NON EXPANSIVE DIRECTIONS FOR $\mathbb{Z}^2$ ACTIONS

MICHAEL HOCHMAN

Abstract. We show that any direction in the plane occurs as the unique non-expansive direction of a $\mathbb{Z}^2$ action, answering a question of Boyle and Lind. In the case of rational directions, the subaction obtained is non-trivial. We also establish that a cellular automaton acting on a subshift can have zero Lyapunov exponents and at the same time act sensitively; and more generally, for any positive real θ there is a cellular automaton acting on an appropriate subshift with $\lambda^+ = -\lambda^- = \theta$.

1. Introduction

Consider a $\mathbb{Z}^2$ action (X, T) on a compact metric space. Let ℓ be a line in the plane and let ℓ^r denote the set of points within distance r of ℓ. Then ℓ is said to be an expansive line if there exist $r > 0$ and $\delta > 0$ such that, for all $x, y \in X$,

$$d(T^u x, T^u y) < \delta \text{ for all } u \in \ell^r \cap \mathbb{Z}^2 \quad \Longrightarrow \quad x = y$$

Expansiveness only depends on the direction of the line and not the line itself, so we may speak of expansive and non-expansive directions. Note that if ℓ contains an integer point $u \in \ell \cap \mathbb{Z}^2$ then expansiveness of ℓ is equivalent to expansiveness of the map $T^u : X \to X$, but for ℓ with irrational slope there is no such interpretation.[1]

Expansive and non-expansive directions were defined by Boyle and Lind in [3], where they were used as a tool in the study of the directional dynamics of an action. Many properties of the dynamics of subactions T^u vary nicely withing connected components of the set of expansive directions. For example, within such a component the entropy of subactions varies linearly. Certain properties are constant within expansive components: for example, if T^u acts as a shift of finite type then so does T^v as long as the directions u, v are in the same expansive component.

One of the basic questions that arose in [3] was to understand what sets can occur as the set of non-expansive directions. It was shown that this set is closed and, when the phase space is infinite, non-empty; and furthermore if C is any closed

---

2000 *Mathematics Subject Classification.* 37B05, 37B10, 37B15.
This research was partially supported by the NSF under agreement No. DMS-0635607.

[1] The first thing one tries is to go to a continuous time analog with flow $\{\sigma^t\}_{t\in\mathbb{R}}$, and set $\varphi = \sigma^\theta$; but this does not work since when one goes back to a discrete action one loses expansiveness.

set of directions of cardinality $|C| \geq 2$, then it is the set of non-expansive directions for some action.

It has been an open problem for some time to determine which direction can occur as the unique non-expansive direction in a non-trivial way. If one begins with an expansive $\mathbb{Z}$-action (X, T) and extends it formally to the $\mathbb{Z}^2$ action $(X, \langle T, \mathrm{id}_X \rangle)$ generated by T and the identity map, then one obtains an action whose unique non-expansive direction is the vertical one; but the action in that direction is trivial. One can construct similarly trivial examples in which an arbitrary rational direction is the only non-expansive one. However, attempts have not succeeded in producing non-trivial examples for rational directions (a proposed example in [6] turned out to be flawed, see [2]), or any examples at all of actions with a single irrational non-expansive direction.

In this paper we resolve this problem as follows:

Theorem. *For every direction ℓ in the plane there is an expansive $\mathbb{Z}^2$ action whose unique non-expansive direction is ℓ. In the case ℓ has rational slope, the corresponding subaction is non-trivial, i.e. none of its elements act as the identity.*

Corollary 1.1. *A set of directions occurs as the set of non-expansive directions for an expansive $\mathbb{Z}^2$-action if and only if it is closed and non-empty.*

This follows by combining the theorem with Boyle and Lind's result for sets of size ≥ 2, but can also be derived directly from our construction by taking unions of the systems it provides. The Boyle-Lind examples are also unions of a similar sort. Thus the constructions we have are quite degenerate, in the sense that they decompose into subsystems with small sets of non-expansive directions. The following question is therefore natural:

Problem 1.2. Can every nonempty closed set of directions occur as the non-expansive directions of $\mathbb{Z}^2$-action that is transitive/minimal/supports a global ergodic measure?

Another consequence of our construction is:

Proposition 1.3. *There exists a cellular automaton f such that, for every t, there is a subshift X_t on which f acts as an automorphism without equicontinuity points and such that the Lyapunov exponents are $\lambda^+ = t, \lambda^- = -t$.*

The case $t = 0$ partially answers to a question of Bressaud and Tisseur [4, Conjecture 3], showing that there is a cellular automata acting on a subshift with sensitive dependence on initial conditions, but in which information propagates at a sub-linear rate.

Theorem 1 and the last proposition are related as follows. Suppose we wish to realize a line ℓ as the unique non-expansive direction of a $\mathbb{Z}^2$-action. We shall do so on a zero-dimensional phase space. In this case we may fix an expansive direction with rational slope, and choose another rational direction so that together the actions in these directions generate the full action (or a finite-index subgroup of it). Using expansiveness we may re-code and identify the first direction with the shift on some symbolic space X, and the second direction with an automorphism of X. Thus the problem has been reduced to one of constructing an appropriate shift space and an automorphism of it; this is the same setting as is studied in the theory of cellular automata.

This reduction highlights an interesting aspect of the problem. Each automorphism is given by a block code. There are only countably many of these, but there are uncountably many directions (or values for Lyapunov exponents). Thus if one is to construct examples of automorphisms which realize any given direction as the unique non-expansive one for the generated action (or if one wants to construct CA with arbitrary Lyapunov exponents), then the direction (or exponents) must be encoded at least in part in the subshift rather than the automorphism. We shall make this encoding quite explicit, effectively designing the automorphism as an interpreter and using the subshift as a program controlling the action of the automorphism.

Our strategy will be to construct an automorphism that, roughly speaking, performs a sequence of shifts on the underlying space at a rate that is encoded in the subshift it is acting on; this rate is what will determine the slope of the non-expansive direction. For example, taking the full shift as our space, σ as the shift and the automorphism $\varphi = \sigma^n$, we see that in the generated $\mathbb{Z}^2$-action every direction is expansive except the line $ny + x = 0$. Here φ shifts at a rate of n symbols per unit time. We would like to control this rate so as to make it an arbitrary real number θ. The implementation of this simple idea, however, is rather involved. Our solution relies on a property that has been called *intrinsic universality* in the cellular automata literature (e.g. [1]). This means that there are automorphisms which, when restricted to an appropriate subshift, can simulate any other automorphism up to a temporal and spacial rescaling. We shall use an infinite hierarchy of such automorphisms, each of which simulates the next, and such that each level in the hierarchy performs a shift on the underlying space at a fixed rate (this construction is somewhat reminiscent of Gacs' error-correcting automata [5]). The sum of these rates will determine the overall shift and the non-expansive direction, and we will control these rates by encoding them into the shift space.

We shall mostly use standard definitions and notation, which can be found e.g. in [10]. We denote by σ the shift map on symbol spaces, and for a point $x \in \Sigma^{\mathbb{Z}}$ we denote its coordinates by x_i. For a symbol a we write a^n for the n-fold concatenation of a. An automorphism φ of a subshift Y is given by a block code, and we shall say that φ has range r if the block code acts on a r-neighborhood $[-r,r]$ (this is also sometimes called the radius of φ or its window width). The notation $O_N(1)$ denotes a constant depending only on N.

The rest of this paper is organized as follows. In the next section we introduce a sufficient condition for the action generated by a shift-automorphism and the shift to have a unique non-expansive direction. Sections 3 outlines the main construction, section 4 supplies further details of the implementation. Section 5 applies the construction to prove the main theorem. Section 6 discusses the relation and applications to Lyapunov exponents. Finally, in section 7 we present an simpler, alternative construction of a system with a unique, rational non-expansive direction.

Acknowledgment: I would like to thank Doug Lind for some very interesting discussions and for his permission to include the example in section 7. This work was done in the fall of 2008 during the special semester on additive combinatorics and ergodic theory at MSRI, and I would like to thank the organizers and hosts for that stimulating event.

2. Prediction shapes

In this section we define a device that quantifies propagation of uncertainty under iteration of an automorphism. This device is related to Shereshevsky's notion of Lyapunov exponents for cellular automata [8], although there are a number of differences. First, we are interested in the propagation of uncertainty both forward and backward in time, although one can easily modify our definitions so that they are one sided, and apply to endomorphisms as well. More importantly we measure uncertainty by fixing a finite block, rather than a one-sided infinite ray (in Shershevsky's case one fixes a leaf of the stable or unstable foliations). We shall discuss the relation to Lyapunov exponents further in section 6.

Definition 2.1. Let Y be a subshift and φ an automorphism of Y. A convex, open subset $\Lambda \subseteq \mathbb{R}^2$ is called a *prediction shape* for $\varphi|_Y$ if $(0,1) \times \{0\} \subseteq \Lambda$ and for every compact set $\Lambda_0 \subseteq \Lambda$ and for all n large enough (in a manner which may depend on Λ_0), if $y, z \in Y$ satisfy

$$y|_{[-n,n]} = z|_{[-n,n]}$$

then

$$(\varphi^t y)_i = (\varphi^t z)_i \text{ for all } (i,t) \in n\Lambda_0 \cap \mathbb{Z}^2$$

where $n\Lambda_0 = \{n \cdot u \,:\, u \in \Lambda_0\}$.

From the definition it is clear that the increasing union of prediction shapes is a prediction shape.

If φ, φ^{-1} have range r then the diamond shaped region with vertices at $(-1,0)$ and $(1,0)$ and with sides of slope $\pm 1/r$ is a prediction shape for $\varphi|_Y$. This bound derives from information about the block-code and one may sometimes get more from the block code, but in general the prediction shapes for $\varphi|_Y$ depend non-trivially on Y. For example, if Y is a finite union of periodic orbits then for large enough n the restriction $y|_{[-n,n]}$ determines $y \in Y$; therefore $\mathbb{R}^2$ is a prediction region. On the other hand, for infinite subshifts it is easy to see that no prediction shape contains $[-1,1] \times \{0\}$ in its interior.

Our application of this notion is the following simple observation.

Theorem 2.2. *Let Y be a subshift with automorphism φ. Let ℓ be a line through the origin distinct from the x-axis, and suppose that*

$$\Lambda = \ell^1 = \{u \in \mathbb{R}^2 \,:\, d(u,\ell) < 1\}$$

is a prediction shape for $\varphi|_Y$. Then every direction except ℓ is an expansive direction for the $\mathbb{Z}^2$-system $(Y, \langle \sigma, \varphi \rangle)$.

Proof. Fix a line ℓ' through the origin in a different direction from ℓ. We must show that if r is large enough then $y|_{(\ell')^r}$ determines y. For this it suffices to show that there is an r such that $y|_{(\ell')^r}$ determines $y|_{(\ell')^{r+1}}$.

Since the slopes of ℓ, ℓ' are different, it is easy to see that there is an $\varepsilon > 0$ and a compact set $\Lambda_0 \subseteq \Lambda$ containing the origin with the property that $(\ell')^{1+\varepsilon} \subseteq (\ell')^1 + \Lambda_0$. The desired conclusion now follows from the fact that Λ is a prediction shape. See figure 2.1. □

Corollary 2.3. *Under the assumptions of the theorem, if Y is infinite then ℓ is the unique non-expansive direction for $(Y, \langle \sigma, \varphi \rangle)$.*

Proof. This follows from the theorem and the fact that every $\mathbb{Z}^2$-action on an infinite space must have at least one non-expansive direction [3]. □

3. Main construction

In this section and the next we construct a subshift X over an appropriate alphabet, and define a pair of endomorphisms π and $\widehat{\pi}$ of X by specifying block codes for them. This section describes their properties and outlines the construction. Some further details appear in the next section.

By construction, X, π and $\widehat{\pi}$ will satisfy the following properties. First, the range of π and $\widehat{\pi}$ will be 1, meaning that $\pi(x)_0$ depends only on x_{-1}, x_0, x_1 and similarly for $\widehat{\pi}$.

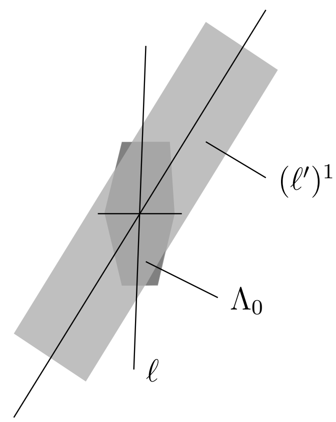

FIGURE 2.1. By shifting the center of the dark region Λ_0 along the line ℓ', one covers $(\ell')^{1+\varepsilon}$ (the area swept out by the darkened corners of Λ_0).

Second, suppose we are given the following parameters:

- An integer N.
- A subshift $Y \subseteq \{1, \ldots, N\}^{\mathbb{Z}}$.
- Block codes of range 1 defining inverse automorphisms φ,φ^{-1} of Y.
- An integer $B \geq 1$ ("Block length"), which is sufficiently large with respect to N, φ.
- An integer $W \geq 1$ ("Wait time").
- An integer D ("Displacement"), which may be positive, negative or 0, the sign indicating displacement to the right or left, respectively.

Then there is a subshift

$$X' = X'(Y, N, \varphi, B, W, D) \subseteq X$$

such that

$$\widehat{\pi}|_{X'} = (\pi|_{X'})^{-1}$$

and an integer

$$T = (B + O_{N,\varphi}(1))(1 + W + |D|) \tag{3.1}$$

so that (X', π) and (Y, φ) are related in the following manner. Each configuration of $x \in X'$ breaks into blocks of length B each representing one symbol from the alphabet $\{1, \ldots, N\}$ of Y, and thus $x \in X'$ encodes a point $y \in Y$. With this interpretation of x, the endomorphism π^T acts on x in the same manner that $\sigma^D \circ \varphi$ acts on Y, i.e. it applies φ to the encoded sequence y without altering the block structure, resulting in a new point $x' \in X$ encoding $\varphi(y)$, and then shifts each block of x' a distance of D blocks, i.e. $D \cdot B$ symbols, mimicking the action of

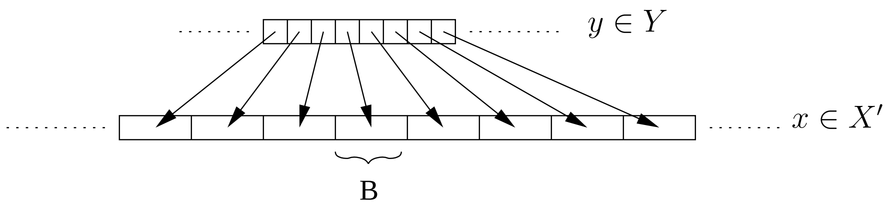

FIGURE 3.1. Encoding a point in y as a sequence of blocks.

σ^D on $\varphi(y)$. More precisely, there is an isomorphism

$$(X', \sigma, \pi) \cong (Y \times \{0, \ldots, B-1\} \times \{0, \ldots, T-1\}, \sigma_B, \varphi_T) \tag{3.2}$$

where σ_B, φ_T are the suspension maps defined by

$$\begin{array}{rcl} \sigma_B(y,b,t) & = & (\quad \sigma^{\{b=0\}}y \quad , \quad b+1 \bmod B \quad , \quad t \quad) \\ \varphi_T(y,b,t) & = & (\quad (\sigma^D\varphi)^{\{t=0\}}y \quad , \quad b \quad , \quad t+1 \bmod T \quad) \end{array}$$

Here we have denoted $\{b=0\} = \delta_{b0}$and $\{t=0\} = \delta_{t0}$. See figure 3.1

What we have required of $\pi, \widehat{\pi}$ is very similar to what is called intrinsic universality, which has been studied in the CA literature (e.g.[1]), although the additional shift by D blocks is special to our construction. Such endomorphisms have been constructed many times, as well as automorphisms [7]. However, we have not found a reference that satisfies all of our requirements exactly, and for this reason and in the interest of completeness we provide an outline of the construction details. In this section we give an overview; in the next we give some of the finer details. However, the properties above are all we shall use about $\pi, \widehat{\pi}$ and one may prefer at this point skip ahead to section 5 where we prove theorem 1.

We construct $\pi, \widehat{\pi}$ in a manner independent of the parameters above; $\pi, \widehat{\pi}$ will operate as "interpreters", and the other parameters will be encoded in the configurations of X' so as to influence the way in which π acts. We shall eventually set $X = \overline{\cup X'}$, where the union ranges over all choices of parameters. Notice that although X is larger than $\cup X'$, nonetheless $\widehat{\pi} = \pi^{-1}$ on X, because this is true on each X' (and $\pi, \widehat{\pi}$ are given by the same block code on all of them).

For the construction we shall assume that the parameters Y, N, φ, B, W, D are given, and describe the block codes for $\pi, \widehat{\pi}$ in a way that is independent of the parameters, and a subshift X' that depends on them.

The **alphabet** of X' consists of quadruples of symbols, which we denote (b, p, s, d): here b stands for Block structure, p for Program, s for State and d for Data. The projection of a sequence onto each of these coordinates are layers: for example the sequence of data components is the data layer.

The symbols used in the data layer will include the symbols $0, 1$, which we call **bits**. We shall represent each symbol of Y by a sequence of $\lceil \log N \rceil$ bits followed by an appropriate terminating symbol; together we call such a sequence a **word**.

The **Data layer** consists of two words, representing symbols from Y, starting at the left side of the block; the remaining space to the right of these words is filled with "blank" symbols. The first word in the pair is interpreted as the symbol of Y currently represented by the block, and the second as the symbol represented by the block in the previous cycle. There will be times when the data will be in a corrupt state, but even then it will be possible to recover the uncorrupted current and previous states from the other layers; see below.

The **Block layer** is a periodic sequence whose period is B and is not modified by π. At this stage we may take the block layer to be a periodic concatenation of the string $1\,0^{B-1}$, and we shall call such a sequence, and also the indices it occupies, simply a block (later on we will add more information to this layer; see section 4). In the identification $X' \cong Y \times \{0, \ldots, B-1\} \times \{0, \ldots, T-1\}$, the second component in the image of $x \in X'$ will be determined by the residue class mod B of the position of 1's in the block layer of x. We remark that since B can be arbitrarily large, X will contain also points whose block layer consists of all 0's, or of a single 1 surrounded by 0's; but as we shall see these will not cause a problem.

The **Program layer** is also periodic with period B and is not modified by π. The repeated sequence, which we call simply the Program, is constant throughout X'. The program begins at the left side with an encoding of the block codes of φ and φ^{-1}. This takes the form of a sequence of 5-tuples of words, representing 5-tuples of symbols from Y. Each 5-tuple represents an input (3 Y-symbols) and the corresponding output symbol of φ and φ^{-1} (recall that both are assumed to have range 1). We separate these 5-tuples from each other with some special symbol, and terminate the sequence of 5-tuples with another special symbol. Next, the program layer contains the parameters W, D encoded as contiguous sequences of 1's, either W or D in number, followed by terminating symbols. The remainder of the program layer is filled with blanks.

Finally, the **State layer** contains auxiliary information used to interpret the Program layer and use it to update the Data layer. We call a sequence of state symbols corresponding to a Block simply a State-block. The state-blocks in different blocks typically differ from each other, since they depend on the data in the block and the neighboring blocks, but they will be synchronized in the following sense: there is a special state-block called the Synchronized State, so that once (and only once) every T applications of π, the synchronized state-block appears in all the blocks of a configuration $x \in X'$. We shall call an $x \in X'$ with all blocks synchronized a synchronized configuration. It is during such a time that the data

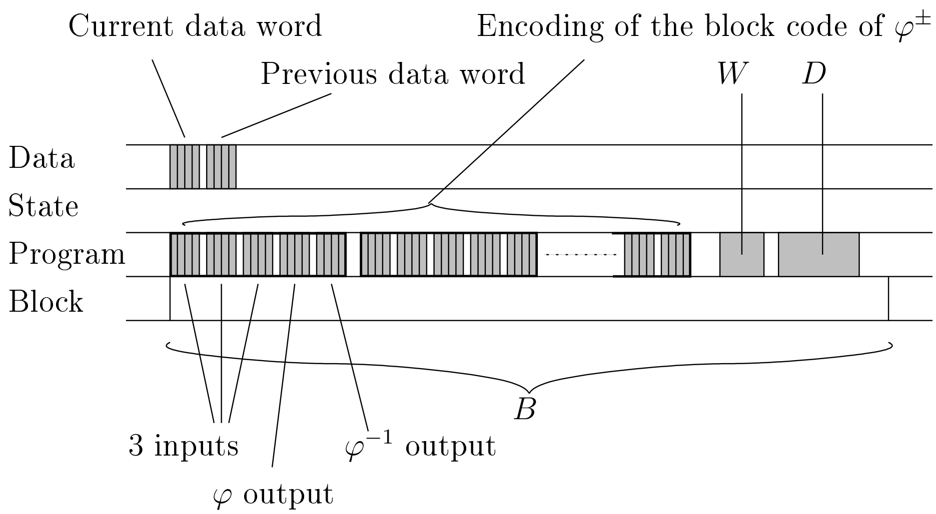

FIGURE 3.2. The arrangement of information inside a block.

layer is guaranteed to represent correctly the current and previous symbols. Thus in the identification $X \cong Y \times \{0, \ldots, B-1\} \times \{0, \ldots, T-1\}$, the third component of the image of x is the number t of applications of π^{-1} needed to bring x to a synchronized configurations, and the first component is the current symbol in the data layer of of $\pi^{-t}x$.

The layout of a block is depicted in figure 3.2

Applying π repeatedly to a synchronized configuration $x \in X'$ leads to the following sequence of events, which we call a Cycle (figure 3.3):

(1) Each block "transmits" its current data word to the neighboring blocks on its left and right, and receives the same information from them. At the end of this stage, the state contains two words, a_L, a_R, representing the current Y-symbol encoded in the neighboring blocks to the left and right of the current block, respectively.

This stage takes $B + O_N(1)$ applications of π to complete (each bit moves a distance of B, taking B time steps, plus a constant amount of time needed too coordinate the transmission which depends only on the number of bits being transmitted, which depends on N).

(2) The two words from the data layer – the current and previous Y-symbols – are copied to the state layer and simultaneously deleted from the data layer. If a_C is the current symbol and a_P the previous symbol of the block, then the state space now contains the 4-tuple $a_L a_C a_R a_P$ of Y-symbols.

This stage takes $O_N(1)$ applications of π to complete.

(3) We now enter a loop in the course of which the 4-tuple in the state layer is translated to the right, stopping opposite each 5-tuple in the program layer.
   (a) For each 5-tuple it checks if the triple of words $a_L a_C a_R$ matches the input-triple in the program layer.
   (b) When a match is found the words a_L, a_R, and a_P in the state layer are erased, and the corresponding output word $b = \varphi_0(a_L a_C a_R)$, which is encoded in the data layer, is copied to the state layer. The state layer now contains the current Y-symbol a_C and future Y-symbol b.
   (c) The comparisons continue also after a match is found. The implementation will be such that each comparison takes the same number of steps, whether or not a match is found. A match will be found exactly once.

   Since the comparisons continue at the same rate after a match is made, the number of applications of π in this stage is independent of the configuration, and this stage ends at the same point in the cycle for all blocks in a configuration. The time for this step is $O_{N,\varphi}(1)$.
(4) The current and future Y-symbols are translated back through the state layer to the left end of the block and transferred to the data layer: b is copied to the "current" slot and a_C to the "previous" slot, and they are simultaneously deleted from the state layer.

   This step takes $O_{N,\varphi}(1)$ applications of π to complete.
(5) The data layer is shifted $|D|$ blocks to the left or right, according to the sign of D. For this, the sign is first determined, and then a loop is performed during each iteration of which the data layer is shifted by one block length.

   This step takes $(B+O_N(1))\,|D|$ applications of π; the B term corresponds to actually transporting the data. The $O_N(1)$ term is the overhead required each cycle. We give further details in the next section.
(6) The state layer "mock-shifts" the data layer W more times, meaning that the state goes through a cycle which takes the same amount of time as shifting it one block, but doesn't result in such a shift taking place.

   This step takes $(B + O_N(1))W$ applications of π.
(7) All blocks return to the synchronized state.

   This takes $O_N(1)$ applications of π.

Part of a cycle is depicted in figure 3.3.

Given the parameters Y, φ etc., let $X'' \subseteq X$ be the set of synchronized configurations whose program layer corresponds to the given parameters and whose sequence of symbols encoded in the data layer correspond to points in $y \in Y$ and $\varphi^{-1}(y)$. We then set $X' = \cup_{t=0}^{T-1} \pi^t X''$, where T is the length of one cycle.

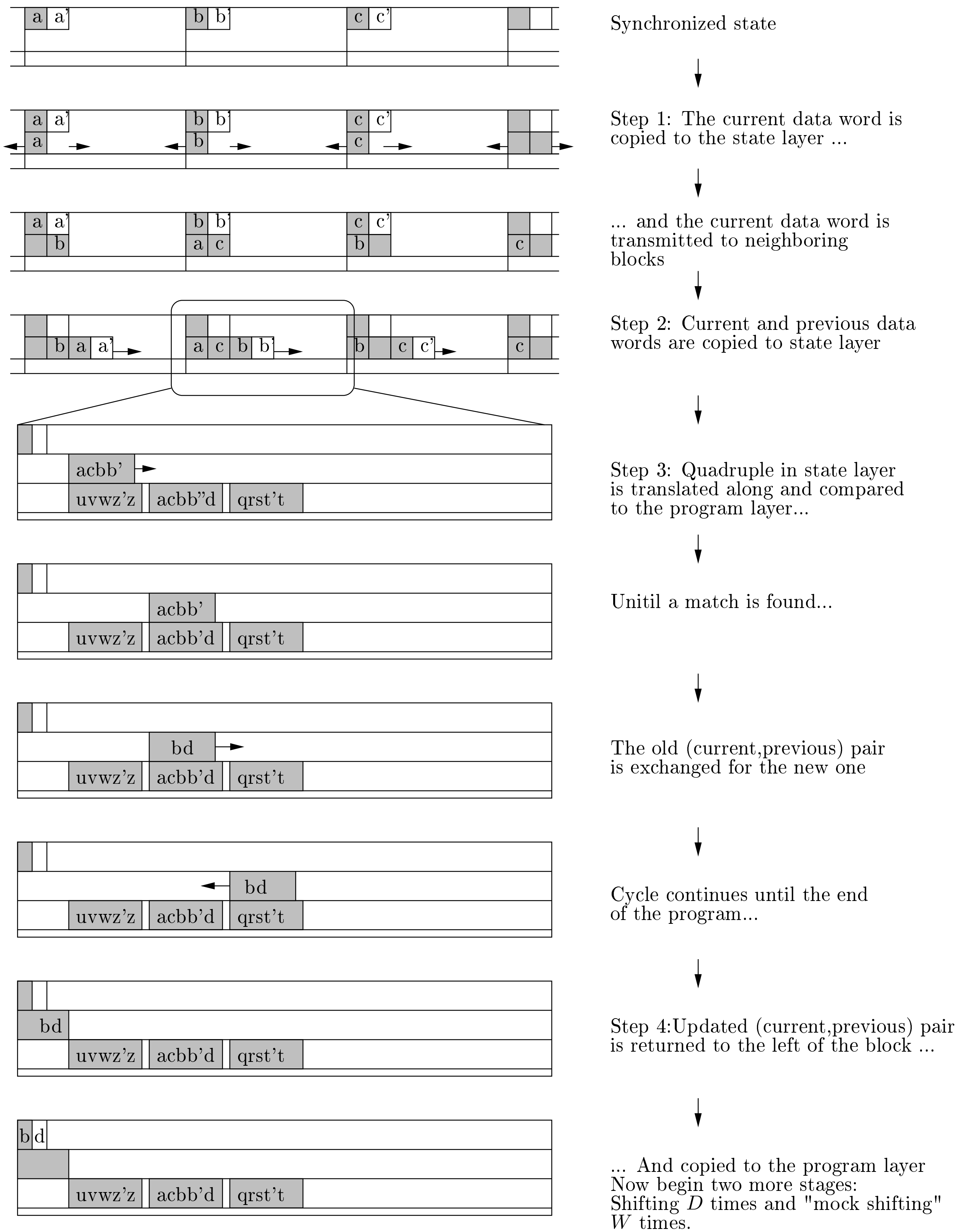

FIGURE 3.3. Part of a cycle, including simulation (but not the shifting).

It is clear from this description that π^T simulates φ, where T is the length of a cycle. The isomorphism (3.2) is given by

$$x \mapsto (y, b, t)$$

where b is the least non-negative integer such that $-b$ is the coordinate of the beginning of a block; t is the number of applications of π^{-1} to x required to bring x to a synchronized configuration; and $y \in Y$ is the sequence encoded by $\pi^{-t}(x)$, where y_0 is the symbol coded in the block to which 0 belongs in $\pi^{-t}(x)$. The length of a cycle is given by (3.1).

Note that the alphabet of X', and therefore of X, is independent of the parameters, and in particular of N. Thus we can simulate systems (Y, σ, φ) on arbitrarily large alphabets by subsystems $X' \subseteq X$. The upshot is that in order to do so the parameter B must be large enough that the blocks can encode the parameters as described above.

The implementation details of π are very similar to those involved in constructing a universal Turing machine, and are completely standard, with one exception: generally Turing machines are not reversible, yet we want π to act invertably on X' and for $\widehat{\pi} = \pi^{-1}$ to have range 1. With care this can be done. Notice that the only place where information is deleted in the scheme above is in the stage (3b), where the 5-tuple in the state layer matches the 4-tuple in the program layer. At this point certain information is erased from and added to the state layer, both are present in the program layer; this allows the process to be reversed locally (this is the reason φ^{-1} is encoded in the program layer; notice that it is not used explicitly in the definition of π). The other steps – namely, the transferring of bits from one place to another, etc. – can be done invertably with inverse having range 1. Thus we have achieved our stated goal. The bound (3.1) for T follows easily from the construction.

4. Implementation details

This section outlines the realization of the automorphism π described in the previous section. It is provided for completeness and readers may prefer to skip ahead to the next section where the main construction is undertaken. CA simulating other CA have been constructed a number of times in the literature, e.g. [1], and are similar to universal Turing machines. The only new ingredient here is a careful analysis of certain aspects of the time complexity of the simulation, and our emphasis on invertability. Although both aspects have been addressed in the literature, it is easier to indicate the construction than to explain how to modify existing ones to meet our specific needs.

As a complete implementation of π would be a very lengthy undertaking, we describe only the part of the implementation responsible for the first stage of the cycle, in which the current data words are transmitted between immediate neighboring blocks. In this stage we already encounter the main ideas needed to complete the rest, and we provide a few hints about the other stages.

The symbols of the state layer represent sets populated by *agents*. An agent is a finite state automaton. Each agent in each state-layer cell will, with each application of π, perform one or more of the following operations: (1) modify the symbols in the data layer, (2) update its internal state, (3) move one cell left or right. By design, not more than one agent per cell will attempt to modify the data layer, so no conflict will arise. The nature of the operation that an agent performs is determined by the other contents of its cell prior to the operation, including its own previous internal state, the states of the other agents in its cell, and the symbols in the data, program and block layers.

There are two types of agents:

- The *main agent*. There is one such agent per block, and it is the only agent capable of modifying the data layer.
- *Data agents*. Used to store and transport data. We allow several types of data agents, which play slightly different roles, but they overall behavior is the same.

We shall also add new symbols to the block layer. We call these *roadsigns*. Their role is to signal some event to the agents at that cell.

Note that, since the alphabet of X may not depend on the parameters N, Y, φ, B, D, W, we may introduce only a finite number of agents and new symbols (roadsigns); but we may arrange them as we wish. This allows us a great deal of flexibility in programming the agents and providing them external cues to modify their behavior.

The data agents role is to store and transport a single bit of data. Their internal state consists of a motion symbol ("left", "right", or "stationary"); and a data symbol ("0", "1" or "empty"). At each step, a data agent updates its state based on roadsigns in the block layer and instructions from the main agent, if it occupies the same cell as the data agent. Then the data agent takes one step left or right or stays in its current cell, according to its motion symbol.

The main agent acts as coordinator, issuing instructions to data agents and modifying the data layer. Its operation is more complex, and it is our goal to describe some of it in detail.

At the beginning of each cycle all agents are arranged as follows. The main agent is located in the leftmost cell of the block in an initial state that we call s_1. At each site where the data layer contains a data bit from the current word, there are two data agents, one of each type Left and one of type Right (not to be confused with their motion state!), with motion symbol "stationary" and empty data.

The main agent is initially in a state s_1. While in this state he moves one cell to the right at each time step, and at each cell has the following effect: the data bit from the data layer is copied to the data agents, and the data agents are "launched", i.e. their motion symbol is set as appropriate (the motion state of the Left agent is set to "left", and that of the other to "right"). The end of the current data word is indicated by a road-sign (special symbol in the block layer), and when reached it causes the main agent to enter internal state s_2.

At this point there are two new copies of the current data word, encoded in sequences of agents who are marching left and right at unit speed, forming what we shall call "caravans". We would like the main agent to meet the caravans arriving in its block and cause them to stop. Thus he should do "nothing" for a while, and arrive at a designated spot at a designated time to receive the first caravan. Since entering a state of inactivity and staying there is not an invertible operation we instead have the agent walk to the right while in state s_2 until it reaches a specially placed road-sign. At this point it enters state s_2', walks left until it reaches another designated road-sign, and enters state s_3. Since the transitions are controlled by encounters with specific roadsigns they are invertible, and by controlling the positions of the roadsigns we can determine the time and place at which the agent enters state s_3.

We assume that the main agent enters state s_3 just as the first data agent in caravan of data agents from the block to the right is arriving at the cell where it is to stop. In state s_3 the main agent walks to the right, and whenever it shares a left-moving data agent it sets its motion symbol to "stationary". Thus the main agent will cause the left-moving caravan to halt. A road-sign indicates to the main agent that it has reached the last data agent in the caravan (this position is completely determined by the parameters and the choice of roadsigns so far); when this road-sign is reached the main agent enters state s_4.

In state s_4 the main agent behaves similarly, moving left and stopping data agents arriving from the right. When the last data agent is halted, a road-sign forces the main agent into a new state s_5.

At this point we have completed our goal: the state layer of each block contains data words from the current block and its two neighbors. The time elapsed from the beginning of the cycle is $B + O_N(1)$; the B term is the time it actually takes each data bit to travel, and the $O_N(1)$ is an adjustment encompassing overhead and the fact that slightly less than B may have been traveled (we have some choice about where to halt the caravans).

With regard invertability, note that given the state s_i in which the main agent is found, the effect on data agents is invertible, and the transitions between s_i to s_{i+1} is controlled by roadsigns, and is invertible as well.

We conclude this outline with some further comments.

In our example we have not demonstrated the deletion of bits from the data or state layers. To make this invertible each deleted bit must be present in some other form in the same cell. For example, when transferring the current data word to the state layer the main agent will delete a bit from the state layer while at the same time recording it in a data agent; thus this operation can be reversed.

The next step in the cycle is to transfer the data-agents, representing three data-words, to the right, stopping opposite each corresponding triple in the program layer. This is similar to the transfer we just performed, except there is no main agent on the receiving end to halt the caravans. This is solved by positioning roadsigns that cause the data agents to reverse direction. These act as "reflecting walls", and allow the main agent who launched the caravan to also halt them (the order of bits in the data words is reversed; one can either work with this reversal, or perform the reflection twice).

Another point that needs some care is the comparison stage, at which the three data words represented by the data agents are compared to the corresponding words in the program layer. When comparing the input triple in the program to the corresponding triple in the state layer, one cannot simply traverse them both from left to right, say, and take note if they differ at some point, because this is not reversible (when going backwards, you are in a state of knowing that there is a differing pair of symbols until you reach the leftmost such pair. After that your state is that of not yet having seen a difference. But there is no way to know, when you reach a differing pair, if it is the leftmost such pair or not). To overcome this, one begins on the left, say, and puts down markings in the state layer: green until the first difference, if there is one, and red thereafter. When the end of the comparison is reached we note the current color, and then go back, right to left, and erase the color markings. This procedure is invertible with range 1.

Lastly, we discuss how to control the number of iterations of the "shifting" stage, which must occur D and W times. Perhaps the simplest is that, during the time that the data is being transferred, the main agent "counts down". This can be done in the simplest of ways, by coding D and W into the program layer as a sequence of D or W special symbols, respectively; transferring them to the state layer (as stationary data agents); and "crossing one out", i.e. resetting one of them, with each "shift" iteration.

5. Realizing unique non-expansive directions

Before continuing, let us make some observations about the construction above. The following is clear from the construction:

Lemma 5.1. *Suppose $X_1 \subseteq X$ is constructed from the parameters N, Y_1, φ, B, W, D. Suppose that $Y_2 \subseteq Y_1$ is σ- and φ-invariant, and let X_2 be constructed using parameters N, Y_2, φ, B, W, D. Then $X_2 \subseteq X_1$.*

Lemma. $\widehat{\pi}|_X = (\pi|_X)^{-1}$

Proof. Recall that $X = \overline{\cup X'}$, the union being over all systems constructed from permissible parameters. The lemma follows from the fact that for each such X' we have $\widehat{\pi}|_{X'} = (\pi|_{X'})^{-1}$, and in all cases π is given by the same block code. □

Next, we relate the prediction shapes of $\varphi|_Y$ and $\pi|_{X'}$.

Lemma 5.2. *Suppose $X' \subseteq X$ is constructed from the parameters Y, φ, N, B, W, D. Let Λ be a prediction shape for $\varphi|_Y$. Then $A(\Lambda)$ is a prediction shape for $\pi|_{X'}$, where $A : \mathbb{R}^2 \to \mathbb{R}^2$ is the linear map fixing $e_1 = (1,0)^T$ and mapping $e_2 = (0,1)^T$ to the vector $(D, T/B)$, or in matrix form,*

$$A = \begin{bmatrix} 1 & D \\ 0 & T/B \end{bmatrix}$$

Proof. This is immediate from the identification

$$(X', \sigma, \pi) \cong (Y \times \{0, \dots, B-1\} \times \{0, \dots, T-1\}, \sigma_B, \varphi_T)$$ □

We now undertake the main construction of this section, and proceed to analyze it. Let N be the number of symbols in the alphabet of X. Fix a sequence B_n, D_n, W_n of parameters for the construction above, with $W_k \geq 2$ (hence $T_k/B_k \geq 2$).

For a subshift $Y \subseteq X$ invariant under π, let $Z_n(Y)$ denote the system X' constructed above with parameters N, Y, π, B_n, D_n, W_n. Define

$$Z_n^{n+k} = Z_n(Z_{n+1}(\dots(Z_{n+k}(X))\dots)).$$

An induction using lemma 5.1 shows that $Z_n^{n+k+1} \subseteq Z_n^{n+k}$; thus

$$Z_n^\infty = \bigcap_{k=1}^{\infty} Z_n^{n+k}$$

is non-empty and σ- and π-invariant. Finally, set

$$Z = Z_1^\infty.$$

It is easy to see that we have the relation

$$Z = Z_1^\infty = Z_1(Z_2^\infty) = Z_1(Z_2(Z_3^\infty)) = \dots \tag{5.1}$$

etc.

The following description of the configurations of Z may be helpful. Each configuration $z = z_1$ encodes a point $z_2 \in Z_2^\infty$ using blocks of size B_1; in turn, z_2 encodes a point $z_3 \in Z_3^\infty$ using blocks of size B_2; and so on, z_i encodes a point $z_{i+1} \in Z_{i+1}^\infty$ using blocks of size b_i. Note that z_2 may be at any point in its cycle independently of where z_1 is in its cycle; and similarly z_3. Also, the alignment of blocks in z_i is independent of the alignment of large blocks in z_{i+1}.

We will now show that there is a (necessarily unique) line ℓ through the origin so that

$$\Lambda = \ell^1 = \{u \in \mathbb{R}^2 \,:\, d(u,\ell) < 1\}$$

is a prediction shape of Z, and calculate the slope of ℓ. It suffices to write Λ as an increasing union of prediction shapes for $\pi|Z$.

Let Δ denote the unit ball in $\mathbb{R}^2$ with the norm $\|\cdot\|_1$, which is a prediction shape for π and any π-invariant subshift of X. Let $A_n : \mathbb{R}^2 \to \mathbb{R}^2$ denote the map associated as in lemma 5.2 to $Z_n(\cdot)$. Since $Z_n^\infty = Z_n(Z_{n+1}^\infty)$ and Δ is a prediction shape for Z_{n+1}^∞, it follows from lemma 5.2 that $A_n(\Delta)$ is a prediction shape for Z_n^∞. Therefore since $Z_{n-1}^\infty = Z_{n-1}(Z_n^\infty)$ the same lemma gives that $A_{n-1}(A_n\Delta)$ is a prediction shape for Z_{n-1}^∞, and iterating we have that

$$\Delta_n := A_1 A_2 \dots A_n(\Delta)$$

is a prediction shape for $Z = Z_1^\infty$.

Notice that each the shape Δ_n is a quadrilateral having two vertices, $(-1,0)$ and $(1,0)$, on the x-axis, one vertex above the x-axis, and one below it. We shall now analyze the asymptotic behavior of these vertices. Let A be the matrix in lemma 5.2, i.e. it is of the form

$$A = \begin{bmatrix} 1 & D \\ 0 & T/B \end{bmatrix}$$

for integer parameters D, T, B and T/B. Let $(x,y)^T \in \mathbb{R}^2$ with $y > 0$ and $(x',y')^T = A(x,y)$. Then

$$\frac{x'}{y'} = \frac{DB}{T} + \frac{B}{T} \cdot \frac{x}{y}$$

Now fix n and let $(x_n, y_n)^T$ denote the vertex of the quadrilateral Δ_n that lies in the upper half plane. Then

$$(x_n, y_n)^T = A_1 \dots A_n(0,1)$$

so x_n/y_n is given by

$$\frac{x_n}{y_n} = \frac{D_1 B_1}{T_1} + \frac{B_1}{T_1}\left(\frac{D_2 B_2}{T_2} + \frac{B_2}{T_2}\left(\dots\left(\frac{D_n B_n}{T_n} + \frac{B_n}{T_n}\cdot\frac{0}{1}\right)\dots\right)\right)$$

or, using (3.1) and writing

$$\alpha_k = \frac{D_k B_k}{T_k} = (1+\varepsilon_k)\frac{D_k}{|D_k|+W_k} \qquad \text{and} \qquad \beta_k = \frac{B_k}{T_k} = \frac{1}{(1+\varepsilon_k)(|D_k|+W_k)}$$

where $\varepsilon_k = O_{N,\varphi}(\frac{1}{B_k})$, we have

$$\frac{x_n}{y_n} = \alpha_1 + \beta_1(\alpha_2 + \beta_2(\ldots(\alpha_n + \beta_n \cdot 0))\ldots)$$

Since $T_k/B_k \geq 2$ for all k by our choice of parameters, $\beta_k < 1/2$, so this sequence converges to some $\lambda \in (-1,1)$. For the same reason we have $y_n \geq 2^n$ and thus for large enough n we have $x_n \to \infty$. Since the slopes of the two sides of Δ_n which lie in the upper half plane are

$$\frac{y_n}{x_n - 1} \text{ and } \frac{y_n}{x_n + 1}$$

it follows that that these slopes converge to the common value $1/\lambda$.

A similar calculation shows that as $n \to \infty$ the remaining vertex of Δ_n grows unboundedly in norm and that the slope of the remaining two sides converges to $1/\lambda$ as well.

Finally, it follows that there is an increasing subsequence of the Δ_k's whose union is (necessarily) the set $\Lambda = \ell^1$, where ℓ is the line with slope $1/\lambda$.

In summary, we have:

Theorem 5.3. $(Z, \langle\sigma,\pi\rangle)$ *has a unique non-expansive direction whose slope is* $1/\lambda$, *where*

$$\lambda = \alpha_1 + \beta_1(\alpha_2 + \beta_2(\alpha_3 + \ldots(\alpha_n + \beta_n(\ldots)))\ldots)$$

and $\alpha_k, \beta_k, \varepsilon_k$ *are as given above.*

Proof. Theorem 2.2 implies that any line with slope different from $1/\lambda$ is expansive (and the horizontal direction is as well, since σ acts expansively by definition). That the line with slope $1/\lambda$ is a non-expansive direction then follows from the fact that any infinite system has non-expansive directions, and the following claim. □

Claim 5.4. For any choice of parameters, the system Z is infinite.

Proof. From the representation (3.2), we see that every choice of $z_2 \in Z_2^\infty$ is represented, modulo σ-shifts, by T_1B_1 points $z_1 = z_1(z_2) \in Z$. Thus $|Z| = |Z_1^\infty| \geq T_1B_1|Z_2^\infty|$. Similarly, $|Z_2^\infty| \geq T_2B_2|Z_3^\infty|$, so $|Z| \geq T_1B_1T_2B_2|Z_3^\infty|$; and so one. Since $T_nB_n \geq 2$ for each n, the conclusion follows. □

We have shown that our construction yields systems with a unique non-expansive direction of the form $1/\lambda$. It remains to show that any direction can be attained. The following an elementary exercise in representing reals:

Lemma 5.5. *For any real number* $|\theta| \leq 1$ *occurs as the number* λ *for some choice of the parameters* B_k, W_k, D_k, *and we may choose* $W_k \geq 2$.

Proof. We assume for convenience that $\theta \geq 0$; the case $\theta < 0$ follows similarly, the only difference being that all the D's are then negative and the endpoints of segments must appear in the reverse order.

For some pair of integers $W \geq 1$ and $D \geq 0$, the number θ lies between $\frac{D}{|D|+W+1}$ and $\frac{D+1}{|D|+W+1}$. Taking $W_1 = W, D_1 = D$ and taking B_1 to be sufficiently large, we can make the error ε_k arbitrarily small and obtain

$$\theta \in (\alpha_1, \alpha_1 + \beta_1)$$

One proceeds inductively to choose B_2, W_2, D_2 so that

$$\frac{\theta - \alpha_1}{\beta_1} \in (\alpha_2, \alpha_2 + \beta_2)$$

implying that

$$\theta \in (\alpha_1 + \beta_1\alpha_2, \alpha_1 + \beta_1(\alpha_2 + \beta_2))$$

and so on (note that the closure of each interval is in the interior of the previous one). □

Theorem 5.3 and lemma 5.5 show that any line with slope θ, $|\theta| > 1$, occurs as the unique non-expansive direction of some action. All other directions can be attained from this result by re-parametrizing the acting group. This completes the proof of the main part of theorem 1.

It remains only to show that in the case of a rational direction the action in that direction is non-trivial. This follows from the following claim, which may provide some further insight into the structure of Z, and gives an alternative proof that Z is infinite:

Proposition 5.6. *(Z, π) is a finite extension of a group extension of an adding machine by an adding machine. Furthermore, π intertwines a map on the base of the group extension whose orbits are infinite, and the shift on Z intertwines a translation on the fiber of the group extension.*

Proof. For $z \in Z$ define the configurations $z_i \in Z_i$ by $z_1 = z$ and the condition that z_i encodes the configuration z_{i+1}. Define $\tau_n : Z \to \{0, \ldots, T_n - 1\}$ with $\tau_n(Z) = t$ if z_n if is at stage t of its cycle. Let $\tau = \times_{n=1}^{\infty} \tau_n$ be the factor map $z \mapsto (\tau_1(z), \tau_2(z), \ldots) \in \prod_{n=1}^{\infty} \{0, \ldots, T_{n-1}\}$; the image $\tau(Z)$ is an adding machine. Next, define $\beta_n : Z \to \{0, \ldots, B_{n+1} - 1\}$ to be the position of the origin (or the $(n-1)$-level block to which the origin belongs) inside the corresponding level-n block of size B_n in z_n; recall that each block in z_{n-1} encodes one symbol of z_n, and this symbol is one symbol in a block of length B_n. We see that $b_1(z)$ is invariant under π and for $n > 1$,

$$\beta_n(\pi^{T_{n-1}} z_n) = \beta_n(z_n) - D_n \bmod B_n$$

which translates to

$$\beta_n(\pi^{T_1\cdot\ldots\cdot T_{n-1}}z) = \beta_n(z) - D_n \bmod B_n$$

because $T_1 \cdot T_2 \cdot \ldots \cdot T_{n-1}$ applications of π to z is the same as T_{n-1} applications of π to z_{n-1} (The i-th level of simulation introduces a "slowdown" by a factor of T_i). Note also that β_n is constant on time stretches of length $T_1 \cdot T_2 \cdot \ldots \cdot T_{n-1}$; the timing of the actual increment of $\beta_n(z)$ depends on $\tau_1(z), \ldots, \tau_{n-1}(z)$, because of the convention which interprets the state of a block according to its state at the beginning of it's current cycle. It follows that $\varphi = \tau \times (\times_{n=1}^{\infty}\beta_n)$ maps Z into the group extension $\prod_{n=1}^{\infty}\{0,\ldots,T_n-1\} \times \prod_{n=1}^{\infty}\{0,\ldots,B_{n+1}-1\}$. The shift action on Z can be seen to intertwine with a rotation on the fiber.

Finally, $\varphi(z)$ contains enough information to reconstruct z except at most one bit. This ambiguity arises in the same way that a Toeplitz sequence decomposes into unions of periodic sequences and possibly a finite number of additional points. Thus, the fibers of φ are finite. □

This implies that $\pi \circ \sigma^n$ intertwines with a translation on the base which has infinite orbits. In particular, no power of it is the identity on Z.

6. Lyapunov exponents

Given a subshift Y and an endomorphism $\varphi : Y \to Y$, Shereshevsky [8] defined the Lyapunov exponents λ^+, λ^- as follows. For $y \in Y$ define

$$I_t^+(y) = \min\left\{n \,:\, \forall z \in Y\ \forall 0 \le s \le t \quad \left(z|_{[-n,\infty)} = y|_{[-n,\infty)} \quad \Longrightarrow \quad (\varphi^s z)|_{[0,\infty)} = (\varphi^s y)|_{[0,\infty)}\right)\right\}$$

and similarly

$$I_t^-(y) = \min\left\{n \,:\, \forall z \in Y\ \forall 0 \le s \le t \quad \left(z|_{(-\infty,n]} = y|_{(-\infty,n]} \quad \Longrightarrow \quad (\varphi^t z)|_{(-\infty,0]} = (\varphi^t y)|_{(-\infty,0]}\right)\right\}$$

Set

$$\Lambda_t^{\pm} = \max_{y\in Y}\max_{i\in\mathbb{Z}} I_t^{\pm}(\sigma^i y)$$

The Lyapunov exponents are then defined by

$$\lambda^{\pm} = \liminf_{t\to\infty}\frac{\Lambda_t^{\pm}}{t}$$

(Shereshevsky's original definition of $\lambda^{\pm}$ differs from the above but is equivalent by [9]).

We omit the proof of the following, which is an immediate consequence of the definitions:

Proposition 6.1. *$\varphi : Y \to Y$ be an automorphism of an infinite subshift Y and let Λ be a prediction shape for $\varphi|_Y$. Let θ^+, θ^- denote the (possibly infinite) slopes* *of the right- and left-tangent rays to* $\partial\Lambda$ *at* $(-1, 0)$ *and* $(1, 0)$*, respectively. Then* $\lambda^+ \leq 1/\theta^+$ *and* $\lambda^- \leq -1/\theta^-$*.*

Corollary 6.2. *If the strip* $\Lambda = \{(x, y) : |x| < 1\}$ *is a prediction shape for* $\varphi|_Y$*, then* $\lambda^+ = \lambda^- = 0$*.*

Tisseur [9] and later Tisseur and Bressaud [4] studied the relation between Lyapunov exponents, particularly the case of zero Lyapunov exponent, and the existence of equicontinuity points for the action of φ. Let us recall some definitions. For an endomorphism φ acting on a subshift $Y \subseteq \Sigma^{\mathbb{Z}}$, we say that a finite word $a \in \Sigma^n$ is a *blocking word* if, for any pair $y, z \in Y$ with $y|_{[1,n]} = z|_{[1,n]} = a$ and $y|_{[1,\infty)} = z|_{[1,\infty)}$, we also have $(\varphi^t y)|_{[1,\infty)} = (\varphi^t z)|_{[1,\infty)}$ for all t; and also for any y, z satisfying $y|_{[-n,-1]} = z|_{[-n,-1]} = a$ and $y|_{(-\infty,-1]} = z|_{(-\infty,-1]}$ we have $(\varphi^t y)|_{(-\infty,-1]} = (\varphi^t z)|_{(-\infty,-1]}$. The condition that φ have equicontinuity points is equivalent to φ having a blocking word. Also, not having equicontinuity points is equivalent to φ acting on Y with sensitive dependence on initial conditions.

Returning to the matter at hand, Bressaud and Tisseur asked if a cellular automaton φ acting sensitively (i.e. without equicontinuity points) one must have $\liminf_{t\to\infty} I_t^+(y) > 0$ or $\liminf_{t\to\infty} I_t^-(y) > 0$ for some $y \in Y$ ([4, Conjecture 3]). Our construction provides a counterexample if one allows action on a subshift rather than the full shift, as we describe next. We continue to use the notation introduced in the previous section during the construction of Z.

Proposition 6.3. *The action of* π *on* Z *does not have equicontinuity points.*

Proof. First, suppose that $z \in Z$ is a point in synchronized state. Suppose that z' differs from z only in the data bits recorded in the block to which the index 0 belongs. Then we claim that there is some t such that $\pi^t z, \pi^t z'$ differ at coordinate 0. Indeed, during the first stage of the cycle the data bits in the current block are transmitted to the block on the other side of the origin. For this to happen the symbols at index 0 must differ at some time between z and z'.

The same argument works if z, z' differ only in data bits of some higher level block. More precisely, define the higher level configurations $z_i \in Z_i$ encoded by $z \in Z$ by $z_1 = z$ and such that z_i encodes z_{i+1}. Define z'_i similarly with $z'_1 = z'$. Suppose that for some i, the points z_i, z'_i are in a synchronized state, their blocks have the same alignment, and they differ only in the data bits in the block containing the index 0. Then a similar argument shows that $\pi^t z, \pi^t z'$ must differ at index 0 for some $t > 0$. One can do this by induction from i down to 1.

Finally, let z be given. Fix k; it is then easy to see that we can find $t_0 > 0$, an integer i and a point $z' \in Z$ such that (a) z, z' agree on $[-k, k]$, and (b) the level-i configurations $(\pi^{t_0} z)_i$ and $(\pi^{t_0} z')_i$ are synchronized, their blocks have the same alignment, and they differ only in the data bits of the block containing index 0. Then, as we saw above, $\pi^t z, \pi^t z'$ differ at coordinate 0 for some $t > 0$. This shows that there are points arbitrarily close to z whose forward orbits draw away from the orbit of z, so z is not an equicontinuity point. □

This last proposition may be surprising since the dynamics of (Z, π) appear at first glance to be almost periodic. Indeed, the orbit closure of every point in Z is an odometer, but Z itself is not transitive. Rather, it is an uncountable union of odometers, glued together in such a way that takes nearby points in different minimal sets far apart after sufficiently many iterations of π. An analogous situation arises when acts on the unit disc by $re^{i\theta} \mapsto re^{i(\theta+r)}$. Then each orbit closure is conjugated to a minimal subset of a rotation on S^1 (either a periodic orbit, or all of S^1), but the action is not equicontinuous because for arbitrarily small $\varepsilon > 0$ the points $e^{i\theta}$, $(1-\varepsilon)e^{i\theta}$ attain distance 1 after some number of applications of the map.

7. An alternative construction in the rational case

We present here a construction that arose in discussions with Doug Lind and provides a simpler example of a system whose unique non-expansive direction is the vertical axis, but no power of the action in that direction is the identity (though in other ways the action is dynamically rather trivial). This example is significantly simpler than the one above and may be adapted to give examples in other rational directions, but we have been unable to get any irrational direction with this method. It is striking to us that the irrational case is so much more difficult than the rational one, and it would be interesting if a simpler construction for that case were found.

As before, we construct a subshift $X \subseteq \Sigma^{\mathbb{Z}}$ and an automorphism $\pi : X \to X$ such that the vertical strip of width 2 around the y-axis is a prediction shape for π.

Fix a parameter $p \in \mathbb{N}$. The alphabet Σ consists of the symbols

$$\begin{array}{lll} - & & \text{(blank)} \\ \rightarrow, \leftarrow & & \text{(arrows)} \\ \overset{k}{[}, \overset{k}{]} & \text{for } 0 \le k \le p & \text{(brackets, with counter } k\text{)} \\ \underset{*}{\overset{k}{[}}, \underset{*}{\overset{k}{]}} & \text{for } 0 \le k \le p-1 & \text{(marked brackets, with counter } k\text{)} \end{array}$$

Each configuration in X will have at most one arrow symbol in it; the rest will be blanks and brackets. Adjacent brackets will not be allowed, instead between any pair of brackets there will always be at least one blank or arrow symbol.

We shall later describe the configurations of X in more detail, but first we define the automorphism π by giving the relevant transitions, from which a range-2 block code may be derived. The transitions are

$$\rightarrow - \quad \text{becomes} \quad - \rightarrow \tag{7.1}$$

$$\rightarrow \overset{p}{[} - \quad \text{becomes} \quad - \underset{*}{\overset{p-1}{[}} \rightarrow \tag{7.2}$$

$$\rightarrow \overset{k}{]} - \quad \text{becomes} \quad \leftarrow \overset{k-1}{]} - \quad \text{if } 0 < k \leq p \tag{7.3}$$

$$\rightarrow \overset{0}{]} - \quad \text{becomes} \quad - \overset{p}{]} \rightarrow \tag{7.4}$$

$$- \underset{*}{\overset{k}{[}} \leftarrow \quad \text{becomse} \quad - \underset{*}{\overset{k-1}{[}} \rightarrow \quad \text{if } 0 < k \leq p \tag{7.5}$$

$$- \underset{*}{\overset{0}{[}} \leftarrow \quad \text{becomes} \quad - \overset{p}{[} \rightarrow \tag{7.6}$$

together with the symmetric rules obtained by reversing left and right, e.g. $-\overset{0}{[} \leftarrow$ becomes $\leftarrow \overset{p}{[} -$ (reversal of (7.4)).

To interpret this, one may imagine that the arrow represents an agent walking in a landscape of brackets, proceeding in the direction the arrow points to (rule (7.1)). The behavior of the agent when it encounters a bracket depends on the orientation and the data on the bracket.

- When the agent approaches a bracket from the "inside",
  - If the bracket has positive counter, the agent decrements the counter by 1 and turns around (rule (7.3),(7.5)).
  - If the counter is 0 then it is reset to p; next, if the bracket was marked the agent removes the mark and turns around, but if it was unmarked the agent passes through (rule (7.4),(7.6)).
- When the agent approaches a bracket from the "outside" it marks the bracket, decrements the counter, and passes through (rule 7.2). We shall arrange that such an encounter only occurs when the counter is p and the bracket is unmarked.

For example, starting from the pattern $\rightarrow \overset{p}{[} - \overset{p}{]} -$ the agent will "enter" the region between the brackets, reverse its direction $2p$ times, and emerge from the right side. Note that upon its exit it leaves behind the configuration as he found it, i.e. the final pattern is $- \overset{p}{[} - \overset{p}{]} \rightarrow$.

We next describe the allowable arrangement of brackets in X. We first define special sequences of the brackets $\overset{p}{[}, \overset{p}{]}$ which are arranged in a hierarchical manner. Begin by choosing a periodic subset $I_1 \subseteq \mathbb{Z}$ of period 2 (there are two ways to do this), and set the symbols $y_i, i \in I_1$ to be alternately [and] (this can again be done in two ways). Half of the symbols in $\mathbb{Z}\setminus I_1$ are now trapped between matching brackets; let I_2' denote the half which is not, which is a coset of $4\mathbb{Z}$. Next, choose a subset $I_2 \subseteq I_2'$ of relative period 2 (a coset of $8\mathbb{Z}$; there are two choices) and define $y_i, i \in I_2$ to be alternately [and] (again two choices). Let $I_3' \subseteq \mathbb{Z}\setminus(I_1 \cup I_2)$ be those indices which have not yet been determined, and which are not trapped between matching brackets, and choose $I_3 \subseteq I_3'$ a subset of relative period 2 (a coset of $32\mathbb{Z}$). Proceed in this manner to define y_i for $i \in I_3$ and I_4', I_4, etc. After carrying this out for all n every $i \in \mathbb{Z}$, with possibly one exception, is trapped between some pair of brackets; the remaining point, if it exists, may be left blank or given the symbol $\overset{p}{[}$, $\overset{p}{]}$. Three steps in the construction of such a y appears below (big brackets indicate the addition at each stage. For readability's sake we have omitted the n).

$$-[-]-[-]-[-]-[-]-[-]-[-]-[-]-[-]-[-]-[-]-[-]-[-]-[-]-$$

$$\Bigg]\,[-]-[-]\,\Bigg[\,[-]-[-]\,\Bigg]\,[-]-[-]\,\Bigg[\,[-]-[-]\,\Bigg]\,[-]-[-]\,\Bigg[\,[-]-[-]\,\Bigg]\,[-]-$$

$$]\,[-]\,\Bigg[\,[-]\,[\,[-]-[-]\,]\,[-]-[-]\,[\,[-]-[-]\,]\,[-]\,\Bigg]\,[-]\,[\,[-]-[-]\,]\,[-]-$$

We define a pre-block to be a subword of an hierarchical arrangement as above, which consists of a matched pair of brackets and the region between them. The pattern $\overset{p}{[}-\overset{p}{]}$ is a pre-block, and we call it the level-0 pre-block; next is $\overset{pp}{[[}-\overset{p}{]}-\overset{p}{[}-\overset{pp}{]]}$, a level-1 pre-block; in general, a pre-block containing level-n pre-blocks but no level-$(n+1)$ pre-block is a level-$(n+1)$ pre-block.

Next, define a level-n block to be the word obtained from a level-n pre-block by inserting a blank in between every pair of symbols.

We now define the admissible words in X. Let a be a block, and consider the patterns $\rightarrow a-$ and $-a \leftarrow$, which we extend with blanks in both directions (but we suppress these blanks notationally). It is easy to verify that after finitely many iterations of π we get the patterns the $-a \rightarrow$ and $\leftarrow a-$, respectively. Let $L(a)$ denote the set of intermediate patterns obtained in this way. We define X to be the subshift such that every finite word in X appears as a subword of some $b \in L(a)$, for some block a.

It is not hard to check that for each $x \in X$ there is a coset of $2\mathbb{Z}$ on which there appears a hierarchical configuration of brackets in the sense above, with brackets now carrying counters and markings. On the complementary coset there appear only blanks and possibly an arrow. One can also verify that if $x \in X$ contains a block that does not contain an arrow, then that block consists of unmarked brackets with counters equal to p.

The point of the construction is the following. The changes that occur in a configuration under π occur only at the site of an arrow or adjacent to an arrow, so in order to understand the propagation of perturbations to a configuration under

π we must understand is the rate at which the arrow moves. For this, note that in order to "pass through" the block

$$a_1 = \overset{p}{[} - - - \overset{p}{]}$$

requires $6n$ steps. Now consider the block

$$a_2 = \overset{p}{[} - \overset{p}{[} - - - \overset{p}{]} - - - \overset{p}{[} - - - \overset{p}{]} - \overset{p}{]}$$

To pass through this requires the arrow to go back and forth $2p$ times between the external brackets; each time it must cross the inner two brackets twice, taking $6n$ steps each time. Thus to cross a_2 requires $2p \cdot (5 + 2 \cdot 6p)$.

Continuing in this way, one may show that the time to cross the level-k block a_k is $(c_p)^k$, where $c_p \to \infty$ with p. On the other hand, the width of a_k is d^k for a constant d independent of p, and the blocks a_k appear periodically with period d^k in any configuration of X. It follows that if we choose p large enough ($p = 10$suffices) then for the agent to travel a distance of Nd^k will require time on the order of $N(c_p)^k$, i.e. over large scales the rate of travel is logarithmic (we remark that this is the slowest possible rate; if the rate were sub-logarithmic we would have, counting configurations, that the action of π were periodic).

In particular, if we know the configuration $x|_{[-N,N]}$ for $x \in X$ we can predict $x|_{[-N+\log N, N-\log N]}$ up to time $O(N)$. It follows that the vertical strip of width 2 is a prediction shape for X, as desired. We omit the details.

Finally, since the arrow does travel arbitrarily far in some configurations (in fact, in any configuration containing the arrow), it follows that the action of π is not periodic. We remark, however, that the dynamics of π are in other ways rather trivial, e.g. all invariant measures are concentrated on fixed points, and there are uncountable many of these (the configurations without an arrow).

References

[1] Jürgen Albert and Karel Culik, II. A simple universal cellular automaton and its one-way and totalistic version. *Complex Systems*, 1(1):1–16, 1987.

[2] Mike Boyle. Open problems in symbolic dynamics. *to appear in Contemporary Mathematics*, 2008. http://www.math.umd.edu/ mmb/papers/openfinalsub2nov2008.pdf.

[3] Mike Boyle and Douglas Lind. Expansive subdynamics. *Trans. Amer. Math. Soc.*, 349(1):55–102, 1997.

[4] Xavier Bressaud and Pierre Tisseur. On a zero speed sensitive cellular automaton. *Nonlinearity*, 20(1):1–19, 2007.

[5] Peter Gács. Reliable cellular automata with self-organization. *J. Statist. Phys.*, 103(1-2):45–267, 2001.

[6] K. M. Madden. A single nonexpansive, nonperiodic rational direction. *Complex Systems*, 12(2):253–260, 2000.

[7] K Morita and M Harao. Computation universality of one-dimensional reversible (injective) cellular automata. *IEICE Trans. Inf. & Syst.*, 72:758–762, 1989.

[8] M. A. Shereshevsky. Lyapunov exponents for one-dimensional cellular automata. *J. Nonlinear Sci.*, 2(1):1–8, 1992.

[9] P. Tisseur. Cellular automata and Lyapunov exponents. *Nonlinearity*, 13(5):1547–1560, 2000.

[10] Peter Walters. *An introduction to ergodic theory*, volume 79 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1982.

Department of Mathematics, Princeton University, Princeton, NJ 08544.
E-mail address: hochman@math.princeton.edu