end{array}$$

Corollary 2.3.6 has several variations, all of which contain h -cycles or v -cycles of length three. We refer to such subgraphs as *graph triangles* and shall take advantage of the appearance of horizontal (vertical) graph triangles in the discussion of periodicity: While not forbidden, graph triangles in $\mathcal{M}_{F(X)}$ indicate much about the labels on certain states and thereby force some localized pictures and forbid others.

3 TRANSITIVITY

Now that we have defined a graph representation for a class of two-dimensional shift spaces, we may examine the relationship between a directed graph and the two-dimensional FPR language that it represents. For example, in one-dimensional symbolic dynamics, it is known that an essential graph (one in which all states have both incoming and outgoing edges) is transitive if and only if its edge shift is transitive [25]. (In graph theory, transitive graphs are referred to as strongly connected graphs.) As we have seen, there are different types of two-dimensional transitivity and mixing, each of which defines an invariant properties for conjugate shift spaces. In this chapter, we employ dot systems over $\mathbb{Z}/2\mathbb{Z}$ to illustrate these various types of two-dimensional transitivity and elaborate on results that are published in [16] regarding transitivity in two-dimensional local languages defined by dot systems. In Section 3.1, it is shown that the factor language of a dot system defined by a rectangular shape lacking free cells can not be transitive in certain directions although the same factor language will be uniformly transitive. It is also shown that the factor language of a dot system defined by a triangular shape will be mixing but not uniformly so. In Section 3.2, transitivity in the graph of $\mathcal{M}_{F(X)}$ is related to (uniform) directional transitivity in the represented language. A main result is given in Theorem 3.2.6, which states that for the shift space X having $F(X) = A(X)$, it is decidable whether $F(X)$ exhibits uniform horizontal transitivity at a given distance K .

3.1 Factor Languages of Dot Systems Defined over $\mathbb{Z}/2\mathbb{Z}$

The investigation into the transitivity of a factor language of a dot system is connected to the existence (or lack thereof) of free cells in the defining shape S . The following example introduces this idea.

Example 3.1.1 Let $S = \{(0, 0), (1, 0), (0, 1), (1, 1)\}$ define a dot system X , which we refer to in this paper as the Full-square System. Define $\beta_0 : S \mapsto \{0\}$ and $\beta_1 : S \rightarrow \{0, 1\}$

with $\beta_1(i, j) = (i + j) \bmod 2$. Then $\{\beta_0, \beta_1\} \subset F(X)$, but these blocks can not meet in direction $(1, 0)$. To see this, suppose we wanted to extend block β_0 in the $(1, 0)$ direction by horizontal 2-concatenation with a 2×1 block β' to produce the block $B_{2,3} \in F(X)$. Since a horizontal translation of the shape S intersects the block β_0 at the cells $\{(1, 0), (1, 1)\}$ and $\beta_0(1, 0) + \beta_0(1, 1) = 0 + 0 = 0$, the block $B_{2,3}$ will be in the factor language only if $\beta'(0, 0) = \beta'(0, 1) \Leftrightarrow \beta'(0, 0) + \beta'(0, 1) \equiv (0 \bmod 2)$. This argument can be repeated indefinitely, where the sum of each new column is required to be equivalent to zero. For β_1 however, $\beta_1(0, 0) + \beta_1(0, 1) = 0 + 1 = 1$ so that β_1 can never appear as a subblock of any block extension of β_0 . Therefore, the factor language $F(X)$ can not be horizontally transitive. The same blocks can be used in a similar argument to demonstrate that $F(X)$ can not be vertically transitive.

We find that for dot systems, the existence of free cells in the (non-singleton) rectangle T that minimally contains the shape S is a necessary condition for the factor language of the dot system to be vertically and/or horizontally transitive. We state the horizontal case only, as the vertical case can be shown analogously.

Proposition 3.1.2 *Suppose X is a dot system defined through some $r \times c$ shape S such that $c > 1$ and let T be a normalized $r \times c$ shape. If T has no free cells, i.e. $T = S$, then $F(X)$ can not be horizontally transitive.*

Proof. (The Wallpaper Pattern) Summations in the following discussion are carried out modulo 2. Suppose X is a dot system defined through an $r \times c$ rectangular shape S having $c > 1$. Note that for all $B'_{r,c} \in F(X)$, since $\sum_{i=0}^{c-1} \sum_{j=0}^{r-1} B'_{r,c}(i, j) = 0$, we have that

$$\forall i' \in \{0, 1, \dots, c-2\}, \sum_{i=0}^{i'} \sum_{j=0}^{r-1} (B'_{r,c}(i, j)) = \sum_{i=i'+1}^{c-1} \sum_{j=0}^{r-1} (B'_{r,c}(i, j)). \quad (3.1.1)$$

Denote with b_i the sum of the bits in the i -th column, i.e., $b_i = \sum_{j=0}^{r-1} B'_{r,c}(i, j)$ for $i = 0, \dots, c-1$. Then the equality given in (3.1.1) implies that $b_0 = \sum_{i=1}^{c-1} b_i$ and $\sum_{i=0}^{c-2} b_i = b_{c-1}$. Now utilize block extension to construct an additional column to the right of the original block, thereby forming $B_{r,c+1} \in F(X)$. Let b_c be the sum of the bits in the $c+1$ -st column.

We see that given any $B'_{r,c} \in F(X)$,

$$B'_{r,c} \subseteq B_{r,c+1} \in F(X) \quad \Leftrightarrow \quad b_0 = \sum_{i=1}^{c-1} b_i = b_c \quad (3.1.2)$$

This is depicted in diagram (a) in Figure 3.1: The shaded columns must each have the same sum as that of the sum taken over the non-shaded columns combined. If we inspect the sequence of 0's and 1's obtained by summing over each column of the original $r \times c$ block, we discover that block extensions will inductively produce this same sequence of columnar sums in a wallpaper-like pattern. In diagram (b) of Figure 3.1, columns with the same shading have the same sum. In particular, if b_i denotes the sum of the bits in the i -th column of a block $B_{r,c+k}$ for $k > 0$, then $b_{c+k-1} = b_q$ where $q \equiv (k \bmod c)$.

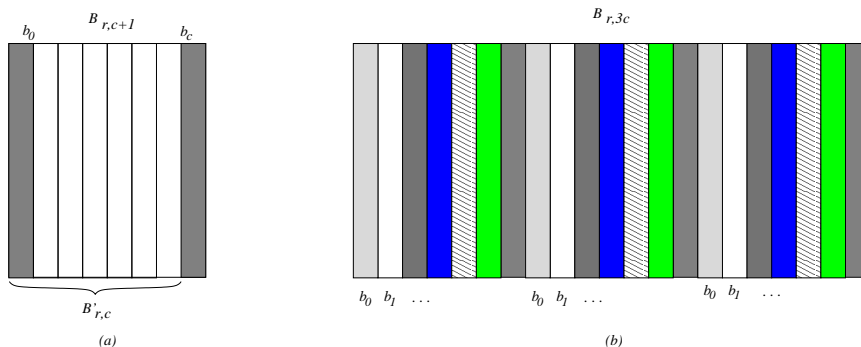


Figure 3.1: The Wallpaper Pattern

Finally, consider the specific block $B'_{r,c} \in F(X)$ that has $B'_{r,c}(0,0) = B'_{r,c}(c-1,r-1) = 1$ and $B'_{r,c}(i,j) = 0$ otherwise. Then for any block $B_{r,n} \in F(X)$ of which $B'_{r,c}$ is a subblock, $B_{r,n}$ cannot contain the $r \times c$ block consisting of all zeros as a subblock. Yet, this subblock of all zeros is contained in the factor language of every dot system. This demonstrates the existence of a pair of blocks that cannot meet in direction $(1,0)$ within $F(X)$. ■

In the sequel, we shall make frequent reference to wallpaper patterns. By this, we refer to the fact that for dot systems defined by an $r \times c$ rectangular shape that lacks free cells, the sums taken over the individual columns of any allowed block having height r and length $n > c$ must obey a repeating pattern of length c . Note, however, that c need not be the minimum pattern length, as certain wallpaper patterns will contain subpatterns.

Example 3.1.3 Suppose $r = 1$ in the conditions of Proposition 3.1.2; for example, say the dot system X is defined by the 1×2 shape $S = \{(0, 0), (1, 0)\}$. Then it is trivial to show that $F(X)$ exhibits vertical transitivity. However, $F(X)$ is not horizontally transitive since $\beta_0 : S \mapsto \{0\}$ and $\beta_1 : S \mapsto \{1\}$ can never meet in direction $(1, 0)$. Rather, for any $B_{m,n} \in F(X)$, $B_{m,n}(i, j) = B_{m,n}(i + 1, j)$ for all $i \in \{0, 1, \dots, n - 2\}$, $j \in \{0, 1, \dots, m - 1\}$. In this case, the wallpaper pattern of length $c = 2$ is also a pattern of length 1.

The existence of free cells in the rectangle T that minimally contains the shape S , while necessary, is not however a sufficient condition for the factor language of the dot system to be vertically and/or horizontally transitive. To see this, we will generalize the rectangular shapes by considering shapes that appear as any set of parallel lines comprised of the same number of dots. More formally, given two non-zero vectors (u', v') and (u'', v'') with integer coordinates having greatest common divisors $d' = \gcd(u', v')$ and $d'' = \gcd(u'', v'')$, we say that a shape S is a *parallelogram with defining non-zero vectors (u', v') and (u'', v'')* if $S = \{\frac{a}{d'}(u', v') + \frac{b}{d''}(u'', v'') : a = 0, 1, \dots, d', b = 0, 1, \dots, d''\}$. By this definition, a rectangular shape with one column or one row (e.g., Example 3.1.3) is not a parallelogram, since one of the defining vectors would be the zero-vector. Similarly, the shape (b) in Figure 1.3 is not technically a parallelogram. Notice that many parallelograms naturally have free cells in the rectangle T that minimally contains the shape: for example, the shape in Figure 1.3 (c) is a parallelogram with defining vectors $(1, 1)$ and $(-1, 1)$ and has free cells in the 3×3 rectangle T . The shape in Figure 1.3 (d) is a parallelogram defined with vectors $(0, 2)$ and $(2, 0)$ that leaves no free cells in the 3×3 rectangle T . The proof of Proposition 3.1.2 can easily be adjusted to show the following Corollary.

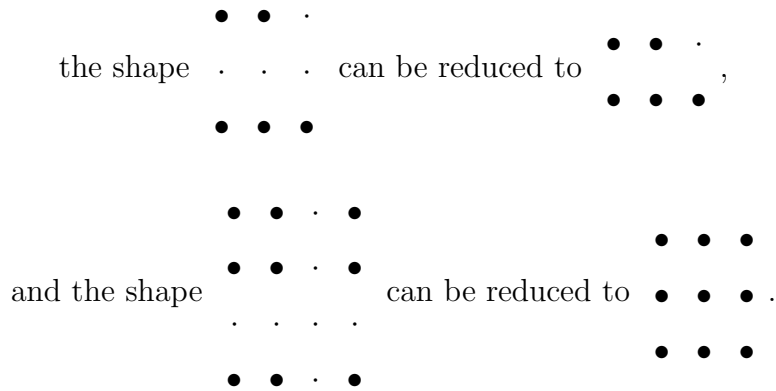
Corollary 3.1.4 *If S is a parallelogram with defining vectors (u', v') and (u'', v'') , then the factor language $F(X)$ of the dot system defined by S is not transitive in direction (u', v') nor in direction (u'', v'') .*

Proof. Suppose S has dimension $r \times c$, let T be the $r \times c$ rectangular shape that minimally contains S , and let $\beta_0 : T \mapsto \{0\}$. Say $i' = \min\{i : (i, 0) \in S\}$, and say $i'' = \max\{i : (i, r - 1) \in S\}$. Toward a counterexample, define a block $\beta_1 \in \{0, 1\}^{r \times c} \cap F(X)$ such that $\beta_1(i', 0) = 1$, $\beta_1(i'', r - 1) = 1$, and $\beta_1(i, j) = 0$ otherwise. Then $\beta_0, \beta_1 \in F(X)$, but it can be proven similarly as in Proposition 3.1.2 that β_0 can not meet β_1 in direction (u', v') nor

direction (u'', v'') within $F(X)$. ■

So in particular, the shape $S = \{(0, 0), (1, 0), (1, 1), (2, 1)\}$ with defining vectors $(1, 0)$ and $(1, 1)$, depicted in Figure 1.3 (e), leaves free cells in the rectangle T , yet the factor language defined by S can not be horizontally transitive. This example serves to illustrate that the existence of free cells in the defining shape does not guarantee horizontal (vertical) transitivity. Another (non-parallelogram) example is given by the following dot system, which has a “hole” in the defining shape $S = \{(0, 0), (1, 0), (2, 0), (0, 1), (2, 1), (0, 3), (1, 3), (2, 3)\}$. Let $\beta_1 \in F_{3,3}(X)$ be such that $\beta_1(1, 1) = 1$, and $\beta_1 = 0$ otherwise; and let $\beta_0 \in F_{3,3}(X)$ be such that $\beta_0(i, j) = 0$ for all $0 \leq i \leq 2, 0 \leq j \leq 2$. Using an argument similar to that of the Wallpaper Pattern, the reader can verify that β_0 and β_1 may never meet in direction $(1, 0)$ within $F(X)$.

The definition of dot systems does not preclude shapes that are disconnected. In fact, shapes that are disconnected may define dot systems having factor languages with the same properties as those corresponding to shapes that are (simply) connected. For example, if a shape S is minimally contained in a rectangle T that has a row or a column comprised entirely of free cells, we say that S can be *reduced* to S' where S' is the shape obtained from S by erasing the rows or columns of T that contain only free cells. For example,



The method of proof employed in the proof of Proposition 3.1.2 also reveals the following.

Corollary 3.1.5 *If a shape S can be reduced to a shape S' that is a rectangle, then S defines a dot system X with factor language that lacks horizontal and/or vertical transitivity.*

Proof. Suppose S has dimension $r \times c$ and without loss of generality suppose that $c > 1$. (We are interested in non-singleton shapes where $r > 1$ or $c > 1$. Note that for a dot

system X defined through a shape having $r = 1$ and $c > 1$, $F(X)$ will trivially exhibit vertical transitivity but not horizontal transitivity.) Define $\beta_0 : T \mapsto \{0\}$, where T is the $r \times c$ rectangular shape that minimally contains S . Note that if S can be reduced to a shape S' that is a rectangle, then $(0, 0), (c-1, r-1) \in S$. Toward a counterexample, define a block $\beta_1 \in \{0, 1\}^{r,c} \cap F(X)$ such that $\beta_1(0, 0) = \beta_1(c-1, r-1) = 1$, and $\beta_1(i, j) = 0$ otherwise. Then $\beta_0, \beta_1 \in F(X)$, but β_0 can not meet β_1 in direction $(1, 0)$ within $F(X)$. To see this, we can modify the proof of Proposition 3.1.2 to include only those cells that intersect horizontal translates of S . First adjust (3.1.1) to reflect the fact that sums are taken only over the cells of a block $B_{r,c}$ that lie in horizontal translates of S .

$$\forall i' \in \{0, 1, \dots, c-2\}, \quad \sum_{w \in \{(i'-(c-1), 0)+S\} \cap T} (B_{r,c}(w)) = \sum_{w \in \{(i'+1, 0)+S\} \cap T} (B_{r,c}(w))$$

In a similar fashion, the remainder of the proof of Proposition 3.1.2 can be modified to reflect that when using block extension to construct additional columns to the right of a block $B_{r,c}$, the sum is taken only over those bits that intersect horizontal translates of S . In this context, β_0 and β_1 have wallpaper patterns that cannot meet in direction $(1, 0)$. Given that $r > 1$, the same blocks can be used to show that $F(X)$ lacks vertical transitivity. ■

Corollary 3.1.6 *If a shape S can be reduced to a shape S' that is a parallelogram, then S defines a dot system with factor language that lacks directional transitivity in the direction of the defining vectors.*

For one class of shapes, it is true that the existence of a free cell is both necessary and sufficient for the corresponding factor language to be both horizontally and vertically transitive. The proof of the following proposition highlights the framework of carefully selecting both the point of origin and the direction in which to begin the inductive process of “filling a block”.

Proposition 3.1.7 *For dot systems defined through a 2×2 shape S , the existence of a free cell is sufficient for the factor language of the dot system to be both vertically and horizontally transitive. The factor language corresponding to such shapes is neither uniformly horizontally transitive nor uniformly vertically transitive.*

Proof. We will prove horizontal transitivity by induction on the number of rows m in a pair of blocks. (Presented with two blocks from $F(X)$ having different heights, we can always use block extension to extend the shorter block to another block in $F(X)$ having greater height.) The vertical case can be shown analogously.

Use $m = 2$ for the basis step, since $\{0, 1\}^{1,n} \subset F(X)$. Let β_0 represent the 2×2 block consisting of all zeros, which is an element of the factor language for all dot systems. To establish the basis step, we will first show that for every block $B'_{2,n} \in \{0, 1\}^{2,n} \cap F(X)$, $B'_{2,n}$ meets β_0 in direction $(1, 0)$ within $F(X)$. In fact, β_0 meets every block in direction $(1, 0)$ at distance ≤ 2 . Say $B'_{2,n} : T' \rightarrow \{0, 1\}$: if $\sum_{w \in \{(n-1,0)+S\} \cap T'} (B'_{2,n}(w)) = 0$, then $B'_{2,n}$ meets β_0 in direction $(1, 0)$ at distance 0 within $F(X)$, and the result is trivial. So consider the case when $\sum_{w \in \{(n-1,0)+S\} \cap T'} (B'_{2,n}(w)) = 1$. We can define a block $B \in \{0, 1\}^{2,n+3} \cap F(X)$ that minimally encloses $B'_{2,n}$ and β_0 in the following way.

$$B(i, j) = \begin{cases} B'_{2,n}(i, j) & \text{for } 0 \leq i \leq n-1, 0 \leq j \leq 1 \\ * & \text{for } i = n, 0 \leq j \leq 1 \\ \beta_0(i - (n+1), j) & \text{for } n+1 \leq i \leq n+2, 0 \leq j \leq 1 \end{cases} \quad (3.1.3)$$

In (3.1.3), $*$ is determined by the location of the free cell(s) in the 2×2 rectangle T : If neither $(1, 0)$ nor $(1, 1)$ is a free cell, let j' denote the ordinate of the free cell $(0, j')$ and then define $B(n, j') = 1, B(n, (j' + 1) \bmod 2) = 0$; If $(1, 0)$ is a free cell, then $B \in F(X)$ only if $B(n, 1) = 1$, so we define $B(n, 0) = 0$ if $(0, 1)$ is also a free cell, but define $B(n, 0) = 1$ otherwise; Similarly, if $(1, 1)$ is a free cell, then $B \in F(X)$ only if $B(n, 0) = 1$, so we define $B(n, 1) = 0$ if $(0, 0)$ is also a free cell, but define $B(n, 1) = 1$ otherwise. By a symmetric argument, we can also show that β_0 meets any block $B_{2,n}$ in direction $(1, 0)$ within $F(X)$, and hence any two blocks of height 2 meet in direction $(1, 0)$ within $F(X)$.

Next, assume by induction that for any two blocks $B', B'' \in F(X)$ having height m , there exists a block B that encloses B' and B'' in direction $(1, 0)$ within $F(X)$. Then suppose we are given $B'_{m+1,u}$ and $B''_{m+1,v} \in F(X)$. The proof is completed by the inspection of four cases based upon the location of the free cell(s).

case i): $(0, 0)$ is a free cell. (See diagram (a) in Figure 3.2, where darkly-shaded regions indicate the shape S .)

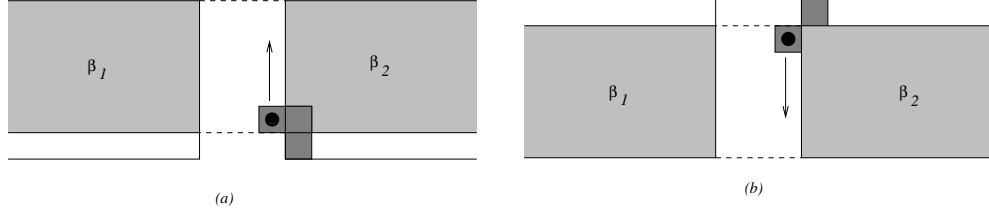


Figure 3.2: The inductive step

Let $\beta_1 = \{B'_{m+1,u}(i, j) : 0 \leq i \leq u - 1, 1 \leq j \leq m\}$, and similarly, let $\beta_2 = \{B''_{m+1,v}(i, j) : 0 \leq i \leq v - 1, 1 \leq j \leq m\}$. We first want to extend β_2 to the left by one column and thereby construct a new block β'_2 . Note that, based on the shape S , the assignment of $\beta'_2(0, 0)$ is arbitrary: we assign the value on this cell based upon entries in the original block $B''_{m+1,v}$ and then inductively define the remainder of the cells in the new column upward from this cell. More formally,

$$\beta'_2(i, j) := \begin{cases} \beta_2(i - 1, j) & \text{for } 1 \leq i \leq v, 0 \leq j \leq m - 1 \\ B''_{m+1,v}(0, j) + B''_{m+1,v}(0, j + 1) & \text{for } 0 \leq j \leq m - 1 \end{cases} \quad (3.1.4)$$

(Use definition (3.1.4) for β'_2 when $(0, 0)$ is the only free cell; If $(1, 1)$ is also a free cell, do not include the addend $B''_{m+1,v}(0, j + 1)$ in the last line of the definition.) By the induction hypothesis, there exists $B^*_{m,n} \in F(X)$ such that $B^*_{m,n}$ encloses β_1 and β'_2 in direction $(1, 0)$ within $F(X)$. Now label an $(m + 1) \times n$ rectangular shape as T , and begin to define $B : T \rightarrow \{0, 1\}$ in the following way.

$$B(i, j) = \begin{cases} B'_{m+1,u}(i, j) & \text{for } 0 \leq i \leq u - 1, 0 \leq j \leq m \\ B^*_{m,n}(i, j - 1) & \text{for } u \leq i \leq n - 2 - v, 1 \leq j \leq m \\ \beta'_2(i - ((n - 1) - v), j - 1) & \text{for } i = n - 1 - v, 1 \leq j \leq m \\ B''_{m+1,v}(i - (n - v), j) & \text{for } n - v \leq i \leq n - 1, 0 \leq j \leq m \end{cases} \quad (3.1.5)$$

To complete the proof of the first case, define $B(i, 0) = B(i, 1) + B(i - 1, 1)$ for $u \leq i \leq (n - 1) - v$. The location of the free cell guarantees that the assignment of these remaining cells does not disrupt the overall acceptance of the block B . Therefore, there exists B that encloses $B'_{m+1,u}$ and $B''_{m+1,v}$ in direction $(1, 0)$ within $F(X)$.

The other three cases are analogous to case (i) when slight modifications are made initially.

case ii): $(1, 0)$ is a free cell.

Define β_1 and β_2 as in case (i), but extend β_1 one column to the right to construct a new block β'_1 before applying the induction hypothesis.

case iii): $(0, 1)$ is a free cell. (For example, see diagram (b) in Figure 3.2.)

Let $\beta_1 = \{B'_{m+1,u}(i, j) : 0 \leq i \leq u - 1, 0 \leq j \leq m - 1\}$, and similarly, let $\beta_2 = \{B''_{m+1,v}(i, j) : 0 \leq i \leq v - 1, 0 \leq j \leq m - 1\}$. Then extend β_2 to the left one column to form β'_2 by first defining the cell $\beta'_2(0, m - 1) = B''_{m+1,v}(0, m - 1) + B''_{m+1,v}(0, m)$. (If $(1, 0)$ is also a free cell as in Figure 3.2 (b), either do not include the addend $B''_{m+1,v}(0, m - 1)$ or use case (ii) discussed above.) Inductively define the remaining cells in the new column downward from this cell before applying the induction hypothesis.

case iv): $(1, 1)$ is a free cell.

Define β_1 and β_2 as in case (iii), but instead extend β_1 one column to the right before applying the induction hypothesis.

Toward a counterexample to the question of uniform directional transitivity, consider the Three-dot System X and suppose that $F(X)$ is uniformly horizontally (vertically) transitive within distance K . Let β_0 be the $(K + 2) \times (K + 2)$ block of all zeros, and define a block $B_{K+2,K+2} \in F(X)$ having all entries zero with the exception of the top-right corner. More formally,

$$B_{K+2,K+2}(i, j) = \begin{cases} 1 & \text{if } (i, j) = (K + 1, K + 1) \\ 0 & \text{otherwise} \end{cases} \quad (3.1.6)$$

For $i = \{1, 2, \dots, K + 1\}$, a block extension of i columns to the right of $B_{K+2,K+2}$ is uniquely determined by translates of S up to height $K - i$ for the i th column. (See Figure 3.4 for an example.) Therefore, $B_{K+2,K+2}$ can meet β_0 in direction $(1, 0)$ (or in direction $(0, 1)$) within $F(X)$ only at some distance $k \geq K + 1$. The same blocks provide a counterexample to uniform vertical and horizontal transitivity in the factor language corresponding to the shape $S = \{(1, 0), (0, 1)\}$. All other shapes are analogous. ■

The Three-dot System is a member of the class of shapes encompassed by Proposition 3.1.7. Since the Three-dot System is known to be mixing (see [19, 20, 30]), it is already known to be both vertically and horizontally transitive. However, not all shapes in this

class have factor language that is mixing. For example, it is easily seen that the factor language of the dot system defined by the shape represented in Figure 1.3 (b) is a member of this class of shapes yet can not be transitive in direction $(1, 1)$. It is interesting to note that while the factor languages corresponding to certain shapes may lack transitivity in specific directions, these languages may still be uniformly transitive in general.

Proposition 3.1.8 *Let X be a dot system defined through an $r \times c$ rectangular shape S (i.e. $T = S$), and let $F(X)$ be its factor language. Then $F(X)$ is uniformly transitive. $F(X)$ is mixing if and only if S is a singleton.*

Proof. We first consider the case where $r = c = 1$. If $r = c = 1$, then S is a singleton and X contains only the one point comprised of all zeros. In this case, $F(X)$ is the set of all rectangular blocks comprised entirely of zeros and as such it is (uniformly) mixing. Conversely, rectangular shapes comprised of more than one dot define languages that lack horizontal and/or vertical transitivity (Proposition 3.1.2) and hence cannot be mixing.

For all other cases, let $B', B'' \in F(X)$ be two rectangular blocks of size $m_i \times n_i$ ($i = 1, 2$), respectively. To ease notation, assume without loss of generality that $m_i \geq r$ and $n_i \geq c$. If not, first use block extensions as needed to extend B' down and/or to the left and then to extend B'' up and/or to the right before finding a block B^* that encloses these larger blocks. The block B that minimally encloses B' and B'' would then be a subblock of B^* .

For the case when $r = 1$ and $c > 1$, assume without loss of generality that $n_1 \leq n_2$. By definition of $F(X)$, there must exist some block $\beta_1 \in F(X)$ that is an extension of B' by $n_2 - n_1$ columns to the right. We define $B \in \{0, 1\}^{(m_1+m_2), n_2} \cap F(X)$ as follows:

$$B(i, j) = \begin{cases} \beta_1(i, j) & \text{for } 0 \leq i \leq n_2 - 1, 0 \leq j \leq m_1 - 1 \\ B''(i, j - m_1) & \text{for } 0 \leq i \leq n_2 - 1, m_1 \leq j \leq m_2 + m_1 - 1 \end{cases} \quad (3.1.7)$$

Since no translate of S will intersect β_1 and B'' simultaneously, we see that $B \in F(X)$. Furthermore, B minimally encloses B' and B'' with distance at which they meet being 0 (the blocks touch). In other words, $F(X)$ is uniformly transitive. The case when $r > 1$ and $c = 1$ is similar.

For the remaining cases, we will show that B' and B'' meet each other at distance 0 by enclosing them in a block B of size $(m_1 + m_2) \times (n_1 + n_2)$ which we will construct by utilizing the set of all translates of S that lie entirely within the block B .

Step 1: Begin defining the block B by defining B' as the bottom-left corner and B'' as the top-right corner. The assignment of the remaining cells is accomplished by the assignment of a centrally-located translate of S : Consider S_α , the translate of S whose top-left cell has coordinates $(n_1 - 1, m_1)$. (See Figure 3.3, where the dotted rectangle denotes the rectangular size of the translate S_α and the top-left cell is shaded.) Other

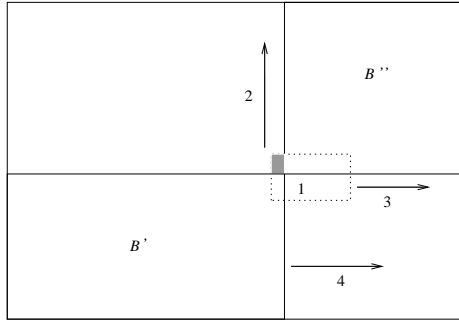


Figure 3.3: The process of filling up the block B

than this cell $(n_1 - 1, m_1)$, all undefined cells of B that intersect S_α are to be populated with zeros. Then define $B(n_1 - 1, m_1)$ as needed to obtain the appropriate sum over S_α .

Step 2: Next we define B inductively on the top-left portion of the shape by using the top-left dot of translates of S_α to guarantee the overall acceptance of the block B . Consider the translate $S_\alpha + (0, 1)$. With the exception of the dot located at $(n_1 - 1, m_1 + 1)$, all dots of this translate are already assigned. Therefore, define $B(n_1 - 1, m_1 + 1)$ as needed to obtain the appropriate sum over $S_\alpha + (0, 1)$. Translating upward for $u = 0, 1, \dots, m_2 - 1$, we may sequentially fill cells by defining $B(n_1 - 1, m_1 + u)$ as needed for the translate $S_\alpha + (0, u)$. Begin the next sequence at $S_\alpha + (-1, 0)$ and for $u = 0, 1, \dots, m_2 - 1$, sequentially define $B(n_1 - 2, m_1 + u)$ as needed for the translate $S_\alpha + (-1, u)$. The remaining cells in the top-left corner of block B are defined sequentially in the same manner.

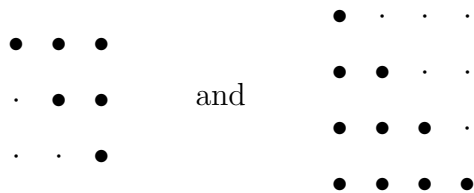
Step 3: Return to S_α and notice that each of the cells originally populated with zeros is the bottom-right dot for some translate of S_α whose top-left dot lies in the region of the block B having already been defined in an acceptable manner. We will define B on the remaining cells in the bottom-right portion of the shape by using the bottom-right dot of translates of S_α in order to guarantee the overall acceptance of block B . First, we define the cells that complete the rows to the right of those cells that were populated with zeros in step 1. Consider $S_{\alpha'} = S_\alpha + (1, r - 2)$. The translate $S_{\alpha'}$ has only the dot located at the bottom-right corner undefined, so at this cell we define $B(n_1 - 1 + c, m_1 - 1)$ as needed

for the appropriate sum over $S_{\alpha'}$. Translating to the right for $v = 0, 1, \dots, n_2 - c$, we may assign $B(n_1 - 1 + c + v, m_1 - 1)$ as needed for the appropriate sum over $S_{\alpha'} + (v, 0)$. Continuing this process for $w = 1, 2, \dots, r - 2$, we can define rows that begin with the translate $S_{\alpha'} + (0, -w)$ and then assign $B(n_1 - 1 + c + v, m_1 - w - 1)$ as needed to obtain the appropriate sum over $S_{\alpha'} + (0, -w) + (v, 0)$ for $v = 0, 1, \dots, n_2 - c$.

Step 4: Finally, we define the remaining cells in the bottom-right portion of the shape. Define $S_{\alpha''} = S_{\alpha} + (-(c - 2), -1)$. Using $w' = 0, 1, \dots, m_1 - r$, we can define rows that begin with the translate $S_{\alpha''} + (0, -w')$ and then assign $B(n_1 + v, m_1 - r)$ as needed to obtain the appropriate sum over $S_{\alpha''} + (0, -w') + (v, 0)$ for $v = 0, 1, \dots, n_2 - 1$.

So the entire $(m_1 + m_2) \times (n_1 + n_2)$ block $B \in F(X)$ can be filled in such a way that B' is in the bottom-left corner and B'' is at the top-right corner. Note that $d(B', B'') = 0$ regardless of the size of the blocks B', B'' . Therefore, $F(X)$ is uniformly transitive. ■

Since it is known that the factor language of the Three-dot System is mixing, the inspection of dot systems that behave like the Three-dot System is a good place to begin further investigations into factor languages which are mixing. We say that an $r \times c$ shape S is *three-dot like* if the dots form a simply-connected isosceles triangle (such that $r = c$ for the $r \times c$ shape S). For example,



are both three-dot like. We can apply the method of proof used in Proposition 3.1.8 to show the following.

Proposition 3.1.9 *If S is a shape that is three-dot like, then S defines a dot system X with factor language that is mixing.*

Proof. Without loss of generality, suppose S is an $r \times r$ triangular shape appearing in the bottom-left corner of the $r \times r$ rectangle T . To begin, let $B_{m,n} \in F(X)$ be such that $m \geq r, n \geq r$. We note some properties of these blocks found in $F(X)$.

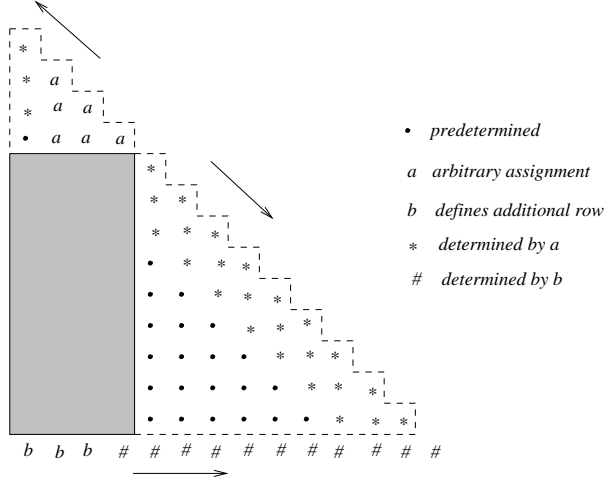


Figure 3.4: Extension of a block into a triangle

i) Any extension to the right of $B_{m,n}$ has $((m - r + 1) + (m - r) + \dots + 1)$ dots predetermined by translates of the shape, and any extension upward of $B_{m,n}$ has $((n - r + 1) + (n - r) + \dots + 1)$ dots predetermined by translates of the shape.

ii) Following the arbitrary assignment of $r - 1$ cells in a column above $B_{m,n}(n - (r - 1), m - 1)$, $r - 2$ cells in a column above $B_{m,n}(n - (r - 2), m - 1), \dots$, and 1 cell above $B_{m,n}(n - 1, m - 1)$, $B_{m,n}$ may be extended along the diagonal from the inside out to take on a triangular design that occurs in some point of the shift space X . (See Figure 3.4 based on a 4×4 triangular shape S , where arrows indicate the direction in which to fill.)

iii) Additional rows below such a triangle may be defined by the arbitrary assignment of a row of $r - 1$ cells directly below the dots in the bottom-left corner of the design (and/or additional columns to the left of such a triangle may similarly be defined).

Now consider any two blocks $B'_{m_1, n_1}, B''_{m_2, n_2} \in F(X)$. We will show that B' meets B'' in every direction (u, v) provided that $d(B', B'') > r + \max\{m_1, m_2, n_1, n_2\}$. Begin by placing the bottom-left corner of B' at $(0, 0)$ and the bottom-left corner of B'' at (u, v) , and then expand both blocks into the triangular designs outlined in (i) and (ii) above. The proof can be completed based on the observation that regardless of the placement, there exist only three shapes that might occur at the point of intersection for two right isosceles triangles.

Consider first the case where $u, v \geq 0$ and expand the triangular extension of B'' below and/or to the left as outlined in (iii) until the bottom-left corner of this triangle touches the triangular extension of block B' . (See diagram (a) in Figure 3.5.) The proof is

completed by filling in the block that minimally contains the two triangles from the inside out in a method similar to that used in Proposition 3.1.8: we will use the bottom-right dot in translates of the shape S to assign values underneath the upper triangle, and we will use the upper-left dot in translates of the shape S to assign values to the left of the upper triangle. Specifically, at the location where the two triangles meet, arbitrarily define enough cells in the left-most portion of the row directly below the upper triangle to begin the induction process that will define the remainder of the cells in this new row. (Note that at the point of intersection there is at most one cell that is already defined by the lower triangle, where $r - 1$ cells must be defined in order to start a new row.) In this way, we may inductively define cells by translating down and starting over on the far left of each new row. Next, the cells to the left of the upper triangle will be defined inductively beginning at the base of the upper triangle and working up each column before translating left to begin at the bottom of the next column. Once we have achieved a single triangular design that encloses the two smaller triangles, the cells above the new diagonal may be filled in a diagonal manner (see shaded regions in Figure 3.5): Begin at the top-left corner with an arbitrary assignment of $r - 1$ cells along the diagonal and then work down each diagonal by checking the bottom-right dot in translates of S before returning to the top to begin the next diagonal.

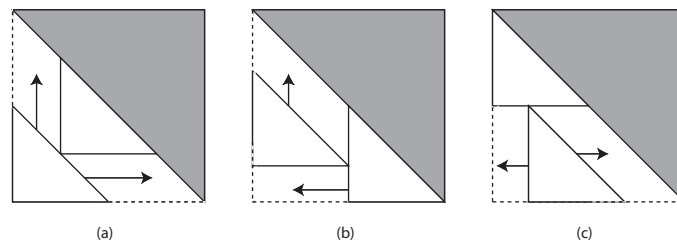


Figure 3.5: Possible positions for intersection of right isosceles triangles

In the cases where $u < 0$ or $v < 0$, extending the triangle(s) and inspecting the point where they meet yields a rectangular shape and one of the same wedge shapes encountered in quadrant one (see diagrams (b) and (c) in Figure 3.5). Each of these shapes may be filled in a manner that satisfies the given dot system structure by beginning at the point of intersection of the triangles and working outward to create a larger triangle that contains the two smaller triangles. (Rectangular shapes at the point of intersection are to be scanned by checking the bottom-left dot in translates of S .) The remaining upper

triangular portion of the block may be filled along the diagonal as in the quadrant one case. ■

In two-dimensional symbolic dynamics, shift spaces that have uniformly mixing factor languages are known to have a certain degree of complexity which is measured by the topological entropy and is based on the number of $n \times n$ blocks that appear in $F(X)$. More specifically, the *topological entropy* of the two-dimensional shift space X is

$$h(X) = \lim_{n \rightarrow \infty} \frac{1}{n^2} \log_2 |F_{n,n}(X)|.$$

Proposition 3.1.10 [6, 27] *If X is an infinite two-dimensional shift of finite type with factor language that is uniformly mixing, then X has a non-zero topological entropy.*

Example 3.1.11 Let X be the Three-dot System and consider $F_{n,n}(X)$ for some $n \geq 2$. By Corollary 1.3.2, $|F_{n,1}(X)| = 2^n$. As in the proof of Proposition 3.1.9, any one-column design of height n can be extended into a unique triangular design over $n - 1$ columns where for $i \in \{1, 2, \dots, n - 1\}$, the i th column has height $n - i$. Again as in the proof of Proposition 3.1.9, the choice of one symbol at the top of the i th column determines the symbol at the top of each consecutive column. (Refer to Figure 3.4 for an example.) Therefore only the top symbol of each of the $n - 1$ columns may be arbitrarily chosen whereby $|F_{n,n}(X)| = 2^{n+(n-1)}$. Then the topological entropy of the Three-dot System is

$$h(X) = \lim_{n \rightarrow \infty} \frac{1}{n^2} \log_2 |F_{n,n}(X)| = \lim_{n \rightarrow \infty} \frac{1}{n^2} \log_2 2^{2n-1} = \lim_{n \rightarrow \infty} \frac{2n-1}{n^2} = 0.$$

So although the Three-dot System has factor language which is mixing, it can not be uniformly so. In fact, it has been shown that all dot systems belong to the class of shifts of finite type having topological entropy equal to zero [20].

Corollary 3.1.12 *If X is a dot system, then the factor language $F(X)$ is not uniformly mixing.*

3.2 Graph Representations

Similar to transitivity in the one-dimensional case, there is a relationship between types of transitivity found in a two-dimensional language and the transitivity of the graph that recognizes the language. In the following, recall that E_h denotes the set of horizontal edges with transitions denoted by \rightarrow and that an automaton is said to be transitive if its graph representation is strongly connected.

Proposition 3.2.1 *Suppose a two-dimensional shift of finite type X is such that $F(X) = L(\mathcal{M}_{F(X)})$ for an automaton $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$. If $F(X)$ is horizontally transitive, then $\mathcal{M}_{F(X)}^h = (Q, E_h, s, t, \lambda)$ is transitive. In a similar fashion, if $F(X)$ is vertically transitive, then $\mathcal{M}_{F(X)}^v = (Q, E_v, s, t, \lambda)$ is transitive.*

Proof. We prove the horizontal case only. Let $q', q'' \in Q = F_{k,k}(X)$ be any two states. By definition of horizontal transitivity, there must exist some block $B_{k,n} \in F(X)$ that encloses q' and q'' . In fact, $B_{k,n}$ can be viewed as a k -phrase P having $s(P) = q', t(P) = q''$, and length $p = n - 2$. Then by the definition of E_h , P can be expressed as a sequence $\Lambda = q_0 \rightarrow q_1 \rightarrow \dots \rightarrow q_{p-1} \rightarrow q_p$ of vertices and horizontal transitions, where for $1 \leq i \leq p$, $q_i \in Q$ are such that $s(\Lambda) = q_0 = q', t(\Lambda) = q_p = q''$, and $q_{i-1} \rightarrow q_i \in F(X)$. In other words, Λ is a path in $\mathcal{M}_{F(X)}$ from q' to q'' . Therefore, $\mathcal{M}_{F(X)}^h$ is a transitive graph. ■

Example 3.2.2 Consider the Full-square System X introduced in Example 3.1.1, where X was defined via $S = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. Then the set $Q = F_{2,2}(X)$ consists of 8 states. We use solid lines to represent horizontal transitions and dashed lines to represent vertical transitions. In the graph representing $F(X)$, two separate components for the horizontal transitions (vertical transitions) illustrate the lack of horizontal (vertical) transitivity. (See Figure 3.6.)

The converse to Proposition 3.2.1 is false. That is, although transitivity in $\mathcal{M}_{F(X)}^h$ is a necessary condition for horizontal transitivity of the local language $F(X)$, it is not a sufficient condition. To see this, consider a graph $\mathcal{M}_{F(X)}$ that displays transitivity in its horizontal component and let $\Lambda' = q'_0 \uparrow q'_1 \uparrow \dots \uparrow q'_p$ and $\Lambda'' = q''_0 \uparrow q''_1 \uparrow \dots \uparrow q''_p$ be any two v -paths in $\mathcal{M}_{F(X)}^v$ that have the same length p . If $F(X) = L(\mathcal{M}_{F(X)})$ is to be horizontally

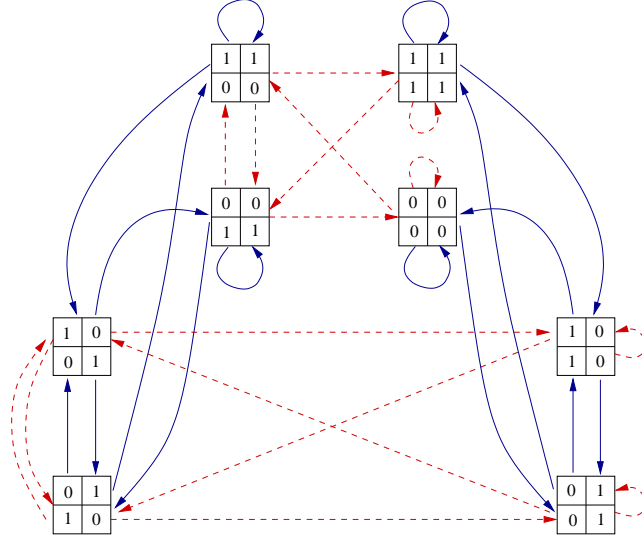


Figure 3.6: Full-square System

transitive, then it is not enough that for all $i \in \{0, 1, \dots, p\}$ there exist h -paths Λ_i in $\mathcal{M}_{F(X)}^h$ having $s(\Lambda_i) = q'_i$ and $t(\Lambda_i) = q''_i$; rather, it must also be the case that the h -paths in the collection $\{\Lambda_i\}$ overlap progressively to form a block path. This is illustrated by the following example.

Example 3.2.3 Let $S = \{(0, 0), (0, 1), (0, 2), (1, 0), (1, 2), (2, 0), (2, 1), (2, 2)\}$ define a dot system X . We shall show that the language $F(X)$ is not horizontally transitive, but that the graph $\mathcal{M}_{F(X)}$ representing X is transitive in both its horizontal and vertical components, $\mathcal{M}_{F(X)}^h$ and $\mathcal{M}_{F(X)}^v$, respectively. By Corollary 1.3.3, the graph $\mathcal{M}_{F(X)}$ representing X would be comprised of 2^8 states. So rather than constructing a graph of size 256, we will demonstrate transitivity in the graph's horizontal component by showing that any 3×3 block $B_{3,3} \in F_{3,3}(X) = Q$ meets the 3×3 block of all zeros $\beta_0 \in Q$ in direction $(1, 0)$ within $F(X)$. This is enough to show that there is a horizontal path connecting any two states in the graph since the vertical axis of symmetry for the shape S allows us to use a symmetric argument to find a path from β_0 to an arbitrary third block $B'_{3,3} \in F_{3,3}(X) = Q$. (By the same token, the horizontal axis of symmetry for the shape S provides a similar argument for showing that β_0 meets every block/state $B_{3,3} \in F_{3,3}(X) = Q$ in direction $(0, 1)$ and vice versa so that $\mathcal{M}_{F(X)}^v$ is transitive.) All sums in this example are carried out modulo 2.

Claim: Any block $B_{3,3} \in F_{3,3}(X)$ meets β_0 in direction $(1, 0)$ within $F(X)$.

Proof of Claim: In a technique similar to that of the Wallpaper Patterns found in the

proof of Proposition 3.1.2, we inspect the sum of the bits in one-column block extensions of a horizontal 3-phrase that has source $B_{3,3}$. Refer to these one-column extensions as B_i , where for $i \geq 1$ each B_i is a 3×1 block whose horizontal 3-concatenation with the existing 3-phrase $B_{3,3} \rightarrow B_1 \rightarrow B_2 \rightarrow \dots \rightarrow B_{i-1}$ produces a $3 \times (3 + i)$ block in $F(X)$. Say $n \geq 2$ is the first index such that the sum of the bits $B_n(0,0) + B_n(0,1) + B_n(0,2) = b_n$ is required to be odd in order for the block to be in the language $F(X)$. (If there is no such n , then the result is trivial since we can define $B_1 = B_2$ to be columns of all zeros so that $B_{3,3}$ meets β_0 in direction $(1,0)$ at distance 0.) To design an odd sum in column B_n , define $B_n(0,0) = B_n(0,1) = B_n(0,2) = 1$. At this discrete time step, B_{n-1} intersects the translate $S + (n,0)$ at the two bits $B_{n-1}(0,0)$ and $B_{n-1}(0,2)$; for the next translate of S , the column B_{n-1} will intersect the translate $S + (n+1,0)$ at the three bits $B_{n-1}(0,0), B_{n-1}(0,1)$ and $B_{n-1}(0,2)$. Therefore, the proof of the claim is completed by the inspection of two cases for the translate $S + (n+1,0)$.

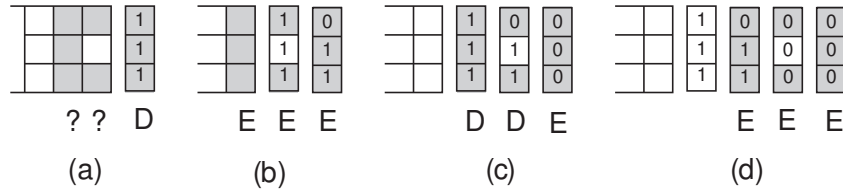


Figure 3.7: One-column extension having even sum

case i): The one-column extension B_{n+1} requires an even sum.

In this case, design an even sum in column B_{n+1} by defining $B_{n+1}(0,0) = B_{n+1}(0,1) = 1$ and $B_{n+1}(0,2) = 0$. At this discrete time step, B_n intersects the translate $S + (n+1,0)$ at the sum of the two bits $B_n(0,0) + B_n(0,2) = 0$, whereas for the next translate of S , the column B_n will intersect the translate $S + (n+2,0)$ at the sum of the three bits $B_n(0,0) + B_n(0,1) + B_n(0,2) = b_n = 1$. In this way, both the $S + (n+2,0)$ translate and the $S + (n+3,0)$ translate will require even sums so that we define $B_{n+2} = B_{n+3}$ as columns of all zeros, which completes the proof of the claim. See Figure 3.7.

case ii): The one-column extension B_{n+1} requires an odd sum.

In this case, design an odd sum in column B_{n+1} by defining $B_{n+1}(0,0) = 1$ and $B_{n+1}(0,1) = B_{n+1}(0,2) = 0$. At this discrete time step, B_n intersects the translate $S + (n+1,0)$ at the sum of the two bits $B_n(0,0) + B_n(0,2) = 0$, whereas for the next translate of S , the column B_n will intersect the translate $S + (n+2,0)$ at the sum of the

three bits $B_n(0, 0) + B_n(0, 1) + B_n(0, 2) = b_n = 1$. The translate $S + (n + 2, 0)$ will then require an even sum: define $B_{n+2}(0, 0) = B_{n+2}(0, 1) = 1$ and $B_{n+2}(0, 2) = 0$. In this way, both the $S + (n + 3, 0)$ translate and the $S + (n + 4, 0)$ translate will require even sums so that we define $B_{n+3} = B_{n+4}$ as columns of all zeros, which completes the proof of the claim. See Figure 3.8.

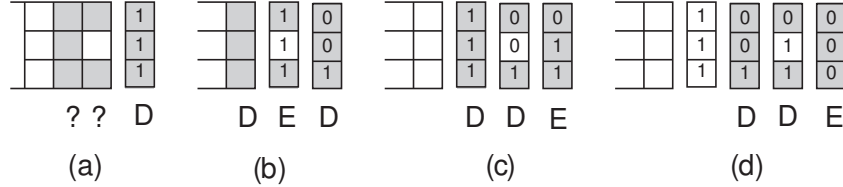


Figure 3.8: One-column extension having odd sum

Now that we have demonstrated the transitivity of the horizontal (and vertical) component of $\mathcal{M}_{F(X)}$, we show that the block $B_{5,3} \in F(X)$ of all 1's can not meet the 5×3 block of all 0's within $F(X)$ in direction $(1, 0)$ at any distance.

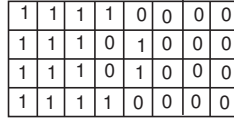


Figure 3.9: A 4×3 block of 1's meets 4×3 block of 0's

Consider first the 4×3 block of all 1's: in order for this block to meet the 4×3 block of all 0's, there must exist two overlapping h -paths in $\mathcal{M}_{F(X)}$ of the same length with both h -paths having source β_1 (the 3×3 state/block of all 1's) and target β_0 . Notice that whenever a one-column block B_n having a sum $B_n(0, 0) + B_n(0, 1) + B_n(0, 2) = b_n = 1$ is horizontally 3-concatenated onto an existing 3-phrase, the symmetry of the shape S results in the column B_n intersecting the translate $S + (n + 2, 0)$ with the same sum $b_n = 1$. Therefore, in order to horizontally 3-concatenate two subsequent columns B_{n+1} and B_{n+2} having columnar sums $b_{n+1} = b_{n+2} = 0$, the one-column block B_{n+1} must be such that $B_{n+1}(0, 1) = 1$ which requires that either $B_n(0, 0) = 1$ or $B_n(0, 2) = 1$ but not both. So for overlapping h -paths to reach state β_0 simultaneously, they must describe a 4-phrase with horizontal reflection symmetry. (See Figure 3.9.) For the 5×3 block of all 1's, such reflection symmetry cannot exist. Therefore, block $B_{5,3} \in F(X)$ of all 1's can not meet the 5×3 block of all 0's within $F(X)$ in direction $(1, 0)$ at any distance.

There do exist conditions on the graph $\mathcal{M}_{F(X)}$ that are sufficient for the factor language $F(X)$ to be horizontally transitive. We state the horizontal case only, as the vertical case is analogous.

Proposition 3.2.4 *Suppose a two-dimensional shift of finite type X is such that $F(X) = L(\mathcal{M}_{F(X)})$ for $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$. If the subgraph $\mathcal{M}_{F(X)}^h$ is transitive and if for every four states q_0, q_1, q'_0, q'_1 having $q_0 \rightarrow q_1, q_0 \uparrow q'_0$, and $q_1 \uparrow q'_1$, it necessarily follows that $q'_0 \rightarrow q'_1$, then $F(X)$ is horizontally transitive.*

Proof. Let $\beta_1, \beta_2 \in F(X)$ be any two blocks in the language defined by $Q = F_{k,k}(X)$. Without loss of generality, we may assume that the two blocks have the same height. The proof by induction on the number of rows m will be similar to the proof of Proposition 3.1.7. Let $m = k$ for the basis step: then B' meets B'' in direction $(1, 0)$ within $F(X)$ since $\mathcal{M}_{F(X)}^h$ is transitive. Now assume by the induction hypothesis that β_1 meets β_2 in direction $(1, 0)$ within $F(X)$ for all blocks $\beta_1, \beta_2 \in F(X) \cap \Sigma^{m,n}$ and consider $B'_{m+1,n'}, B''_{m+1,n''} \in F(X)$. Let $\beta_1 = \{B'_{m+1,n'}(i, j) : 0 \leq i \leq n' - 1, 0 \leq j \leq m - 1\}$, and similarly, let $\beta_2 = \{B''_{m+1,n''}(i, j) : 0 \leq i \leq n'' - 1, 0 \leq j \leq m - 1\}$. Then by the induction hypothesis, there exists $B^*_{m,n+n''} \in F(X)$ such that $B^*_{m,n+n''}$ encloses β_1 and β_2 in direction $(1, 0)$ within $F(X)$. Now label an $(m+1) \times (n+n'')$ rectangular shape as T , and begin to define $B : T \rightarrow \Sigma$ in the following way.

$$B(i, j) = \begin{cases} B'_{m+1,n'}(i, j) & \text{for } 0 \leq i \leq n' - 1, 0 \leq j \leq m \\ B^*_{m,n+n''}(i, j) & \text{for } n' \leq i \leq (n-1), 0 \leq j \leq m - 1 \\ B''_{m+1,n''}(i - (n), j) & \text{for } n \leq i \leq n + n'' - 1, 0 \leq j \leq m \end{cases} \quad (3.2.8)$$

(Here, we may assume that as defined thus far, block B occurs in X : as in the proof of Proposition 3.1.7, before defining B we may first extend the blocks β_1 and/or β_2 by $k - 1$ columns to ensure that this happens.) The completion of block B is guaranteed by states from $\mathcal{M}_{F(X)}$, beginning at the left-most undefined cell in the top row of B and working across the top row by following edges in E_h . More formally, in $\mathcal{M}_{F(X)}$ there exists states $q_0 = B\{(m-2, n'-2), (m-2, n'-1), (m-1, n'-2), (m-1, n'-1)\}$, $q'_0 = B\{(m-1, n'-2), (m-1, n'-1), (m, n'-2), (m, n'-1)\}$, and $q_1 = B\{(m-2, n'-1), (m-2, n'), (m-1, n'-1), (m-1, n')\}$; the existence of state q'_1 is guaranteed by $A(X) = F(X)$

and supplies the label for $B(m, n')$, while the condition “for every four states q_0, q_1, q'_0, q'_1 having $q_0 \rightarrow q_1, q_0 \uparrow q'_0$, and $q_1 \uparrow q'_1$, it necessarily follows that $q'_0 \rightarrow q'_1$ ” guarantees the acceptance of the block with this additional label. When we reach the right-most undefined cell in the top row of B , the state $q'_1 = B\{(m-1, n-1), (m-1, n), (m, n-1), (m, n)\}$ is already predetermined, while the acceptance of all transitions is guaranteed by the condition involving the four states of $\mathcal{M}_{F(X)}$. ■

The conditions of Proposition 3.2.4 are sufficient but not necessary for the language $F(X)$ to be horizontally transitive. For example, the graph of the Three-dot System provided in Figure 2.4 does not meet the conditions of Proposition 3.2.4, although the Three-dot System is known to be mixing and is therefore both horizontally and vertically transitive.

While the graph representations discussed here employ horizontal and vertical transitions, these graphs can still provide information about transitivity in other directions. However, it will be necessary to place bounds on the neighborhood in question.

Proposition 3.2.5 *Suppose a two-dimensional shift of finite type X is such that $F(X) = L(\mathcal{M}_{F(X)})$ for an automaton $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$. If $F(X)$ is uniformly transitive in direction (u, v) , then $\mathcal{M}_{F(X)}$ is transitive.*

Proof. Consider any two states $q', q'' \in Q = F_{k,k}(X)$. As $F(X)$ is uniformly transitive in direction (u, v) , there must exist a block $B \in F(X)$ that minimally encloses q' and q'' in direction (u, v) such that $d(q', q'') < K$ for some positive integer K . (Recall that by definition, q' meets q'' within $F(X)$ in direction (u, v) if and only if q' meets q'' within $F(X)$ in direction $(-u, -v)$.) The proof is completed by the inspection of two cases based on the direction (u, v) .

case i) If $u, v \geq 0$, then in B there exists some k -phrase P such that $s(P) = q', t(P) = q''$ and P is accepted by $\mathcal{M}_{F(X)}$. In other words, there must exist some path Λ in $\mathcal{M}_{F(X)}$ from q' to q'' . (See diagram (a) in Figure 3.10.) The case when $u, v \leq 0$ is similar.

case ii) Suppose $u > 0, v < 0$. (If $u = 0$, then the result is trivial, as there would exist some path connecting q' and q'' in $\mathcal{M}_{F(X)}^v = (Q, E_v, s, t, \lambda)$.) We shall use extension to create a block too large to be contained in the K -neighborhood. Consider vector $(K+k, h')$

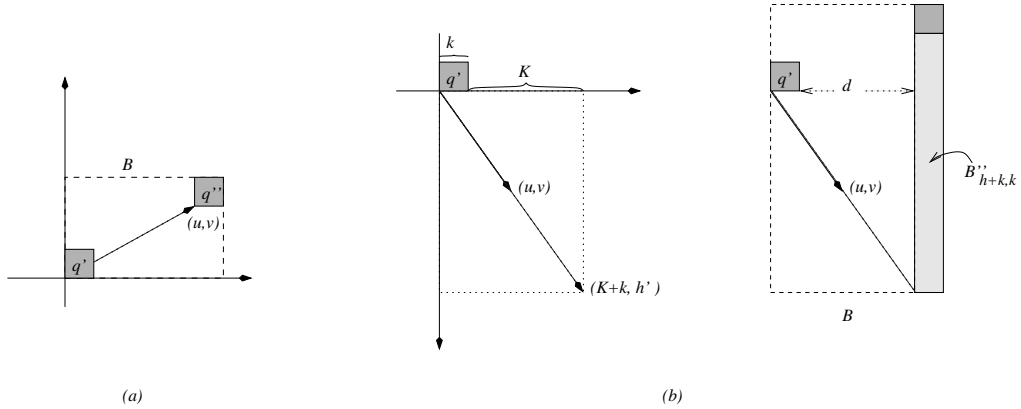


Figure 3.10: Uniform transitivity in direction (u, v)

that is parallel to (u, v) . Then $h' = \frac{v}{u}(K + k)$. (See left portion of diagram (b) in Figure 3.10.) Consider a block $B''_{h+k, k} \in F(X)$ where $h > h'$ by extending q'' below by h -many columns so that q'' appears in the top portion of the new block. Then q' meets B'' within $F(X)$ in direction (u, v) at distance $d < K$. (See right portion of diagram (b) in Figure 3.10.) Therefore, it must be the case that q'' appears above and to the right of q' in the block B that minimally encloses q' and B'' . So there must exist some path Λ in $\mathcal{M}_{F(X)}$ from q' to q'' .

The case when $u < 0$ and $v \geq 0$ is similar. ■

If X is a shift of finite type having property $A(X) = F(X)$, then given $K \geq 0$ it is decidable whether $F(X)$ exhibits uniform horizontal (vertical) transitivity at distance K . Let us first outline how the $\mathcal{M}_{F(X)}$ construction facilitates the proof.

Suppose we are given an $m' \times n'$ block $B' \in F(X)$ and an $m'' \times n''$ block $B'' \in F(X)$. To show uniform horizontal transitivity, we must demonstrate that we can find a block $B \in F(X)$ that encloses B' and B'' in such a way that the bottom left corners of B' and B'' appear at vertices $(0, 0)$ and $(n' + k, 0)$, respectively, for some $k \leq K$. (See diagram (a) of Figure 3.11: since $A(X) = F(X)$, we may always extend blocks upward so that a given pair of blocks have the same height m .) Let $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$ be a graph representation of $F(X)$, where we assume that 2×2 states were used in the construction. For every $m \times n$ block $B \in F(X)$, there must exist some $(m - 1) \times (n - 1)$ block path in $\mathcal{M}_{F(X)}$ that recognizes B . (Refer to Figure 2.6 for the dimension of a block versus a block

path.) So suppose that B' is recognized by some $(m - 1) \times (n' - 1)$ block path π' and that B'' is recognized by some $(m - 1) \times (n'' - 1)$ block path π'' , where we assume without loss of generality that $n', n'' \geq 2$. Horizontal transitivity can be illustrated by finding a block path π in $\mathcal{M}_{F(X)}$ such that the states in the initial (left) column of π agree with the states in the final (right) column of π' and such that the states in the final column of π agree with the the states in the initial column of π'' . For example, if a block path of length 3 overlaps the given blocks in the desired way, then the distance at which the two blocks meet would be zero, since this implies that the original two blocks touch. Uniform horizontal transitivity at distance K can be satisfied by finding a $(m - 1) \times (k + 3)$ block path π for some $k \leq K$ such that $\forall j \in \{0, 1, \dots, m - 2\}$, $\pi'(n' - 2, j) = \pi(0, j)$ and $\pi(k + 2, j) = \pi''(0, j)$. A block path of length $k + 3$ is needed to represent a block of length $k + 4$ that overlaps the final two columns of symbols (comprising one v -path in π') and initial two columns of symbols in blocks B' and B'' , respectively. (See diagram (b) of Figure 3.11, where the block path π that overlaps with π' and π'' is indicated with dashed lines, while the darker shading indicates the initial and final v -paths of π that overlap with v -paths of π' and π'' .)

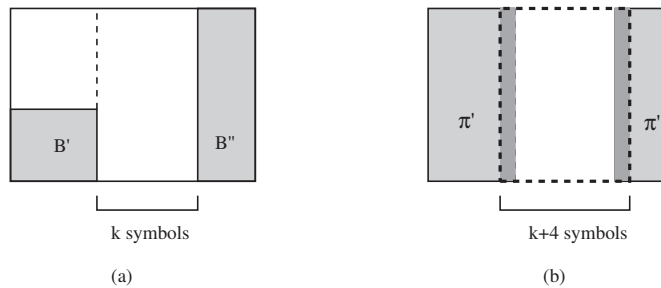


Figure 3.11: Blocks and block paths over 2×2 states

Theorem 3.2.6 *If X is a two-dimensional shift of finite type having property $A(X) = F(X)$, then given a distance K , there is an algorithm which decides whether $F(X)$ has uniform horizontal transitivity at distance K .*

The proof will be comprised of four steps. To begin, the uniformity condition is used to place the question into the framework of one-dimensional languages so that several well-known results concerning one-dimensional recognizable languages may be applied. We first use the fact that if two one-dimensional languages are recognizable, then their union

is also known to be recognizable. The second application concerns the product L_1L_2 of two one-dimensional languages L_1 and L_2 given by $L_1L_2 := \{x_1x_2 : x_1 \in L_1, x_2 \in L_2\}$. (Informally, a string of symbols belongs to L_1L_2 if it can be written as a string in L_1 followed by a string in L_2 .) Finally, we will use the result that it is decidable whether two one-dimensional languages are equal or not.

Proof. Step 1: Let $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$ be a graph representation of $F(X)$. For $i \in \{1, 2, \dots, k+2\}$, form the set \mathcal{H}_i of all h -paths of length i found in $\mathcal{M}_{F(X)}$. For each \mathcal{H}_i , form the one-dimensional finite automaton \mathcal{M}_i in the following way.

- Define the states of \mathcal{M}_i to be the set \mathcal{H}_i .
- Define a transition from state $h = q_0 \rightarrow q_1 \rightarrow \dots \rightarrow q_{i-1}$ to state $h' = q'_0 \rightarrow q'_1 \rightarrow \dots \rightarrow q'_{i-1}$ if and only if $\forall j \in \{0, 1, \dots, i-1\} \exists e_v \in E$ such that $s(e_v) = q_j$ and $t(e_v) = q'_j$.
- An edge from h to h' is given the label h .

Note that each \mathcal{M}_i is essentially a one-dimensional vertex shift representing the set of all block paths of length $i+1$ that represent blocks in $F(X)$.

Step 2: For each finite automaton \mathcal{M}_i , form the language \mathcal{L}_i by taking products of the sequences of states found in the lower portions of the first and last columns of the represented block paths in the following way. First distinguish between states in the left and right columns of a block by forming two different alphabets Q and Q' . Then let $\mathcal{L}_i := \{\alpha_j \omega_j\}$, where if $\pi = h_0 h_1 \dots h_m \in L(\mathcal{M}_i)$ is a path of “height” m , then for all $j \in \{0, 1, \dots, m-1\}$, $\alpha_j = \pi(0, 0) \pi(0, 1) \dots \pi(0, j) = q_0^0 q_0^1 \dots q_0^j$ is a sequence of states found in the bottom part of the first column of π ; that is, q_0^s is the source of the h -path h_s for $s \in \{0, 1, \dots, j\}$. In a similar fashion, $\omega_j = [\pi(i-1, 0)]' [\pi(i-1, 1)]' \dots [\pi(i-1, j)]' = q_i'^0 q_i'^1 \dots q_i'^j$ is a sequence of states found in the bottom part of the last column of the same block path π ; that is, $q_i'^s$ is the target of the h -path h_s for $s \in \{0, 1, \dots, j\}$. Now define the recognizable language \mathcal{L} to be $\mathcal{L}_1 \cup \mathcal{L}_2 \cup \dots \cup \mathcal{L}_{k+2}$. Thus defined, \mathcal{L} represents the set of all blocks that can meet in direction $(1, 0)$ at distance $k \leq K$ in $F(X)$. As the language \mathcal{L} is formed from the labels of bi-infinite sequences of the vertex shifts \mathcal{M}_i , the language \mathcal{L} is not necessarily finite.

Step 3: Consider $\mathcal{M}_{F(X)}^v = (Q, E_v, s, t, \lambda)$, the restriction of the graph representation of $F(X)$ to vertical transitions only. Relabel $\mathcal{M}_{F(X)}^v$ to be a vertex shift where the transition $q \uparrow r$ is labeled with q . (We are interested in columnar block paths rather than the symbols from the alphabet Σ .) Denote by L the language recognized by this relabeled finite automaton, and create a second language L' from L by attaching a prime to each state's symbol. So if the alphabet for L is $Q = \{q_1, q_2, \dots, q_f\}$, then the alphabet for L' is $Q' = \{q'_1, q'_2, \dots, q'_f\}$. Now define the recognizable language LL' to be the product of L and L' . This language LL' represents the possibility of any two blocks in $F(X)$ meeting in direction $(1, 0)$.

Step 4: It is decidable whether $LL' = \mathcal{L}$.

■

Theorem 3.2.6 can be analogously reworded to determine if $L(\mathcal{M}_{F(X)}) = F(X)$ exhibits uniform vertical transitivity at distance K . Transitivity in the horizontal and/or vertical direction is of particular interest, since the graph $\mathcal{M}_{F(X)}$ is defined based on these types of transitions. Furthermore, the definition of periodicity in the two-dimensional case is based on horizontal and vertical movement.

4 PERIODICITY

In one-dimensional symbolic dynamics, it is known that any shift space X that can be represented by a finite automaton must contain a periodic point under the \mathbb{Z} action of coordinate-wise translation [25]. For a two-dimensional shift space X , we are interested in the \mathbb{Z}^2 action of translating elements of X and the periodic points that may occur as a result of this action. In the two-dimensional case, however, even a shift of finite type can be aperiodic (see [17] for an example of such a shift space). In this chapter, we limit our focus to two-dimensional shift spaces where the factor language $F(X)$ is equivalent to the local language $A(X)$. For such two-dimensional FPR languages, periodic points of a specified period can be identified by the presence of h -cycles and v -cycles in the graph $\mathcal{M}_{F(X)}$ that represents X . In particular, Theorem 4.1.4 states that if $F(X) = A(X)$ exhibits uniform horizontal transitivity at distance K , then X must contain a periodic point with a bounded least period. Proposition 4.1.5 provides an algorithm for finding all periodic points up to a specified bound in the shift of finite type X represented by graph $\mathcal{M}_{F(X)}$. Using this algorithm, various examples of two-dimensional periodic points and the (sub)graphs that represent them are provided in Section 4.2.

Before discussing periodic points in two-dimensional shift spaces, we need a more precise notion of what it means for a point to be periodic under the action of \mathbb{Z}^2 . Given the two-dimensional shift space X , $x \in X$ is periodic of period $(a, b) \in \mathbb{Z}^2 \setminus \{(0, 0)\}$ iff $x_{(i,j)} = x_{(i+a,j+b)}$ for every $(i, j) \in \mathbb{Z}^2$. Furthermore, if x is periodic, then there exists $a, b > 0$ such that $x_{(i,j)} = x_{(i+a,j)} = x_{(i,j+b)} = x_{(i+a,j+b)}$ for every $(i, j) \in \mathbb{Z}^2$, in which case we say that x is *doubly periodic* of period (a, b) . If a and b are such that

- $a = \min\{a : x \text{ is doubly periodic of period } (a, b)\}$ and
- $b = \min\{b : x \text{ is doubly periodic of period } (a, b)\}$,

then we say that (a, b) is the *least double period* of x . For example, the point x in Figure 4.1 is not doubly periodic of period $(3, 2)$ since $x_{(0,0)} = A \neq B = x_{(3,0)}$, although x is

periodic of period $(3, 2)$ since $x_{(i,j)} = A = x_{(i+3,j+2)}$ for every $(i, j) \in \mathbb{Z}^2$ as indicated by the boxed symbols in Figure 4.1.

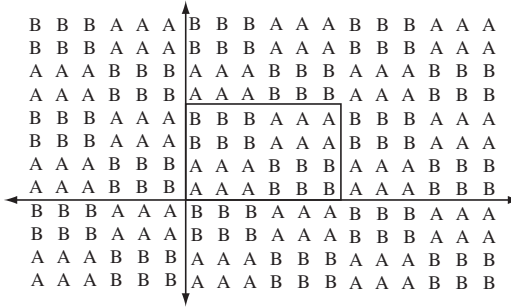


Figure 4.1: Point with least double period $(6, 4)$

4.1 Doubly Periodic Points in Two-dimensional Shift Spaces

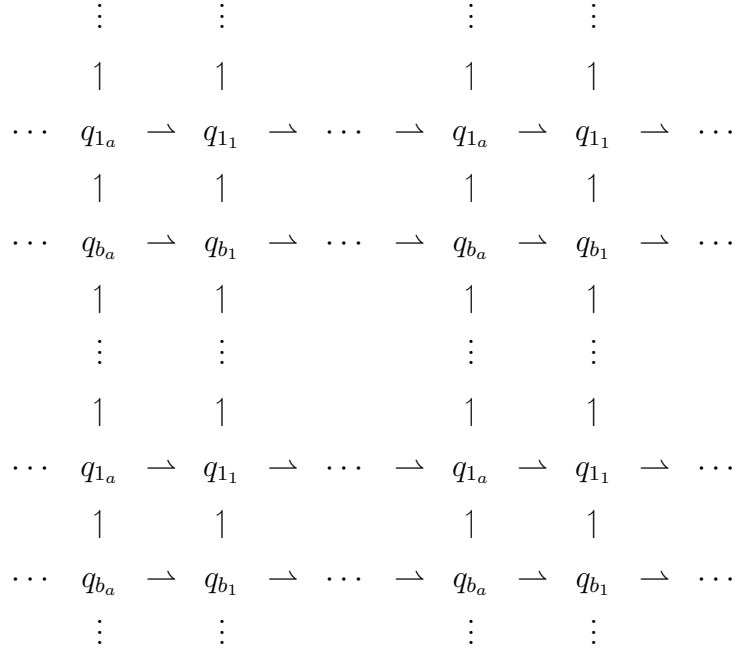
Proposition 4.1.1 guarantees the existence of a periodic point under certain conditions. The proposition begins with identifying h -cycles of a fixed length in $\mathcal{M}_{F(X)}$. (Alternatively, the proposition could identify v -cycles in $\mathcal{M}_{F(X)}$.)

Proposition 4.1.1 *Let X be a two-dimensional shift of finite type having property $A(X) = F(X)$ and graph representation $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$. There exists a point $x \in X$ being doubly periodic of period $(a, b) \in \mathbb{Z}^2 \setminus \{(0, 0)\}$ iff there exist h -cycles $\{\Lambda_1, \Lambda_2, \dots, \Lambda_b\}$ in $\mathcal{M}_{F(X)}$ (where for $1 \leq \rho \leq b$ Λ_ρ is denoted by $\Lambda_\rho = q_{\rho_1} \rightarrow q_{\rho_2} \rightarrow \dots \rightarrow q_{\rho_a} \rightarrow q_{\rho_1}$) such that*

- i) for $1 \leq \rho \leq b$, $|\Lambda_\rho| = a$,*
- ii) for $1 \leq i \leq b, 1 \leq j \leq a$, there exists $e_v \in E$ such that $e_v = q_{i_j} \uparrow q_{(i+1)_j}$, and*
- iii) for $1 \leq j \leq a$, there exists $e_v \in E$ such that $e_v = q_{b_j} \uparrow q_{1_j}$.*

Proof. Say x is doubly periodic of period (a, b) , and consider a design on the normalized $b \times a$ shape T which we denote $x_T = B_{b,a}$. Then by the definition of doubly periodic, $x_T = x_{T+(a,0)} = x_{T+(0,b)} = B_{b,a}$. (See, for example, Figure 4.1). In particular, for the normalized $(b+1) \times a$ shape T' , the block $B'_{b+1,a}$ that occurs in x must be accepted by $\mathcal{M}_{F(X)}$ in such a way that conditions (i), (ii), and (iii) are fulfilled, since the states of $\mathcal{M}_{F(X)}$ have distinct labels.

For the converse, let cycles $\{\Lambda_1, \Lambda_2, \dots, \Lambda_b\}$ be as given and consider the following repetition of these cycles:



This collection of edges satisfies the requirements for a grid-infinite path, and as such, Corollary 2.1.7 guarantees the existence of some point $x \in X$ having the property that $x_{(i,j)} = x_{(i+a,j)} = x_{(i,j+b)} = x_{(i+a,j+b)}$ for every $(i, j) \in \mathbb{Z}^2$. In other words, x is doubly periodic of period (a, b) . ■

In Proposition 4.1.1, neither cycles nor states (even within the same cycle) need be distinct. Consider a point having least double period (a, b) : If each state in some $b \times a$ block path representing the point were distinct, then the number of states required to represent the point would be ab .

Corollary 4.1.2 *Given the graph representation $\mathcal{M}_{F(X)}$ of a two-dimensional shift of finite type X , the maximum number of states required to represent a point having least double period (a, b) is ab .*

It is possible for a subgraph of two states to be both necessary and sufficient for a graph $\mathcal{M}_{F(X)}^\Phi$ to represent a shift space Y that contains a fixed point. For example, consider the subgraph of 2 states connected in both an h -cycle and a v -cycle as in Corollary 2.3.5 for

some non-deterministic $\mathcal{M}_{F(X)}^\Phi$. If $A = B$ so that the pair of states have the same labels, then this pair of states suffice to represent a fixed point of the shift space. When viewed in the context of $\mathcal{M}_{F(X)}^\Phi$, however, the states may be distinct although their labels are not. (Say one of the states is distinguished by a vertical transition that the other state lacks.) If the sofic shift space Y represented by $\mathcal{M}_{F(X)}^\Phi$ has no single state representing a fixed point (that is, no state has both an h -loop and a v -loop associated with it), then the pair of states represent a subgraph of minimum size capable of representing a fixed point in Y . This simple example illustrates that in Proposition 4.1.1, the conditions for a point x to be doubly periodic of period (a, b) - while still sufficient - are no longer necessary in the more general sofic case. That is, the maximum number of states required to represent a point having least double period (a, b) might exceed ab if the labels on the states are not distinct.

Proposition 4.1.3 *Let Y be a two-dimensional shift space with graph representation $\mathcal{M}_{F(X)}^\Phi = (Q, E, s, t, \lambda)$. If there exists a point $y \in Y$ being doubly periodic of period $(a, b) \in \mathbb{Z}^2 \setminus \{(0, 0)\}$, then for some $k > 0$, there exist h -cycles $\{\Lambda_1, \Lambda_2, \dots, \Lambda_{kb}\}$ in $\mathcal{M}_{F(X)}^\Phi$ (where for $1 \leq \rho \leq kb$, the h -cycle Λ_ρ is denoted by $\Lambda_\rho = q_{\rho_1} \rightarrow q_{\rho_2} \rightarrow \dots \rightarrow q_{\rho_{ka}} \rightarrow q_{\rho_1}$) such that*

- i) for $1 \leq \rho \leq kb$, $|\Lambda_\rho| = ka$,*
- ii) for $1 \leq i \leq kb, 1 \leq j \leq ka$, there exists $e_v \in E$ such that $e_v = q_{i_j} \uparrow q_{(i+1)_j}$, and*
- iii) for $1 \leq j \leq ka$, there exists $e_v \in E$ such that $e_v = q_{kb_j} \uparrow q_{1_j}$.*

The existence of uniform horizontal transitivity (or uniform vertical transitivity) for $L(\mathcal{M}_{F(X)}) = F(X)$ guarantees that the shift space X has a periodic point.

Theorem 4.1.4 *Let X be a two-dimensional shift of finite type having property $F(X) = A(X)$. If $F(X)$ exhibits uniform horizontal transitivity at some distance K , then X has a periodic point of least double period (a, b) for some $a \leq K + 2$.*

Proof. Let $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$ be the graph representation of $F(X)$ and consider some v -cycle contained in $\mathcal{M}_{F(X)}^v$, say $\theta = q_0 \uparrow q_1 \uparrow \dots \uparrow q_p = q_0$. (Such a v -cycle must exist since $\mathcal{M}_{F(X)}^v$ represents a one-dimensional sofic shift space, which must contain a periodic point.) Denote by B the $(p + 2) \times 2$ block described by $\lambda(\theta)$, and for $i \geq 1$ consider the set $\{B_i\}$ of $(ip + 2) \times 2$ blocks that result from repeatedly traveling the v -cycle θ . Since

$F(X)$ exhibits uniform horizontal transitivity, B_i meets B_i at a distance $k \leq K$. As the set $\{B_i\}$ is infinite, there must exist $m \leq K$ such that blocks from a (countably) infinite subset $S \subseteq \{B_i\}$ all meet themselves at distance m . For each $B_i^j \in S$, let $C_i^1, C_i^2, \dots, C_i^j \dots$ be the blocks that connect B_i^j with B_i^j . Since $\{C_i^j\}$ is also infinite and since each block C_i^j has length m , there must exist a block C_I^J with the property that the h -path connecting q_0 to q_0 whose label describes the bottom row of C_I^J equals the h -path connecting q_0 to q_0 whose label describes the top row of C_I^J . (There exist only a finite number of h -paths connecting q_0 to q_0 since the graph $\mathcal{M}_{F(X)}$ is finite.) Finally, define the block $B_{Ip+1, m+2}$ by

$$B_{Ip+1, m+2}(i, j) = \begin{cases} B_I(i, j) & \text{for } 0 \leq i \leq 1, 0 \leq j \leq Ip \\ C_I^J(i-2, j) & \text{for } 2 \leq i \leq m+2, 0 \leq j \leq Ip \end{cases} \quad (4.1.1)$$

Then $B_{Ip+1, m+2}$ contains all but the top row of symbols in blocks B_I and C_I^J so that $B_{Ip+1, m+2}(i, j) = B_{Ip+1, m+2}(i+a, j+b)$ with $a = m+2, b = Ip+2$ is a block that makes up a doubly periodic point of period (a, b) . ■

Notice that the converse to Theorem 4.1.4 is false. For example, one can easily find periodic points in the Full-square System, yet it was shown that the factor language of the Full-square System lacks horizontal (and vertical) transitivity. When they exist, doubly periodic points of a given period (a, b) can be located for a two-dimensional shift of finite type X through use of the $\mathcal{M}_{F(X)}$ graph representation regardless of whether $L(\mathcal{M}_{F(X)})$ exhibits uniform transitivity or not.

Proposition 4.1.5 *Given $a, b > 0$ and a graph representation $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$ of the two-dimensional shift of finite type X , there is an algorithm which locates all points of X having double period (a, b) .*

Proof. By inspection of the graph $\mathcal{M}_{F(X)}$, first find all h -cycles of length a in $\mathcal{M}_{F(X)}$. Distinguish cycles by their initial state, making permutations of the same cycle distinct, so that there will be at most $|Q|^a$ cycles. Denote by Θ this collection of cycles having length a , and set $|\Theta| = z$. For $1 \leq \rho \leq z$ denote $\theta \in \Theta$ by $\theta_\rho = q_{\rho_1} \rightarrow q_{\rho_2} \rightarrow \dots \rightarrow q_{\rho_a} \rightarrow q_{\rho_1}$.

Next inspect pairs of cycles $\theta_f, \theta_g \in \Theta$. (Here $\theta_f = \theta_g$ is not precluded, so that there exists at most z^2 comparisons.) If it is true that $\forall j \in \{1, 2, \dots, a\}, \exists e_v \in E$ such that

$e_v = q_{f_j} \uparrow q_{g_j}$, then denote this pair as γ_{f_1g} . The collection of all such pairs is to be denoted by Γ .

Say that γ_{f_1g} overlaps $\gamma_{f'_1g'}$ iff $g = f'$. Periodic points can be found by progressively overlapping pairs found in Γ : If there exists $\gamma_{f_1g} \in \Gamma$ such that $f = g$, then there exists a point $x \in X$ being doubly periodic of period $(a, 1)$; if there exists $\gamma_{f_1g}, \gamma_{f'_1g'} \in \Gamma$ such that γ_{f_1g} overlaps $\gamma_{f'_1g'}$ in such a way that $g' = f$, then there exists a point $x \in X$ being doubly periodic of period $(a, 2)$; and so on. ■

Given a graph representation \mathcal{M} of a one-dimensional sofic shift, it is fairly easy to locate periodic points, as one need only check the graph for cycles (which must exist in the one-dimensional case). In the two-dimensional case, even a shift of finite type may be aperiodic; when doubly periodic points do exist, it is apparent from Proposition 4.1.5 and Proposition 4.1.3 that the process of locating these points will not be so straightforward, as one must inspect the graph $\mathcal{M}_{F(X)}^\Phi$ for overlapping h -cycles of length ka . More specifically, for a sofic subshift Y and a point y being doubly periodic of period (a, b) , there may not exist a grid-infinite path Π such that $\Pi(i, j) = \Pi(i + a, j + b)$ for all $(i, j) \in \mathbb{Z}^2$ although $y(i, j) = y(i + a, j + b)$ for all $(i, j) \in \mathbb{Z}^2$. Rather, it may be the case that we need to find some $k > 1$ such that $\Pi(i, j) = \Pi(i + ka, j + kb)$ for all $(i, j) \in \mathbb{Z}^2$, as the states need no longer have distinct labels. (If $k = 1$, the conditions of Proposition 4.1.3 become the necessary conditions of Proposition 4.1.1.) For our discussion of periodic points, we therefore focus on those doubly periodic points of period (a, b) that are represented by a grid-infinite path Π having $\Pi(i, j) = \Pi(i + a, j + b)$ for all $(i, j) \in \mathbb{Z}^2$, such as those found in $\mathcal{M}_{F(X)}$ representing a shift of finite type X .

4.2 Examples

In this section, we provide examples of subgraphs representing doubly periodic points of least period (a, b) given that $a, b \in \{1, 2, 3\}$. In particular, we comment on graphs of minimum size capable of representing a point of least double period (a, b) given that $a, b \in \{1, 2, 3\}$, and then we generalize these graphs of minimum size to include certain other types of doubly periodic points.

Given least double period (a, b) , Corollary 4.1.2 provides an upper bound on the number

of states required to represent such a point in the graph of a two-dimensional shift of finite type. Furthermore, for all $a, b \geq 1$ there exists some graph representing a point for which this upper bound ab is strict, provided the alphabet is large enough for the necessary number of distinct states to exist. The task of finding a lower bound on the number of states needed to represent a point of least double period (a, b) is a bit more arduous.

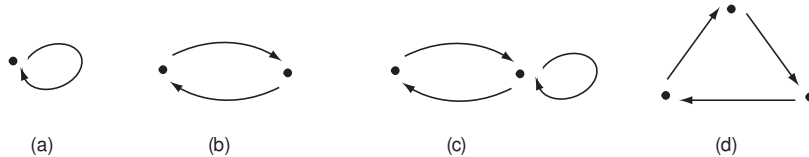


Figure 4.2: Vertex shifts in one-dimensional case

In the one-dimensional case, suppose we consider the set of all bi-infinite sequences (i.e. points in some one-dimensional shift space) having least period a . The maximum number of states required for a vertex shift to represent such a point would be a . For example, let $|\Sigma| = a$ and let X be the shift space represented by a graph \mathcal{M} consisting of a states connected in a simple cycle. (Figure 4.2 (d) is such an example for $a = 3$.) At the other extreme, the vertex shift of size 3 having the underlying graph depicted in Figure 4.2 (c) is the subgraph of minimum size capable of representing some point of least period $a \geq 3$. (Graphs (a) and (b) of Figure 4.2 are the subgraphs of minimum size capable of representing points of least period 1 and 2, respectively.) If we limit ourselves to vertex shifts representing one-dimensional subshifts, then in terms of graphs having minimum size, there are exactly four non-isomorphic subgraphs capable of depicting a point having least period $a \in \{1, 2, 3\}$. These are documented in Figure 4.2.

According to the standard definition of doubly periodic, a fixed point in a two-dimensional shift space has least double period $(1, 1)$. Using the $\mathcal{M}_{F(X)}$ construction with 2×2 states, fixed points are the only doubly periodic points that can be represented by a subgraph comprised of a single state due to the need for both an h -loop and a v -loop at the same state. For two-dimensional shifts of finite type then, $ab = 1$ is both the maximum and minimum graph size needed to represent a point of least double period $(a, b) = (1, 1)$.

Since a graph of size 1 can only represent a fixed point, $ab = 2$ is both the maximum and minimum graph size needed to represent a point of least double period $(2, 1)$ or $(1, 2)$ for two-dimensional shifts of finite type. In regards to other graphs of size 2, recall that according to Proposition 2.3.1, graph (c) of Figure 4.2 is a forbidden (sub)graph for any

$\mathcal{M}_{F(X)}$ having 2×2 states that represents a two-dimensional shift space. Therefore, doubly periodic points represented by two states can only have h -cycles of length 1 or 2 -not both, and likewise for v -cycles. For example, if we can locate a subgraph of $\mathcal{M}_{F(X)}$ comprised of two distinct states that have v -loops and that are connected in a h -cycle of length 2, then we will have located a point in the shift space having double period $(2, 1)$. (See graph (a) of Figure 4.3.) These two-dimensional points are rather uninteresting, as they take on the appearance of an infinite number of copies of a one-dimensional point of period 2 that are “stacked” vertically one upon the other. For example,

$$\begin{array}{ccccccc} & & \vdots & \vdots & \vdots & \vdots & \\ & \cdots & a & b & a & b & \cdots \\ & \cdots & a & b & a & b & \cdots \\ & & \vdots & \vdots & \vdots & \vdots & \end{array}$$

is such a point. Therefore, unless otherwise noted, when discussing points having least double period (a, b) we shall assume that $a, b > 1$.



Figure 4.3: Points of double period $(2, 1)$ and $(1, 2)$

It is possible, however, for two states in a graph $\mathcal{M}_{F(X)}$ to be capable of representing a two-dimensional point of double period $(2, 2)$. This graph of minimum size capable of representing a point of least double period $(2, 2)$ is not the graph of maximum size capable of representing a point of least double period $(2, 2)$.

Proposition 4.2.1 *Using the $\mathcal{M}_{F(X)}$ construction to represent a shift of finite type X , there exist exactly two (up to isomorphism) subgraphs capable of representing a point having least double period $(2, 2)$.*

Proof. Consider graph (b) of Fig 4.4 containing four states. By Proposition 4.1.1, any $x \in X$ having least double period $(2, 2)$ has such a set of four states with the corresponding transitions implied by some 2×2 factor Π' of the grid-infinite path representing the point x . Denote by \mathcal{G} this graph corresponding to the block path Π' . To find a smaller graph

representing a point of double period $(2, 2)$, we could identify multiple appearances of the same state within \mathcal{G} and then “glue” these states together to create a graph with distinct states as required in the construction of $\mathcal{M}_{F(X)}$. By inspection, however, we can see that combining any one pair of states in graph (b) of Figure 4.4 would “collapse” a cycle and lead to the creation of a forbidden subgraph. (For example, if we glue state 1 to state 4, nondisjoint h -cycles of length 2 are formed.) We may, however, glue multiple pairs of states even if glueing these states produced forbidden subgraphs when applied independently of one another. Notice, though, that if we glue state 1 to state 2 and glue state 3 to state 4, we create graph (b) of Figure 4.3. That is, the graph no longer represents a point having double period $(2, 2)$. So we see that the only way to glue states while maintaining cycles of length 2 without creating a forbidden subgraph is to glue state 1 to state 4 and glue state 2 to state 3. The result is graph (a) of Figure 4.4. Any further reduction in the size of the graph would collapse both cycles into a single state representing a fixed point. ■

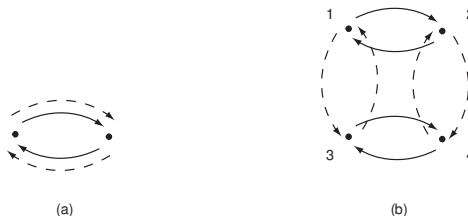


Figure 4.4: Subgraphs representing point of double period $(2, 2)$

We will continue to employ the construction introduced in the proof of Proposition 4.2.1 to create a deterministic graph from a block path. That is, starting with a graph of size ab representing the maximum number of possible states as provided by Corollary 4.1.2, we seek out possible multiple occurrences of the same state. The graph is then decreased in size by glueing together multiple copies of the state into a single state that preserves all transitions. More formally, if $\mathcal{G} = \{Q, E, s, t, \lambda\}$ is such that $\lambda(q_i) = \lambda(q_j)$ for some $i \neq j$, then construct a deterministic graph $\mathcal{G}' = \{Q', E', s, t, \lambda\}$ from \mathcal{G} in the following way. Define the set of states Q' based upon the set of states Q : if $q_i, q_j \in Q$ are such that $\lambda(q_i) = \lambda(q_j)$ (here, $i = j$ is not precluded), then this defines a unique state $q' \in Q'$ having $\lambda(q') = \lambda(q_i) = \lambda(q_j)$. Define the edge set E' based upon the edge set E : if $e \in E$ has source $s(e) = q_s$ and target $t(e) = q_t$, then this defines a unique edge $e' \in E'$ having

$s(e') = q'_s$ and target $t(e') = q'_t$, where $\lambda(q'_s) = \lambda(q_s)$ and $\lambda(q'_t) = \lambda(q_t)$. In particular, when $\lambda(q_i) = \lambda(q_j)$ for some $i \neq j$, the edge in \mathcal{G} having $s(e) = q_i$ and $t(e) = q_j$ will appear as a loop in \mathcal{G}' . We refer to this process of constructing a smaller, deterministic graph from an existing (non-deterministic) graph as *state amalgamation*. The case where an entire cycle is reduced to a single state is referred to as *collapsing a cycle*.

A two-dimensional point of period $(2, 2)$ represented by 4 states as depicted in graph (b) of Figure 4.4 can be located in Figure 2.9 over the four q_i states ($i = 1, 2, 3, 4$). A subgraph over two states as pictured in graph (a) of Figure 4.4 can be located in the Diagonal-shift System. (See Figure 2.7, states q_4 and q_5 .)

Alternately, we could have applied Corollary 2.3.5 to negate the existence of any doubly periodic point of period $(2, 2)$ over a subgraph of size 3. We do so here for future reference.

Lemma 4.2.2 *Given a shift of finite type X , no $x \in X$ of least double period $(a, 2)$ can be represented by a subgraph of $\mathcal{M}_{F(X)}$ comprised of exactly 3 states.*

Proof. By Proposition 4.1.1, a graph representing a point of double period $(a, 2)$ would need to exhibit v -cycles of length 2, but Corollary 2.3.2 states that v -cycles of length 2 must be disjoint. This implies that the only valid combination of v -cycles of length 2 over 3 states would be one v -loop and one v -cycle of length 2. Let q, r and s be the three states in the subgraph, and suppose that the v -cycle of length 2 connects q and r , while s has the v -loop. The proof is completed by the presence - or lack thereof - of a horizontal transition between q and r . Suppose such a transition exists. In this case, the labels of the states must be those given in Corollary 2.3.5. If this were so, then there would be no state that could satisfy the condition for s (has a v -loop) while at the same time coexisting in an h -cycle with q and/or r . On the other hand, if no horizontal transition exists between q and r , then one of q or r must have an h -loop for an edge, while the other must comprise an h -cycle of length 2 with s . (Otherwise, either h -cycles of length 2 are connected, or the three states form two separate subgraphs.) Without loss of generality, suppose q has the h -loop. Then the v -loop at s and the v -cycle of length 2 connecting q and r imply that

$$s = \begin{array}{cc} C & D \\ C & D \end{array} \quad q = \begin{array}{cc} A & A \\ B & B \end{array} \quad \text{and} \quad r = \begin{array}{cc} B & B \\ A & A \end{array} .$$

Then $s \rightarrow r \Rightarrow A = B = D$ and $r \rightarrow s \Rightarrow A = B = C$. That is, $A = B = C = D$ so that

$q = r = s$, and the only points of a shift space that can be represented by a single state are fixed points. ■

Naturally, Lemma 4.2.2 could also be stated in terms of points having least double period $(2, b)$. However, here and in the sequel, we need only focus on points of least double period (a, b) where $a \geq b$, since all other discussions are analogous.

When $\max\{a, b\} = 3$, one must take care not to create graph diamonds of size 2 during the amalgamation process.

Proposition 4.2.3 *Using the $\mathcal{M}_{F(X)}$ construction to represent a shift of finite type X , there exist exactly three (up to isomorphism) subgraphs capable of representing a point having least double period $(3, 2)$.*

Proof. Begin with the collection of 6 states that would appear in a 2×3 block path Π' of a grid-infinite path representing some doubly periodic point of period $(3, 2)$. Form graph \mathcal{G} from the block path Π' as before. States within the h -cycles of length 3 must all be distinct - otherwise there would exist nondisjoint v -cycles of length 2. A valid subgraph is created by collapsing one of the v -cycles of length 2, resulting in a graph consisting of 5 states. (Such a subgraph can be found in the graph given in Figure 2.9, where the edge set does not contain the h -loop at the state labeled p .) However, if we attempt to collapse a second v -cycle, the resulting graph comprised of 4 states would contain a horizontal graph diamond of size 2, while collapsing all three v -cycles would yield the uninteresting point of period $(3, 1)$. By returning to the original graph \mathcal{G} comprised of 6 states, another option is discovered through the amalgamation of certain states in adjacent v -cycles. For example, as depicted in Figure 4.5, we could amalgamate state 1 with state 5 (denote this as the $1/5$ state) and amalgamate state 2 with state 4 (denoted as the $2/4$ state).

Finally, by Lemma 4.2.2, it has been shown that a doubly periodic point of period $(3, 2)$ can not exist over three states, which prohibits any further amalgamation possibilities. So including the original graph of size 6, there exist three distinct graphs capable of representing doubly periodic points of period $(3, 2)$. ■

In particular then, for a point x having least double period $(3, 2)$, a graph presentation

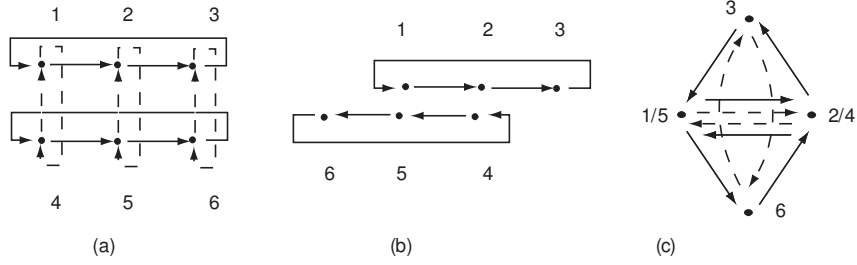


Figure 4.5: State amalgamation

of x must be comprised of at least four states. The Full-square System contains a subgraph of this size representing a point in the shift space having least double period $(3, 2)$.

Example 4.2.4 Consider the Full-square System X defined via the shape $S = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$ and graphed in Figure 3.6. Say we were to define a subgraph by first bisecting the given graph with a vertical line l drawn through the center of the graph and then discarding the right side of the bisected graph as well as the two h -loops on the left side of the graph and all vertical edges intersecting l . Then the remaining subgraph would represent a two-dimensional point of period $(3, 2)$.

In the proof of Proposition 4.2.3, a pair of amalgamations were applied to nearby non-adjacent states located in adjacent v -cycles. An application of Proposition 2.3.3 reveals other general graph structures that appear when states of a block path are amalgamated with nearby states.

Proposition 4.2.5 *Let Π be some grid-infinite path in $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$ representing the point $x \in X$ having double period (a, b) for $a, b \geq 2$, and let Π' be any 2×2 block path of Π where*

$$\Pi' = \begin{array}{ccc} q_{[i,j+1]} & \rightarrow & q_{[i+1,j+1]} \\ \uparrow & & \uparrow \\ q_{[i,j]} & \rightarrow & q_{[i+1,j]} \end{array},$$

is such that $q_{[i,j+1]} = q_{[i+1,j]}$. Then the subgraph representing x must contain a subgraph of 3 states $q, r, s \in Q$ such that $q \rightarrow r, q \uparrow r, r \rightarrow s$, and $r \uparrow s$.

Proof. Set $r = q_{[i,j+1]} = q_{[i+1,j]}$. Then r is the source for both horizontal and vertical transitions having target in state $s = q_{[i+1,j+1]}$, and likewise, r is target for both horizontal and vertical transitions having source in state $q = q_{[i,j]}$. ■

Corollary 2.3.4 can be applied to block paths in the same manner.

Proposition 4.2.6 *Let Π be some grid-infinite path in $\mathcal{M}_{F(X)} = (Q, E, s, t, \lambda)$ representing the point $x \in X$ having double period (a, b) for $a, b \geq 2$, and let Π' be any 2×2 block path of Π where*

$$\Pi' = \begin{array}{ccc} q_{[i,j+1]} & \rightarrow & q_{[i+1,j+1]} \\ \uparrow & & \uparrow \\ q_{[i,j]} & \rightarrow & q_{[i+1,j]} \end{array},$$

is such that $q_{[i,j]} = q_{[i+1,j+1]}$. Then the subgraph representing x must contain a subgraph of 3 states $q, r, s \in Q$ such that $q \rightarrow r, r \uparrow q, r \rightarrow s$, and $s \uparrow r$.

Note that when $a = b = 2$, if the conditions of both Proposition 4.2.5 and Proposition 4.2.6 appear in the same 2×2 block path, then the resulting subgraph is that depicted in Figure 4.4 (a). For example, graph (a) of Figure 4.4 that represents a point of double period $(2, 2)$ is contained as a subgraph of the graph depicted in Figure 4.5 (c) that represents a point of double period $(3, 2)$.

Doubly periodic points having period $(3, 3)$ can assume several different graph representations. As before, form \mathcal{G} from some 3×3 factor Π' of a grid-infinite path Π that represents such a point. The graph \mathcal{G} will contain three h -cycles of length 3, for a total of 9 states (not all of which need be distinct). However, due to the implications of Proposition 2.3.1, we see that each h -cycle of length 3 is either composed of three distinct states (a horizontal graph triangle) or is the same state repeated three times (an h -loop).

It seems feasible that a subgraph of minimum size representing a point x of least double period $(3, 3)$ might be obtained by using state amalgamation to create h -loops and/or v -loops in the graph representing x . So suppose the edge set of the subgraph representing a point of period $(3, 3)$ contains h -loops and/or v -loops. Such a subgraph could be constructed from \mathcal{G} by collapsing one or more of the cycles into a single state. For example, if we collapse one of the h -cycles into a single state, then the resulting subgraph is a valid subgraph over 7 states with exactly one h -loop. In a similar fashion, one of the v -cycles could be collapsed. However, if we attempt to collapse two of the h -cycles, a forbidden graph diamond would be created, while collapsing all three h -cycles would yield the uninteresting point having period $(1, 3)$. It is possible, though, to achieve a valid subgraph by collapsing one of the h -cycles and one of the v -cycles (the choice of

which is irrelevant, due to symmetry). The resulting subgraph will contain a single state that has both h -loops and v -loops. Such a subgraph of 5 states appears in the graph in Figure 2.9 as does a subgraph of size 5 representing a doubly periodic point of period $(3, 2)$, but the two subgraphs have differing edge sets. In particular, we have seen that a graph representing a point of period $(3, 2)$ can not have an edge set containing both h -loops and v -loops.

However, the graph of size 5 discussed above need not be the smallest graph capable of representing a point of least double period $(3, 3)$. To explore further possibilities, suppose the edge set of the subgraph representing a point of period $(3, 3)$ contains neither h -loops nor v -loops, and let us attempt to construct such a graph from \mathcal{G} through state amalgamation. We can apply Propositions 4.2.5 and 4.2.6 to a block path representing a point of double period $(3, 3)$ if we are careful with respect to the intersection of horizontal (vertical) graph triangles that restrict the set of valid labels that may appear on the states involved. To ease further discussion, let us refer to the states within the 3×3 block path Π' numerically. That is, say

$$\Pi' = \begin{array}{ccccc} 1 & \rightarrow & 2 & \rightarrow & 3 \\ & & 1 & & 1 \\ 4 & \rightarrow & 5 & \rightarrow & 6 \\ & & 1 & & 1 \\ 7 & \rightarrow & 8 & \rightarrow & 9 \end{array} ,$$

and let graph \mathcal{G} correspond to this block path.

Now suppose we amalgamate state 1 with state 5 so that the corresponding graph triangles intersect at a single state. (Denote this single state with $1/5$ as before.) The resulting subgraph will then contain a pair of vertical graph triangles and a pair of horizontal graph triangles intersecting with opposite orientations on the h -cycles and v -cycles that pass through the newly formed $1/5$ state. (See graph (a) in Figure 4.6: as depicted, the h -cycles passing through the $1/5$ state are oriented counterclockwise, while the v -cycles passing through the $1/5$ state are oriented clockwise.) Alternatively, we could amalgamate states 5 and 7 so that the resulting subgraph will contain a pair of vertical triangles and a pair of horizontal triangles intersecting with the same orientation on the h -cycles and v -cycles that pass through the newly formed $5/7$ state. (See graph (b) in Figure 4.6.) We apply Proposition 4.2.5 and Proposition 4.2.6 to find valid labels for the states of such

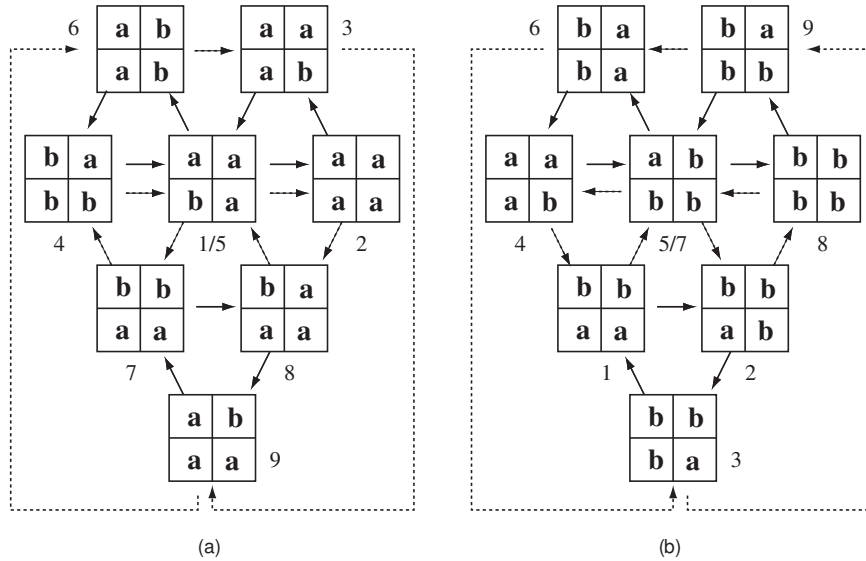


Figure 4.6: Non-isomorphic graphs representing points of period $(3, 3)$

subgraphs: For example, the two non-isomorphic subgraphs depicted in Figure 4.6 both appear in the graph representing the shift of finite type defined by the set of 2×2 blocks populated with symbols a and b according to the restriction that each block contain at least one a . Other amalgamations involving a single pair of states are either forbidden or isomorphic to one of the underlying graphs in Figure 4.6. (For example, amalgamation of state 5 with state $q \in \{2, 4, 6, 8\}$ would create a graph diamond; amalgamation of states 5 and 3 creates a subgraph isomorphic to that of graph (b) in Figure 4.6; and so on.)

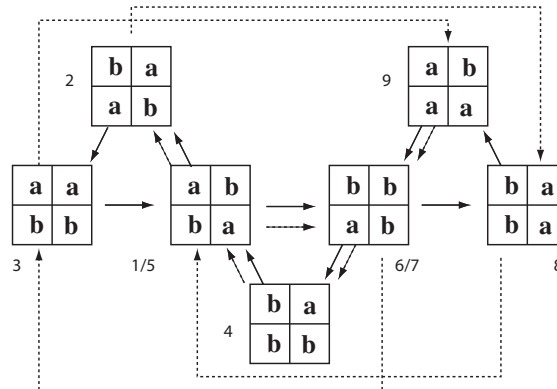


Figure 4.7: Amalgamation of two pairs of states

We next consider pairs of amalgamations involving pairs of states. We can apply Proposition 4.2.5 and Proposition 4.2.6 to two pairs of states and thereby create a subgraph of size 7 that is non-isomorphic to the previously-created subgraph of size 7 that contained

a loop. (See the Full-square System graphed in Figure 3.6 for such a subgraph: Remove the state labeled with all zeroes, remove all edges associated with that state, and remove any remaining h -loops or v -loops.)

Other possible amalgamations involving two pairs of states fall into three distinct cases. Using properties of symmetry, we may without loss of generality first form state $1/5$ and then inspect all other possible amalgamation pairs that may occur within the same subgraph.

- i) The additional amalgamation of $2/6$ or $3/4$ would create a horizontal graph diamond (for example, $2/6 \rightarrow 3 \rightarrow 1/5$ and $2/6 \rightarrow 4 \rightarrow 1/5$), while the additional amalgamation of $4/8$ or $2/7$ would create a vertical graph diamond.
- ii) The additional amalgamation of $6/7$ or $3/8$ would link the three horizontal graph triangles in a way that would maintain constant entries on the main diagonals of five of the seven states in the subgraph. This allows transitions to travel through the amalgamated states without forcing additional labels on nearby states. See Figure 4.7 for an example of a subgraph with valid labels.
- iii) In contrast to case (ii), the additional amalgamation of $2/9$, $4/9$, $6/8$ or $3/7$ would link the three horizontal graph triangles in a way that would force additional labels on other states, thereby reducing the size of the graph. See Example 4.2.7 for the case involving the amalgamation of states 6 and 8.

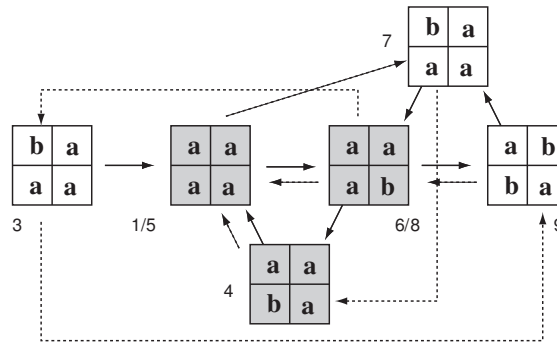


Figure 4.8: Subgraph showing forced labels

Example 4.2.7 Refer to Figure 4.8 for a subgraph of the graph in question. Here, labels on the shaded states are dictated by Corollary 2.3.6. Other labels are forced by first

following the vertical graph triangle $1/5 \uparrow 7 \uparrow 4$, then following the horizontal graph triangle $6/8 \rightarrow 9 \rightarrow 7$, and finally following the vertical triangle $6/8 \uparrow 3 \uparrow 9$. The result is that states 3 and 7 are forced to bear the same label. This further reduces the size of the graph since states 3 and 7 must be amalgamated. The resulting graph of six states would have labels as provided in Figure 4.9.

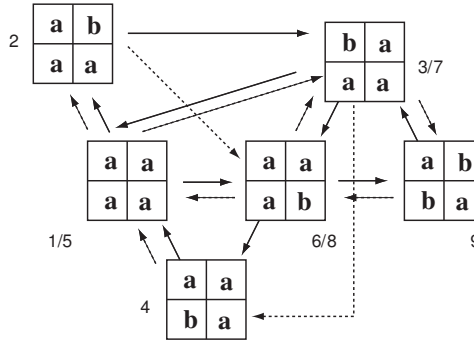


Figure 4.9: Amalgamation of three pairs of states

In search of a graph of minimum size capable of representing some point of least double period $(3, 3)$, the only other inquiry is to check whether three states from three different cycles might be amalgamated. Without loss of generality, consider the amalgamation of states 1, 5, and 9 into a single state. (The amalgamation of states 1 and 5 with any other state would imply the collapse of a cycle.) In fact, a valid subgraph does result from the sequence of amalgamations suggested by initially amalgamating states 1, 5, and 9.

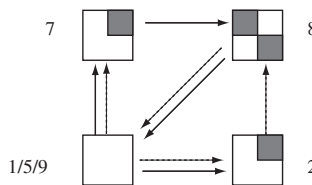


Figure 4.10: Amalgamation of states 1, 5, and 9 dictates other labels

- i) States 2 and 7 may differ only in the upper-right quadrant (Refer to the subgraph in Figure 4.10, which excludes states 3, 4, and 6 since these states are not pertinent.) State 8 must have matching entries along its diagonal. (see Proposition 2.3.3.) Therefore, all entries in the shaded boxes must agree, which implies that states 2 and 7 must have the same labels.

- ii) By the same argument as that found in step i), it can be determined that states 6 and 7 have the same label. Therefore, states 2, 6, and 7 must be amalgamated. (See graph (a) of Figure 4.11.)
- iii) To avoid the existence of horizontal graph diamonds between states 1/5/9 and 2/6/7, states 3, 4, and 8 must be amalgamated. (See graph (b) of Figure 4.11, where one set of valid labels is supplied.)

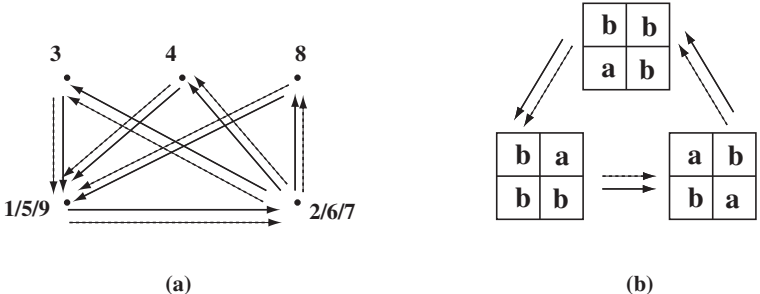


Figure 4.11: Multiple amalgamations suggested by 1/5/9 amalgamation

So it has been demonstrated that 3 states is the minimum number required for a graph to represent the point having least double period $(3, 3)$. There are, however, two non-isomorphic graphs of size 3 capable of this: that is, we could amalgamate the sets of states that appear along the counter-diagonal in the block path representing \mathcal{G} instead. This would create a 3/5/7 state, a 1/6/8 state, and a 2/4/9 state. Such a subgraph of size 3 can be found in the Three-dot System, representing the point

$$\begin{array}{cccccc}
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \cdots & 1 & 0 & 1 & 1 & 0 & 1 & \cdots \\
 \cdots & 0 & 1 & 1 & 0 & 1 & 1 & \cdots \\
 \cdots & 1 & 1 & 0 & 1 & 1 & 0 & \cdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots &
 \end{array}
 \tag{4.2.2}$$

These graph triangles in the automaton representing the Three-dot System take the appearance of h -cycles and v -cycles that exhibit opposite orientations as dictated by Corollary 2.3.4. In this case, there are constant entries along the counter diagonals of points in the shift space. In contrast, the graph of size 3 depicted in diagram (b) of Figure 4.11 has h -cycles and v -cycles with the same orientation, which produces constant entries along the

main diagonals of points in the shift space.

Over the course of examining points having least double period (a, b) for $a, b \in \{1, 2, 3\}$, it was determined that the only doubly periodic points of this type that could be represented by a graph of size 3 were the point $(3, 3)$ and the uninteresting points $(1, 3)$ and $(3, 1)$. We can generalize doubly periodic points with respect to graphs of size 3 by the inclusion of one additional graph.

Example 4.2.8 Consider the graph \mathcal{G} of size 3 given in Figure 4.12. (Such a graph appears as a subgraph of the graph in Figure 3.6 representing the Full-square System.)

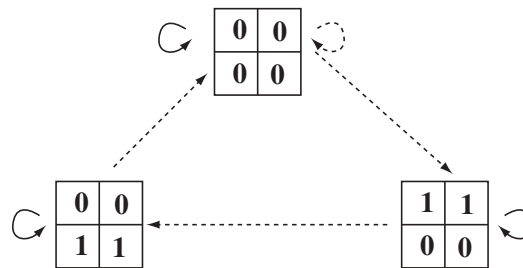


Figure 4.12: Represented points have period $(1, n)$ for all $n \geq 3$

Continuously following the vertical graph triangle of \mathcal{G} describes a point in the shift space X having least double period $(1, 3)$. However, other points of least double period $(1, 3+n)$ are described by repetitions of an extended v -cycle that results from the inclusion of n trips around the v -loop located at the state of all 0's. For example, the following point of least double period $(1, 4)$ results when trips around the vertical graph triangle are interrupted with single trips around the v -loop.

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

Naturally, Figure 4.12 and Example 4.2.8 could be reworked in terms of horizontal

graph triangles and least double periods $(m, 1)$ for $m \geq 3$. This, along with Lemma 4.2.2, verifies the following.

Proposition 4.2.9 *Suppose x is a doubly periodic point of the two-dimensional shift of finite type X . If x can be represented by the subgraph \mathcal{G} of $\mathcal{M}_{F(X)}$ having $|\mathcal{G}| = 3$, then x has least period $(a, b) \in \{(1, n) : n \geq 3\} \cup \{(n, 1) : n \geq 3\} \cup \{(3, 3)\}$.*

Two-dimensional points of the type expressed by (4.2.2) can be viewed as infinite copies of a one-dimensional point having least period a where each row is a copy of the row directly below it that has been shifted diagonally to the right one space. That is, for a point x in the shift space, $x_{(i,j)} = x_{(i+1,j+1)} \forall (i, j) \in \mathbb{Z}^2$. A great variety of these types of doubly periodic points are naturally found in the Diagonal-shift System.

Example 4.2.10 Consider the graph \mathcal{G} of size 4 given in Figure 4.13. (Such a graph appears as a subgraph of the graph in Figure 2.7 representing the Diagonal-shift System.)

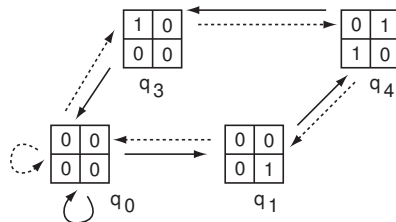


Figure 4.13: Represented points have period (n, n) for all $n \geq 4$

The subgraph \mathcal{G} is capable of representing points having least double period (a, a) for all $a \geq 4$. For example, the point of least double period $(4, 4)$ expressed in (4.2.3) results when trips around the h -cycle (v -cycle) of length 4 are repeated without the inclusion of the h -loops (v -loops) at the state of all 0's.

The periodic point in (4.2.3) is expressed through repetition of the 4×4 block having 1's along the diagonal and 0's elsewhere. Other points of least double period $(4 + n, 4 + n)$ are described by repetitions of an extended h -cycle (v -cycle) that results from the inclusion of n trips around the h -loop (v -loop) located at the state of all 0's. These points of least double period $(4 + n, 4 + n)$ are expressed through repetition of the $(4 + n) \times (4 + n)$ block having 1's along the diagonal and 0's elsewhere.

$$\begin{array}{cccccccccc}
& & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\
\cdots & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \cdots & & \\
\cdots & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \cdots & & \\
\cdots & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & & \\
\cdots & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & & \\
\cdots & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \cdots & (4.2.3) & \\
\cdots & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \cdots & & \\
\cdots & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & & \\
\cdots & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & & \\
& & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & &
\end{array}$$

Proposition 4.2.11 *Consider the set of all shifts of finite type having the property that $F(X) = A(X)$. Then for $a \in \{1, 2, 3\}$, the minimum size of a graph capable of representing some point of least double period (a, a) is a , and for $a \geq 4$, the minimum size of a graph capable of representing some point of least double period (a, a) is 4.*

5 MONOIDS AND FOLLOWER SETS

To each one-dimensional language, one may associate a *monoid* (a set with an associative binary operation and an identity) called the syntactic monoid of the language, which is finite precisely when the language is recognizable. Furthermore, with each one-dimensional recognizable language one may associate a finite automaton of minimal size and a corresponding transition monoid that takes the words of the language as a set of functions acting on the automaton. For a one-dimensional language that is recognizable, it is known that the syntactic monoid of the language is isomorphic to the transition monoid of its minimal automaton. (See, for example, [24].) In one-dimensional symbolic dynamics, a similar notion is that of follower sets, which are the infinite sets of words that can follow any given word in the language of the shift space. (In the one-dimensional case, follower sets are always infinite since words in the allowed language of a one-dimensional shift space are always prolongable and therefore words can always be extended indefinitely.) A one-dimensional shift space is sofic (recognizable) if and only if it has a finite number of follower sets [25]. In a labeled graph recognizing a sofic shift space, the follower set of a state is therefore defined to be the collection of labels of paths originating at that state.

In this chapter, we introduce the idea of monoids and follower sets in the two-dimensional case. This allows us to investigate notions of equivalence in the blocks of a two-dimensional language and in the states of a graph representing a two-dimensional shift space. In Section 5.1 we define monoids based on the allowed concatenations of blocks having the same height. Bounds are then placed on the size of these monoids for certain dot systems. In Section 5.2 we generalize the notion of one-dimensional follower sets to $\mathcal{M}_{F(X)}$ graphs that represent two-dimensional shift spaces. Example 5.2.4 reviews all the key elements of this dissertation.

5.1 Monoids for Dot Systems

In defining a binary operation for two-dimensional blocks, the immediate difficulty is one of closure. If we desire the “product” of two blocks to yield another block (in the sense of concatenation, to be made more precise in a moment), we must return to the convention of inspecting the horizontal case and the vertical case separately, since the product of two blocks having different heights could produce a non-block shape. Some effort has been made to define a type of diagonal product producing non-block shapes, but such research has been limited to one-letter alphabets [1]. To take full advantage of the rich structure inherent in two-dimensional languages, we will look more closely at the definition of follower sets in the next section. Here, we mention some results achieved by limiting the discussion to blocks of a uniform height.

Two blocks of the same height may be *horizontally concatenated* to form a new block. We denote the binary operation by $B_{m,n_1} \ominus B'_{m,n_2}$ and define the resulting block by

$$B_{m,n_1+n_2}(i, j) = \begin{cases} B'_{m,n_1}(i, j) : 0 \leq i \leq n_1 - 1, 0 \leq j \leq m - 1 \\ B''_{m,n_2}(i - n_1, j) : n_1 \leq i \leq n_1 + n_2 - 1, 0 \leq j \leq m - 1 \end{cases} .$$

Informally, the symbols for B''_{m,n_2} are copied to the right of the symbols for B'_{m,n_1} in such a way as to create a new block having dimension $m \times (n_1 + n_2)$. In the same manner, two blocks of the same length may be *vertically concatenated*. That binary operation will be denoted by $B_{m_1,n} \oplus B'_{m_2,n}$ with the resulting block defined by

$$B_{m_1+m_2,n}(i, j) = \begin{cases} B'_{m_1,n}(i, j) : 0 \leq i \leq n - 1, 0 \leq j \leq m_1 - 1 \\ B''_{m_2,n}(i, j - m_1) : 0 \leq i \leq n - 1, m_1 \leq j \leq m_1 + m_2 - 1 \end{cases} .$$

Informally, the symbols for $B''_{m_2,n}$ are copied above the symbols of $B'_{m_1,n}$ in such a way as to create a new block having dimension $(m_1 + m_2) \times n$.

To each picture language L , we may associate semigroups based on the allowed concatenations between blocks. (A semigroup is a set with a binary associative operation.) Consider $R_m = \{\Sigma^{m,n} : n \geq 0\}$, i.e. the set of all blocks having m rows; In particular, the empty block ε is an element of R_m for all m since we can view ε as having dimension $m \times 0$. We define *horizontal syntactic equivalence* \sim_h on R_m relative to L as follows. Given

$\beta_1, \beta_2 \in R_m$, say that $\beta_1 \sim_h \beta_2$ iff $\forall \beta_i, \beta_j \in R_m, \beta_i \ominus \beta_1 \ominus \beta_j \in L \Leftrightarrow \beta_i \ominus \beta_2 \ominus \beta_j \in L$. Equivalence classes shall be denoted by $[\beta]$ and their contexts by $\zeta([\beta])$, e.g. if $\beta_1 \ominus \beta \ominus \beta_2 \in L$, then we would denote this by $(\beta_1, \beta_2) \in \zeta([\beta])$. Right and left contexts are defined in the obvious way. The *m-horizontal syntactic semigroup* of L with the operation $[\beta_1] \ominus [\beta_2] := [\beta_1 \ominus \beta_2]$ is denoted $H_m(L) = R_m / \sim_h$. Note that unless $\Sigma^{m,*} \subseteq L$, the *m-horizontal syntactic semigroup* of L will always contain an element representing the class of blocks that are not in the language; we denote this *zero* by $\mathbf{0} = \{\beta : \beta \in R_m, \beta \notin F(X)\}$. More importantly, for all m , the semigroup $H_m(L)$ will contain the *identity element* $\mathbf{e} = [\varepsilon]$. (For any block $\beta_1 \in R_m$, $[\beta_1] \ominus [\varepsilon] = [\beta_1 \ominus \varepsilon] = [\beta_1]$ and $[\varepsilon] \ominus [\beta_1] = [\varepsilon \ominus \beta_1] = [\beta_1]$.) So in fact, $H_m(L)$ is a monoid.

Vertical semigroups may be defined on $C_n = \{\Sigma^{m,n} : m \geq 0\}$, the set of all blocks having n columns, in a similar manner. Define *vertical syntactic equivalence* \sim_v on C_n relative to L as follows Given $\beta_1, \beta_2 \in C_n$, say that $\beta_1 \sim_v \beta_2$ iff $\forall \beta_i, \beta_j \in C_n, \beta_i \oplus \beta_1 \oplus \beta_j \in L \Leftrightarrow \beta_i \oplus \beta_2 \oplus \beta_j \in L$. The *n-vertical syntactic semigroup* of L with the operation $[\beta_1] \oplus [\beta_2] := [\beta_1 \oplus \beta_2]$ can be denoted $V_n(L) = C_n / \sim_v$. However, we shall discuss horizontal syntactic equivalence only, as vertical syntactic equivalence is analogous.

Now suppose X is a dot system defined through some $r \times c$ shape S . For ease of notation, when discussing the factor language of a dot system we denote $H_m(F(X)) = H_m(A(X))$ by simply H_m . For $m < r$, any two blocks β_1, β_2 of height $m < r$ may be horizontally concatenated within the language. (See Lemma 1.3.1.) In other words, for all $\beta_1, \beta_2 \in R_m$, the fact that $\beta_1 \ominus \varepsilon \ominus \beta_2 \in L$ implies that $(\beta_1, \beta_2) \in \zeta(\mathbf{e})$ so that H_m must be comprised of the single class $[\varepsilon] = \mathbf{e}$. Therefore, when discussing the cardinality of H_m in the sequel, we shall assume $m \geq r$. On the other hand, for $m \geq r$, H_m will always contain both the zero $\mathbf{0}$ and the identity \mathbf{e} , and these two classes are included in the size of the *m-horizontal monoid*.

Example 5.1.1 Suppose the dot system X is defined by some $r \times 1$ shape S . (Here, it is irrelevant whether or not free cells exist in the minimal rectangle T that contains S .) Since $c = 1$, horizontal translations of the shape S can not intersect two columns simultaneously. Therefore if we fix m , we see that $\forall \beta_1, \beta_2 \in F(X) \cap R_m, \beta_1 \ominus \beta_2 \in F(X)$. So the only two equivalence classes in H_m are $\mathbf{0}$ and \mathbf{e} . Therefore $|H_m| = 2$ for all m .

It is helpful to think of \mathbf{e} as representing all pairs of blocks (of equal height) that may

be horizontally concatenated within the language $F(X)$. That is, if $(\beta_1, \beta_2) \in \zeta(\mathbf{e})$, then β_1 and β_2 meet in direction $(1, 0)$ at distance 0 within $F(X)$. Unlike the dot systems defined by one-column shapes as in Example 5.1.1, for most dot systems horizontal (and vertical) translations of the defining shape S dictate whether blocks may meet in direction $(1, 0)$ and if so at what distance. This affects the size of the horizontal syntactic monoids. It should be noted that the zero $\mathbf{0}$ representing concatenations involving blocks $B \in R_m \setminus F(X)$ also represents pairs of blocks from the language that do not meet in direction $(1, 0)$ within $F(X)$. So if $\beta_1, \beta_2 \in F(X) \cap R_m$ and β_1 does not meet β_2 in direction $(1, 0)$ within $F(X)$, then $(\beta_1, \beta_2) \in \zeta(\mathbf{0})$.

To determine the size of the r -horizontal syntactic monoid H_r , it must be determined how many horizontal translations of S will affect $|H_r|$ since there is no bound on the length of the blocks that may represent the equivalence classes. For a given shape, we therefore inspect blocks of length $n = 1, 2, \dots$ to find “good” representatives for the equivalence classes of H_r . Let us first inspect two examples where horizontal translates of S do not affect the size of H_r regardless of the length of the blocks used to represent the equivalence classes. This makes it possible to easily determine the size of both H_r and H_m .

Example 5.1.2 Suppose the dot system X is defined by the 1×2 shape $S = \{(0, 0), (1, 0)\}$. An example of a point contained in this shift space is:

$$\begin{array}{ccccccc}
 & & \vdots & \vdots & \vdots & \vdots & \\
 \dots & 1 & 1 & 1 & 1 & \dots & \\
 \dots & 0 & 0 & 0 & 0 & \dots & \\
 \dots & 1 & 1 & 1 & 1 & \dots & \\
 \dots & 1 & 1 & 1 & 1 & \dots & \\
 & & \vdots & \vdots & \vdots & \vdots &
 \end{array}$$

For a block $B_{m,n} \in F(X)$, all horizontal translations of S within B must maintain a sum equivalent to zero. This requires that one-row designs be populated with the same symbols as those found in the initial column of any given block; that is, $\forall j \in \{0, 1, \dots, m-1\}$, either $B_{m,n}(i, j) \mapsto \{0\}$ for all $i \in \{0, 1, \dots, n-1\}$ or $B_{m,n}(i, j) \mapsto \{1\}$ for all $i \in \{0, 1, \dots, n-1\}$. On the other hand, for one-column designs there is no restriction on the string of symbols that may appear since according to Corollary 1.3.2, $|F_{m,1}(X)| = 2^m$. Therefore, the

set $F_{m,1}(X)$ can be used to represent 2^m distinct equivalence classes for H_m : that is, for $B_{m,1}, B'_{m,1} \in F_{m,1}(X)$, $B_{m,1} \sim_h B'_{m,1}$ iff $B_{m,1} = B'_{m,1}$. So including $\mathbf{0}$ and \mathbf{e} , $|H_m| = 2 + 2^m$.

Example 5.1.3 generalizes Example 5.1.2 to a rectangular shape comprised of two rows. For this shape, one must consider the sum of the symbols (rather than just a single symbol) within each column of S . Although the symbols appearing in a one-row design of length n need not be uniform as in Example 5.1.2, the columns that may appear in horizontal translates of S are restricted by a wallpaper pattern.

Example 5.1.3 Recall that the 2×2 shape $S = \{(0, 0), (1, 0), (0, 1), (1, 1)\}$ defines the Full-square System X . An example of a point contained in this shift space would be:

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

If $B \in F_{2,n}(X)$, then taking the sum over any column of B must yield the same result: that is, either $B(i, 0) + B(i, 1) \equiv 0 \pmod{2}$ for all $i \in \{0, 1, \dots, n-1\}$ or $B(i, 0) + B(i, 1) \equiv 1 \pmod{2}$ for all $i \in \{0, 1, \dots, n-1\}$. Let $T = \{(0, 0), (1, 0)\}$ and define the two one-column designs $\beta_0 : T \mapsto \{0\}$ and $\beta_1 : T \rightarrow \{0, 1\}$ with $\beta_1(0, 0) = 0$ and $\beta_1(1, 0) = 1$. Note that both (β_0, β_0) and (β_1, β_1) are in $\zeta(\mathbf{e})$. However, $(\beta_0, \beta_0) \in \zeta([\beta_0])$ but $(\beta_1, \beta_1) \notin \zeta([\beta_0])$ whereas $(\beta_1, \beta_1) \in \zeta([\beta_1])$ but $(\beta_0, \beta_0) \notin \zeta([\beta_1])$. Therefore H_2 can be represented by $[\beta_0], [\beta_1], \mathbf{e}$, and $\mathbf{0}$ so that $|H_2| = 2 + 2^1$.

For the size of the m -horizontal syntactic monoid H_m when m is allowed to exceed $r = 2$, first consider one-column designs of height m and the equivalence classes that they represent. In this case, there will be $m - 2$ vertical translations of the shape S along any one-column design $B \in R_m$. The original (normalized) shape S and the $m - 2$ vertical translations of S that intersect the one-column block B will therefore produce a (vertical) sequence of length $m - r + 1 = m - 2 + 1 = m - 1$ over the alphabet $\Sigma = \{0, 1\}$ as sums are taken within the initial column of translates of S . By Corollary 1.3.2 there will be no restriction on the string of symbols that may appear in a one-column design so that

2^{m-1} distinct sequences are produced. Furthermore, due to the lack of a free cell in the two-column shape S , this sequence of sums dictates the sequence of sums for all other columns appearing in any horizontal block extension of B . (For the 2×2 shape S , there are only two wallpaper patterns: either the sum of the bits of each column is even or the sum of the bits in each column is odd.) Therefore, blocks in the left and right context of block B must exhibit the same pattern of columnar sums as each other and as B itself. Note that \mathbf{e} is a distinct class since all one-column blocks of height m are in the right context of \mathbf{e} ; that is, $(\varepsilon, B) \in \zeta(\mathbf{e})$ for all $B \in F_{m,1}(X) = \{0, 1\}^{m,1}$. So with the exception of \mathbf{e} and $\mathbf{0}$, there exists a one-to-one correspondence between these 2^{m-1} sequences and equivalence classes of H_m . Therefore, for $m \geq 2$, $|H_m| = 2 + 2^{m-1}$.

Note that in Example 5.1.3, all one-column blocks are in the right context of \mathbf{e} . However, when a one-column block is used to represent an equivalence class, it discriminates between other one-column blocks that may be contained in its right context. In this way, the length c of the shape S affects the size of the r -horizontal syntactic monoid for all $r \times c$ shapes, since blocks of length $n < c - 1$ represent equivalence classes that do not contain enough information about the shape S to discriminate between other blocks of insufficient length ($\eta < c - n$) in their right or left context. Lemma 5.1.4 describes these “undersized” blocks and the equivalence classes that they represent. The Lemma pertains to all dot systems, not just those defined by rectangular shapes lacking free cells.

Lemma 5.1.4 *Let X be a dot system defined through an $r \times c$ shape S having $c \geq 2$. For $0 \leq n < c - 1$, if $B'_{r,n'} \sim_h B_{r,n}$, then $n' = n$.*

Proof. Consider some design $B_{r,n}$ for a fixed n having the property $0 \leq n < c - 1$. We shall use two conditions to sort blocks of arbitrary length into the (right) context of the equivalence class represented by $B_{r,n}$.

Suppose $\eta \in \{0, 1, \dots, c - n - 1\}$. Then since $n + \eta < c$, Lemma 1.3.1 guarantees that any undersized block $\beta \in \{0, 1\}^{r,\eta}$ of length η is in the right context of $[B_{r,n}]$ regardless of the design β . So if $B'_{r,n'} \sim_h B_{r,n}$, then it must be the case that $n' \leq n$ in order for $n' + \eta \leq n + \eta < c$ for all η .

Now suppose $\eta \geq c - n$. In this case, $n + \eta \geq c$ implies that a block $\beta \in \{0, 1\}^{r,\eta}$ of length η is in the right context of $[B_{r,n}]$ only if $\sum_{w \in S} (B_{r,n} \ominus \beta)_w \equiv 0 \pmod{2}$. So if

$B'_{r,n'} \sim_h B_{r,n}$, then it must be the case that $n' \geq n$ in order for $n' + \eta \geq n + \eta \geq c$ for all η . ■

Lemma 5.1.4 suggests a lower bound on the size of the r -horizontal syntactic monoid for any dot system. The bound is based on the need for distinct equivalence classes for blocks of length $n \leq c - 1$.

Proposition 5.1.5 *If X is a dot system defined through an $r \times c$ shape S , then $|H_r| \geq 2 + 2(c - 1)$.*

Proof. Two elements of H_r always exist and are always distinct: the zero $\mathbf{0}$ always exists (the full shift does not exist for dot systems defined by finite shapes S) and is always distinct (a block $B \in R_m \setminus F(X)$ cannot belong to the right context of any other element); and the identity \mathbf{e} always exists (we can always concatenate zero blocks of any height) and is always distinct (\mathbf{e} is the only element of H_r to contain all one-column blocks in its right context unless S has only one column as in Example 5.1.1, in which case Example 5.1.1 provides that $|H_2| = 2 \geq 2 + 2(c - 1)$ since $c = 1$). Lemma 5.1.4 indicates that blocks of length $c - 1$ may not belong to any equivalence class $[B_{r,n}]$ having $0 < n < c - 1$. On the other hand, for a block $B_{r,c-1}$ of length $c - 1$, Corollary 1.3.2 guarantees the existence of $2^{r(c-1)}$ designs $B_{r,c-1} : T \rightarrow \{0, 1\}$, so that there exist at least two distinct equivalence classes for each n based on the cases $\sum_{w \in S \cap T} B_{r,c-1}(w) \equiv 0 \pmod{2}$ and $\sum_{w \in S \cap T} B_{r,c-1}(w) \equiv 1 \pmod{2}$. In particular, $|H_r| \geq 4$ when $c = 2$. (See Examples 5.1.2 and 5.1.3 for shapes that provide a sharp lower bound for the case when $c = 2$.) When $c \geq 3$, Lemma 5.1.4 indicates the existence of distinct equivalence classes for blocks of length $n \in \{1, 2, \dots, c - 2\}$. For all such n , Corollary 1.3.2 guarantees the existence of 2^{rn} designs $B_{r,n} : T \rightarrow \{0, 1\}$, so that there exist at least two distinct equivalence classes for each n based on the cases $\sum_{w \in S \cap T} B_{r,n}(w) \equiv 0 \pmod{2}$ and $\sum_{w \in S \cap T} B_{r,n}(w) \equiv 1 \pmod{2}$. ■

The size of the r -horizontal syntactic monoid H_r is affected not only by the length c of S but also by the location (or the lack thereof) of free cells within the minimal rectangle T that contains S . Free cells in the shape S contribute to the size of the horizontal syntactic

monoid since they may disrupt the symmetry of the shape and/or may contribute to the dot system having factor language that is horizontally transitive. For example, the 2×2 diagonal shape defines a dot system having m -horizontal syntactic monoid larger than that of the 2×2 rectangular (square) shape.

Example 5.1.6 Let X be the Diagonal-shift System defined via the 2×2 shape $S = \{(0, 0), (1, 1)\}$. An example of a point contained in this shift space would be:

$$\begin{array}{ccccccc} & & \vdots & \vdots & \vdots & \vdots & \\ \dots & 1 & 0 & 1 & 1 & \dots & \\ \dots & 0 & 1 & 1 & 0 & \dots & \\ \dots & 1 & 1 & 0 & 1 & \dots & \\ \dots & 1 & 0 & 1 & 1 & \dots & \\ & & \vdots & \vdots & \vdots & \vdots & \end{array}$$

Points in this shift space can be placed in a one-to-one correspondence with points in the one-dimensional full 2-shift: First map any bi-infinite sequence $\dots x_{-1}x_0x_1\dots$ onto the x -axis according to the map $P : x_i \mapsto (x_i, 0)$ for $i \in \mathbb{Z}$; then map shifts of the bi-infinite sequence in discrete time steps according to the map $P_t : x_i \mapsto (x_{i+t}, t)$ for $t \in \mathbb{Z}$. Note that since any two blocks may be concatenated in the language of the one-dimensional full 2-shift, the one-dimensional full 2-shift has only one element, the identity, in its syntactic monoid. We will demonstrate that in the two-dimensional case, this shape creates an m -horizontal syntactic monoid of substantial size.

Define the four blocks $\beta_i : \{(0, 0), (0, 1)\} \rightarrow \{0, 1\}$ by

$$\beta_0(0, 0) = 0, \beta_0(0, 1) = 0$$

$$\beta_1(0, 0) = 1, \beta_1(0, 1) = 0$$

$$\beta_2(0, 0) = 0, \beta_2(0, 1) = 1$$

$$\beta_3(0, 0) = 1, \beta_3(0, 1) = 1$$

Each of these four blocks represents a distinct equivalence class in H_2 . For example, the reader can easily verify that while β_0 and β_2 have the same right context, their left contexts

differ; on the other hand, β_1 and β_3 have the same right context (which differs from that of β_0 and β_2), but their left contexts differ. Other than \mathbf{e} and $\mathbf{0}$, there are no other equivalence classes for $|H_2|$, since for all $n \geq 1$, $(B''_{2,n'}, B'_{2,n'}) \in \zeta([B_{2,n}])$ iff $B''_{2,n'}(n' - 1, 0) = B_{2,n}(0, 1)$ and $B_{2,n}(n - 1, 0) = B'_{2,n'}(0, 1)$. Therefore, $|H_2| = 2 + 2^1 2^1$.

Now consider the size of the m -horizontal syntactic monoid H_m when m is allowed to exceed $r = 2$. Unlike Example 5.1.3, given $B_{m,n} \in F(X)$ representing a equivalence class in H_m , the columns of block $B_{m,n}$ appearing within horizontal translates of S need not carry any particular pattern of sums. However, blocks in the factor language do exhibit a discernible pattern that must cycle through a certain number of columns. For example, if we consider three blocks $B_{m,n}, B'_{m,n'}, B''_{m,n''} \in F(X) \cap R_m$, then $(\varepsilon, B'_{m,n'}) \in \zeta([B_{m,n}])$ iff for $j \in \{0, 1, \dots, m - 2\}$ the cells $B'_{m,n'}(n' - 1, j) = B_{m,n}(0, j + 1)$, and $(B''_{m,n''}, \varepsilon) \in \zeta([B_{m,n}])$ iff for $j \in \{0, 1, \dots, m - 2\}$ the cells $B_{m,n}(n - 1, j) = B''_{m,n''}(0, j + 1)$. So for any two blocks, $B_{m,n} \sim_h B'_{m,n'}$ if and only if $B_{m,n}(n - 1, j) = B'_{m,n'}(n' - 1, j)$ for $j \in \{0, 1, \dots, m - 2\}$ and $B_{m,n}(0, j) = B'_{m,n'}(0, j)$ for $j \in \{1, 2, \dots, m - 1\}$. Therefore there are 2^{m-1} different one-column designs β_L corresponding to distinct equivalence classes $[\beta_L]$ according to $(\varepsilon, B'_{m,n'}) \in \zeta([\beta_L])$ and there are 2^{m-1} different one-column designs β_R corresponding to distinct equivalence classes $[\beta_R]$ according to $(B''_{m,n''}, \varepsilon) \in \zeta([\beta_R])$. Since $F(X)$ is horizontally transitive, for every ordered pair of one-column blocks (β_L, β_R) , there exists a block of length $n = m + 1$ such that for $0 \leq j \leq m - 1$, $\beta_L = B_{m,m+1}(0, j)$ and $\beta_R = B_{m,m+1}(m, j)$. That is, since the initial column of $B_{m,m+1}$ affects only $B_{m,m+1}(i, j) = B_{m,m+1}(i + 1, j + 1)$ for $0 \leq i \leq m - 1$ and $i \leq j \leq m - 1$, the design of the initial (far-left) column of $B_{m,m+1}$ is independent of the design of the final (far-right) column of $B_{m,m+1}$. (The initial column only affects the cells in and above the counterdiagonal.) Therefore, the blocks of length $m + 1$ represent $2^{m-1} 2^{m-1}$ distinct equivalence classes. Furthermore, any block of length $n \geq 1$ can be placed into one of these equivalence classes based on its initial and final columns. For example, since $F(X)$ is horizontally transitive, there exists some block of length $m + 1$ where the design of the initial column is the same as that of the final column, so that such blocks are equivalent to the set of one-column designs. So for $m \geq 3$, $|H_m| = 2 + (2^{m-1})^2$.

Consider the *translation sequence* $x = x_1, x_2, \dots, x_{c+n-1}$ of even and odd sums taken over the cells that lie in the $c + n - 1$ distinct translates of S that intersect blocks of height

r and length n . For the diagonal shape $S = \{(0, 0), (1, 1)\}$, horizontal translates of the shape S intersect pairs of (diagonal) cells in a representative block once and only once, so that for this shape each of 2^{c+n-1} sequences is realized. Proposition 5.1.7 extends this process to $r \times r$ shapes comprised of a single diagonal through a square region. Although $r = c$, we denote by c the values that pertain to the length of S .

Proposition 5.1.7 *Suppose $r = c \geq 2$ and let X be a diagonal-shift system defined via the $r \times c$ shape $S = \{(0, 0), (1, 1), \dots, (r - 1, c - 1)\}$. Then*

$$|H_r| = 2 + \sum_{n=1}^{c-2} 2^{c+n-1} + (2^{c-1})^2.$$

Proof. By Lemma 5.1.4, for all shapes and for each $n \in \{1, 2, \dots, c-2\}$ there exist distinct equivalence classes for blocks of length n , since only undersized blocks can indiscriminately accept other undersized blocks in their context. In particular, for the $r \times c$ diagonal shape, there exists 2^{c+n-1} distinct equivalence classes for each $n \in \{1, 2, \dots, c-2\}$, since an equivalence class represented by an undersized block $B_{r,n}$ corresponds to a sequence of length $c+n-1$ based upon the sum of the bits in $B_{r,n}$ that allows $(B'_{r,n'}, B''_{r,n''}) \in \zeta([B_{r,n}])$ whenever $n' + n + n'' = c$. That is, each of $c+n-1$ horizontal translations of S can potentially intersect a design of length n with a sum that is either even or odd. (See Figure 5.1.)

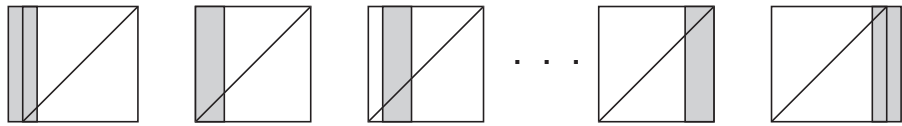


Figure 5.1: Horizontal translations of S

■

Now consider the equivalence classes for blocks of length $n \geq c - 1$. Such classes are best represented by blocks of length $c + 1$ since for blocks of this length, the initial column of $B_{r,c+1}$ is such that

$$B_{r,c+1}(0, 0) = \sum_{i=1}^{i=r-1} B_{r,c+1}(i, i).$$

Since only the cells on the counterdiagonal of the underlying $c \times c$ square shape are affected, the design of the initial (far-left) column of $B_{r,c+1}$ is independent of the design of the final (far-right) column of $B_{r,c+1}$. Therefore, the blocks of length $c+1$ comprise exactly $2^{c-1}2^{c-1}$ distinct equivalence classes since blocks accepted in the left context of a class represented by a block of length $c+1$ are independent of blocks accepted in the right context of the class. Finally, any block β of length $n = c - 1$, $n = c$ or $n > c + 1$ can be placed into an equivalence class represented by a block of length $c + 1$ that has the same initial and final columns as β .

Two blocks $B_{r,n}, B'_{r,n'}$ are equivalent in the r -horizontal syntactic monoid only if they bear the same sum for all horizontal translates of S . The process of counting distinct translation sequences in order to determine the size of the r -horizontal syntactic monoid can be extended to any shape, regardless of the height r of the shape. We can place an upper bound on the size of H_r by assuming that all horizontal translation sequences are achieved. For example, the Diagonal-shift System provides a sharp upper bound for $|H_r|$ related to dot systems defined by $r \times 2$ shapes. The general diagonal system of Proposition 5.1.7 provides a sharp upper bound for all other shapes.

Corollary 5.1.8 *Let X be a dot system defined by an $r \times c$ shape S . Then for $c \geq 3$,*

$$|H_r| \leq 2 + \sum_{n=1}^{c-2} 2^{c+n-1} + (2^{c-1})^2.$$

At the other extreme are dot systems defined by rectangular shapes lacking free cells. For example, let a dot system X be defined by an $r \times c$ rectangular shape and consider the undersized blocks of length $n \leq c - 2$ as representatives for an equivalence class. The condition that two blocks $B_{r,n}, B'_{r,n}$ bear the same sum for all horizontal translates of S , while necessary, is not sufficient for the two blocks to be equivalent in the r -horizontal syntactic monoid. For the purpose of matching wallpaper patterns, two blocks $B_{r,n}, B'_{r,n}$ are equivalent in H_r if and only if they bear the same sum over each column. For reference, Lemma 5.1.9 restates a useful result found within the proof of Proposition 3.1.2. (Refer to Figure 3.1 as needed.)

Lemma 5.1.9 *Let X be a dot system defined through some $r \times c$ rectangular shape S that*

lacks free cells. Then for $B_{r,n}$ with $n \geq c$, $B_{r,n} \in F(X) \Leftrightarrow \forall i' \in \{1, \dots, n - c\}$,

$$\sum_{i=i'}^{i'+c-2} \sum_{j=0}^{r-1} (B_{r,n}(i, j)) = \sum_{j=0}^{r-1} (B_{r,n}(i' - 1, j)) = \sum_{j=0}^{r-1} (B_{r,n}(i' + c - 1, j)).$$

Because of these wallpaper patterns, it is possible to determine the exact size of the m -horizontal syntactic monoids associated with dot systems defined by rectangular shapes that lack free cells. The case when $c = 1$ is encompassed by Example 5.1.1, and all rectangular shapes having $c = 2$ are analogous to Example 5.1.3. (Regardless of the height r of the shape, when $c = 2$ there are only four equivalence classes for H_r : $[\beta_0]$, in whose context all elements have columns that sum to 0 (mod 2); $[\beta_1]$, in whose context all elements have columns that sum to 1 (mod 2); \mathbf{e} , in whose context both (ε, β_0) and (ε, β_1) appear; and $\mathbf{0}$, in whose context (β_0, β_1) appears. For cases where $m \geq r$, there are two possibilities for each one-row extension so that $|H_m| = 2 + 2^{m-r+1}$.) For wallpaper patterns defined by rectangular shapes of length $c \geq 3$, we isolate the case H_r in Lemma 5.1.10 before we examine the more general case H_m . Lemma 5.1.10 uses the notion of primitive words: A *primitive word* is a word that can not be written in the form u^i for any word u and number $i > 1$. So the primitive word is not a power of any other word. (Here, concatenation is taken as multiplication.) If we use D to denote a one-column design with bits that sum to 1 (mod 2) and E to denote a one-column design with bits that sum to 0 (mod 2), then we can refer to wallpaper patterns as words in a one-dimensional language over the alphabet $\Sigma = \{D, E\}$.

Lemma 5.1.10 *For $c \geq 3$, let X be a dot system defined through some $r \times c$ rectangular shape S that lacks free cells. Let $\Delta = \{d : 1 \leq d \leq c - 1, c \equiv 0 \pmod{d}\}$, and let \mathcal{W} be the set representing all wallpaper patterns of length c where the words $w_1 w_2 \cdots w_c = w \in \mathcal{W}$ are such that for $i \in \{1, 2, \dots, c\}$, the symbol $w_i \in \{D, E\}$. Let $W_c \subset \mathcal{W}$ denote the primitive words of length c , and for $d \in \Delta$ let the set $W_d \subset \mathcal{W}$ be the set of all non-primitive words that may be written in the form $w = (w_1 w_2 \cdots w_d)^{c/d}$. Then $|H_r| = 2 + 2^1 + 2^2 + \dots + 2^{c-2} + c|W_c| + (d - 1)|W_d|$.*

Proof. We place arbitrary blocks into equivalence classes based upon their length n . First consider an equivalence class represented by a block $B_{r,n}$ of length $0 \leq n < c - 1$. If we inspect the right context of the equivalence class represented by this undersized block, we

will find that other undersized blocks of varying lengths are grouped into the right context based solely on their lengths; that is, Lemma 1.3.1 guarantees that $(\varepsilon, B'_{r,n'}) \in \zeta([B_{r,n}])$ whenever $n' \leq c - n - 1$, regardless of the design $B'_{r,n'}$. On the other hand, $[B_{r,n}]$ will also contain blocks of length $n' > c - n - 1$ in its right context, but these longer blocks belong to the right context of an equivalence class only if their wallpaper pattern agrees with the partial pattern specified by $B_{r,n}$. For example, if $n' > c$, $(\varepsilon, B'_{r,n'}) \in \zeta([B_{r,n}])$ only if block $\beta = B_{r,n} \ominus B'_{r,n'}$ has the property that $\sum_{j=0}^{r-1} \beta(i, j) = \sum_{j=0}^{r-1} \beta(i + c, j)$ for all $i \in \{0, 1, \dots, n - 1\}$ as in Lemma 5.1.9. Therefore, each wallpaper pattern of length $n < c - 1$ represents a distinct equivalence class, with Corollary 1.3.2 providing for the existence of 2^n such wallpaper patterns. For a block $B'_{r,n'}$ in the right context of $[B_{r,n}]$, if $B'_{r,n'}$ is such that $n' + n \geq c$, then blocks in the left context of $[B_{r,n}]$ must exhibit the same wallpaper pattern as blocks in the right context. Therefore, there exists a one-to-one correspondence between the wallpaper patterns of length $n < c - 1$ and equivalence classes in H_r represented by blocks of length $n < c - 1$.

Now consider an equivalence class that is represented by a block $B_{r,n}$ having length $n \geq c - 1$. By Lemma 5.1.4, these equivalence classes can not be represented by undersized blocks of length $0 \leq n < c - 1$. Furthermore, for blocks of length $c - 1$, $(\varepsilon, B'_{r,c}) \in \zeta([B_{r,c-1}])$ only if block $\beta = B_{r,c-1} \ominus B'_{r,c}$ has the property that $\sum_{j=0}^{r-1} \beta(i, j) = \sum_{j=0}^{r-1} \beta(i + c, j)$ for all $i \in \{0, 1, \dots, c - 1\}$ as in Lemma 5.1.9. So in some sense, blocks of length $n = c - 1$ are the blocks of minimum length needed to hold all the information regarding a particular wallpaper pattern of length c . However, to find a block $B'_{r,c}$ contained in the (right) context of $[B_{r,n}]$ requires not only that the wallpaper pattern of $B'_{r,c}$ agree with that of $B_{r,n}$, but that it match at the point of concatenation (where translates of S simultaneously intersect $B_{r,n}$ and $B'_{r,c}$). Therefore a single wallpaper pattern can spawn multiple (distinct) equivalence classes that are represented by blocks of varying length for some $n \geq c - 1$. To determine how many classes are created by each pattern, first form the set \mathcal{W} of all wallpaper patterns of length c and thereby identify each word w that is not primitive, i.e., $w = (w_1 w_2 \cdots w_d)^{c/d}$. For each $d \in \{2, 3, \dots, c - 1\}$ having the property that $c \equiv 0 \pmod{d}$, denote by W_d the collection of all non-primitive words that may be written in the form $w = (w_1 w_2 \cdots w_d)^{c/d}$. If a word $\omega \in w_d$, then whenever that word is represented by some block of length $n \geq c - 1$, it is also represented by some block of length $n' = c - 1 + d$. Therefore, the word ω produces precisely d distinct equivalence classes, and these can be

best represented by blocks of length $c - 1, c, \dots, c - 1 + (d - 1)$; that is, each non-primitive word (wallpaper pattern) of length c is represented by a block of length $c - 1$ and then creates $(d - 1)$ other distinct equivalence classes until it reaches a block of length $c - 1 + d$ which permits removal of a block of length d and still leaves a block of length $c - 1$ to represent the equivalence class. Since these wallpaper patterns repeat every d translates, blocks of length $n \geq c + d - 1$ permit removal of a block of (some multiple of) length d and still leave a block of length $n \geq c - 1$. In the same way, a primitive word (wallpaper pattern) of length c is represented by a block of length $c - 1$ and then creates c other distinct equivalence classes until it reaches a block of length $2c - 1$ which permits removal of a block of length c and still leaves a block of length $c - 1$ to represent the equivalence class. Since all patterns including the primitive wallpaper patterns repeat every c translates, blocks of length $n \geq 2c - 1$ permit removal of a block of (some multiple of) length c and still leave a block of length $n \geq c - 1$. ■

For an $r \times c$ rectangular shape S , a block $B_{m,n}$ of height $m > r$ still displays a wallpaper pattern of length c , as any subblock $B'_{r,n} \subset B_{m,n}$ must itself display a wallpaper pattern of length c . This makes it possible to determine the exact size of the m -horizontal syntactic monoid. To do so, we consider words (wallpaper patterns) formed from an alphabet of one-column blocks. The alphabet represents the potential sequence of sums that may exist for vertical translates of S within a representative block.

Proposition 5.1.11 *For $c \geq 3$, let X be a dot system defined through some $r \times c$ rectangular shape S that lacks free cells. Let $\Delta = \{d : 1 \leq d \leq c - 1, c \equiv 0 \pmod{d}\}$, and let \mathcal{W}' be the set representing all wallpaper patterns of length c where the words $w'_1 w'_2 \cdots w'_c = w' \in \mathcal{W}'$ are such that for $i \in \{1, 2, \dots, c\}$, the symbol $w'_i \in \{D, E\}^{m-r+1,1}$ is a one-column block. Let $W'_c \subset \mathcal{W}'$ denote the primitive words of length c , and for $d \in \Delta$ let the set $W'_d \subset \mathcal{W}'$ be the set of all non-primitive words that may be written in the form $w' = (w'_1 w'_2 \cdots w'_d)^{c/d}$. Then for $m \geq r$, $|H_m| = 2 + (2^1)^{m-r+1} + (2^2)^{m-r+1} + \dots + (2^{c-2})^{m-r+1} + c|W'_c| + (d-1)|W'_d|$*

Proof. The proof is by induction on $m \geq r$. The case $m = r$ is shown in Lemma 5.1.10. Now assume the result is true for $m > r$. We want to show that for blocks of height $m + 1$, $|H_{m+1}| = 2 + (2^1)^{m-r+2} + \dots + (2^{c-1})^{m-r+2} + c|W'_c| + (d-1)|W'_d|$. Using

the induction hypothesis, place blocks from R_m into equivalence classes based on the wallpaper pattern that appears over their columns. Note that all blocks of length $n \geq c$ have wallpaper patterns that appear over their rows as well. Say we now want to use vertical concatenation to add a new row to some block $B_{m,n}$, producing the new block $B'_{m+1,n}$. For $0 \leq n \leq c-1$ there are no restrictions on the symbols that may appear in the new row of the block extension, but for $n \geq c$ there will be restrictions based on the (vertical) wallpaper pattern (unless $r = 1$, which is inconsequential). Similar to the proof of Lemma 5.1.10, we shall consider cases based on the length n of blocks placed in the equivalence classes.

First consider an equivalence class represented by a block $B_{m+1,n}$ of length $1 \leq n \leq c-2$. If we inspect the right contexts of the equivalence classes represented by these undersized blocks, we again find that other undersized blocks of varying lengths are grouped into the right context based solely on their lengths; that is, Lemma 1.3.1 guarantees that $(\varepsilon, B'_{m+1,n'}) \in \zeta([B_{m+1,n}])$ whenever $n' \leq c-n-1$. On the other hand, $B_{m+1,n}$ will again contain blocks of length $n' > c-n-1$ in its right context, but now these longer blocks will belong to the right context of an equivalence class only if their wallpaper pattern agrees with the partial patterns specified by $B_{m+1,n}$ as S is translated both horizontally and vertically: that is, for $n' \geq c$, $(\varepsilon, B'_{m+1,n'}) \in \zeta([B_{m+1,n}])$ only if block $\beta = B_{m+1,n} \ominus B'_{m+1,n'}$ has the property that for all $i \in \{0, 1, \dots, n-1\}$ and all $j \in \{0, 1, \dots, m-r+1\}$, $\sum_{y=0}^{r-1} \beta(i, j+y) = \sum_{y=0}^{r-1} \beta(i+c, j+y)$. For all $n \leq c-2$, Corollary 1.3.2 guarantees that $|F_{1,n}(X)| = 2^n$ and Lemma 1.3.1 guarantees that $B_{m,n} \oplus \beta_n \in F(X)$ for all $B_{m,n} \in F_{m,n}(X)$ and all $\beta_n \in \{0, 1\}^{1,n}$. Therefore, each equivalence class in H_m represented by a block $B_{m,n}$ of length $n \leq c-2$ creates 2^n additional equivalence classes in H_{m+1} , giving $((2^n)^{m-r+1})2^n = ((2^n)^{m-r+2})$ as desired.

Now consider an equivalence class represented by a block $B_{m+1,n}$ of length $n \geq c-1$. Since horizontal translates of S must always follow a distinct wallpaper pattern, $B_{m+1,n}$ will exhibit a pattern of length c even though the height is increased by one row. Therefore we can apply the argument found in the proof of Lemma 5.1.10 by adjusting the one-dimensional alphabet to be comprised of the symbols $\{D, E\}^{m-r+1,1}$ and then forming the sets \mathcal{W}' , \mathcal{W}'_c , and \mathcal{W}'_d accordingly. ■

There is an inherent difficulty associated with efforts to determine H_m precisely. Due to the presence of both horizontal and vertical translates of S within the representative blocks, a transition from H_m to H_{m+1} does not properly account for the richness that results from interlacing horizontal translates with vertical ones. For example, in a dot system X defined by an $r \times c$ rectangular shape, consider a block of height m that can be represented by a non-primitive word of the form $w' = (w'_1 w'_2 \cdots w'_d)^{c/d}$ as in Proposition 5.1.11. When this block is extended to height $m + 1$, it need no longer be represented by $w' = (w'_1 w'_2 \cdots w'_d)^{c/d}$; rather, it is more likely to now be represented by a primitive word w'_c of length c . Therefore, the process of checking for primitive words must be readjusted at each height.

5.2 Follower Sets and Predecessor Sets

To take full advantage of the $\mathcal{M}_{F(X)}$ construction and the machine's ability to read symbols in a non-linear fashion, the following definitions are put forth.

- The follower set of a state $q \in \mathcal{G} = \mathcal{M}_{F(X)}^\Phi$ is the collection of labels of block paths starting at q . That is,

$$\mathcal{F}_{\mathcal{G}}(q) = \{B_{m,n} : \exists \text{ block path } \Pi' \text{ in } \mathcal{G} \text{ with } \lambda(\Pi') = B_{m,n}, \Pi'(0, 0) = q\}.$$

- The predecessor set of a state $q \in \mathcal{M}_{F(X)}^\Phi$ is the collection of labels of block paths terminating in q . That is,

$$\mathcal{P}_{\mathcal{G}}(q) = \{B_{m,n} : \exists \text{ block path } \Pi' \text{ in } \mathcal{G} \text{ with } \lambda(\Pi') = B_{m,n}, \Pi'(n - 2, m - 2) = q\}.$$

We shall say that graph \mathcal{G} is *follower separated* if each state has a distinct follower set. For a vertex shift \mathcal{G} with distinct labels on the states, \mathcal{G} is naturally follower separated: in this case, $q' \in \mathcal{F}_{\mathcal{G}}(q)$ if and only if $q = q'$. However, when the labels on the states are not distinct, it may be the case that two different states have the same follower set. If we consider the (strictly) sofic shift space Y of Example 2.2.2, a quick inspection reveals that the graph corresponding to Y is follower separated although the labels on the states are

not distinct.

Example 5.2.1 Recall the strictly sofic shift space Y of Example 2.2.2 and the four states q_1, q_2, q_3 , and q_4 of Figure 2.9. For ease of reference, we include here the relabeled graph $\mathcal{M}_{F(X)}^\Phi = \mathcal{G}$.

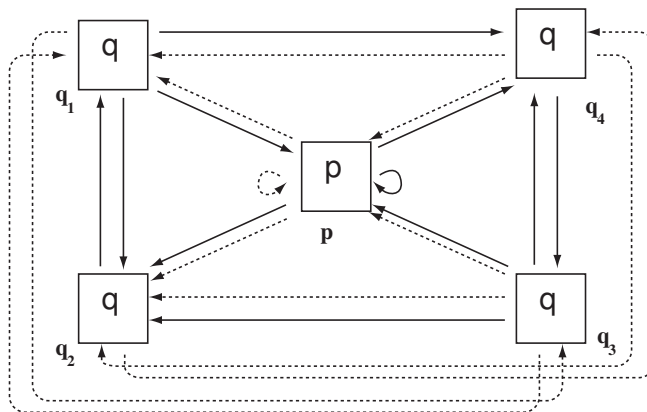


Figure 5.2: Strictly sofic shift with follower-separated graph

Now define the following four blocks from $F(Y)$.

$$\beta_1 = \begin{matrix} p & p \\ q & p \\ q & p \end{matrix} \quad \beta_2 = \begin{matrix} p & p & p \\ q & q & p \\ q & q & p \end{matrix} \quad \beta_3 = \begin{matrix} p & p \\ q & p \end{matrix} \quad \text{and} \quad \beta_4 = \begin{matrix} p & p & p \\ q & q & p \end{matrix}$$

Then for $j \in \{1, 2, 3, 4\}$, $\beta_j \in \mathcal{F}_{\mathcal{G}}(q_i) \Leftrightarrow i = j$. Furthermore, only state p can be followed by the block $B_{2,2}$ defined by $B_{2,2}(i, j) = p$ for $0 \leq i \leq 1, 0 \leq j \leq 1$. Therefore, graph $\mathcal{G} = \mathcal{M}_{F(X)}^\Phi$ is follower separated.

We say that states with the same follower set are *equivalent*. In one-dimensional symbolic dynamics, equivalent states of a graph can be merged to create a smaller graph presenting the same sofic shift. Equivalent states in $\mathcal{M}_{F(X)}^\Phi$ can also be merged without affecting the represented shift space.

Proposition 5.2.2 *Given a graph $\mathcal{G} = \mathcal{M}_{F(X)}^\Phi = \{Q, E, s, t, \lambda\}$ and the equivalence relation $q \sim q'$ iff $\mathcal{F}_{\mathcal{G}}(q) = \mathcal{F}_{\mathcal{G}}(q')$, partition the set of states Q into disjoint equivalence classes Q_1, Q_2, \dots, Q_v . Define a graph \mathcal{G}' with states $Q' = \{Q_1, Q_2, \dots, Q_v\}$ and an edge from Q_i to Q_j exactly when there are vertices $q \in Q_i$ and $q' \in Q_j$ and an edge in \mathcal{G} from*

q to q' . (The labeling function and the definition of acceptance are unchanged.) Then $L(\mathcal{G}) = L(\mathcal{G}')$.

Proof. Denote by Y the two-dimensional shift space represented by $\mathcal{M}_{F(X)}^\Phi$. By definition, the language of the original graph $L(\mathcal{G}) = L(\mathcal{M}_{F(X)}^\Phi) = F(Y)$ is the union of all follower sets of states in \mathcal{G} . (All blocks in the language of the graph start with $\beta_\alpha = q_\alpha$ for some state $q_\alpha \in Q$.) The language of all blocks recognized by \mathcal{G}' is also the union of the follower sets of states in \mathcal{G}' : denote this by $L(\mathcal{G}') = F(Y')$. Since the follower sets are the same, the set of all recognized blocks are the same so that $F(Y) = F(Y')$. Finally, since the language of a shift space uniquely determines the shift space, it must be the case that $Y = Y'$. ■

The merged graph \mathcal{G}' need not conform to the forbidden and forced structures outlined in Section 2.3: In particular, it is possible that graph diamonds will be created during the merging process. This is only possible because the merged graph of a strictly sofic shift space Y will have fewer states than the original graph $\mathcal{M}_{F(X)}$ representing the preimage X of Y . For this reason, the technique of merging states as in Proposition 5.2.2 must be distinguished from that of state amalgamation used in Chapter 4. State amalgamation was applied to merge states with the same label in a graph derived from the block path representing a point in a shift of finite type. In that setting, state amalgamation was used to reduce a graph of size ab representing a periodic point of double period (a, b) in order to determine the size of the subgraph contained within $\mathcal{M}_{F(X)}$, where states are known to have distinct labels. In the present setting, states are merged in a graph representing a sofic shift space in an effort to find a graph of smaller size that is capable of recognizing the same shift space. Note that as constructed, $\mathcal{M}_{F(X)}$ recognizing a two-dimensional shift of finite type X can not be further reduced since each state is a distinct element in the set of blocks that describe the space. It is only possible to reduce the size of the graph $\mathcal{M}_{F(X)}^\Phi$ representing a (strictly) sofic shift that is the image of X under a block code since several states may now have the same label. However, states in a graph representing a sofic shift that have the same label can be merged only if it has been determined that these states have the same follower set. In general, this can be difficult to determine. The following is a simple example in the one-dimensional case that can be generalized to a two-dimensional

setting in order to illustrate the merging of states.

Example 5.2.3 Let X be the one-dimensional strictly sofic shift space comprised of the bi-infinite sequences that contain no more than one 1. (If X were a shift of finite type, there would be a finite set of blocks B_X such that a bi-infinite sequence x would be a point in X if and only if all subblocks of x belong to B_X . For a finite set of blocks, there would be a maximum length N for blocks belonging to the set B_X . Then the bi-infinite sequence $0^*10^N10^* \notin X$ although all subblocks of x would belong to B_X . So there is no finite set of blocks describing X so that X must be strictly sofic.) A graph representing X via labeled states is provided in Figure 5.3. There is neither vertex shift nor edge shift representing X , since both vertex shifts and edge shifts are essentially shifts of finite type [25]. Notice, however, that the states of the graph are follower separated.



Figure 5.3: Graph of one-dimensional strictly sofic shift

Example 5.2.3 can be generalized to two dimensions to create a graph with states whose follower sets are easy to determine.

Example 5.2.4 Consider the shift of finite type X presented by the vertex shift $\mathcal{M}_{F(X)}$ of Figure 5.4. Now define the 1×1 block code ϕ by

$$\phi(z) = \begin{cases} 1 & \text{if } z = 1 \\ 0 & \text{otherwise} \end{cases}$$

Then the sofic shift space Y is the set of all configurations of the plane containing at most one 1. In fact, Y is strictly sofic. To see this, suppose towards a contradiction that Y is a shift of finite type. Then there must exist some $N \geq 1$ such that

$$Y := \{y \in \{0, 1\}^{\mathbb{Z}^2} : \forall v \in \mathbb{Z}^2, \sigma^v(y)_{[-N, N]} \in Q = F_{N, N}(Y)\}. \quad (5.2.1)$$

Now consider the point

$$y(i, j) = \begin{cases} 1 & \text{for } (i, j) \in \{(0, 0), (N + 2, N + 2)\} \\ 0 & \text{otherwise} \end{cases}. \quad (5.2.2)$$

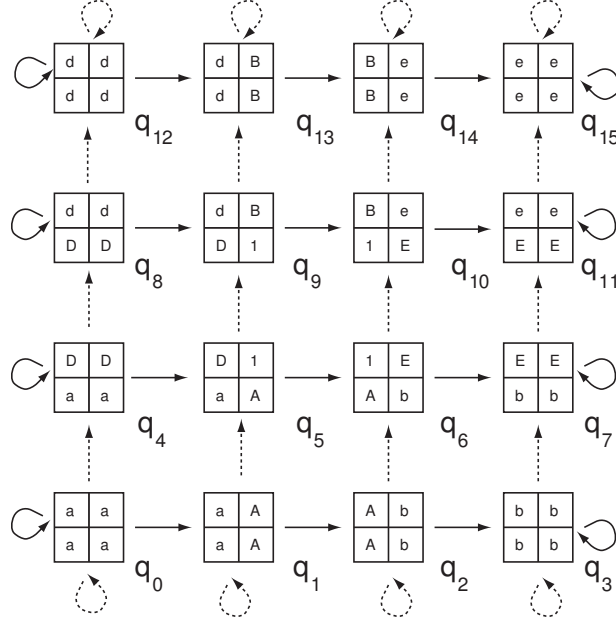


Figure 5.4: Strictly sofic shift; graph not follower separated

Then y is such that $\forall v \in \mathbb{Z}^2, \sigma^v(y)_{[-N, N]} \in Q = F_{N, N}(Y)$, but $y \notin Y$ since this configuration of the plane contains more than one 1.

The graph $\mathcal{M}_{F(X)}^\Phi = \mathcal{G}$ is not follower separated. To see this, let y_0 be the configuration of the plane populated entirely with 0's and let $F(y_0)$ denote all factors of this point, i.e., the set of all blocks populated entirely with 0's. Then for $i \in \{3, 7, 11, 12, 13, 14, 15\}$, $\mathcal{F}_{\mathcal{G}}(q_i) = \{F(y_0)\}$. We can merge these states in graph \mathcal{G} to create a smaller graph \mathcal{G}' representing the same shift space Y . Notice that several states q_i have the same predecessor set as well: if $i \in \{0, 1, 2, 3, 4, 8, 12\}$, then $\mathcal{P}_{\mathcal{G}}(q_i) = \{F(y_0)\}$. Alternatively then, we could have merged these states in graph \mathcal{G} to create a smaller graph representing the same shift space Y . However, if we attempt to merge these states now in the graph \mathcal{G}' , the new graph will no longer represent the shift space Y as points will be accepted that contain more than a single 1. Notice that only states q_3 and q_{12} have both predecessor and follower set equal to $F(y_0)$. So if we merge states q_i for $i \in \{0, 1, 2, 4, 8\}$ to create graph \mathcal{G}'' from graph \mathcal{G}' , we will have reduced the size of the graph while still maintaining distinct initial and terminal states for blocks in the language $F(Y)$ that contain a single 1. In other words, the graph \mathcal{G}'' represents Y . See Figure 5.5 for the merged graph with states labeled according to ϕ .

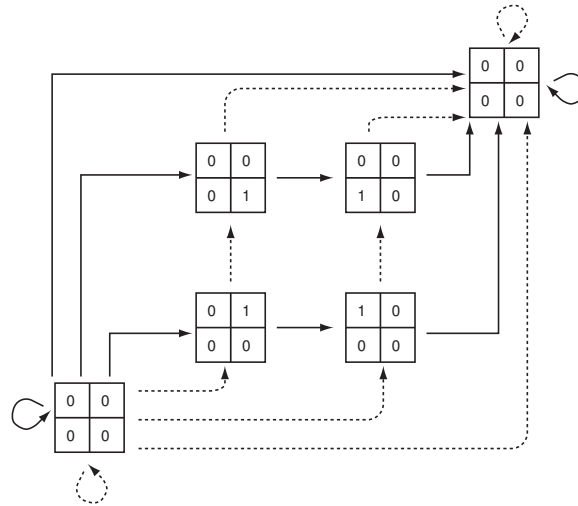


Figure 5.5: Graph of reduced size recognizing Y

With this introduction to the topic of follower sets, we now have all the tools necessary to discuss the properties of graphs representing two-dimensional sofic shift spaces and their corresponding factor languages. The graph recognizing Y in Figure 5.5 is not transitive - nor is the language $F(Y)$, since a block containing the symbol 1 can not meet itself in any direction. The graph is also not deterministic since (for example) the bottom-left state has two horizontal transitions with the same label. Although the graph is follower separated, the bottom-left and top-right states both contain blocks of arbitrary size populated with all zeros in their follower sets. This means that there is not a one-to-one relationship between grid-infinite paths of the graph and points of the shift space, as evidenced by multiple grid-infinite paths representing the same periodic point of all zeros. Example 5.2.4 has brought us full-circle in one other sense: blocks in the language $F(Y)$ may be surrounded by a border of uniform symbols (in this case, the symbol 0) in a fashion reminiscent of the $\#$ symbol used to surround pictures in REC. The study of recognizable picture languages has now been successfully placed within the context of two-dimensional sofic shift spaces.

CONCLUSION

In the literature, investigations surrounding the class REC of recognizable picture languages have been carried out in terms of language theory. This motivated the current research with respect to the set of all factors (blocks) of a two-dimensional sofic shift space. In one-dimensional language theory, there are a great many results regarding FTR (factorial-transitive-recognizable) languages, so that the early focus of this research was on transitivity. It quickly became apparent that there was a need to clarify what was meant by transitivity in the two-dimensional case. The different notions of transitivity in the class of recognizable languages have not been studied before. Even for dot systems there are several questions about transitivity that remain unanswered, as many shapes still elude classification at this time. Dot systems have provided examples of languages that have directional transitivity but not uniformly so, and languages that are mixing but not uniformly so. Missing is an example of a language that is transitive but not uniformly so. (I thank Antonio Restivo for posing the question at the 2006 conference on Advances on Two-dimensional Language Theory.) The classification of two-dimensional languages based on a hierarchy of transitivity promises to be an interesting experiment; for example, Proposition 3.1.8 provides a language that is uniformly transitive yet fails to be transitive in certain directions and therefore cannot be mixing. The characterization of (uniformly) transitive dot systems and, in general, local and recognizable languages remains open.

The graph that is introduced with this paper is the first to successfully represent shifts of finite type as well as their sofic images. Recognition of the points in a (strictly) sofic shift space requires that the scanning mechanism be able to alternate the reading of symbols in the horizontal and vertical directions. This makes the graphs constructed herein an important tool for the study of transitivity in the represented factor languages and periodicity in the represented shift spaces. For example, the graph provides a nice

mechanism to prove Theorem 4.1.4 relating uniform horizontal transitivity to periodicity. Because of the emptiness problem, questions concerning periodicity are central to the discussion of two-dimensional sofic shift spaces. Further work needs to be done with respect to periodic points and their appearance in graphs representing two-dimensional shift spaces. It remains to be seen what other properties of the shift space can and cannot be observed from the graph presentation constructed here. (However, any property that can be observed from the graph structure will necessarily be decidable.) In particular, it remains unclear what the necessary and sufficient conditions are for a two-dimensional graph representation to be a representation of a transitive picture language.

The use of a syntactic monoid of equivalence classes for blocks in the factor language highlights the relevance of transitivity to the properties of a two-dimensional language. It also exposes the shortcomings of treating two-dimensional languages as one-dimensional languages by limiting the height of blocks under consideration. This avenue of research now seems peripheral to the study of recognizable languages defined by two-dimensional shift spaces, although the topic bears some interest in its own right.

The use of equivalence classes for states in a graph can lead to the merging of states, which makes the graph size more manageable and also opens the door for questions regarding the minimal deterministic graph representation of a shift space. A main result in one-dimensional symbolic dynamics is that for a sofic shift space X with factor language that is transitive, a deterministic graph is the minimal deterministic presentation of X if and only if the graph is transitive and the states of the graph have distinct follower sets [25]. It should be determined whether a similar statement applies for two-dimensional sofic shift spaces. The shift space of Example 5.2.4, while not transitive, is a step in the right direction toward linking these concepts in the two-dimensional case. More general languages resulting from block codes of the set of languages that fulfill the property $A(X) = F(X)$ need to be studied in regards to the topic of merging states.

Two-dimensional symbolic dynamics is varied enough from the one-dimensional case to warrant a complete treatment (e.g. textbook) along the lines of the outstanding resource, *An Introduction to Symbolic Dynamics and Coding* by Lind and Marcus [25]. The work in this dissertation provides a solid foundation to connect the key elements of language theory, automata theory, and symbolic dynamics as they relate to the class of two-dimensional factorial and prolongable recognizable languages.

REFERENCES

- [1] M. Anselmo, D. Giammarresi and M. Madonia, New operations and regular expressions for two-dimensional languages over one-letter alphabet, *Theoretical Computer Science* **340** (2005): 408-431.
- [2] R. Berger, The undecidability of the Domino Problem, *Memoirs AMS* **66** (1966).
- [3] F. Blanchard and G. Hansel, Système Codés, *Theoretical Computer Science* **44** (1986): 17-49.
- [4] M. Blum and C. Hewitt, Automata on a two-dimensional tape, *IEEE Symp. Switching Automata Theory* (1967): 155-160
- [5] E.M. Coven, A. Johnson, N. Jonoska, and K. Madden, The symbolic dynamics of multidimensional tiling systems, *Ergodic Theory Dynam. Systems* **23** no. 2 (2003): 447-460.
- [6] E. Coven and N. Jonoska, Uniform transitivity and entropy in multidimensional subshifts, unpublished manuscript.
- [7] D. Giammarresi and A. Restivo, Recognizable picture languages, in: M. Nivat, A. Saoudi and P.S.P. Wang, eds., *Proc. 1st Internat. Colloq. on Parallel Image Processing, Internat. J. Pattern Recognition Artif. Intell.* **6** (1992): 231-256.
- [8] D. Giammarresi and A. Restivo, Two-dimensional languages, *Handbook of formal languages* **3** Springer, Berlin (1997): 215-267.
- [9] W.H. Gottschalk and G.A. Hedlund, Topological Dynamics, *AMS Colloquium Pubs.* **36** (1955).
- [10] J.E. Hopcroft and J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, *Addison-Wesley, Reading* (1979).

- [11] K. Inoue and A. Nakamura, Some properties of on-line tessellation acceptors, *Inform. Sci.* **13** (1977): 95-121.
- [12] K. Inoue and I. Takanami, A survey of two-dimensional automata theory, *Information Sciences* **55** (1991): 99-121.
- [13] K. Inoue and I. Takanami, A characterization of recognizable picture languages, in : *Proc. 2nd Internat. Colloq. on Parallel Image Processing*, Lecture Notes in Computer Science **654** Springer, Berlin (1992): 133-143.
- [14] A.S. Johnson and K. Madden, The decomposition theorem for two-dimensional shifts of finite type, *Proc. Amer. Math. Soc.* **127** no. 5 (1999): 1533-1543.
- [15] N. Jonoska, Sofic shifts with synchronizing presentations, *Theoretical Computer Science* **158** (1996): 81-115.
- [16] N. Jonoska and J.B. Pirnot, Transitivity in two-dimensional local languages defined by dot systems, *International Journal of Foundations of Computer Science* **17** no. 2 (2006): 435-464.
- [17] J. Kari, A small aperiodic set of Wang tiles, *Discrete Mathematics*, **160** (1996): 259-264.
- [18] J. Kari and C. Moore, New results on alternating and non-deterministic two-dimensional finite automata, in *Proceedings of STACS 2001*, LNCS **2010** (2001): 396-406.
- [19] B. Kitchens and K. Schmidt, Automorphisms of compact groups, *Ergod. Th. & Dynam. Sys.* **9** (1989): 691-735.
- [20] B. Kitchens and K. Schmidt, Markov subgroups of $(\mathbb{Z}/2\mathbb{Z})^{\mathbb{Z}^2}$, in *Symbolic Dynamics and its Applications* (P. Walters ed.) *Contemporary Mathematics* **135** (1992): 265-283.
- [21] S.C. Kleene, Representation of events in nerve nets and finite automata, in *Automata Studies* (editors C.E. Shannon and J. McCarthy, Princeton University Press (1956): 3-42.

- [22] G. Lallement, Semigroups and Combinatorial Applications, *John Wiley & Sons* New York (1979).
- [23] M. Latteux and D. Simplot, Recognizable picture languages and domino tiling, *Theoretical Computer Science* **178** (1997): 275-283.
- [24] M. Lawson, Finite Automata, *Chapman & Hall/CRC* (2004).
- [25] D. Lind and B. Marcus, An Introduction to Symbolic Dynamics and Coding, *Cambridge University Press* (1995).
- [26] N. Markley and M. Paul, Matrix subshifts for \mathbb{Z}^{ν} symbolic dynamics, *Proc. London Math. Soc.* **43** (1981): 251-272.
- [27] R. Meester and J.E. Steif, Higher-dimensional subshifts of finite type, factor maps and measures of maximal entropy, *Pacific J. Math.* **200** no. 2 (2001): 497-510.
- [28] D. Perrin, Finite automata, in *Handbook of Theoretical Computer Science* (editor J. van Leeuwen), Elsevier Science Publishers B.V. (1990): 3-57.
- [29] J. E. Pin, Varieties of Formal Languages, *North Oxford Academic* (1986).
- [30] K. Schmidt, Dynamical systems of algebraic origin, *Birkhäuser* (1995).
- [31] H. Wang, *Bell System Tech. Journal* **40** (1961): 1-42.
- [32] H. Wang, Games, logic and computers, *Scientific American* (1965): 98-106.

ABOUT THE AUTHOR

Joni Burnette Pirnot received her Bachelor of Arts in Mathematics from New College of Florida as a New College Foundation Scholar and a Florida Academic Scholar. In 1997, she received her Master of Arts in Mathematics from the University of South Florida and began teaching at Manatee Community College (MCC). She currently lives in Sarasota, Florida, with her husband and two teenage daughters and is an Assistant Professor of Mathematics at MCC.