Santa Fe Institue Technical Report SFI 93–03–010 To appear in Physica D

Turbulent Pattern Bases for Cellular Automata

James P. Crutchfield
James E. Hanson

Department of Physics
University of California
Berkeley CA 94720 USA*

Abstract

Unpredictable patterns generated by cellular automata (CA) can be decomposed with respect to a turbulent, positive entropy rate pattern basis. The resulting filtered patterns uncover significant structural organization in a CA's dynamics and information processing capabilities. We illustrate the decomposition technique by analyzing a binary, range-2 cellular automaton having two invariant chaotic domains of different complexities and entropies. Once identified, the domains are seen to organize the CA's state space and to dominate its evolution. Starting from the domains' structures, we show how to construct a finite-state transducer that performs nonlinear spatial filtering such that the resulting space-time patterns reveal the domains and the intervening walls and dislocations. To show the statistical consequences of domain detection, we compare the entropy and complexity densities of each domain with the globally averaged quantities. A more graphical comparison uses difference patterns and difference plumes which trace the space-time influence of a single-site perturbation. We also investigate the diversity of walls and particles emanating from the interface between two adjacent domains.

Keywords:
cellular automata, coherent structure, complexity, pattern recognition, turbulence

*Internet addresses: JPC: chaos@gojira.berkeley.edu; JEH: hanson@gojira.berkeley.edu

Contents

List of Figures iii
List of Tables v
Section 1 Pattern Bases, Coherent Structures, and Turbulence 1
Section 2 Raw and Filtered Space-time Data 2
Section 3 Domain Properties 7
3.1 Regular Domains 7
3.2 Pattern Quantifiers 8
3.3 Domain Difference Patterns 11
3.4 Domain Difference Plumes 12
Section 4 Domain Interfaces 15
4.1 Interface Plots 16
4.2 Interface Plumes 18
Section 5 Discussion and Conclusion 21
Section 6 Acknowledgments 23
Appendix A CA Invariant Set Design (with Dan Upper) 24
Appendix B Filter Construction 25
Bibliography 26

List of Figures

Figure 1 Space-time plot of the evolution of binary range-2 CA 2614700074, starting from a random initial condition on a 200 site lattice, over 200 time steps. The spatial configuration is displayed from left to right, with nonzero cells represented as black squares. Time is increasing down the page. 3

Figure 2 Process graphs for the two domains: (a) Λ^0; (b) Λ^1. Nodes or states of the graph are shown as circles. States with inscribed circles and squares denote start and final states, respectively; by definition, all states in a process graph are both start and final states. The number inscribed in each state gives the phase of the underlying pattern when the machine is in that state. An edge labelled with the symbol Σ is shorthand for a pair of edges with labels $\{0, 1\}$. 4

Figure 3 (a) Space-time plot of the evolution of binary range-2 CA 2614700074, starting from a random initial condition on a 100 site lattice over 100 time steps. (b) Filtered space-time plot of the same data shown in (a). Space-time regions in domain Λ^0 are shown in white, regions in Λ^1 are gray, and all walls are shown in black. All four varieties of wall are present. This figure was generated using the transducer of Figure 4. . . 6

Figure 4 Domain transducer. Cells in domain Λ^0 are mapped onto 0's, cells in domain Λ^1 are mapped onto 1's, and walls Γ^{ij} are mapped onto symbol $(2i + j + 2)$. States in the transducer corresponding to cells in a domain are labelled with that domain. Edges are labeled $\sigma_{in}|\sigma_{out}$ for input and output symbols, respectively. For the first few symbols read, the transducer is synchronizing to the data stream, and no output is generated; this is indicated by the null symbol λ. The edges shown in bold correspond to cells within a domain. 7

Figure 5 Difference patterns $\mathbf{\Delta}_t^{\mathcal{L}}$: (a) $\mathcal{L} = \mathcal{A}^N$. Global difference pattern in which the initial condition was randomly selected over the entire state space: $\mathrm{s}_0 \in \mathcal{A}^N$; (b) random IC in domain $\mathcal{L} = \Lambda^0$; (c) random IC in domain $\mathcal{L} = \Lambda^1$. In each plot, evolution was on a lattice of size $N = 100$ and the initial condition was perturbed in cell $j = 50$. The patterns of (b) and (c) are identical, and are also identical to an "expanded" space-time diagram of elementary CA rule 90 in which a 0 is inserted between every horizontal pair of cells. 13

Figure 6 Difference plumes $\langle \Delta_t^i \rangle$: (a) $\mathcal{L} = \mathcal{A}^N$. Global difference plume averaged over initial conditions randomly selected over the entire state space; (b) initial conditions in domain $\mathcal{L} = \Lambda^0$; (c) initial conditions in domain $\mathcal{L} = \Lambda^1$. Note that while (a) appears to be a smooth surface, both (b) and (c) show structure. In fact, (b) and (c) are identical to the patterns of Figure 5(b) and 5(c). For each plot, $M = 1000$ pairs of states were iterated on a lattice of size $N = 100$. As above, each initial condition was perturbed in cell $j = 50$. 14

Figure 7 Difference plume volumes $\zeta_{\mathcal{L}}(t)$ versus t for the entire state space $\mathcal{L} = \mathcal{A}^*$ (dashed line) and for domain $\mathcal{L} = \Lambda^0$ (solid line). The plume volume for the other domain, $\mathcal{L} = \Lambda^1$, is identical to $\zeta_{\Lambda^0}(t)$. 15

Figure 8 Evolution of domain interfaces. Each state starts out with a single wall at cell $j = 500$. The initial conditions were chosen randomly from the following ensembles: (a) $\Lambda^{0,0}$ with relative phase $\delta\phi^{00} = 1$; (b) $\Lambda^{0,1}$ with $\delta\phi^{01} = 0$; (c) $\Lambda^{1,0}$ with $\delta\phi^{10} = 0$; (d) $\Lambda^{1,0}$ with $\delta\phi^{10} = 1$; (e) $\Lambda^{1,1}$ with $\delta\phi^{11} = 1$; and (f) $\Lambda^{1,1}$ with $\delta\phi^{11} = -1$. Cells in the interior of Λ^0 are shown in white, cells in Λ^1 are gray, and all walls are black. 17

Figure 9 Interface plumes for ensembles of the six types of state shown in Figure 8. The labels (a) — (f) denote the same ensembles as in that Figure. 19

Figure 10 Interface plume volumes $\xi_{\mathcal{L}}(t)$ versus t. The labels (a) — (f) denote the same ensembles as in Figures 8 and 9. 21

List of Tables

Table 1 Quantifiers for the two domains Λ^0 and Λ^1, and for the full state space $\mathcal{A}^*$. See the text for definitions of the quantifiers. Fractional numbers and integers without decimal points are exact quantities. All others were estimated numerically, and have an uncertainty of ± 1 in the last decimal place. Numbers marked with the symbol (†) are nonstationary quantities, and were measured on a lattice of size $N = 1 \times 10^6$ at time $t = 200$. . 10

1 Pattern Bases, Coherent Structures, and Turbulence

One of the fundamental properties of any spatially-extended dynamical system is the variety of spatial configurations or patterns that arise during its evolution. Accordingly, the analysis of such systems usually starts with the choice of a representation that fits these patterns—in other words, a *pattern basis*. Once chosen, the pattern basis forms the underlying structure in terms of which the system is understood. Expansion in Fourier modes, the common choice of pattern basis for continuum systems, implicitly assumes that harmonic waves are appropriate for representing the system's spatial configurations. In many instances, this assumption is justified; for example, in Rayleigh-Bénard convection when the stationary state (heat conduction) destabilizes, the fluid motion organizes itself into spatially periodic roll patterns of a single (spatial) frequency. Representing the system in terms of its Fourier modes, therefore, projects the infinite-dimensional state space down onto a space of two dimensions: the amplitude and phase of the rolls. The small nonlinearities in the evolution then show up as modulations of the amplitude, which are governed by an "envelope" equation. Points at which the periodic pattern breaks down are "dislocations", and can be treated as quasiparticles with their own equations of motion.[1,2]

If the system is turbulent, however, the periodic representation is less appropriate. The spatial Fourier spectrum typically exhibits a continuum of excited modes. The diversity of observed configurations is so great that one is forced into the position of disregarding the details of the evolution in order to adopt some sort of stochastic model, such as in the case of fully-developed "hard" turbulence.[3] The problem with these models is that they virtually ignore the possibility of *any* regularity existing in the system, effectively masking it with purely stochastic dynamics. It is by now well-appreciated that for chaotic systems, this modeling procedure throws away important structural information; in particular, it fails to capture the possibility that coherent structures may be present in the system. What is needed instead is a *turbulent pattern basis*: a way of representing spatial configurations that captures the structural properties of the dynamics—e.g. via an appropriate notion of pattern—but that still allows for sufficient stochasticity that it is not burdened with describing the unpredictable details of high-entropy-rate "turbulent" behavior. In this way, coherent structures, if they exist, would show up as points at which the *turbulent* pattern breaks down. The system's evolution can then be decomposed into a turbulent background and coherent structures evolving against it. Recent work on two-dimensional turbulence[4,5] has shown how effective this type of decomposition can be.

The same situation appears in a much simpler class of spatial system, cellular automata (CA). Many CA evolve to patterns in which there is an easily-recognized spatio-temporally periodic background on which "particles" or "dislocations" move. In the same way as for the convection example above, the "particles" are defined *in terms of* the background pattern: they are points at which the pattern breaks down. Two relatively well known examples of this, for which the particles have been studied, are elementary CA rules 54 and 110.[6,7] But in chaotic CA, no periodic background pattern exists. If there is any regularity in the spatial configurations, it is hidden in the "turbulent" behavior of the CA. Again, what is needed is a turbulent pattern basis for the system, in terms of which its behavior is to be analyzed.

In Ref. [8] we introduced the notion of a *regular domain* as an organizing principle for the analysis of discrete spatially-extended dynamical systems, in particular for one-dimensional

CA, though it is also appropriate to other systems such as map lattices.[9] A regular domain is a regular language—i.e. a collection of symbol strings[10]—with the properties, defined more precisely below, of temporal invariance and spatial homogeneity. Regular domains can be spatially or temporally chaotic, or both, yet they can still capture the structural regularities that may exist. They give a precise notion of "spatial pattern" and the complexity of their patterns is determined by the size of the minimal stochastic automaton that describes all allowed domain configurations. It is these domains that serve as the turbulent pattern bases for cellular automata. For more information on regular domains in cellular automata, the reader is referred to Refs. [8] and [11].

In this article, we examine a CA having two distinct domains, each of which is chaotic, and in doing so we present a variety of generally applicable analytical and numerical techniques. Having reconstructively identified the domains from the space-time plot of the CA's evolution, we construct a filter for decomposing arbitrary spatial configurations into domains and intervening walls or coherent structures. We then analyze the structural and statistical properties of the domains and contrast them with those of the global state space. Finally, we briefly investigate the behavior of the coherent structures by tracing the evolution of inter-domain interfaces. Taken together, the techniques we develop represent a fundamental set of tools for analyzing the qualitative pattern dynamics of the system.

The domains have different spatial signatures, different entropy densities, and different complexities, yet they coexist on a single lattice. In analogy with statistical mechanics, we regard these domains as two distinct "thermodynamic phases" of the spatial dynamical system. Among CA, the phenomenon of multiple distinct domains appears to be a very common one. The methods we develop, and the conclusions we draw concerning the importance of domain detection for this one example, apply generally to a wide class of CA.

The CA is a binary-alphabet, range-2 CA with rule number 2614700074 in the conventional numbering scheme.[12] The rule was selected because it has two domains, both high-entropy "turbulent" processes, but with different internal structure. (See Appendix A.) Nevertheless, our formalism enables us to identify, separate and analyze them. This reveals more structural information there than would otherwise be apparent, especially when compared to what is afforded by the global view.

2 Raw and Filtered Space-time Data

To fix notation, we briefly recall the definition of a cellular automaton. The state of the CA at time t, denoted s_t, consists of a one-dimensional array or lattice of cells σ_t^i with values chosen from a finite alphabet $\mathcal{A} = \{0, 1, ..., k-1\}$. The local site-update function operating on neighborhoods of radius r is written $\sigma_{t+1}^i = \phi\left(\sigma_t^{i-r}, ..., \sigma_t^{i+r}\right)$. On a lattice of N cells, $\mathrm{s}_t \in \mathcal{A}^N$. The global update operator $\Phi : \mathcal{A}^N \to \mathcal{A}^N$ applies ϕ in parallel to all neighborhoods in the lattice: $\mathrm{s}_{t+1} = \Phi(\mathrm{s}_t)$. For finite N, it is also necessary to specify the behavior at the edges of the array; in the following, we will always use periodic boundary conditions.

A space-time plot of the CA we are concerned with, binary range–two ($k = 2$, $r = 2$) rule 2614700074, is shown in Figure 1. The evolution is down the page, starting with a random

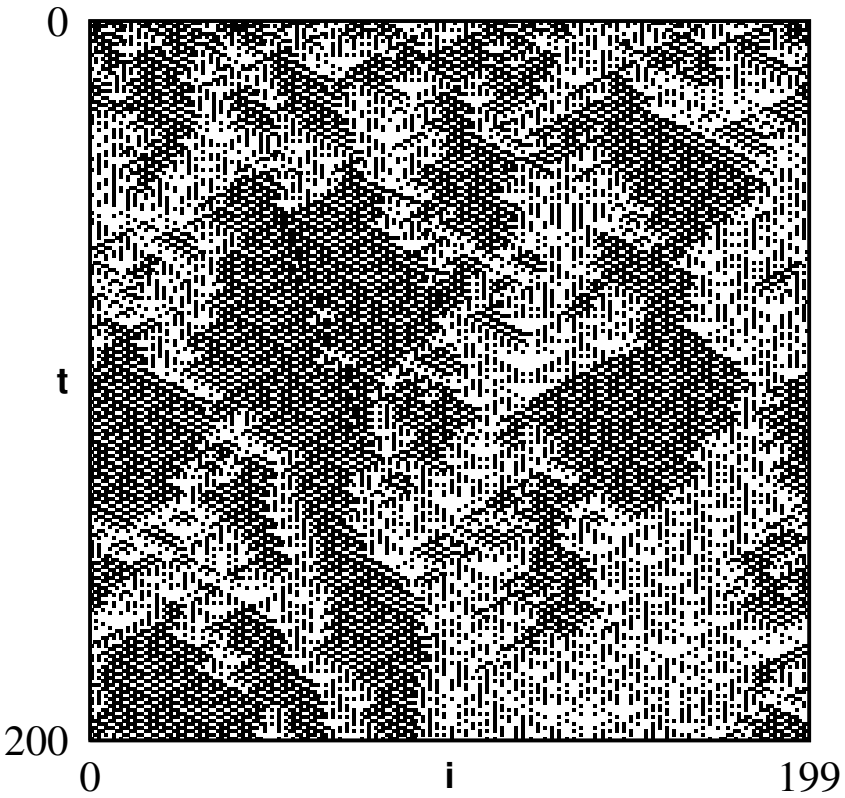

Figure 1 Space-time plot of the evolution of binary range-2 CA 2614700074, starting from a random initial condition on a 200 site lattice, over 200 time steps. The spatial configuration is displayed from left to right, with nonzero cells represented as black squares. Time is increasing down the page.

initial condition on a lattice of $N = 200$ cells with periodic boundary conditions. Cells with $\sigma_t^i = 0$ are white; $\sigma_t^i = 1$ are shown in black.

This sequence of spatial configurations was then fed as input into the machine reconstruction algorithm[13,14]. Machine reconstruction inductively constructs explicit models of the patterns that are present in the data. It acts as a substitute for visually identifying the patterns, and hence provides a method of discovering patterns that are too complex to be discovered by eye.

In this case, it is quite easy to visually identify space-time regions of two different types: one type of region is mostly white, in which there is a greater percentage of cells with $\sigma_t^i = 0$ than with $\sigma_t^i = 1$, while the other is predominantly black. Applying machine reconstruction to this data then serves to corroborate the visual evidence and to construct exact representations of the actual patterns present—and, incidentally, to illustrate the effectiveness of the reconstruction technique. But it is important to stress that machine reconstruction is an *alternative* to visual pattern-recognition, designed to operate on data for which visual identification is not effective.

Taking spatial sequences from each of the two types of space-time region in turn and applying machine reconstruction to them, we found that the first type of region consists of a tiling of the spatial pattern (0Σ), and the second consists of a tiling of (110Σ). Here, $\Sigma = \{0, 1\}$ denotes a "wild-card" cell that may be either 0 or 1. It is the appearance of the wild-card that indicates these patterns are unpredictable and chaotic. Using the notation $(s)^*$ to mean an arbitrary number of concatenations of the string s and $\mathrm{sub}\mathcal{L}$ to indicate the set of all subwords of a language $\mathcal{L}$, we identify the two regions with the regular languages $\Lambda^0 \equiv \mathrm{sub}(0\Sigma)^*$ and $\Lambda^1 \equiv \mathrm{sub}(110\Sigma)^*$.

As we show in the next section, Λ^0 and Λ^1 are in fact regular domains. Each domain language represents the set of all spatial configurations with the given pattern. Depending on context, we will use the word "domain" to mean the domain language itself, or any single configuration in that language.

The process graphs for the domain languages Λ^0 and Λ^1 are shown in Figure 2. The circles represent the nodes or states of the graph. A symbol in $\mathcal{A}$ is assigned to each edge connecting two states; an edge labelled with the symbol Σ is shorthand for a pair of parallel edges with symbols 0 and 1. States with inscribed circles or squares denote legal initial or final states, respectively. The graphs are interpreted as follows: a word in a language $\mathcal{L}$ is generated from its process graph $\mathrm{M}(\mathcal{L})$ by starting in any state of $\mathrm{M}(\mathcal{L})$ and taking any path along the graph's edges, reading off the edge labels as they are traversed. Each path on the graph corresponds to a single word in $\mathcal{L}$, although there can be several distinct paths generating the same word. The indices inscribed in the states of $\mathrm{M}\big(\Lambda^j\big)$ are the values of the domain's *phase* ϕ^j. A symbol σ^i in a word $\omega = ...\sigma^{i-1}\sigma^i\sigma^{i+1}... \in \Lambda^j$ has phase n if $\mathrm{M}\big(\Lambda^j\big)$ is in the state with index n after reading σ^i; this is written

$$\phi^j(i) = n \tag{1}$$

By convention, we have chosen the $\phi^j = 0$ state of each domain Λ^j to be the state after the edge labelled with the wild-card Σ.

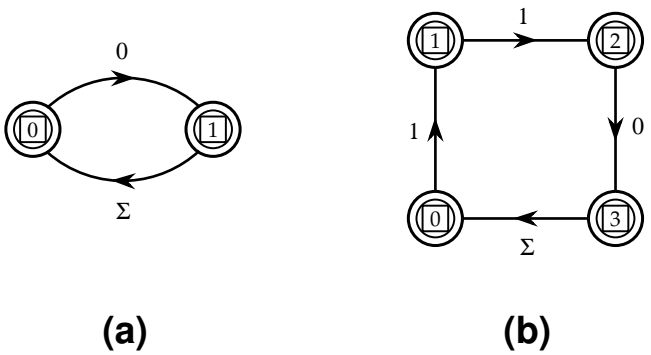

Figure 2 Process graphs for the two domains: (a) Λ^0; (b) Λ^1. Nodes or states of the graph are shown as circles. States with inscribed circles and squares denote start and final states, respectively; by definition, all states in a process graph are both start and final states. The number inscribed in each state gives the phase of the underlying pattern when the machine is in that state. An edge labelled with the symbol Σ is shorthand for a pair of edges with labels $\{0, 1\}$.

Once the domains have been identified, the next step is to find the domain walls, which are the boundaries between adjacent domains (see appendix B for the exact definition. From the two domains, we constructed a spatial filter T_Λ that maps arbitrary spatial configurations $\mathrm{s} \in \mathcal{A}^*$ onto the sequence of domains and walls the string contains. Each cell σ^i of s is classified according to whether it is in a given domain or is a wall: cells in subsequences $\omega \in \Lambda^0$ are mapped to the symbol 0, cells in subsequences $\omega \in \Lambda^1$ to the symbol 1, and each wall between domains is mapped to another symbol whose value depends on the domains to either side. At each time-step t, T_Λ is run over the spatial configuration s_t from left to right, and the output string $\mathrm{T}_\Lambda(\mathrm{s}_t)$ is plotted. Using this filter on the space-time data shown in Figure 3(a), we generated

the plot shown in Figure 3(b). In the latter, cells in Λ^0 are represented as white squares, cells in Λ^1 are gray squares, and all walls are black. Comparison of Figures 3(a) and 3(b) shows that the filtered pattern highlights the domains and walls present in the spatial configurations. The walls between different domains show up clearly, and are more precisely pinpointed than by eye in the original space-time data. Dislocations, which are walls separating domains of the same type, are made evident.

The filtered pattern represents a significant simplification of the apparent dynamics of CA rule 2614700074. The domains are temporally invariant, reducing the essential part of the temporal evolution to tracing the trajectories of the various walls. Additionally, each pattern is seen to be the concatenation of subpatterns from the two domains.

We traced the evolution of the initial condition shown in Figure 3 beyond time $t = 100$. Both types of domains were present up until $t = 700$, at which time the state as a whole fell into domain Λ^1 and remained there. In a survey of several dozen other initial conditions on a variety of lattice sizes, all of the states eventually fell into one or the other domain, with approximately 80% falling into Λ^0. This is a preliminary indication that Λ^0 and Λ^1 may be regular attractors (defined in Ref. [8]), with Λ^0 having the larger basin measure. In any case, it is abundantly clear that the Λ^0 and Λ^1 are the fundamental structures organizing the high-dimensional state space.

The filter T_Λ we constructed is shown in Figure 4. It is a finite-state transducer that takes as input any sequence $\mathrm{s} \in \mathcal{A}^*$ and parses it into the sequence of domain and wall labels. In this way, it recognizes the structures implicit in the spatial configuration s. There are three different types of state shown in Figure 4. States without inscribed labels are synchronization states, devoted to processing the first few symbols in s. The machine starts in the state with the double circle at the top of the Figure, then moves along edges according to the input symbols. The first few symbols read serve to synchronize the machine to the underlying pattern, driving it from the start state of total ignorance into one of the asymptotic states. The edges are labelled $\sigma_{in}|\sigma_{out}$ for input and output symbols, respectively; however, during synchronization the machine produces no output, which is signified by putting $\sigma_{out} = \lambda$. Besides the synchronization states, there are two types of asymptotic state, one for each domain. States labelled Λ^0 or Λ^1 correspond to states visited when the input cell is in domain Λ^0 or Λ^1, respectively. Having synchronized to the pattern data stream, the transducer is in the asymptotic part of the graph, where it remains for the rest of its operation.

During synchronization, the transducer has yet to read in a sufficient amount of information to unambiguously distinguish the two domains and the several wall types. Once the machine has synchronized, every cell read is either in a domain or is a wall. Edges corresponding to cells in either domain are printed in bold lines, and have output symbols identifying the domain. Walls are printed in lighter lines, and have output symbols 2, 3, 4, or 5. These symbols denote the four types of wall, corresponding to the four possible combinations of domains on either side of the wall. Denoting a wall between domain Λ^i and Λ^j by Γ^{ij}, the four possible walls between domains Λ^0 and Λ^1 are Γ^{00}, Γ^{01}, Γ^{10}, and Γ^{11}. The complete domain transducer shown in Figure 4 assigns a unique symbol to each type of wall; in Figure 3(b), however, we have plotted all four types of walls as black squares. In making Figure 3(b), the transducer traversed the lattice from left to right, using the periodic boundary conditions to wrap around at the end in order

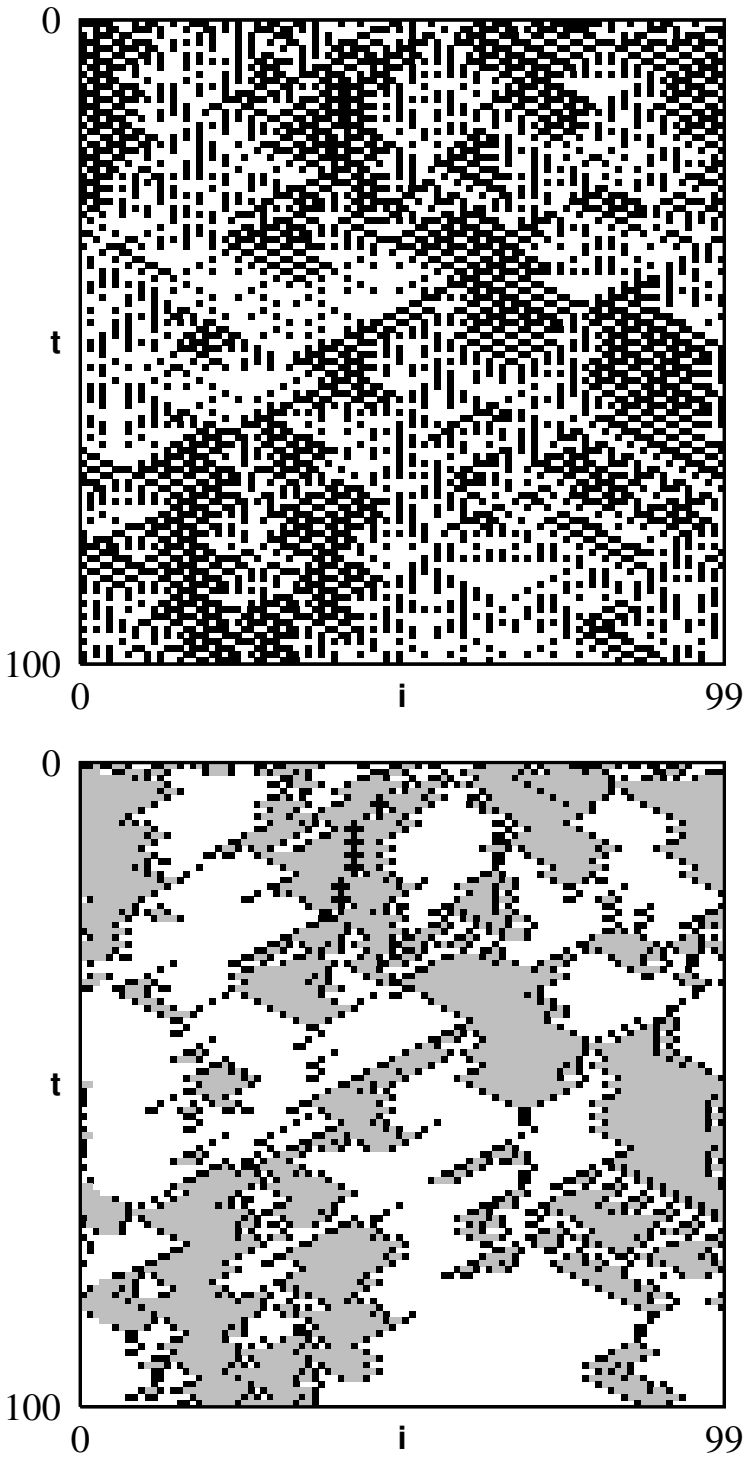

Figure 3 (a) Space-time plot of the evolution of binary range-2 CA 2614700074, starting from a random initial condition on a 100 site lattice over 100 time steps. (b) Filtered space-time plot of the same data shown in (a). Space-time regions in domain Λ^0 are shown in white, regions in Λ^1 are gray, and all walls are shown in black. All four varieties of wall are present. This figure was generated using the transducer of Figure 4.

to classify the first few cells, i.e., the cells used for synchronization. Except for the 0^* pattern, each cell in the lattice can be classified in this way with at most two passes over the data.

This machine T_Λ was built using a general construction procedure whereby, from any set of domain languages $\left\{\Lambda^0, \Lambda^1, ...\Lambda^{m-1}\right\}$, a minimal transducer is constructed that maps each cell in domain Λ^i to symbol i, and maps each cell on wall Γ^{ij} to symbol $(m + mi + j)$. (See Appendix B for more details on the construction.) Thus for a set of m domains, there are $m(m + 2)$ symbols in the transducer's output alphabet.

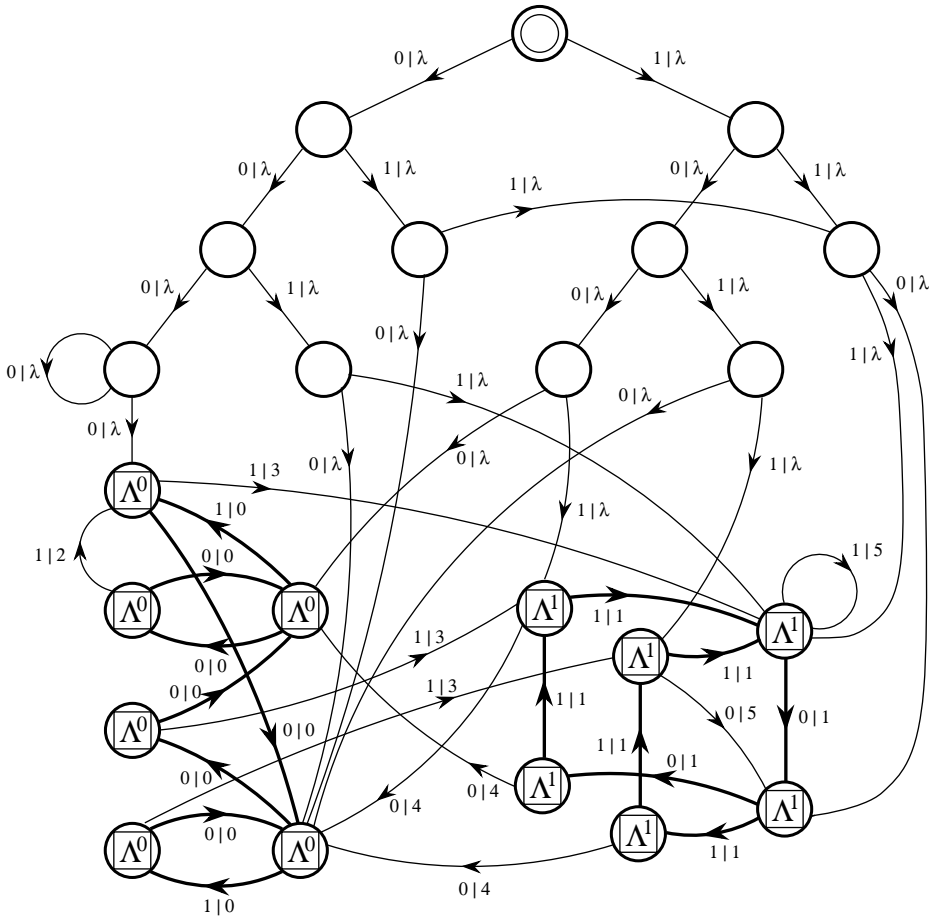

Figure 4 Domain transducer. Cells in domain Λ^0 are mapped onto 0's, cells in domain Λ^1 are mapped onto 1's, and walls Γ^{ij} are mapped onto symbol $(2i + j + 2)$. States in the transducer corresponding to cells in a domain are labelled with that domain. Edges are labeled $\sigma_{in}|\sigma_{out}$ for input and output symbols, respectively. For the first few symbols read, the transducer is synchronizing to the data stream, and no output is generated; this is indicated by the null symbol λ. The edges shown in bold correspond to cells within a domain.

3 Domain Properties

3.1 Regular Domains

Having reconstructed the machines for the two regular languages Λ^0 and Λ^1 from the observed space-time data, we must now prove that they are domains. As mentioned in the introduction, this entails showing that each language satisfies two requirements: temporal invariance and spatial homogeneity. For more details on the notion of domain used here, including the formal definition of regular domain, see Ref. [8].