Chaos

Traveling patterns in cellular automata

Cite as: Chaos **6**, 493 (1996); https://doi.org/10.1063/1.166190
Submitted: 05 December 1994 . Accepted: 16 May 1996 . Published Online: 04 June 1998

Jesús Urías, G. Salazar-Anaya, Edgardo Ugalde, and Agustín Enciso

Traveling patterns in cellular automata

Jesús Urías
Instituto de Investigación en Comunicación Optica, Universidad Autónoma de San Luis Potosí, 78000, San Luis Potosí, SLP, México

G. Salazar-Anaya
Department of Mathematics and Statistics, Carleton University, Ottawa, Ontario, Canada

Edgardo Ugalde
CPT, Luminy, Case 907, F-13288 Marseille, Cedex 9, France

Agustín Enciso
Escuela de Física, Universidad Autónoma de Zacatecas, 98000 Zacatecas, Zac., México

(Received 5 December 1994; accepted for publication 16 May 1996)

A method to identify the invariant subsets of bi-infinite configurations of cellular automata that propagate rigidly with a constant velocity ν is described. Causal traveling configurations, propagating at speeds not greater than the automaton range, $|\nu|\leq r$, are considered. The sets of traveling configurations are presented by finite automata and its topological entropy is calculated. When the invariant subset of traveling configurations has nonzero topological entropy, the dynamics is dominated by the interaction of domains, composed of traveling patterns of finite size. The sets of traveling patterns and domains are presented by finite automata. End-resolving CA are shown to always have sets of traveling configurations that are spatially periodic with zero entropy, except possibly for traveling configurations at top speed. The elementary CA are examined exhaustively along these lines. © *1996 American Institute of Physics.* [S1054-1500(96)00603-9]

In the two-dimensional space–time plots that represent the evolution of a one-space dimensional cellular automaton (CA), the human eye readily discerns considerable structure and pattern. Often the pattern contains easily visible "particle-like" objects, localized regions of finite spatial extent in which a particular configuration (or set of configurations) persists and propagates through the space–time plot. These objects are reminiscent of the spatially extended solitary waves in nonlinear continuum equations. Combining concepts from the theory of computation and formal languages with those on dynamical systems, we identify and study a particular class of persistent, spatially extended objects which travel rigidly at constant speed: we call these tropons. Understanding the dynamics of these tropons is crucial to understanding the full dynamics of the CA in which they occur and hence is important to potential applications of CA, including filtering.

I. INTRODUCTION

From computer phenomenology it is known that the dynamics of certain cellular automata is dominated by patterns that behave as interacting particles.[1–4] Sometimes the particle-like patterns behave as kinks[1] with diffusive motion. In some other cases they are just small defects on a fixed background, traveling at a constant speed.[2,3] Objects more complicated than particle-like defects are extended domains,[5–7] appearing with almost any length. Intuitively, domains are blocks of cells presenting configurations with a persistent, e.g., particle-like, behavior under the CA dynamics. Generally, domains in an automaton configuration interact through their, not necessarily short, interfaces. The finite length configurations conforming the domains are termed patterns. The central role of domains, in the sense of dynamical system theory, is studied in Ref. 6 in the framework of the theory of computation. Techniques for the identification of dynamically relevant domains are applied to a particular automaton in Ref. 7. Here, we characterize patterns and domains that travel rigidly at a constant speed in automaton configurations.

To be relevant for the automaton dynamics, domains appearing in a CA configuration must be persistent in time and in space. Two necessary conditions for this to happen are the following. First, time persistence means a long lifetime for domains. So, the set of patterns that build up the body of domains must be in the language associated to some *invariant subset* of configuration space.[8] Second, for space persistence, the invariant subset must have nonzero topological entropy. Indeed, any invariant subset is included in the limit set of the automaton, if it has a nonvanishing topological entropy we are sure that the invariant subset under consideration samples an important part of the limit set. These two conditions for dynamical relevance of domains were introduced in Ref. 6 under the names of temporal invariance and spatial homogeneity, respectively. When they are fulfilled, domains in the automaton configuration will be persistent in time and in space.

To identify the invariant subsets of traveling configurations that satisfy the criteria for dynamical relevance we look at the defining presentation of the automaton. A cellular automaton is a map $f:X\to X$, defined over configuration space $X\equiv Z_k^Z$ by a block function $f_L:Z_k^L\to Z_k$ that maps words of length L to symbols in the alphabet $Z_k=\{0,1,\ldots,k-1\}$. The length of blocks for f_L is the odd integer $L=2r+1$, and r is the range of the automaton.[9] If we are able to identify an invariant subset from the defining presentation of the automaton, the dynamics within the domains would be understood and the domains could be filtered out of the game, leaving only their interfaces. The unknown part of the dynamics then resides just in the play of the interfaces. So, we proceed in two steps. First, we identify invariant subsets of positive topological entropy; the patterns that conform the domains are in the associated language. Second, we construct the filter that eliminates from the game all the patterns belonging to the language.[10] This we do for the subset of configuration space X that is invariant and collects all the automaton configurations that slide rigidly under the action of the cellular map f. This invariant subset is the set of traveling configurations of f. The language that is associated[8] to the set of traveling configurations contains all the patterns that, when appearing in any CA configuration, will travel with constant velocity, during all of its lifetime. We call these patterns tropons. To fix ideas, we must mention that soliton-like patterns, discussed in Ref. 11, are the special type of tropons that do not interact, i.e., soliton-like patterns are tropons that travel without colliding. Here, we do not distinguish tropons by their interactions. Finally, domains are tropons with the particularity of having both ends matched to nontropons. Hence, the domain language is a subset of the language of tropons. To illustrate the terms introduced so far, consider the left shift σ that corresponds to the elementary CA with standard number 170. For the left shift σ the set of traveling configurations is the whole of configuration space, the language of tropons is the set of all possible strings on Z_2, and the language of domains is the empty set.

For an automaton f, the invariant subset of traveling configurations with velocity ν is the largest subset, S_ν, of configuration space X, such that the restriction of f to S_ν behaves as the νth power of the left shift, i.e.,

$$f|S_\nu(x)=\sigma^\nu x. \tag{1}$$

By definition, a traveling configuration has positive velocity when it advances to the left.

A given automaton f of range r can show up traveling configurations with arbitrarily large velocities. However, information does not propagate faster than the range r of the automaton and the appearance of configurations that do travel faster than r is rather accidental. We will limit ourselves to consider just *causal* traveling configurations, with speeds not greater than the automaton's range r. This is the top speed for causal traveling configurations. The set S_ν is invariant and the union $\cup_{|\nu|\leqslant r}S_\nu$ is the set of all the traveling configurations of f. For the elementary CA the only speed values allowed are $|\nu|\leqslant 1$.

In Sec. II, we briefly review the presentation of cellular automata by transducers in order to derive in Sec. III a general algorithm to find the invariant set of traveling configurations S_ν and its associated regular languages of tropons, $\mathscr{L}(S_\nu)$, and domains, $\mathscr{D}(S_\nu)\subset\mathscr{L}(S_\nu)$. The topological entropy of S_ν is computed.[12] When it is nonzero, the associated language $\mathscr{L}(S_\nu)$ will provide a dynamically relevant set of tropons. The algorithm presented in Sec. III works fine for any cellular automata and provides a transducer that presents the restriction $f|_{S_\nu}$ of the automaton f to the set S_ν.

Once we have found the invariant set of traveling configurations, S_ν, tropons are filtered by a cellular automaton $\varphi{:}X\rightarrow X$, defined by the block function $\varphi_L:Z_k^L\rightarrow Z_k$ such that $\phi_L(e)=0$ if e is a tropon and it is the identity $\varphi_L(e)=\mathrm{id}_L(e)$ otherwise. When an automaton configuration x is seen through the filter as the configuration $\varphi(x)$, any tropon from $\mathscr{L}(S_\nu)$ that is present in x is mapped to a null pattern. In Sec. III, we explain how to present the filter φ as a transducer. Dynamics is further reduced by a simple boost of the automaton as to see tropons at rest. In the Appendix, elementary cellular automata CA are analyzed exhaustively in terms of tropons.

Finally, we consider some particular cases. Configurations that are the bi-endless repetition of a unit pattern of length $\lambda>0$ are eventually periodic. But, if such a configuration is a traveling one then it is periodic (in time). The ratio of its time period T to its unit pattern length λ is an integer independent of the automaton rule. The sets of these traveling configurations always have zero topological entropy, and the corresponding language of tropons is dynamically irrelevant for the automaton. For the important class of end-resolving CA[9] we prove a theorem in Sec. V stating that they have invariant subsets of traveling configurations with speeds slower than the range of the automaton, $|\nu|<r$, that *are spatially periodic with zero topological entropy*. End-resolving CA were first introduced by Hedlund,[9] they are a key element in the cryptosystems engineered by Gutowitz[13] and random walks of topological defects in end-resolving CA are studied in Ref. 14. The theorem we prove in Sec. V explains the empirical remark made in Ref. 14, that domains are rarely seen in end-resolving CA. In order to prove the theorem, first in Sec. IV, apparently dissimilar properties of CA are proved to be equivalent to the end-resolving property.[9]

II. GRAPH PRESENTATIONS OF CA

Several types of finite state machines[15] have been used in the literature to emphasize different aspects of cellular automata.[6,16–20] In this section, we review very briefly the basics we will need of transducers. For a thorough discussion of the application of transducers to cellular automata, see Ref. 6.

The idea behind transducers is to present by means of a directed graph a function that maps words from a language to words in another language. The advantage of transducers is that simultaneously they provide a presentation of the domain and image sets of the function. We will make an extensive use of this dual property of transducers.

A cellular map $f{:}X\rightarrow X$ is defined by a block function f_L, with domain Z_k^I. So, to compute $f(x)$ one has to scan the configuration x, block after block, i.e., one needs to replace x by a sequence of overlapping blocks.[21] Hence, part of the description of f is advanced if configurations on Z_k are encoded as configurations on a larger alphabet consisting of k^L

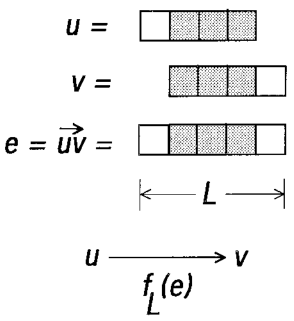

FIG. 1. An edge in the L-block encoding graph. The edge $e = \vec{uv}$ represents a word $a_{L-1} \cdots a_0$ of length L, with $u = a_{L-1} \cdots a_1$ and $v = a_{L-2} \cdots a_0$. The symbols a_i in the shaded cells are the same for u, v, and e. In the transducer $\mathcal{T}_f$, the edges are labeled by the block-function f_L (see Fig. 2).

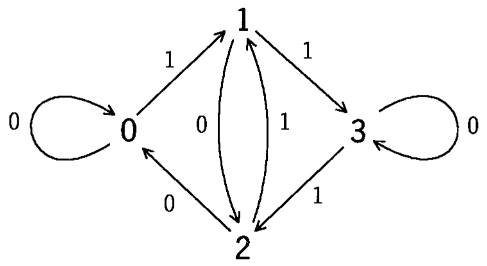

FIG. 2. The f-transducer for elementary CA 106. This automaton is right resolving with permutations $\pi(11) = \chi$ and $\pi(ab) = \text{id}$ for $ab \neq 11$; where id is the identity and χ is the exchange defined over {0,1}.

symbols. Each symbol e in the new alphabet represents a word of length L. After the L-block encoding, the L-block function f_L reduces to a function that maps single symbols from Z_{k^L} to symbols in Z_k. The scheme just described is realized by transducers.[6]

The L-block encoding of X is done with the help of a directed graph $\mathcal{G}_L(k)$ with vertices labeled by words of length $L-1$ and edges connecting pairs of vertices iff they are labeled by words that overlap over a length $L-2$. Figure 1 shows this overlapping condition on two words u and v as vertex labels to be connected by the edge $e = \vec{uv}$ in the L-block encoding graph, $\mathcal{G}_L(k)$. The symbol e identifies the edge $\vec{uv}$ in the graph $\mathcal{G}_L(k)$ and represents a word $a_{L-1} \cdots a_1 a_0$ of length L, with symbols $a_i \in Z_k$. Notice that we are numbering the symbols in a word from right to left and we shall denote by $e|_i$ the symbol a_i implicit in e. Hence, every vertex in $\mathcal{G}_L(k)$ has exactly k incoming edges, as well as k outgoing edges. In every case, two edges, $\vec{uv}$ and $\vec{uw}$, that are rooted at the same vertex u are distinguished by the rightmost symbol, i.e., $\vec{uv}|_0 \neq \vec{uw}|_0$ for $v \neq w$. The set $E = Z_{k^L}$ of edges of $\mathcal{G}_L(k)$ is identified as the set Z_k^L of L blocks.

The block encoding graph $\mathcal{G}_L(k)$ is the skeleton for the transducer presenting a cellular map f. The block function f_L is encoded into $\mathcal{G}_L(k)$ by attaching to every edge e the label $f_L(e)$. The decorated graph that results is the transducer $\mathcal{T}_f$, with output by the edges, and it presents the automaton: $f{:}X \rightarrow X = Z_k^Z$ as defined by f_L. Besides presenting f, the transducer provides a presentation of a vertex language, an edge language and an edge-symbol language in the following way. The vertex language is the set of words made up of vertex symbols that correspond to a continuous path of vertices in the graph. The edge and edge-symbol languages are defined in a similar fashion.

As an example that will be discussed thoroughly in the Appendix, Fig. 2 shows the f transducer for the elementary CA 106. The underlying 3-block encoding graph $\mathcal{G}_3(2)$ is the same for all elementary CA, the decoration of the edges is what makes the difference. The vertices of $\mathcal{G}_3(2)$ in Fig. 2 represent 2-blocks and the vertex language presented by $\mathcal{G}_3(2)$ is a 2-block encoding of the regular language {0,1}*. For instance, the vertex path 132 in Fig. 2 corresponds to the word 0110. Edges represent 3-blocks on Z_2, e.g., the edge $e = 7 = \vec{33}$ represents the word 111. The edges are mapped to edge labels by f_L, e.g., $f_L(\vec{33}) = 0$ in Fig. 2. The edge language presented by $\mathcal{G}_3(2)$ is a 3-block encoding of {0,1}*. For instance, 0110 is encoded by the edges as 36.

For a transducer $\mathcal{T}_f$, the bunch of vertices, together with their edge labels, that are fastened together as incoming edges to a vertex v is called the incoming bouquet $\mathcal{B}_{\text{in}}(v)$ reaching vertex v; formally, $\mathcal{B}_{\text{in}}(v) = \{[u, f_L(\vec{uv})] | \vec{uv} \in \mathcal{T}_f\}$. Similarly, the bouquet stemming out from the vertex u is defined as $\mathcal{B}_{\text{out}}(u) = \{[f_L(\vec{uv}), v] | \vec{uv} \in \mathcal{T}_f\}$. For instance, the f transducer on Fig. 2, presenting the elementary $|f| = 106$, has $\mathcal{B}_{\text{in}}(3) = \{(1,1),(3,0)\}$ and $\mathcal{B}_{\text{out}}(1) = \{(1,3),(0,2)\}$.

The domain of a cellular map f can be restricted to some subset, $Y \subset X$, of configuration space. In this case the set of configurations presented by the edges of the $f|_Y$ transducer is the block encoded domain Y of $f|_Y$. The image of $Y, f(Y)$, is the set of configurations presented by the edge labels of the f transducer. Keep in mind that *edges* and *edge labels* are two different kinds of objects.

III. TRAVELING CONFIGURATIONS AND TROPONS

The set $S_\nu \subset f(X) \subset X$ of causal traveling configurations of f, defined in Eq. (1), is f invariant, i.e., $f(S_\nu) = S_\nu$. This property means that the transducer for $f|_{S_\nu}$ provides a finite presentation of the set S_ν, either by the edges or the edge labels. Indeed, the transducer for $f|_{S_\nu}$ is a subgraph of the transducer presenting the unrestricted automaton f, e.g., Fig. 2. The subgraph for $f|_{S_\nu}$ has the property that *the configurations presented by its edges are the L-block encoding of the configurations presented by its edge labels.*

The problem of identifying the set of configurations that travel with velocity ν, S_ν, is thus to find the transducer on $\mathcal{G}(S_\nu)$ as a subgraph of $\mathcal{T}_f$. The edges and edge labels of the subgraph must satisfy the following conditions:

$$\begin{aligned} &\text{(i):} \quad f_L(e) = \sigma^\nu(e) = e|_{r-\nu}, \quad |\nu| \leq r, \\ &\text{(ii):} \quad e \text{ is part of a closed loop.} \end{aligned} \tag{2}$$

In condition (i), $e|_r$ is the central symbol of the L-block e and we are using the same symbol σ to denote the 3-block

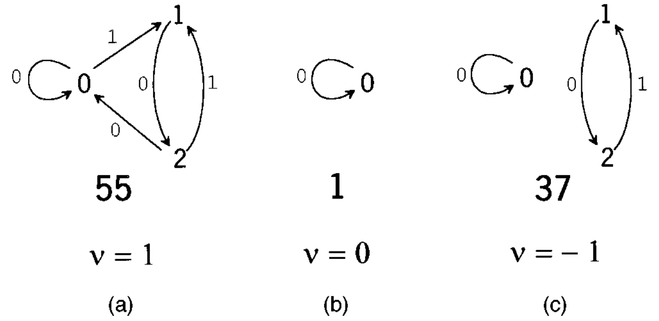

FIG. 3. Transducers presenting the sets of traveling configurations for the elementary $|f|=106$, numbered by their characteristic number $|S_\nu|$. Only the set S_1 has nonzero entropy.

function that generates the left shift, σ. Equation (2) is an effective definition of a traveling configuration, equivalent to the one in Eq. (1). Condition (ii) is just a convention to eliminate from the transducer presenting $f|_{S_\nu}$ all the edges that are irrelevant for bi-infinite configurations. This convention normalizes the transducer.

Introducing the integers $|e|=\Sigma_i e|_i k^i$ and $|f|=\Sigma_e f_L(e)k^{|e|}$, the usual valuation of the cellular map f, the important quantities in condition (i) of Eq. (2) can be written in modulo k arithmetics as

$$f_L(e)\equiv\left[\frac{|f|}{k^{|k|}}\right](\text{mod }k)\quad\text{and}\quad e|_{r-\nu}\equiv\left[\frac{|e|}{k^{r-\nu}}\right](\text{mod }k). \tag{3}$$

The square brackets in Eq. (3) denote the integer part. The condition (i) in Eq. (2) for e to be an edge in $\mathscr{G}(S_\nu)$ reads

$$\left[\frac{|e|}{k^{r-\nu}}\right]\equiv\left[\frac{|f|}{k^{|e|}}\right](\text{mod }k), \tag{4}$$

in modulo k arithmetics.

Conditions (i) and (ii) are the basis of the algorithm to find the transducer for $f|_{S_\nu}$. For a given automaton value $|f|$ and velocity ν, Eq. (4) is used to find the set of edges of $\mathscr{G}(S_\nu)$ as follows. For $|e|=0$ to k^L-1, if Eq. (4) is satisfied the edge e belongs to $\mathscr{G}(S_\nu)$. The set of edges so obtained is then trimmed as to satisfy condition (ii). The trimming is done by eliminating the vertices that do not have any outgoing or incoming edge. The trimming process is repeated until the set of edges stabilizes. The required number of repetitions to stabilize the set of edges is always smaller than the number of vertices of $\mathscr{G}_L(k)$. Figure 3 shows the transducers, obtained with the algorithm, that present $f|_{S_\nu}$ for the three invariant sets of traveling configurations, $\nu=\pm 1,0$, for the elementary CA $|f|=106$.

The transducer presenting $f|_{S_\nu}$ is a subgraph of $\mathscr{T}_f$. Accordingly, we give to S_ν the valuation

$$|S_\nu|=\sum_{e\in\mathscr{T}_f} c_e k^{|e|}, \tag{5}$$

with coefficients c_e given as $c_e=1$ if edge e is in the transducer for $f|_{S_\nu}$ and $c_e=0$ otherwise. The integer $|S_\nu|$ in Eq. (5) defines the skeleton graph $\mathscr{G}(S_\nu)$ as a subgraph of $\mathscr{G}_L(k)$. We call it the characteristic number of the $f|_{S_\nu}$ transducer. The total number of subgraphs of $\mathscr{G}_L(k)$ is $2^{k^L}-1$, 255 for the elementary CA.

The simplest graphs that are relevant for bi-infinite configurations consist of disjoint edge loops. The configurations presented by these graphs are the bi-endless concatenation of a single unit pattern. The length of the unit pattern in a spatially periodic x is the smallest integer $\lambda>0$ such that $\sigma^\lambda x=x$. A spatially periodic traveling configuration is periodic in time. The time period of x is defined as the smallest integer T such that $f^T(x)=x$. The time period is given by the next lemma.

Lemma 1. Let x be a spatially periodic traveling configuration of f with unit pattern length $\lambda>0$ and speed $|\nu|>0$. Then, it has period $T=\lambda/(\lambda,|\nu|)$, where (n,m) denotes the greatest common divisor of the integers n and m.

Proof. First, x is a traveling configuration: $f(x)=\sigma^\nu x$, with $\nu\neq 0$. Then, there exists a positive integer T such that $f^T(x)=x$. Let T be the smallest such integer. Then, there exists $n>0$, relative prime to T, such that $n\lambda=|\nu|T$. Since $(n,T)=1$ (otherwise T is not the smallest positive integer), it follows that $T=\lambda/(\lambda,|\nu|)$.

The time sequence of spatially periodic configurations $f^t(x)$ appears to jump back and forth, relative to x, but at the end it takes one time period T to shift x one unit pattern length λ. The net velocity, defined as the quotient λ/T, is the integer $(\lambda,|\nu|)$. Spatially periodic traveling configurations are presented by the simple loops of the transducer on $\mathscr{G}(S_\nu)$. The set S_ν is spatially periodic if it is presented by a graph consisting of disjoint loops.

Definition 1. Tropons are the patterns in the language $\mathscr{L}(S_\nu)$ of edge labels of the transducer on $\mathscr{G}(S_\nu)$. Every vertex is an initial as well as an acceptance state.

Tropons are the words built up of edge labels $f_L(e)$, corresponding to paths in $\mathscr{G}(S_\nu)$ connecting any pair of vertices. The language of tropons $\mathscr{L}(S_\nu)$ is a regular language.

Domains are the tropons that have both ends connected to nontropon patterns in a CA configuration. Not every tropon is a domain and $\mathscr{D}(S_\nu)\subset\mathscr{L}(S_\nu)$, strictly. To be a domain, a tropon must start at any vertex of $\mathscr{G}(S_\nu)$ that has lost at least one incoming edge, respect to $\mathscr{T}_f$. So, the set of vertices that are initial states to define the domain language $\mathscr{D}(S_\nu)$ is

$$\Sigma_{\text{init}}=\{v\in\mathscr{G}(S_\nu)\,|\,\exists\,\overrightarrow{uv}\ \text{ in }\ \mathscr{G}_L(k)\ \text{ but not in }\ \mathscr{G}(S_\nu)\}.$$

Similarly, a domain must finish at a vertex of $\mathscr{G}(S_\nu)$ that has lost, respect to $\mathscr{G}_L(k)$, at least one outgoing edge. So, the set of vertices that are acceptance states to define $\mathscr{D}(S_\nu)$ is

$$\Sigma_{\text{acc}}=\{v\in\mathscr{G}(S_\nu)\,|\,\exists\,\overrightarrow{vw}\ \text{ in }\ \mathscr{G}_L(k)\ \text{ but not in }\ \mathscr{G}(S_\nu)\}.$$

The initial- and acceptance-state sets are identified by direct inspection of $\mathscr{G}(S_\nu)$ against $\mathscr{G}_L(k)$.

Definition 2. The language of domains $\mathscr{D}(S_\nu)$, associated to the invariant set of traveling configurations S_ν, is the

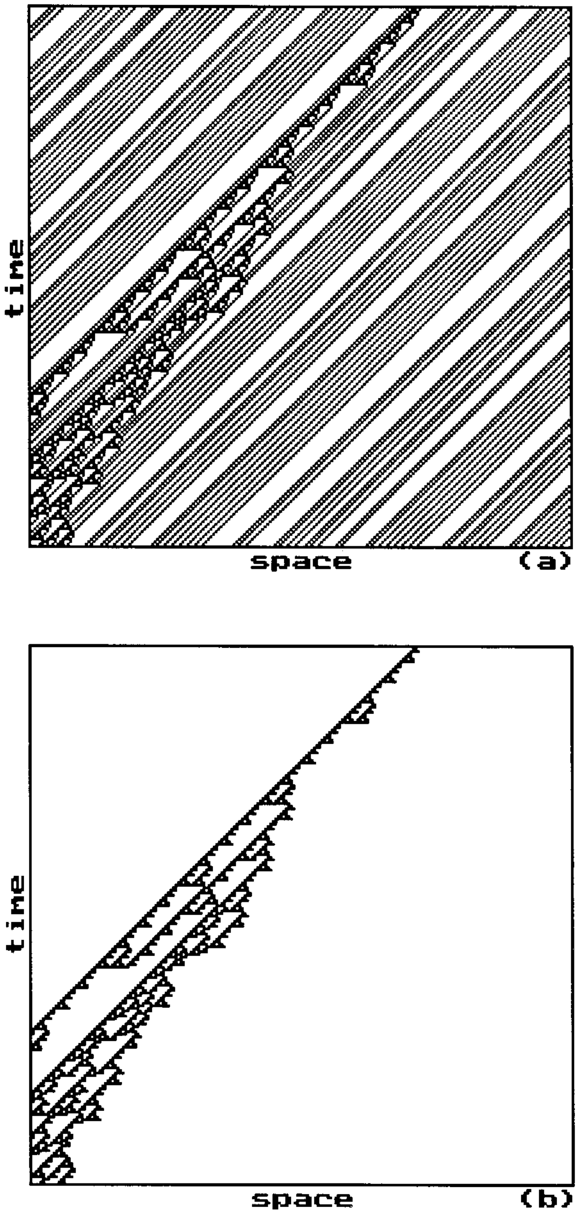

FIG. 4. (a) Space–time pattern for a single interface between two semi-infinite traveling domains with $\nu=1$ for ECA 106. Time runs downwards. (b) The tropons are identified and filtered out to zero with the filter of characteristic number $|S_1|=55$.

edge-label language accepted by the transducer on $\mathscr{G}(S_\nu)$ with Σ_{init} the set of initial states and Σ_{acc} the set of acceptance states.

The space–time pattern in Fig. 4(a) is a 210×210 patch of the evolution of the elementary $|f|=106$ from an initial configuration prepared by tying together, with a 1 in between, two semi-infinite traveling domains, generated as random walks in the graph for S_1, shown in Fig. 3(a). The single interface in Fig. 4(a) goes growing as it travels towards the left side. For details see the Appendix.

Next, we explain how to filter tropons off the space–time patterns. For a given set S_ν, the filter is a cellular map $\varphi{:}X\rightarrow X$ that is defined by the block function $\varphi_L:Z_k^L\rightarrow Z_k$ such that $\varphi_L(e)=c_e\,\mathrm{id}_L(e)$. The block function id_L defines the identity cellular map, $\mathrm{id}(x)=x$, and the coefficients c_e are retrieved from the integer $|S_\nu|$ by the equation

$$c_e\equiv 1+\left[\frac{|S_\nu|}{2^{|e|}}\right](\mathrm{mod}\ 2). \tag{6}$$

Hence, the characteristic number of S_ν provides the filter to make tropons invisible. This is nice, since we do know how tropons behave. If we see the configuration x through the filter as $\varphi(x)$ we see the actual configuration only at sites where the patterns are not tropons. Figure 4(b) shows the space–time pattern of Fig. 4(a) filtered with $|S_1|=55$ [see Fig. 3(a)].

For sets of traveling configurations with large values of the entropy, e.g., the elementary $|f|=106$, randomly generated configurations present a high density of domains. In the case that $\nu\neq 0$, the presence of domains imposes an overall drift velocity to the space–time pattern generated from any random initial configuration. The global drift is eliminated from the dynamics by a simple boost. The automaton $\tilde{f}=\sigma^{-\nu}f$ is a boosted version of f; boosted to the velocity $-\nu$, as to compensate the drifting introduced by domains of velocity ν. For any f, the boosted automaton $\tilde{f}$ can be encoded using blocks of length $L'=L+2|\nu|$.

The natural generalization of a traveling configuration is a coherent succession of traveling configurations (a flock). We say that the configuration $x\in X$ belongs to a traveling flock of the CA f if

$$f^{\tau}(x)=\sigma^{\nu}x. \tag{7}$$

The duration (number of different successive configurations) and speed of a flock are the smallest τ and $|\nu|$, respectively, that satisfy Eq. (7). A flock is then the set of configurations

$$\beta(\tau,\nu)=\{x^{(i)}=f^{i}(x)|i=0,\cdots,\tau-1;x=x^{(0)};f^{\tau}(x)=\sigma^{\nu}x\}. \tag{8}$$

The flock $\beta(\tau,\nu)$ slides rigidly with velocity ν under f, $f\beta(\tau,\nu)=\sigma^{\nu}\beta(\tau,\nu)$. As for configurations, we say a flock is spatially periodic if its configurations are formed from a unit pattern of length λ, defined as the smallest positive integer such that $\sigma^{\lambda}x=x$, for every x in the flock. The set of all traveling flocks of velocity ν of an automaton f is $\mathscr{A}_\nu=\cup_\tau\beta(\tau,\nu)$. The extension of Lemma 1 to flocks follows.

Lemma 2. Let $\beta(\tau,\nu)$ be a spatially periodic flock of automaton f, with unit pattern length $\lambda>0$. Then, it is periodic in time with time period $T=\tau\lambda/(\lambda,|\nu|)$.

Proof. Let $\tilde{f}=f^{\tau}$ and let m be the smallest positive integer such that $\tilde{f}^{m}(x)=x$, i.e., $m\nu=n\lambda$ for some integer n such that $(m,n)=1$. It then follows that $m=\lambda/(\nu,\lambda)$. The time period of x is $T=m\tau$. i.e., $T=\tau\lambda/(\nu,\lambda)$.

IV. END-RESOLVING CA

They were introduced by Hedlund,[9] they are a key element in the cryptosystems engineered by Gutowitz,[13] and random walks of topological defects in end-resolving CA have been studied in Ref. 14. In the next section, the invariant subsets of traveling configurations and flocks are characterized for the important class of end-resolving cellular automata. But first, in this section, we define them and prove two lemmas that are necessary to characterize their traveling configurations.

In the literature one finds several definitions to describe the same end-resolving property of a block function. The following lemma establishes the equivalence of four definitions that are apparently dissimilar.

Lemma 3. The next four statements about the block function f_L are equivalent.

(a) Let $f_L(e)=a$ and let the 2-block ab appear in the image of $f, f(X)$. Then, there is just one cell value $c \in Z_k$ such that $f_{L+1}(ec)=ab$. In this case we say that f_L is *right resolving*.

(b) Let $\pi(w)$ map the $(L-1)$ block w to a permutation of the cell symbols in Z_k. The block function f_L is *right permuting* if $f_L(wa)=\pi(w)a$.

(c) The block function f_L is presented by an f transducer such that at every one of its vertices all of the outcoming edges are labeled differently, i.e., $f_L(\overrightarrow{uv}) \neq f_L(\overrightarrow{uw})$, for $v \neq w$. Equivalently, for every vertex u of $\mathscr{T}_f$, an edge label does not appear twice in $\mathscr{B}_{\mathrm{out}}(u)$.

(d) The block function f_L is presented by an f transducer; $\mathscr{T}_f$, of disjoint incoming bouquets, i.e., $\mathscr{B}_{\mathrm{in}}(u) \cap \mathscr{B}_{\mathrm{in}}(v) = \emptyset$, for $u \neq v \in \mathscr{T}_f$.

Proof. (a)≡(b). Let f_L be right permuting and ab a 2-block appearing in $f(X)$. Then, there exists a $(L+1)$ block $c'(wc)$ such that $f_L(c'w)=a$ and $f_L(wc)=b$. But $f_L(wc)=\pi(w)c=b$, i.e., the cell symbol c is unique. This shows that f_L is right resolving. Next, let f_L be right resolving. Then, for $ab \in f(X)$ and $f_L(e=c'w)=a$ there is only one c such that $b=f_L(wc)$, i.e., for w fixed, the resulting map $c \mapsto b$ is a permutation. This shows that f_L is right permuting.

(b)≡(c). Let f_L be right permuting: $f_L(\overrightarrow{uv}) = \pi(u)\overrightarrow{uv}|_0$. Together with the property of block encoding graphs, the skeleton of $\mathscr{T}_f$, that $\overrightarrow{uv}|_0 \neq \overrightarrow{uw}|_0$ for $v \neq w$; we conclude that $f_L(\overrightarrow{uv}) \neq f_L(\overrightarrow{uw})$. Next, let f_L be as in (c). Then, for a fixed vertex u, the assignment $f_L(\overrightarrow{uv}) \mapsto \overrightarrow{uv}|_0$ implied in $\mathscr{T}_f$ is a permutation of cell symbols.

(c)≡(d). Let f_L be as in (c) and let $[w, f_L(\overrightarrow{wu})] \in \mathscr{B}_{\mathrm{in}}(u)$ and $[w, f_L(\overrightarrow{wv})] \in \mathscr{B}_{\mathrm{in}}(v)$ for $u \neq v$. Then, given that $f_L(\overrightarrow{wu}) \neq f_L(\overrightarrow{wv})$, we conclude that $\mathscr{B}_{\mathrm{in}}(u) \cap \mathscr{B}_{\mathrm{in}}(v) = \emptyset$. Next, let f_L be as in (d) and let the vertex w appear in $\mathscr{B}_{\mathrm{in}}(u)$ and $\mathscr{B}_{\mathrm{in}}(v)$ for $u \neq v$. Given that $\mathscr{B}_{\mathrm{in}}(u) \cap \mathscr{B}_{\mathrm{in}}(v) = \emptyset$, we conclude that $f_L(\overrightarrow{wu}) \neq f_L(\overrightarrow{wv})$, for $u \neq v$.

An example of right-resolving CA is the elementary $|f| = 106$, presented by the f transducer shown in Fig. 2. Right-resolving CA are $(k!)^{k^{L-1}}$ in number. Left-resolving CA are defined to have a block function given by $f_L(aw)=\pi(w)a$. Left-resolving CA are $(k!)^{k^{L-1}}$ in number also. A lemma equivalent to Lemma 3 holds true for left-resolving CA. One just replaces ''right'' by ''left'' and exchanges ''leaving'' and ''incoming'' in Lemma 3.

TABLE I. Characteristic numbers of the sets of traveling configurations, of the indicated velocities, for all the end-resolving elementary CA. The rules are organized in equivalence classes (Ref. 23) of the group generated by reflection (r) and conjugacy (c). A letter L (R) means that the corresponding automation f is left resolving (right resolving). The topological entropy h of the sets is zero, except as indicated.[a]

$\lvert f\rvert$	end	$v=1$	$v=0$	$v=-1$
15	L	90	36	0
85(r)	R	0	36	90
30	L	1	37	1
86(r)	R	1	37	1
135(c)	L	128	164	128
149(b)	R	128	164	128
45	L	104	22	0
75(c)	L	22	104	0
89(b)	R	0	104	22
101(r)	R	0	22	104
60	L	105	1	1
102(r)	R	1	1	105
153(b)	R	128	128	150
195(c)	L	150	128	128
90	L,R	1	105	1
165(c)	L,R	128	150	128
105	L,R	36	90	36
106	R	55(−)	1	37
120(r)	L	37	1	55(−)
169(c)	R	236(−)	128	164
225(b)	L	164	128	236(−)
150	L,R	129	165	129
154	R	139(*)	129	129
166(c)	R	209(*)	129	129
180(b)	L	129	129	139(*)
210(r)	L	129	129	209(*)
170	R	255(+)	129	165
240(r)	L	165	129	255(+)

[a] (−) $h=0.694\,238$, (+) $h=1.0$, (*) connected loops with $h=0$, (r) = reflection, (c) = conjugacy, (b) = the composition of r and c.

End-resolving CA are easy to find among the linear automata. An automaton f is linear if its block function is given by

$$f_L(e) \equiv C_0 + \sum_{i=1}^{L} C_i e|_i (\mathrm{mod}\ \ k). \tag{9}$$

In the set of k^{L+1} linear CA, $\phi(k)k^L$ of them are left resolving (the function ϕ is Euler's). They are the linear CA with coefficient C_1 relative prime to k, i.e., such that $(C_1, k)=1$. Similarly, there are $\phi(k)k^L$ linear CA that are right resolving. They have coefficient C_L such that $(C_L, k)=1$. Linear CA with both coefficients C_1 and C_L relative primes to k are resolving at both ends.

Table I is the complete list of elementary end-resolving CA. The second column indicates whether it is a right (R) or a left (L) resolving automaton. The 16 elementary CA with rule numbers that are multiples of 15 ($15n$, $n=1,\ldots,16$) are the left-resolving elementary CA. All the sets of traveling configurations of the end-resolving elementary CA are spatially periodic, except for those marked with (−), (+) and (*) in Table I. The sets marked (*) are not spatially periodic but still have zero entropy. They are presented by graphs that have singly connected loops.

The property of f being end resolving is preserved by the composition of f with itself.

Lemma 4. Let f be end resolving. Then, for $n>1$, f^n is end resolving.
Proof. (right-resolving CA). Let f be right resolving. First, we prove that f^2 is right resolving. If f_L is the block function defining f, then f^2 is defined by blocks of length $L_2=2L-1$, an odd integer:

$$\begin{aligned} f_L[f_L(e_{L-1})\cdots f_L(e_0)] &= f_L[\pi(w_{L-1})a_{L-1}\cdots\pi(w_0)a_0] \\ &= \pi(u)[\pi(w_0)a_0], \end{aligned} \tag{10}$$

where $e_0=a_{L-1}\cdots a_0$, $e_i=w_ia_i$, and $u=\pi(w_{L-1})a_{L-1}\cdots\pi(w_1)a_1$, is a word of length $L-1$. In Eq. (10) the product $\pi(u)\pi(w_0)$ is a permutation that depends on the word of length L_2-1 that is a prefix of the path of edges $e_{L-1}\cdots e_0$ in the f transducer. That f^n is right resolving follows by induction.

The proof for left resolving follows very similar steps. Mixed compositions of left- and right-resolving automata do not transmit any end-resolving property to the composite automaton. For instance, the elementary $|f|=106$ is right resolving and the right shift σ^{-1} is left resolving. The boosted version of f, $\tilde{f}=\sigma^{-1}\circ f$, is not end resolving.

V. TRAVELING CONFIGURATIONS IN END-RESOLVING CA

The main result of this section is that for $|\nu|<r$, end-resolving CA always have nonempty sets of traveling flocks and they always are spatially periodic. For $|\nu|=r$ nothing can be concluded.

The entropy of spatially periodic sets is zero and this makes tropons to have a secondary (if any) role in the dynamics of end-resolving CA. If an end-resolving CA has a set of traveling configurations with positive entropy, then they must be running at top speed $|\nu|=r$. This is the case of the right-resolving elementary $|f|=106$ (see Table I), that has a set of traveling configurations with entropy $h=0.694$ (and characteristic number 55). Besides elementary CA 106, elementary CA 154 in Table I has a set of traveling configurations that are not spatially periodic with characteristic number 139. However, all CA in Table I have sets of standing configurations, $\nu=0$, and they all are spatially periodic.

Lemma 5. Let f be right resolving and let $\nu<r$. Then f has a set of traveling configurations S_ν that is spatially periodic.

Proof. (right resolving). We prove that the transducer presenting $f|_{S_\nu}$ consists of disjoint loops. Let f be right resolving. After Lemma 3 we know that all the edges $e_i = \overrightarrow{uv}_i$, $i=1,\cdots,k$, stemming out from any root vertex u and aiming vertex v_i in the transducer $\mathscr{T}_f$ are labeled according to $f_L(\overrightarrow{uv}_i) = \pi(u)e_i|_0$ and $e_i|_0\neq e_j|_0$ for $i\neq j$ (e.g., Fig. 1). By definition of $f|_{S_\nu}$, the only edges from $\mathscr{T}_f$ that appear in the transducer presenting $f|_{S_\nu}$ are those labeled as $f_L(\overrightarrow{uv}) = e_i|_{r-\nu}$, i.e.,

$$\pi(u)e_i|_0=e_i|_{r-\nu}. \tag{11}$$

Since all the edges e_i have the same root vertex u, the symbol $e_i|_{r-\nu}\equiv b$ is the same for every i if $\nu<(L-1)/2=r$. Hence, condition in Eq. (11) is $\pi(u)e_i|_0=b$, with symbol b fixed. The condition always has a unique solution for $e_i|_0$, i.e., only one of the edges that outcome from u in the f-transducer survives in the graph presenting $f|_{S_\nu}$. We conclude that all the vertices in $\mathscr{G}(S_\nu)$ have just one outgoing edge. This result is enough to assure that $\mathscr{G}(S_\nu)$ is a collection of disjoint loops.

In the proof of Lemma 5 is clear that if $\nu=r$, condition Eq. (11) for an edge to survive is $\pi(u)a_i=a_i$. This condition may have no solution, one solution or more than one solution. For instance, when $\pi=\mathrm{id}$ the set S_ν happens to be X, e.g., σ and σ^{-1} in Table I. Lemma 5 is generalized to flocks by the following.

Theorem (right resolving and $\nu<r$ imply spatial periodicity). Let f be right resolving and let $\nu<r$. Then, the set of traveling flocks $\mathscr{A}_\nu$ is spatially periodic.

Proof. By Lemma 4, $\tilde{f}=f^\tau$ is end resolving. From Lemma 5 it follows that $\tilde{f}$ has a set of traveling configurations, $\tilde{S}_\nu$, that is spatially periodic. The equivalence relation on $\tilde{S}_\nu$, $x\sim y$ iff $x=f^n(y)$ for some n, identifies flocks as equivalence classes in $\tilde{S}_\nu$ [see Eq. (8)]. Evidently, they are spatially periodic.

Equivalent results hold for left-resolving CA. The proof of the left-resolving equivalent to Lemma 5 is quite similar. In this case one proves that all the vertices in $\mathscr{G}(S_\nu)$, for $\nu>-r$, have just one incoming edge. Again, this condition is enough to assure that the graph $\mathscr{G}(S_\nu)$ consists of disjoint loops, only. The left-resolving equivalent of the theorem requires that $\nu>-r$. The theorems explain, at least for tropons, the remark made in Ref. 14 that particle-like patterns are rarely seen in end-resolving cellular automata.

VI. CONCLUSIONS

The problem of finding the full invariant subsets of causal traveling configurations for cellular automata was solved with the help of transducers. We showed how to identify the sets of traveling configurations from the defining presentation of the automaton. A general algorithm to find transducers to present the restricted automaton $f|_{S_\nu}$, the invariant set of traveling configurations, and the corresponding regular languages of tropons, $\mathscr{L}(S_\nu)$, and domains, $\mathscr{D}(S_\nu)\subset\mathscr{L}(S_\nu)$ was given. Tropons are patterns of finite length that propagate rigidly with constant velocity in a CA configuration. Domains are special tropons that have their ends matching nontropons.

The criterion for dynamical relevance of an invariant subset is to have nonzero topological entropy. In this way, the automaton has a nontrivial limit set and the invariant subset under consideration samples an important part of it. The topological entropy of the set of traveling configurations is computed from its presentation.

Dynamics within domains is a simple shift. It was explained how to filter tropons out of the game, leaving just the interfaces in a CA configurations. A cellular automaton $\varphi{:}X\to X$ provides the filter for tropons. When an automaton configuration x is seen through the filter as the configuration $\varphi(x)$, any tropon that is present in x is mapped to a null pattern. The filter φ was presented by a transducer. Dynamics was further reduced by a simple boost of the automaton as to move along with tropons. The part of the dynamics left unknown resides just in the play of the interfaces.

We paid special attention to the important class of end-resolving CA. First, several properties of CA were proved to be equivalent to the end-resolving property. Then, we proved that their invariant sets of traveling configurations with speeds slower than the range of the automaton, $\nu<r$ (the top speed excluded), *are spatially periodic and have zero topological entropy*. This result explains for tropons the fact that domains are rarely seen in end-resolving CA.

In the Appendix, the criterion of positive entropy is used to identify all the elementary cellular automata with dynamically relevant sets of tropons.

In conclusion, we have discussed how to understand and to extract tropons out of the dynamics. Tropons are not the only type of structure that is dynamically relevant in CA. For instance, the dynamics of the elementary CA 18 is organized in Ref. 6 with a very simple domain that is not of the kind we have discussed. However, we are certain that any insight towards the understanding of the dynamics of extended objects will be useful when devising applications of CA.[13,22]

ACKNOWLEDGMENTS

The authors G.S.A. and E.U. are CONACyT (México) fellows at Carleton U. (Ottawa, Canada) and CPT-Luminy (Marseille, France), respectively. A.E. is a participant in the doctoral program of SEP-CONACyT (México). This collaboration was made possible by CONACyT (México) through Contract No. 2109-E.

APPENDIX: TROPON ANALYSIS OF ELEMENTARY CA

1. End-resolving elementary CA

The sets of traveling configurations ($|\nu|\leq 1$) for end-resolving elementary CA, discussed in Secs. IV and V, are listed in Table I. The table is organized in equivalence classes of the group generated by reflection (r) and conjugacy (c);[23] rules 105 and 150 are single in their respective classes. As explained in Sec. V, most of the sets of traveling configurations for end-resolving elementary CA are spatially periodic with zero topological entropy. The left and right shifts, elementary CA nos. 170 and 240, respectively, have the whole of configuration space X as set of traveling configurations. The set X has maximum entropy $h=1$ and is presented by $\mathscr{T}_f$ itself, with characteristic number 255. Besides the well known left and right shifts, only end-resolving elementary CA Nos. 106, 120, 169, and 225, all of them within the same equivalence class,[23] have sets of traveling configurations with nonzero entropy. We discuss in the following the minimal representative, elementary CA 106.

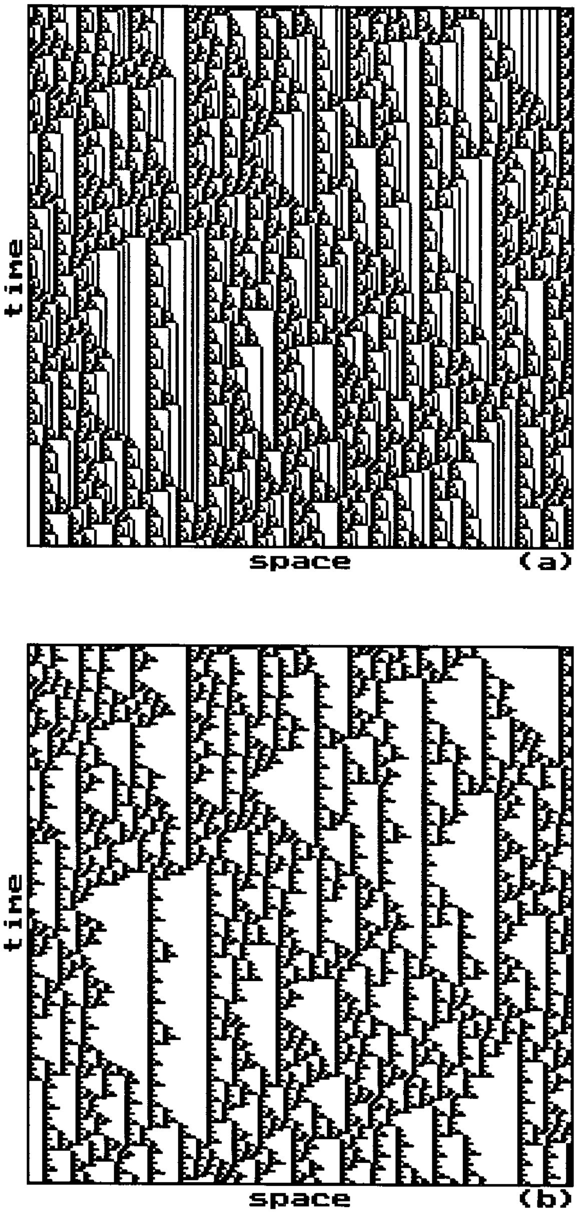

FIG. 5. (a) Space–time pattern for the boosted automaton $\tilde{f}=\sigma^{-1}f$, with f the elementary CA 106. The initial configuration was generated at random, with equal probabilities for 0's and 1's. (b) Same as (a) with tropons filtered out.

The transducer on $\mathscr{S}(S_\nu)$ presenting $f|S_\nu$, $\nu=\pm 1,0$, for the elementary CA 106 are obtained by using the methods of Sec. III. The result is shown in Fig. 3. The null configuration $\underline{0}=\cdots 000\cdots$ is degenerate. It is presented by all three transducers shown in Figs. 3(a)–3(c) since $f(\underline{0})=\sigma^{-1}\underline{0}=\sigma^{0}\underline{0}=\sigma^{1}\underline{0}$. Similarly, the alternating configurations $\underline{01}$ and $\underline{10}=\sigma\underline{01}$ appear both in set S_1 and set S_{-1}, respectively. The nontrivial set is S_1, with topological entropy $h=0.694$ and is presented by the edge labels of the transducer with characteristic number 55 in Fig. 3. This invariant subset provides an ample sampling of the limit set of the automaton. As seen in Fig. 5, a randomly generated configuration presents a high density of domains. The presence of the traveling domains imposes an overall drift velocity to the space–time pattern generated from any random initial configuration. The global drift is eliminated from the dynamics by a simple boost, $\tilde{f}=\sigma^{-1}f$. For any f of range $r=1$, the boosted version $\tilde{f}$ has block function $\tilde{f}_5$, on blocks of length $L=5$. To find $\tilde{f}$, for every 3-block e_3 one produces four 5-blocks by the rule

$$|e_3|\rightarrow|e_5|=4|e_3|+n, \quad n=0,1,2,3, \tag{A1}$$

and the block function $\tilde{f}_5$ is assigned values such that

$$\tilde{f}_5(4e_3+n)=f_3(e_3). \tag{A2}$$

After the boost, the set of traveling configurations S_1 for elementary CA 106 appears as the set S_0 of standing configurations of the boosted map $\tilde{f}$. Figure 5 shows a 210×210 patch of the space–time pattern of automaton $\tilde{f}$ for a randomly generated initial configuration. The drifting is eliminated in both Figs. 5(a) and 5(b). One just sees domains that fluctuate in length. In Fig. 5(b) the domains have been filtered out to zero using coefficients c_e retrieved from $|S_1|=55$, as explained in Sec. III. In the filtered pattern, Fig. 5(b), domains are shown as white strips of fluctuating width and black dots correspond to edges not in the transducer presenting $f|_{S_1}$ (see Fig. 3).

Tropons are words in $\mathscr{L}(S_1)$ and domains are the tropons that have both ends matching to edges not in $\mathscr{L}(S_1)$. The domain language $\mathscr{D}(S_1)$ is a subset of $\mathscr{L}(S_1)$. A comparison of $\mathscr{G}(S_1)$ in Fig. 3(a) with $\mathscr{G}_L(2)$ in Fig. 2 shows that the edges that are missing in $\mathscr{G}(S_1)$ are $\overrightarrow{13}$, $\overrightarrow{33}$, and $\overrightarrow{32}$. Compare Fig. 3(a) to Fig. 2. The missing piece of the graph accepts the vertex language $\cdots 1[3^+]2\cdots$, the 2-block encoding of $\cdots 0[1^+]10\cdots$; the interface language is the expression between brackets. The regular language $\mathscr{T}=1^+$ is the set of nontraveling patterns that act as interfaces between domains. To produce an interface, one exits the graph in Fig. 3(a) at vertex 1, loops around vertex 3, and reenters the graph at vertex 2. So, for the language of domains $\mathscr{D}(S_1)$ the sets of initial and acceptance states in $\mathscr{G}(S_1)$ includes just vertex 2 and vertex 1, respectively. The domain language is $\mathscr{D}(S_1)=0^*1(0^+1)^*$. Finally, any configuration in X is an alternating concatenation of domains from $\mathscr{D}(S_1)\subset\mathscr{L}(S_1)$ with interfaces from $\mathscr{T}$. However, in Fig. 5 one notices that long interfaces are not very frequent in the space–time pattern. This is because in one time step long interfaces either decay to 010*10 [a block of zeros is a tropon, see Fig. 3(a)], or it fuses with the domain next to its right as to become a tropon in a largest domain. Hence, long interfaces, once they are produced, either they decay in one time step into a domain or they become tropons that fuse with the domain next to its right. It means that interfaces are vulnerable only by its right end and domains fluctuate in size only by its left end, its right end being pinned.

So, the most abundant interface in any space–time pattern of elementary CA 106 is $\cdots 0[1]10\cdots$. This short interface shows a complicated interaction with tropons. To see this, we prepared an initial configuration, consisting of two semi-infinite configurations, generated as random walks in the transducer on $\mathscr{G}(S_1)$ [see Fig. 3(a)], separated by a 1 interface. A 210×210 patch of the evolution of such a configuration is shown in Fig. 4.

TABLE II. The elementary CA that have two sets of traveling configurations with nonzero entropy. The sets of traveling configurations are entered by its characteristic number $|S_\nu|$ and the entropy value is in parentheses. The rules are organized in equivalence classes as in Table I.

$\|f\|$	$v=1$	$v=0$	$v=-1$
56	109(0.405 687)	1(0.0)	55(0.694 238)
98(r)	55(0.694 238)	1(0.0)	109(0.405 687)
185(b)	236(0.694 238)	128(0.0)	182(0.405 687)
227(c)	182(0.405 687)	128(0.0)	236(0.694 238)
57	108(0.405 687)	0(0.0)	54(0.405 687)
99(c)	54(0.405 687)	0(0.0)	108(0.405 687)
172	249(0.551 62)	159(0.551 62)	129(0.0)
202(c)	159(0.551 62)	249(0.551 62)	129(0.0)
216(b)	129(0.0)	235(0.551 62)	215(0.551 62)
228(r)	129(0.0)	215(0.551 62)	235(0.551 62)
184	237(0.694 238)	139(0.0)	183(0.694 238)
226(c)	183(0.694 238)	209(0.0)	237(0.694 238)

TABLE III. Minimal representatives of the equivalence classes of elementary CA with sets of standing configurations that have nonzero topological entropy. The transducers are entered by their characteristic numbers.

$\|f\|$	$v=1$	$v=0$	$v=-1$
4	1	55(0.694 238)	1
5	0	54(0.405 687)	0
12	1	55(0.694 238)	1
13	0	54(0.405 687)	0
36	1	23(0.551 62)	1
44	121	23(0.551 62)	1
72	1	123(0.551 62)	1
73	0	122(0.287 756)	0
76	1	127(0.879 149)	1
77	0	126(0.694 238)	0
78	1	125(0.405 687)	1
94	1	109(0.405 687)	1
104	53	91(0.464 961)	39
108	1	95(0.694 238)	1
132	209	183(0.694 238)	139
140	209	191(0.694 238)	129
164	209	151(0.551 62)	139
200	129	251(0.551 62)	129
204	129	255(1.0)	129
232	189	219(0.694 238)	231

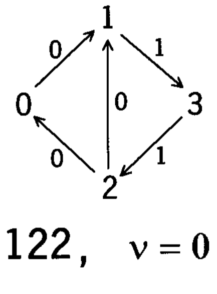

FIG. 6. Transducer presenting the set of standing configurations of the elementary $|f|=73$. The characteristic number is $|S_0|=122$ and the topological entropy is $h=0.287$. The edges of the graph are decorated by the block function of the CA.

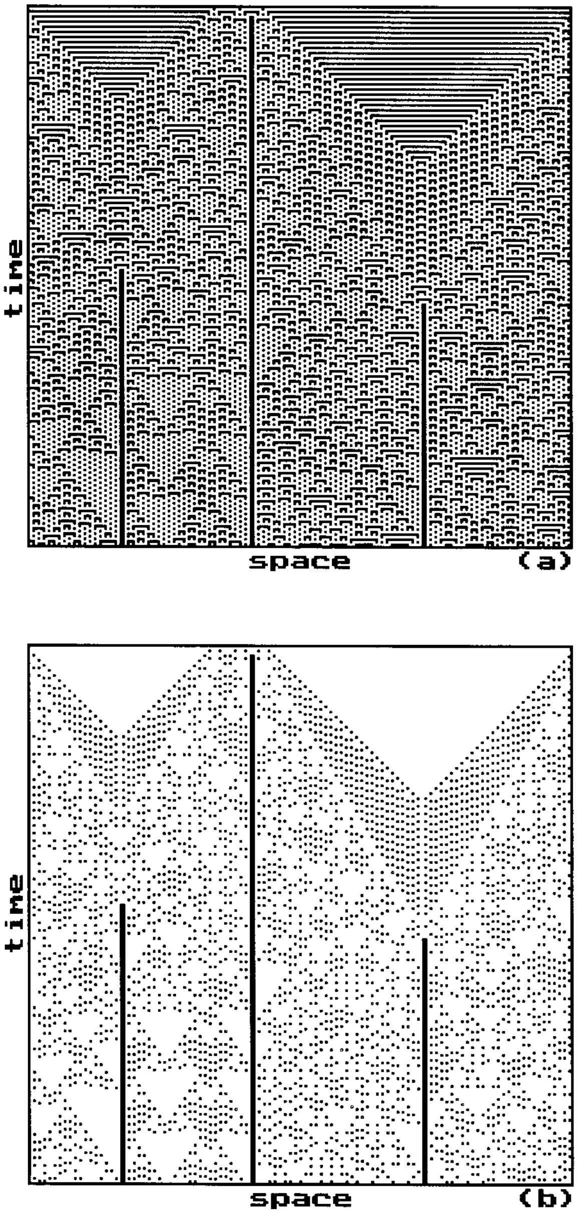

FIG. 7. (a) A 210×210 patch of the space–time pattern of the elementary $|f|=73$. The initial configuration was generated at random, with a very low density (0.02) of ones. (b) A filtered version of (a) that marks with a black dot the presence of the patterns $011=\overrightarrow{13}$ and $110=\overrightarrow{32}$. When two such dots appear next to each other, they become a rigid wall (the pattern 0110) that severs the configuration in two independent portions, forever.

In conclusion, the dynamics of the elementary CA, $|f|=106$, is dominated by the interaction of the traveling domains in $\mathscr{D}(S_1)=0^*1(0^+1)^*\subset\mathscr{L}(S_\nu)$ with interfaces in $\mathscr{T}=1^+$.

2. Elementary CA with two sets of tropons

Next, we examine the elementary CA that have two sets of traveling configurations with nonzero entropy. They are organized in four classes of equivalence[23] in Table II. The class of elementary CA 172 is qualitatively different from the other three classes. It is the only class having a set of standing configurations with nonzero entropy $h=0.551$.

The other three classes in Table II, with minimal representatives 56, 57, and 184, have sets of both left ($\nu=1$) and right ($\nu=-1$) traveling configurations with nonzero entropy. Traveling domains having opposite velocities annihilate between them in a CA configuration, except for degenerate domains. In all three classes the alternating patterns from the regular set $(10)^+$ are degenerate so that, *a fortiori*, it is a surviving pattern, creating a checkerboard background where the few tropons that survive an annihilating transient, either left running or right running, slide endlessly.[2,3]

3. Elementary CA with standing configurations

To exhaust the description of tropons in elementary CA, next we consider the elementary CA that have a set S_0 of standing configurations ($\nu=0$) of nonzero entropy. There are 20 equivalence classes with one such S_0. Only the minimal representative for each equivalence class[23] is entered in Table III. Except for the class of the elementary CA 73, the rest have a very simple behavior: any random configuration ends up in the set S_0. Only elementary CA 73 is worthy of attention. The transducer on $\mathscr{S}(S_0)$ for the elementary CA 73 is shown in Fig. 6. It has characteristic number 122 and entropy $h=0.287$. This entropy value is not large enough as to make of S_0 an invariant set stable under perturbations, as found for the rest of elementary CA in Table III. In Fig. 7(a) we show the first 210 time steps of the space–time pattern for elementary CA 73 with an initial configuration generated at random with a very low density (0.02) of ones. The set of domains $\mathscr{D}(S_0)\subset\{0,1\}^*$ is the language not presenting the words 11^+1, 00^+0, 00100. The domains in elementary CA 73 show a complicated interaction, except for the pattern $0110=\overrightarrow{13}\overrightarrow{32}$ that consists of the coalescence of the patterns $e_1=\overrightarrow{13}$ and $e_2=\overrightarrow{32}$. Figure 7(b) shows the same space–time pattern as in Fig. 7(a), filtered as to show (as black dots) the sites where the patterns e_1 and e_2 appear. When the e_1e_2 pattern is formed, it behaves as a rigid wall that severs the configuration in two independent portions. Whatever happens between two such rigid walls is independent of the rest of the configuration. Figure 7 shows first the appearance of one and then of other two of these rigid walls. The interaction of the cells at both sides of the rigid wall, $\cdots[0110]ba$ and $ab[0110]\cdots$, is given by the rule $b^{t+1}\equiv(b^t+1)+a^t$ (mod 2). Nothing can destroy this wall. Once it appears, it stays forever.

[1] P. Grassberger, ‘‘New mechanism for deterministic diffusion,’’ Phys. Rev. A **28**, 3666 (1984).

[2] J. Krug and H. Spohn, ‘‘Universality classes for deterministic surface growth,’’ Phys. Rev. **38**, 4271 (1988).

[3] N. Boccara, J. Nasser, and M. Roger, ‘‘Particle-like structures and their interactions in spatio-temporal patterns generated by one-dimensional deterministic cellular automaton rules,’’ Phys. Rev. A **44**, 866 (1991).

[4] J. P. Crutchfield and J. E. Hanson, ‘‘Attractor vicinity decay for a cellular automaton,’’ Chaos **3**, 215 (1993).

[5] E. Jen, ‘‘Transience and dislocations in one-dimensional cellular automata,’’ in *Cellular Automata and Cooperative Systems*, edited by N. Boccara *et al.* (Kluwer Academic, London, 1991).

[6] J. E. Hanson and J. P. Crutchfield, ''The attractor-basin portrait of a cellular automaton,'' J. Stat. Phys. **66**, 1415 (1992).

[7] J. P. Crutchfield and J. E. Hanson, ''Turbulent pattern bases for cellular automata,'' Physica D **69**, 279 (1993).

[8] A. Maass, ''On the sofic limit sets of cellular automata,'' Ergod. Th. Dynam. Syst. **15**, 663 (1995).

[9] G. A. Hedlund, ''Endomorphisms and automorphisms of the shift dynamical system,'' Math. Syst. Th. **3**, 320 (1969).

[10] A general approach of hierarchical modeling is introduced and formalized in J. P. Crutchfield, ''The calculi of emergence: computation, dynamics and induction,'' Physica D **75**, 11 (1994).

[11] Y. Aizawa, I. Nishikawa, and K. Kaneko, ''Soliton turbulence in one-dimensional cellular automata,'' Physica D **45**, 307 (1990).

[12] E. M. Coven and M. E. Paul, ''Sofic systems,'' Israel J. Math. **20**, 165 (1975).

[13] H, Gutowitz, ''Cryptography with dynamical systems,'' in *Cellular Automata and Cooperative Systems*, edited by N. Boccara *et al.* (Kluwer Academic, London, 1991).

[14] K. Eloranta, ''Partially permutative cellular automata,'' Nonlinearity **6**, 1009 (1993).

[15] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation* (Addison–Wesley, Reading, MA, 1987).

[16] S. Wolfram, ''Computation theory of cellular automata,'' Commun. Math. Phys. **96**, 15 (1984).

[17] E. Ugalde and J. Urias, ''Symmetry groups of automata,'' Physica D **70**, 178 (1994).

[18] K. Sutner, ''Complex Systems, De Bruijn graphs and linear cellular automata,'' Complex Systems **5**, 19 (1991).

[19] R. Fisher, ''Sofic systems and graphs,'' Monatsh. Math. **80**, 179 (1975).

[20] J. Uri´as, ''Arithmetic representations of cellular automata,'' Physica D **68**, 437 (1993).

[21] B. Weiss, ''Subshifts of finite type and sofic systems,'' Monatsh. Math. **77**, 462 (1973).

[22] M. Mitchell, J. P. Crutchfield, and P. T. Hraber, ''Evolving cellular automata to perform computations: mechanisms and impediments,'' Physica D **75**, 361 (1994).

[23] L. P. Hurd, ''Rule forms and equivalencies,'' in *Theory and Applications of Cellular Automata*, edited by S. Wolfram (World Scientific, Singapore, 1986), pp. 487–492.