This paper appeared in *J. Stat. Phys.*, 66:1415, 1992.

The Attractor-Basin Portrait of a Cellular Automaton

James E. Hanson
James P. Crutchfield

Department of Physics
University of California
Berkeley CA 94720 USA*

Abstract

Local space-time structures, such as domains and the intervening dislocations, dominate a wide class of cellular automaton (CA) behavior. For such spatially-extended dynamics regular domains, vicinities, and attractors are introduced as organizing principles to identify the discretized analogs of attractors, basins, and separatrices: structures used in classifying dissipative continuous-state dynamical systems. We describe the attractor-basin portrait of nonlinear elementary CA rule 18, whose global dynamics is largely determined by a single regular attracting domain. The latter's basin is analyzed in terms of subbasin and portal structures associated with particle annihilation. The conclusion is that the computational complexity of such CA is more apparent than real. Transducer machines are constructed that automatically identify domain and dislocation structures in space-time, count the number of dislocations in a spatial pattern, and implement an isomorphism between rule 18 and rule 90. We use a transducer to trace dislocation trajectories, and confirm that in rule 18, isolated dislocation trajectories, as well as a dislocation gas, agree extremely well with the classical model of annihilating diffusive particles. The CA efficiently transforms randomness of an initial pattern ensemble into a random walk of dislocations in space-time.

*JPC's Internet address is chaos@gojira.berkeley.edu; JEH's is hanson@gojira.berkeley.edu.

Contents

Section 1	Introduction	1
Section 2	Definitions	2
2.1	Elementary CA and Rule 18	2
2.2	Finite Automata and Regular Languages	3
2.3	Transducers	5
2.4	CA as Regular Language Processors	7
Section 3	Structures in Language Space	9
3.1	Domains	10
3.3	Vicinities	13
3.4	Attractors	16
3.5	Other Structures	21
Section 4	Global Structure of Rule 18	22
4.1	Transducers for Global Structure Analysis	24
4.2	Basins and Separatrix Structure	26
4.3	The Attractor-Basin Portrait	27
Section 5	Dislocation Trajectories	29
5.1	Detection	30
5.2	Diffusion	32
5.3	Computing with Dislocations	37
Section 6	Concluding Remarks	39
Section 7	Acknowledgments	41
Appendix A	Dislocation Decay Statistics	42
Bibliography		43

1 Introduction

Cellular automata (CA) are models of spatially extended dynamical systems with the particular feature of being as discrete as possible: in space (cells), in site value (e.g. binary), and in time.[1] The discrete nature of CA makes the techniques of formal language and computation theories, which deal with strings of discrete symbols, particularly appropriate tools for their analysis. Computational and formal language properties of CA global behavior have been investigated during the last two decades.[2–5] Discussions of computational capability have often centered on constructing specialized cellular automata to mimic features of serial computation models: Turing machines embedded in, for example, Conway's Game of Life,[6] or digital circuitry implicit in logic gate, billiard computer, and particle flow simulations.[7] Such constructions, however, fail to address the basic question of how to identify the generic computational properties of any given CA.

Similarly, analyses of CA in the context of dynamical systems theory have tended toward specialized results, focusing, for example, on very small lattices and explicit or Monte Carlo enumeration of periodic orbits and their transients.[8–11] Nonetheless, the last decade has witnessed a substantial increase in our appreciation of the diversity of CA behavior. There have been extensive phenomenological studies, including classification,[12–14] identification of propagating structures,[15,16] and measurement of bulk statistical properties.[1,17–19] These unfortunately have taken advantage of neither the rich potential implicit in the well-developed theory of computation nor the geometric underpinnings of dynamical systems theory.

The following attempts a synthesis of these two approaches by considering a view of CA dynamics based on the local space-time and state-space structures that organize the system's evolution. In doing so, it introduces a rather literal analysis of CA as dynamical systems. For a range of related CA one can identify the analogs of attractors, basins and separatrix structures.[20,21] These ideas will be illustrated by focusing on a particular elementary CA, rule 18 in the conventional numbering scheme. Although our discussion is restricted to this one example, it serves to illustrate a number of properties of CA generally, but in a simple setting. A more detailed development and more complete survey of CA will appear elsewhere.[22]

The basic phenomenology of the CA we study here has been known for some time.[17,23] It organizes itself into local regions in which every other cell in space and time is 0 and the remaining cells are unrestricted. The boundaries between adjacent "phase-locked" regions are dislocations, where the spatial phase slips a cell. When evolved from a random initial condition consisting of a rarified dislocation gas, numerical results indicate that the number density d of dislocations decays in time as $d \approx (8\pi Dt)^{-1/2}$, with $D \approx 1/2$. Furthermore, the excursion x of a set of initially distant dislocations grows as $\langle x^2 \rangle \approx 2Dt$. The interpretation of this is that the dislocations move as if following a space-time random walk with diffusion constant D,

annihilating in pairs when their paths cross.[24] In short, starting from random configurations the dislocations behave like a diffusive gas of annihilating particles.

The focus in the present work is on the computational analysis of this behavior, on a reconstructive method for identifying domain structure, and on the implications of these in treating CA as dynamical systems. The techniques introduced below for detecting, analyzing and manipulating general structural elements such as domains and dislocations point the way toward a generally applicable computation-analytical tool.

The presentation is organized as follows. The next section gives definitions and introduces notation for elementary CA, finite automata and regular languages, and transducers. We define an operator governing the time evolution of ensembles of CA states, and give an algorithm for calculating the iterates of such ensembles. The following section addresses the central issue of space-time structures, giving definitions and applying them to rule 18, which serves as an illustrative example throughout. Next we describe the global structure of rule 18, and give its attractor-basin portrait. Finally, as an application of the ideas introduced, we examine the statistical properties of dislocation trajectories.

2 Definitions

2.1 Elementary CA and Rule 18

Definition.[12] A **cellular automaton** consists of a countable array of discrete sites or cells i and a discrete-time update rule ϕ operating in parallel on local neighborhoods of a given radius r. At each time the sites take on values in a finite alphabet $\mathcal{A}$ of primitive symbols: $\sigma_t^i \in \{0, 1, ...k-1\} \equiv \mathcal{A}$. The local site-update function is written

$$\sigma_{t+1}^i = \phi\big(\sigma_t^{i-r}, \ldots, \sigma_t^{i+r}\big)$$

The **state** s_t of the CA at time t is the configuration of the finite or infinite spatial array: $\mathrm{s}_t \in \mathcal{A}^N$, where $\mathcal{A}^N$ is the set of all possible site value configurations on a lattice of N cells. The "extended state space", denoted $\mathcal{A}^*$, is the union of all states of any N:

$$\mathcal{A}^* = \bigcup_{N \geq 0} \mathcal{A}^N$$

with $\mathcal{A}^0 = \emptyset$.

The CA global **update rule** $\Phi : \mathcal{A}^N \to \mathcal{A}^N$ applies ϕ in parallel to all sites in the lattice: $\mathrm{s}_t = \Phi \mathrm{s}_{t-1}$. For finite N it is also necessary to specify a boundary condition. In the following, we will always use periodic boundary conditions.

The automaton we are concerned with is an "elementary" CA for which $(k, r) = (2, 1)$; that is, nearest-neighbor interactions and binary cell values, giving a total of $k^{2r+1} = 8$ possible

neighborhood patterns. Following the conventional numbering scheme,[12] we arrange these patterns in the order below, write the value obtained by applying ϕ to each pattern, and interpret the resulting string of 8 symbols as a binary number giving the rule index. This paper analyzes the CA having rule index 18, for which the rule table is as follows:

Neighborhood	$\underline{111}$	$\underline{110}$	$\underline{101}$	$\underline{100}$	$\underline{011}$	$\underline{010}$	$\underline{001}$	$\underline{000}$	$\underline{\sigma_t^{i-1}\sigma_t^i \quad \sigma_t^{i+1}}$
Next site value	0	0	0	1	0	0	1	0	σ_{t+1}^i

As is readily seen from the rule table this CA is dissipative in the sense that the global update rule is many-to-one: $\Phi\mathcal{A}^N \subset \mathcal{A}^N$.

It will turn out that for finite periodic lattices, there are a few values of N for which the evolution of the CA is radically different. Since we are presently interested in the *typical* behavior of a given CA, we define the notion of a **sparse set** of lattice sizes, on which the behavior is allowed to be atypical, but which are themselves so rare as to be insignificant. This allows us to investigate the qualitative dynamics of the attractor-basin portrait without worrying about complications introduced by rare number-theoretic fluctuations in the behavior. Such fluctuations are an interesting subject in themselves, but do not concern us here.

Definition. Given a set $R = \{N_1, N_2, ...\}$ of distinct CA lattice sizes, let $I_R(n)$ be the number of elements $N_i \in R$ such that $N_i \leq n$. Then R is said to be **sparse** if and only if

$$\lim_{n\to\infty} \frac{I_R(n)}{n} = 0\,.$$

If a property P holds for all but a sparse set of lattice sizes, we will say that P holds for "**almost all**" lattices, or equivalently, for "**typical**" lattices.

2.2 Finite Automata and Regular Languages

In this section, we briefly review the basic definitions of automata and languages we will need. For a thorough presentation of this topic, see any standard text on the subject.[25] It is important not to confuse the automata defined here, which represent ensembles of symbol strings, with cellular automata, which are the dynamical systems we are studying.

Definition. An **automaton**, or "machine", is formally defined as a quintuple $\mathrm{M} = \{Q, \Sigma, \delta, q_0, F\}$. Here Q is a set of states; Σ is an alphabet of tape symbols; $\delta : Q \times \Sigma \to Q$ is the transition rule, taking the current state and a symbol read from a tape into a new state; q_0 is the initial state; and F is the set of final or accepting states. If the set Q of states is finite, $\|Q\| < \infty$, then M is a **finite automaton**.

Finite automata can be **deterministic** or **nondeterministic** depending on whether for all states $q \in Q$, each transition from q to another state is labeled with a unique tape symbol. These

two automaton classes are abbreviated DFA and NFA, respectively. An automaton is said to **recognize** a string if, upon reading it from the tape, no disallowed transitions are seen. An automaton **accepts** a string if, after recognizing it, the automaton is in one of the accepting states. Strings that are not accepted are **rejected**. Two finite automata are **equivalent** if they accept identical sets of strings. Two automata are **disjoint** if they accept disjoint sets of strings.

A formal language L is a set of strings over some alphabet Σ: $L \subseteq \Sigma^* = \bigcup_{i \geq 0} \Sigma^i$. The language $L(\mathrm{M})$ of a machine M is the set of all strings, sometimes called words, accepted by that machine. This definition is independent of whether or not M is finite.

Definition. A **regular language** is that $L(\mathrm{M})$ accepted by some finite automaton M. We define **regular language space**, or just "language space", as the set $\mathcal{R}$ of all regular languages.

For each regular language L there is a unique finite automaton $\mathrm{M}_{\min}$ with the minimal number $\|Q\|$ of states that accepts only and all of the strings in L. Since we will be focusing almost exclusively on minimal machines, we drop the subscript and simply write M, with the understanding that we mean $\mathrm{M}_{\min}$ unless it is explicitly stated otherwise. Consistent with this we denote by $\mathrm{M}(L)$ the minimal DFA accepting L. This allows us to use finite automaton and regular language representations interchangeably.

An important subclass of the regular languages we will be studying is the class of process languages. That is, for all words $\omega \in L$, all subwords $\nu \in \left\{ s^i \ldots s^{i+j} : s^k = (\omega)_k, i + j \leq \|\omega\|, i, j \geq 0 \right\}$ of ω are in L. We form the subword closure sub(L) of a language L by including in sub(L) all subwords of each $\omega \in L$. Finite automata for regular process languages have the property that all states are accepting: $F = Q$.[26]

Definition. The **process graph** of a process language L is the graph formed by allowing the associated machine $\mathrm{M}(L)$ to start in any state, and minimizing the result.

The process graph is the "state transition graph" of ref. [4]. It clarifies the structure of the language by absorbing the transient states devoted to "phase-locking" of individual strings to the machine's recurrent states.[27]

Of the many measures of infinite languages, the one we will find most useful is the **size**, defined below.

Definition. Let $N_L(n)$ be the number of strings of length n in a language L. The **relative size** of a language L_1 with respect to another language L_2 such that $L_1 \subseteq L_2$ is defined as

$$S(L_1|L_2) \equiv \lim_{n \to \infty} \frac{N_{L_1}(n)}{N_{L_2}(n)}$$

$S(L_1|L_2)$ is just the fraction of strings in L_2 that are also in L_1, in the limit of large n.

2.3 Transducers

The automata defined above can be used either as input devices for classifying existing strings or as output devices that generate strings belonging to a given language. These two modes of operation are easily combined by modifying the machine so that on each transition a symbol is emitted as well as read. The resulting "transducer" machines, also called Mealy machines,[25] are functions from strings to strings. In the information theory literature the process versions are referred to as channels.[28] In ergodic theory, they are endomorphisms of subshifts.[29] Transducers give a very compact notation for functions that map complicated sets, e.g. Cantor sets, to complicated sets. Of equal importance, they are constructive in the sense that they indicate literally how the function is to be implemented and so computed.

In this section we formally define transducers and develop a few basic tools for manipulating them. Transducers will turn out to be central to the analysis of structural properties of CA state space.

Definition. A **transducer** $\mathrm{T} = \{Q, \Sigma_{in}, \delta, q_0, F, \Sigma_{out}, \beta\}$ is an automaton where Q, δ, q_0, F are as defined above and where Σ_{in} is the previous input alphabet Σ. Σ_{out} is an output alphabet. The observation function β maps $Q \times \Sigma_{in}$ to Σ_{out}. A transducer operates as a finite automaton with the extra feature of writing a symbol in Σ_{out} to an output tape upon reading a symbol in Σ_{in} from the input tape.

The effect of running a transducer over a space-time pattern, say, is to recode the data by replacing original symbols with those emitted by the machine. Formally, the transducer induces a function f_{T} that maps strings over Σ_{in} to strings over Σ_{out}.

Definition. Let T be a transducer. The **output machine**, which we denote $[\mathrm{T}]_{\mathrm{out}}$, is the minimal DFA accepting the set of all strings that can be emitted from T, on all possible inputs. $[\mathrm{T}]_{\mathrm{out}}$ can be constructed by the following series of steps:

i. The input symbols on all transitions of T are disregarded, leaving only the output symbols labeling transitions.
ii. The resulting NFA is converted to a DFA using the standard algorithm.[25]
iii. The DFA is then minimized to obtain the unique machine with the minimal number $\|Q\|$ of states. This operation is also standard.[25]

We call this construction **minimization with respect to the output**.

Given a machine $\mathrm{M}(L)$ and a transducer T, one can define the composition operator $\mathrm{T} \circ \mathrm{M}(L)$ such that the resulting machine, itself a transducer, accepts as input exactly the set of strings in L, and for each such input string $s \in L$, it emits the encoded string $f_{\mathrm{T}}(s)$. In effect, $\mathrm{M}(L)$ and T are connected in series, with $\mathrm{M}(L)$ emitting strings in L which are then used as inputs to T. This corresponds to restricting the domain of f_{T} to the language L. The operator is formally expressed in the following.

Definition. Let M and T be a finite automaton and a transducer, respectively:

$$\mathrm{M} = \left\{ Q^{\mathrm{M}}, \Sigma^{\mathrm{M}}, \delta^{\mathrm{M}}, q_0^{\mathrm{M}}, F^{\mathrm{M}} \right\}$$
$$\mathrm{T} = \left\{ Q^{\mathrm{T}}, \Sigma_{in}^{\mathrm{T}}, \delta^{\mathrm{T}}, q_0^{\mathrm{T}}, F^{\mathrm{T}}, \Sigma_{out}^{\mathrm{T}}, \beta^{\mathrm{T}} \right\} \text{ with } \Sigma_{in}^{\mathrm{T}} = \Sigma^{\mathrm{M}}$$

Then the **composition** of T and M, denoted $\mathrm{T}' = \mathrm{T} \circ \mathrm{M}$ is a transducer

$$\mathrm{T}' = \left\{ Q', \Sigma'_{in}, \delta', q'_0, F', \Sigma'_{out}, \beta' \right\}$$

such that: Q' are ordered pairs of the states in Q^{T} and Q^{M}: $Q' = Q^{\mathrm{T}} \times Q^{\mathrm{M}}$; the input alphabet is $\Sigma'_{in} = \Sigma_{in}^{\mathrm{T}} = \Sigma^{\mathrm{M}}$; the start state is $q'_0 = \left(q_0^{\mathrm{T}}, q_0^{\mathrm{M}}\right)$; the accepting states are $F' = \left\{ \left(q^{\mathrm{T}}, q^{\mathrm{M}}\right) : q^{\mathrm{T}} \in F^{\mathrm{T}}, q^{\mathrm{M}} \in F^{\mathrm{M}} \right\}$; the output alphabet is $\Sigma'_{out} = \Sigma_{out}^{\mathrm{T}}$; the transition rule $\delta' : Q' \times \Sigma'_{in} \to Q'$ is given by

$$\delta'\left(q', \sigma_{in}\right) = \left(\delta^{\mathrm{T}}\left(q^{\mathrm{T}}, \sigma_{in}\right), \delta^{\mathrm{M}}\left(q^{\mathrm{M}}, \sigma_{in}\right) \right) \text{ , where } q' = \left(q^{\mathrm{T}}, q^{\mathrm{M}}\right) \text{ and } \sigma_{in} \in \Sigma'_{in}$$

and the observation function $\beta' : Q' \times \Sigma'_{in} \to \Sigma'_{out}$ is

$$\beta'\left(q', \sigma_{in}\right) = \beta^{\mathrm{T}}\left(q^{\mathrm{T}}, \sigma_{in}\right), \text{ with } q' = \left(q^{\mathrm{T}}, q^{\mathrm{M}}\right)$$

A transducer can also be used to define a "machine metric" that measures the distance of a string from a given language. The action of this transducer is to encode a given pattern into a string of 1's that represents in unary form the number of disallowed transitions observed in the pattern. We will represent the fiducial language by its corresponding automaton M. Then the metric with respect to this language is defined as follows.

Definition. Let M be a machine, and ω be any string, not necessarily in $L(\mathrm{M})$. The **machine metric** is

$$\|\omega\|_{\mathrm{M}} = \begin{cases} \|f_{\mathrm{T_M}}(\omega)\| & \|\omega\| < \infty \\ \lim\limits_{n \to \infty} \frac{\|f_{\mathrm{T_M}}(\omega_n)\|}{\|\omega_n\|} & \text{otherwise} \end{cases}$$

where $\mathrm{T_M}$ is the transducer formed from automaton M by augmenting it to output the null symbol λ on allowed transitions and adding new edges that emit a 1 on previously disallowed transitions. The new edges are "reset" edges, connected to the state representing the effect of having read the disallowed symbol when in the start state. $f_{\mathrm{T_M}}(\omega)$ is the string emitted by $\mathrm{T_M}$ upon reading the input ω. In the infinite case, ω_n denotes the string consisting of the first n symbols of ω. The finite and infinite cases are related by dividing the former by $\|\omega\|$.

In a later section, we will use a machine metric that counts dislocations in spatial patterns generated by rule 18.

2.4 CA as Regular Language Processors

The connection between CA and regular languages is readily made by identifying the CA alphabet $\mathcal{A}$ with the automaton alphabet Σ. Then a CA state $\mathrm{s}_t \in \mathcal{A}^N$ is treated as a word in some language $L \in \mathcal{R}$. It is important to note that this makes the analysis independent of lattice size, since L is a subset of the extended state space $\mathcal{A}^* \equiv \Sigma^*$, and hence may contain strings of all lengths. When discussing CA with lattices of arbitrary but fixed size N, we will often use the regular language notation. It is understood that we mean only the strings in the language that are of length N.

The operation of the global CA update rule Φ is to recode a parent string s_{t-1} into a child string s_t. Stated in this way, it is clear that Φ can be implemented as a transducer, which we denote T_Φ. Since a transducer reads the symbols one by one, the full definition must include a specification of the direction in which it is to be passed over the spatial array; in the following, we will always take this direction to be from left to right. The machine representation of T_Φ for an arbitrary elementary CA is shown in Fig. 1. It operates as a queue with two memory registers, in which a child symbol is emitted when the final member of its parent neighborhood is read. The output symbols are given by the local update rule operating on the current parent neighborhood: $\sigma_{out}^{i-1} = \phi(\sigma_{in}^{i-2}, \sigma_{in}^{i-1}, \sigma_{in}^{i})$. This is in fact just a description of the most memory-efficient way of programming one-dimensional CA dynamics for numerical experiments.

As defined above, the CA update rule Φ operates on individual configurations. Alternatively, we can look at how a CA processes regular language representations of spatial configurations.[4] This is analogous in dynamical systems theory to the study of the evolution of state ensembles, rather than individual orbits. To this end we define the ensemble evolution operator $\mathbf{\Phi}$ by

$$\Omega_t = \mathbf{\Phi}\Omega_{t-1} = \mathbf{\Phi}_t\Omega_0$$

where Ω_0 is any regular language of spatial configurations used as initial conditions:

$$\Omega_t = \{\mathrm{s} \,:\, \mathrm{s} = \Phi\mathrm{s}_{t-1},\ \forall\, \mathrm{s}_{t-1} \in \Omega_{t-1}\}$$

When $\Omega_0 = \mathcal{A}^*$, the spatial languages are regular for all finite t: $\Omega_t \in \mathcal{R}, \forall t < \infty$.[4] The FME algorithm defined below shows that the iterate of *any* regular language is regular: $\Omega_0 \in \mathcal{R} \Rightarrow \Omega_t \in \mathcal{R}, \forall t < \infty$. We also note here that, since any finite language is regular, the spatial language generated by any CA over a finite spatial lattice $(N < \infty)$ is also always regular. Thus, if either space or time is finite, the set of CA configurations is a regular language.[30] As a consequence $\mathbf{\Phi}$ is an automorphism that maps regular language space into itself: $\mathbf{\Phi} : \mathcal{R} \to \mathcal{R}$. For infinite lattices, however, the asymptotic spatial language, found in the limit $t \to \infty$, need not be regular—or even recursively enumerable.[5] Somewhat surprisingly, this property often does not hamper our computational analysis, as will be seen in the next section.

Figure 2 shows a comparison of the operation of $\mathbf{\Phi}$ in configuration space $\mathcal{A}^N$, where it operates on sets of configurations, and in regular language space $\mathcal{R}$, where it operates on

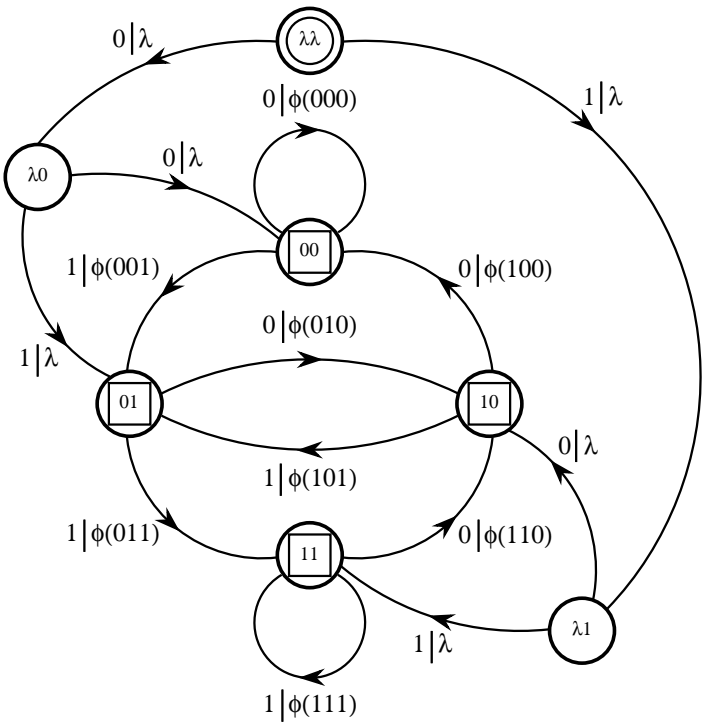

Fig. 1. The CA update rule transducer T_Φ that reads a state s_{t-1} and emits its iterate $\mathrm{s}_t = \Phi(\mathrm{s}_{t-1})$. T_Φ is passed over the spatial array from left to right. In this and the following figures, inscribed circles and squares denote start and accepting states, respectively. Edges are labeled with $\sigma_{in}|\sigma_{out}$ to denote the input and output symbols. λ is the null symbol; for example, when used for output, it means that no symbol is emitted. The labels inside the states correspond to the previous two symbols read. This is the machine for the infinite lattice case; finite lattices require additional constraints due to boundary conditions.

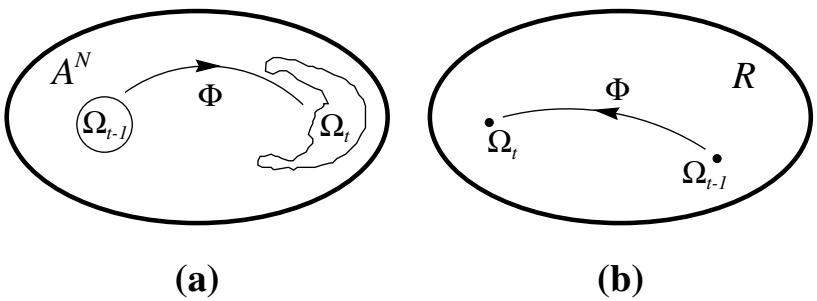

Fig. 2. Schematic of the operation of the language operator $\mathbf{\Phi}$ (a) in CA state space and (b) in regular language space. Note that case (b) subsumes all values of N simultaneously.

individual languages. The important advantage of the latter approach is that it allows us to investigate the dynamics of ensembles of structurally similar CA states without having to deal with the highly nontrivial complications introduced by tracing the evolution of individual states.

Definition. The **Finite Machine Evolution (FME) algorithm.** Given the CA update transducer T_Φ for a specific CA rule Φ, and a machine $\mathrm{M}_{t-1} = \mathrm{M}(\Omega_{t-1})$, we calculate $\mathrm{M}_t = \mathrm{M}(\Omega_t) = \mathrm{M}(\mathbf{\Phi}\Omega_{t-1})$ by using machine-composition and minimizing the resulting transducer with respect to the output symbols:

$$\mathrm{M}_t = [\mathrm{T}_\Phi \circ \mathrm{M}_{t-1}]_{\mathrm{out}}$$

3 Structures in Language Space

The global machine $\mathrm{M}_t = \mathrm{M}(\Omega_t)$, when $\Omega_t = \mathbf{\Phi}_t(\mathcal{A}^*)$, describes the ensemble of the iterates of all possible initial conditions at some time t. It necessarily captures all of the information processing accessible to the CA, which includes correlations and information transmission as well as computation. Many previous investigations of CA language properties have tended to focus on the global machine.[4,31]

In terms of dynamical systems theory, though, the global machine describes the entire attractor-basin portrait.[20] In many cases, this description is seen to be prohibitively difficult to construct because, for example, M_t grows too large with time. In fact, such complete global descriptions are rarely pursued in dynamical systems theory. The main exceptions to this come in the study of flows on manifolds of constant negative curvature, uniform hyperbolic systems, and some very simple periodic systems: all of which are highly idealized systems. Similarly for CA, the complexity of the global machine is reflected in the fact that the asymptotic language can be nonregular, as noted above. Even for elementary CA, there are many cases in which the size of M_t increases extremely rapidly in time. This is true in particular of rule 18: $\mathrm{M}(\Omega_3)$ has $\|Q\| = 143$ states; $\mathrm{M}(\Omega_4)$ is estimated to have $\|Q\| > 20,000$.[4] Furthermore, there are other CA rules for which the growth is even faster.[31]

Another shortcoming of the global approach is that machines for qualitatively similar CA can be vastly different. Comparison of the space-time patterns generated by rule 18 and rule 90 shows a marked similarity; indeed, for an important class of initial conditions they are identical. Yet the global machine for rule 90, far from exhibiting the radical growth seen in rule 18, has just one state for all t.

The apparent intractability due to unbounded growth and the contrast between global machines for rules with such similar behavior suggest that an alternative view is required. It is not necessary, for example, to use $\Omega_0 = \mathcal{A}^*$; nor can we expect it to be particularly useful, as we just noted. Characterizing the entire attractor-basin portrait of any dynamical system with a single expression is implausible for anything but the simplest dynamics. Instead, one standard approach is to identify the *structures* that dominate and organize the system's behavior. The most basic of these for dissipative systems are attractors, basins, and separatrices. The state space is then understood in terms of those structures. In the case of CA, we want to find the dominant, dynamically-homogeneous space-time patterns and their associated *languages*.

Yet previous investigations from the viewpoint of pure dynamical systems theory have tended to overlook computationally equivalent space-time behavior. For example, the study of periodic orbits of very small lattices,[8–11] perturbation growth rates as the analog of Lyapunov characteristic exponents,[32] and information transmission in space or in time[33] have failed to provide a formulation of attractors, basins, and separatrices that integrates them with a description of the intrinsic computational capability of the underlying CA.

This is a reflection of a basic problem in the literal application of dynamical systems theory to CA and other spatially-extended systems. The theory does not provide a formalism for the investigation of a state's internal structure. A state of a dynamical system is a structureless point. But it is exactly the internal, now spatial, structure of a CA configuration that is of interest. This deficit can be addressed by interpolating between the literal notion of a CA state as a position-dependent configuration of cells and the intuitive notion of a translation-independent pattern. Generally speaking, a pattern is (i) a position-independent configuration and (ii) a typical sequence generated by the process, where "typical" is used in the sense of the Shannon-McMillan theorem of information theory.[28] The use of a regular language or finite automaton description of processes captures exactly these features.

For CA rule 18, the dominant structure is the *domain* described above: every other site has value 0, while the remaining sites can be either 0 or 1.[17,23] The iterate of a domain consists of a domain shifted by one site. This observation is consistent with the intuitive notion of a domain as a dynamically-homogeneous region of the spatial array. We formalize the preceding comments in the following subsections. First we consider the spatial structure via the notion of a regular domain. Then we address temporal structure and stability by defining regular vicinity and regular attractor.

3.1 Domains

Definition. A **regular domain** or "domain language" $\Lambda \in \mathcal{R}$ of a CA is a process language representing a set of spatial regions with the following two properties:
(i) *Temporal invariance:* Λ is mapped onto itself by the dynamic, i.e., $\mathbf{\Phi}\Lambda = \Lambda$.
(ii) *Spatial homogeneity:* The process graph of Λ is strongly connected.[34] This is similar to statistical stationarity of the spatial pattern.

For general CA there can be any number of different domains Λ^i, each with its own computational structure.

It is straightforward to generalize this to more complicated dynamical behavior than single iteration invariance. A domain can be defined as a nonwandering orbit of languages. This includes those whose temporal behavior is periodic or even chaotic. In each case, all languages in the orbit must be spatially homogeneous.

A domain describes a computationally-homogeneous *region* of the spatial array. In this way we take "domain" to mean the domain language Λ or, alternatively, any configuration $\mathrm{s} \in \Lambda$.

Definition. A **wall** is a boundary between domains. Two domains Λ^i and Λ^j separated by a wall will be denoted $\Lambda^{i,j}$. Walls may consist of any number of cells, including zero. A **dislocation** is a wall of zero thickness separating two identical domains.

Definition. The **phase** of site i in a domain Λ^j is the state q of the machine $\mathrm{M}\left(\Lambda^j\right)$ when σ^i is read in a chosen direction.

In the case of rule 18, the observed behavior described in the introduction defines a domain language Λ^0. All spatial states are made up of concatenations of words $\omega \in \Lambda^0$ separated by dislocations. In particular, the two-domain language $\Lambda^{0,0}$ consists of all states in which there is a single dislocation. For reasons that will become clear later, we also include in $\Lambda^{0,0}$ the null language 0^* consisting of the null states for all lattice sizes, i.e., all strings of consecutive 0's. The process graph of Λ^0 is shown in Fig. 3. The DFAs for both Λ^0 and $\Lambda^{0,0}$ are shown in Fig. 4. Note from the figures that Λ^0 consists of strings made up of isolated 1's separated by an odd number of 0's.

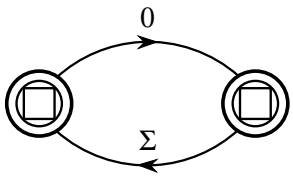

Fig. 3. The process graph of Λ^0 in which both states are start and final states. The symbol Σ denotes the symbol set {0, 1}.

Proposition. Λ^0 is a regular domain.

Proof: (i) Temporal invariance: by explicit construction using the FME algorithm. Figure 5 shows an intermediate stage of the construction of the first iterate of $\mathrm{M}\left(\Lambda^0\right)$, immediately after composing the machine with the rule 18 update transducer T_{18}. After minimizing with respect to the output, the final machine $\mathrm{M}\left(\mathbf{\Phi}\Lambda^0\right) = \left[\mathrm{T}_{18} \circ \mathrm{M}\left(\Lambda^0\right)\right]_{\mathrm{out}}$ is seen to be identical to $\mathrm{M}\left(\Lambda^0\right)$. (ii) Spatial homogeneity: this follows immediately since the process graph of Λ^0, shown in Fig. 3, is strongly connected. ▮

Proposition. $\Lambda^{0,0}$ is a time-invariant (fixed-point) regular language, although it is not a domain.

Proof: We begin by calculating the image of $\mathrm{M}\left(\Lambda^{0,0}\right)$ under rule 18 by means of the FME algorithm. Figure 6 shows the transducer $\mathrm{T}_{18} \circ \mathrm{M}\left(\Lambda^{0,0}\right)$. After finishing the FME algorithm, we find that the resulting machine is slightly different from $\mathrm{M}\left(\Lambda^{0,0}\right)$: there is a set of input strings accepted by $\mathrm{M}\left(\Lambda^{0,0}\right)$ whose images under rule 18 are accepted by $\mathrm{M}\left(\Lambda^0\right)$. This is a spurious effect due to the spatial inhomogeneity of $\mathrm{M}\left(\Lambda^{0,0}\right)$ in combination with the spatial correlation of the input and output strings of the CA update rule transducer. The input to $\mathrm{T}_{18} \circ \mathrm{M}\left(\Lambda^{0,0}\right)$ consists

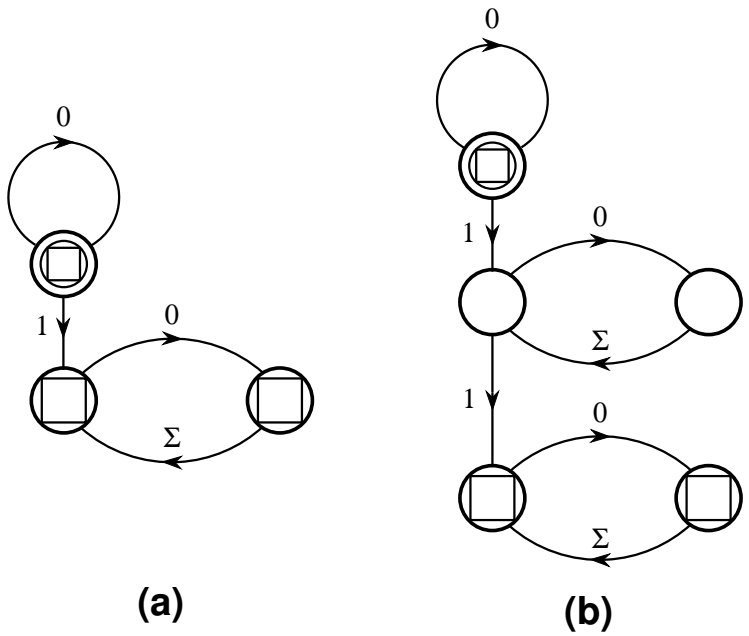

Fig. 4. DFAs (a) $\mathrm{M}(\Lambda^0)$ for single domain language Λ^0 and (b) $\mathrm{M}(\Lambda^{0,0})$ for domain-wall-domain language $\Lambda^{0,0}$. Note that $\mathrm{M}(\Lambda^0)$ differs from the process graph of Fig. 3 in that it is a DFA, which necessarily has a single start state. In this case, the start state is a transient that shows how $\mathrm{M}(\Lambda^0)$ synchronizes to the string it is reading.

of some portion of the spatial state s_{t-1}, starting an arbitrary distance from the dislocation. For input strings of the form $(01)^*10\Sigma0\Sigma$..., the observed region before the parent dislocation maps to a sequence of adjacent 0's. There is no nonzero cell to the left of the child dislocation to fix the spatial phase, and so the dislocation does not show up. As we discuss in the next section, dislocations can only disappear in pairs, except when the state evolves to the null state 0^*. Thus the dislocation is still present, but merely unseen. We remove this "phase-ambiguity" artifact by requiring that for all non-null states s_{t-1} the machine observe the dislocation before it can enter a final state. After the result is minimized, the final machine is seen to be identical to $\mathrm{M}(\Lambda^{0,0})$, shown in Fig. 4(b). Finally, $\Lambda^{0,0}$ is not a domain because it is not spatially homogeneous: its process graph has two strongly connected components. ▮

An essential issue that arises here is the question of discovering domains in the first place. Even with unlimited space-time data, the domain language(s) for a given CA may be sufficiently complex as to defy visual identification. For rule 18, the dominant structure had already been found; but this is not true in general. One consequence of the foregoing analysis is that a general approach to identifying these structures, once developed, would provide a basis for automated dynamical systems investigation of CA computational capability. A promising answer to this problem is what we have used here: the machine reconstruction technique.[26,35] As a starting point to this study we applied machine reconstruction to the state reached by 10^5 iterations of rule 18 on a periodic array of 10^5 sites, starting from a random initial condition. The

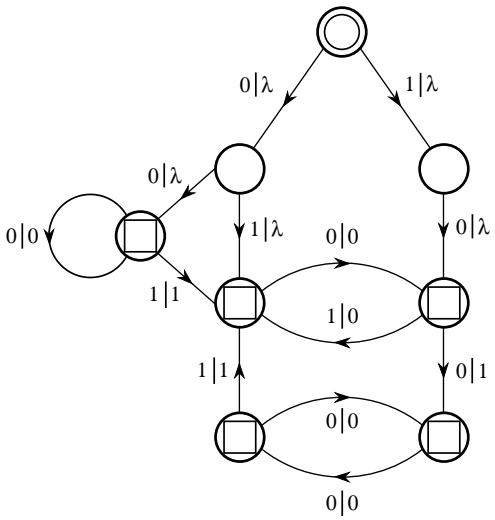

Fig. 5. Λ^0 is a regular domain: Machine $\mathrm{T}_{18} \circ \mathrm{M}(\Lambda^0)$ obtained after composing the rule 18 update transducer with the single-domain machine of Fig. 4(a). Minimization with respect to the output results in a machine identical to that shown in Fig. 4(a).

results of a series of reconstructions were trivial variants of the two-domain machine $\mathrm{M}(\Lambda^{0,0})$ shown in Fig. 4. The variations consisted of missing edges or nondeterministic transitions to "dangling" states from which there was no exiting edge, both of which are common signatures of incomplete statistics. With machine reconstruction it was quite straightforward to identify the regular domains and attractors and proceed with the dynamical systems analysis. Plans include an expert-system-like environment to facilitate the discovery of domains and to perform automated construction of vicinities and basins and proofs of invariance and attraction. The knowledge base for such a system consists of the properties, rules of manipulation, and theorems for regular languages and finite automata. It also incorporates basic notions from dynamical systems, such as those used here.

3.3 Vicinities

Before turning to the discussion of regular attractors, it is necessary to develop the notion of the *vicinity* of a language. In dynamical systems theory, an attractor is required to be stable to small perturbations within an open neighborhood.[20] To adapt this to the present context it is necessary to define first what is meant by "open neighborhood" and second what is meant by "small".

An open neighborhood is an open set in the topology of the state space. We must find a topology that allows us to formulate open sets appropriate for the qualitative dynamics of CA. To this end, we make the following observation.

Remark. Regular language space induces a topology on the extended CA state space.

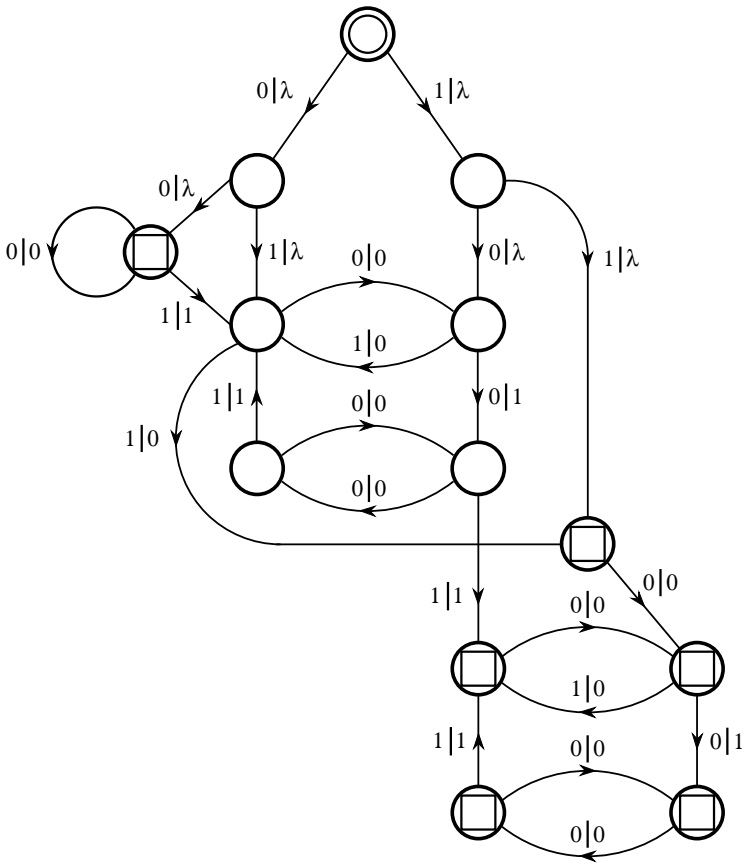

Fig. 6. $\Lambda^{0,0}$ is a fixed-point language: Machine $\mathrm{T}_{18} \circ \mathrm{M}(\Lambda^{0,0})$ obtained after composing the rule 18 update transducer with the two-domain machine of Fig. 4(b). On completion of the remaining steps of the proof, Fig. 4(b) is found.

Proof: The set $\mathcal{R}$ of regular languages satisfies the two requirements of the basis of a topology on $\mathcal{A}^*$:
(i) $\bigcup_{L \in \mathcal{R}} L = \mathcal{A}^*$, since $\mathcal{A}^*$ is itself a regular language; and
(ii) $L \cap L' \in \mathcal{R}, \ \forall\, L, L' \in \mathcal{R}$ by the closure of regular languages under intersection.[25] ▮

All regular languages are open sets in this topology. Thus, the extended CA state space possesses the necessary topological structure. This allows us to define the regular vicinity of a language as an open set containing that language, and which is itself a regular language. The idea is to use the vicinity of a language the way ϵ-balls are used in qualitative dynamics.

Definition. A **regular vicinity** $V(L)$ of a language L is (i) a regular language and (ii) an open set in $\mathcal{A}^*$ such that $L \subset V(L)$.

To define "small perturbation" we use Hamming distance, which leads to the following notion of Hamming vicinity. The discrete nature of CA state space precludes carrying over the

standard definition, since it would demand continuity of site values.

Definition. The **Hamming vicinity** $V_H(L)$ of a regular language L is the regular vicinity generated by adding a unit Hamming distance perturbation each string in L: $V_H(L) = L \dot{+} \delta L$ where $\|\delta L\|_{\text{Hamming}} \leq 1$, that is, $\delta L = \{0^* + 0^*10^*\}$, and $\dot{+}$ is sum modulo the alphabet size. Figure 7(a) shows the Hamming vicinity transducer T_{V_H}.

Some problems arise with perturbations of patterns on infinite lattices. For rules with a quiescent state 0^*, one can define "finite initial conditions"[36] as the set of states having a finite nonzero support. Then the effect of the perturbation on these states may depend sensitively on whether it occurs within the nonzero region or outside it. For example, a single bit flip inside the nonzero support of a finite pattern in Λ^0 generates 0 or 2 dislocations, whereas a bit flip outside the nonzero region can create either 0 or 1 dislocation. It is convenient to restrict the vicinity to include only perturbations in the nonzero region. Note that the new vicinity type can be used with finite lattices with no additional consequences, since in that case the support is the whole finite lattice.

Definition. The **restricted Hamming vicinity** $V_{H_{restricted}}(L)$ of a regular language L is the regular vicinity generated by adding a unit Hamming distance perturbation somewhere within the region of nonzero support of each string in L: $V_{H_{restricted}}(L) = L \dot{+} \delta_r L$ where $\|\delta_r L\|_{\text{Hamming}} \leq 1$ and the perturbation δ_r is restricted to operate within the nonzero support of each string in L. Figure 7(b) shows the restricted Hamming vicinity transducer $\mathrm{T}_{V_{H_r}}$.

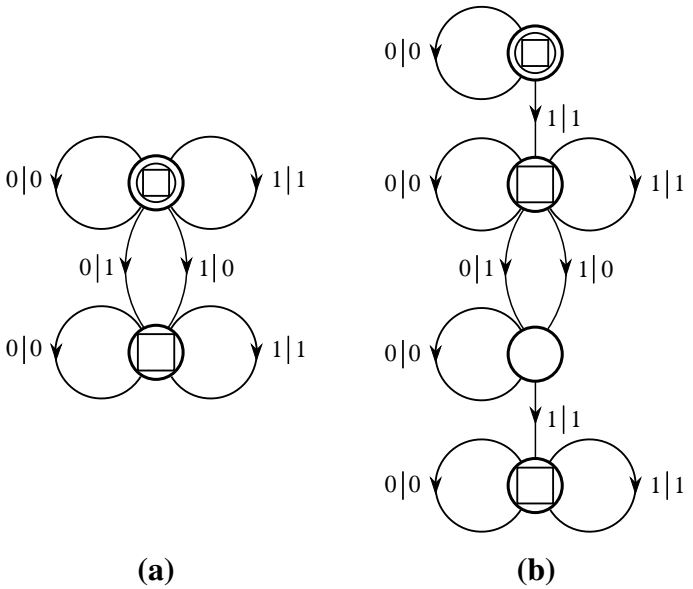

Fig. 7. (a) The Hamming vicinity transducer T_{V_H}. (b) The restricted version $\mathrm{T}_{V_{H_r}}$. Note that these transducers are nondeterministic over inputs.

Lemma. $V^0 \equiv V_H\big(\Lambda^0\big)$, shown in Fig. 8(a), and $V^0_{\rm supp} \equiv V_{H_{restricted}}\big(\Lambda^0\big)$, shown in Fig. 8(b), are the regular Hamming vicinity and the restricted Hamming vicinity of Λ^0, respectively. The subscript $\rm supp$ denotes the restriction of the perturbations to the region of nonzero support.

Proof: By explicit construction using the definitions: $V^0 = \big[\mathrm{T}_{V_H} \circ \Lambda^0\big]_{\rm out}$ and $V^0_{\rm supp} = \big[\mathrm{T}_{V_{H_r}} \circ \Lambda^0\big]_{\rm out}$. ∎

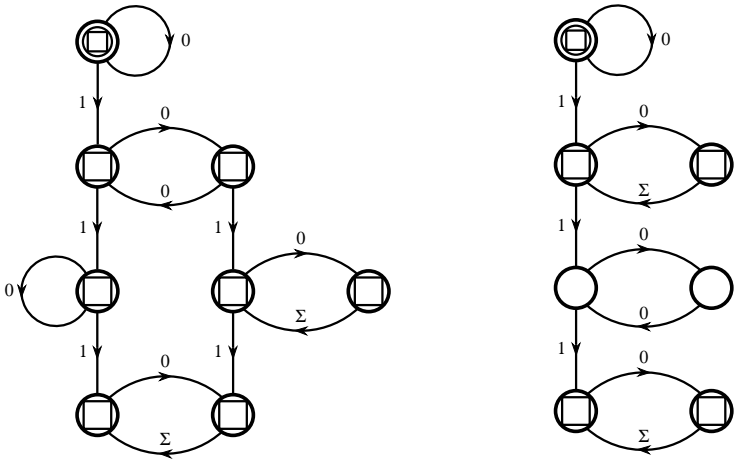

Fig. 8. (a) The Hamming vicinity V^0 of the regular domain Λ^0, shown in Fig. 4(a). (b) The machine for the language $V^0_{\rm supp}$, formed from Λ^0 by flipping a single bit in the nonzero region of each state.

Note that $V^0_{\rm supp}$ consists of two disjoint parts: (i) states that remain in Λ^0 and (ii) states in which the perturbation generates two dislocations separated by an even number of 0's.

3.4 Attractors

Having set up the necessary geometrical prerequisites, it is now possible to define the regular attractor of a CA. First we define the necessary temporal behavior, and then we combine that with a stability analysis using regular vicinities, to arrive at the final definition.

Definition. An **invariant language** L is one that is mapped onto itself by the dynamic: $\mathbf{\Phi}(L) = L$. A **periodic language** is a language L such that $\mathbf{\Phi}_p(L) = L$ for some finite integer period p. A **language orbit** Γ is the set of iterates of a language: $\Gamma = \{\mathbf{\Phi}_t(L) : \ t \geq 0\}$.

Note that since any finite union of regular languages is itself a regular language, any periodic regular language orbit can be turned into a regular invariant language by taking the union over

the orbit's elements. Aperiodic language orbits, which can occur only on infinite lattices, may be nonregular, however.

Definition. An invariant language L is a **fixed-point regular attractor** if it is a regular language and there exists a regular vicinity $V(L)$ such that

1. All states in V remain in V forever: $\Phi_t(V) \subseteq V, \ \forall\ t \geq 0$.
2. All states in V, except possibly for a set R such that $S(R|V) = 0$, eventually collapse down onto L: thus for almost all $\mathbf{s} \in V$, $\exists\ t \geq 0$ such that $\Phi_t(\mathbf{s}) \in L$.

Letting a set of zero size remain outside the attractor indefinitely is absolutely essential to domain-wall dominated CA, as will be seen below. This is analogous in continuous-state dynamical systems to allowing unstable or marginally stable fixed points and periodic orbits to exist in the vicinity of an attractor. One of the benefits of this type of definition is that the simple Hamming vicinity is an adequate tool in establishing the existence of attractors. An alternative formulation would have introduced a machine metric that implicitly excluded such nonattracting states; the net result would be the same.

In conjunction with numerical investigations, the definitions and propositions formulated so far allow us to analyze in some detail the state space structure of rule 18. Our goal is to establish that Λ^0 and $\Lambda^{0,0}$ are fixed-point regular attractors. The argument breaks up into three cases depending on the CA lattice size and the class of initial conditions: (i) finite periodic lattices; (ii) infinite lattices with finite initial conditions; and (iii) infinite lattices with infinite initial conditions. After analyzing the cases via several observations, a proposition, and a conjecture, we summarize the remaining steps for a direct proof.

Observation. For finite periodic lattices with an even number of sites N, Λ^0 is a fixed-point regular attractor.

Discussion: A previous proposition established that Λ^0 is time-invariant. We take as the regular vicinity of Λ^0 the unrestricted Hamming vicinity V^0. The periodic boundary conditions ensure that on all lattices of even N, the number of dislocations is even; thus all states in V^0 have either $d = 0$ or 2 dislocations. In the next section we will prove that the number of dislocations is nonincreasing with time and dislocations annihilate in pairs, which means that all states in V^0 remain in V^0 for all t. To show attraction, it would be sufficient to show that the fraction of states in the nonattracted set goes to zero with increasing lattice size. Numerical simulations, to be discussed in a later section, indicate that the upper bound on the fraction of nonattracted states obeys a power law $N^{-\alpha}$ with $\alpha \approx \frac{1}{2}$. This worst-case fraction is already below 3% on lattices with as few as 400 cells. This indicates that the size of the nonattracted set relative to V^0 vanishes, and so Λ^0 is an attractor.

Observation. For finite periodic lattices with N odd, $\Lambda^{0,0}$ is a fixed-point regular attractor.

Discussion: The discussion is quite similar to the case for N even. $\Lambda^{0,0}$ has been proven to be time-invariant. The periodicity of the lattice ensures that the unrestricted regular Hamming vicinity $V^{0,0} \equiv V_H\left(\Lambda^{0,0}\right)$ consists, aside from the single pattern 0^*, of states with $d = 1$ or 3 dislocations. Since for all states $\mathbf{s} \in V^{0,0}$, either $\Phi_{18}(\mathbf{s}) = 0^*$ or d decreases by 0 or 2, all states in $V^{0,0}$ remain in $V^{0,0}$ for all t. As in the case of Λ^0, the numerical evidence shows rapid decay of the size of the nonattracted set with increasing lattice size. Thus for periodic lattices with odd size N, almost all states in $V^{0,0}$ eventually fall into $\Lambda^{0,0}$, and so $\Lambda^{0,0}$ is an attractor.

Note that the behavior on finite lattices depends on the parity of the lattice. For even N, the periodic boundary conditions require that there be an even number of dislocations in the state at all times. Since $\Lambda^{0,0}$, apart from the single state 0^*, has a single dislocation, it will never be present on an even-sized lattice. Similarly, Λ^0 (again, apart from 0^*) will never exist on a lattice with N odd. Thus, one or the other may be present, but never both together.

By taking advantage of the domain isomorphism discussed in the next section, we can apply the well-understood algebraic analysis of elementary CA rule 90 to rule 18 [37] to learn more about the attractors. The principal result of concern to us here is that on periodic lattices of size $N = 2^k$, all initial conditions evolve under rule 90 to the null state 0^*.[8] For $N \neq 2^k$, this is not the case. As we will see, the implication of this for rule 18 is that for states with d dislocations evolving on lattices of size $N = 2^k - d$, a relatively large number of states will fall directly into 0^* before any dislocations have annihilated. For example, for $N = 2^k$ all states in Λ^0 eventually evolve to the single pattern 0^*, and for $N = 2^k - 1$ all states in $\Lambda^{0,0}$ evolve to 0^*. Thus 0^* is the attracting invariant set for these lattices, and the regular attractors Λ^0 and $\Lambda^{0,0}$ are "degenerate" in the sense that only the state 0^* is actually present. Similarly, for $N = 2^k - 2$ a significant number of states in V^0 will fall directly into 0^*, and for $N = 2^k - 3$ many states in $V^{0,0}$ evolve to 0^*. However, this "leakage" into 0^* is typically a zero-sized effect, as the following observation indicates.

First we define $\mathcal{B}(0^*)$ as the set of all states that map to 0^*, i.e., the basin of 0^*. In a later section, we will define the basin of a language more formally.

Observation. Let (Λ, V) denote the attractor and its vicinity on a finite periodic lattice of any size. That is, $(\Lambda, V) = \left(\Lambda^0, V^0\right)$ for N even, and $(\Lambda, V) = \left(\Lambda^{0,0}, V^{0,0}\right)$ for N odd. Then on all but a sparse set of lattices, $S(\mathcal{B}(0^*) \cap V|V) = 0$ and $S(\mathcal{B}(0^*) \cap \Lambda|\Lambda) = 0$.

Discussion: First we show that the set of lattices discussed above, for which 0^* is known to be an important attracting invariant set, is sparse. The set of these lattice sizes is $R_d = \left\{N : N = 2^k - d,\ d \in \{0, 1, 2, 3\} \text{ and } k \in \mathbb{Z}\right\}$. The number $I_{R_d}(n)$ of lattice sizes N in R_d such that $N \leq n$ grows as $I_{R_d}(n) = 4 \times \lfloor \log_2(n) \rfloor - \text{const}$ for large n, where $\lfloor x \rfloor$ is the integer part of x. In the limit of large n, we have

$$\lim_{n\to\infty} \frac{I_{R_d}(n)}{n} = \lim_{n\to\infty} \frac{4 \log_2(n)}{n} = 0$$

and so R_d is sparse. We next investigate numerically the behavior on the remaining lattices. Simulations on lattices not in the above sparse set have shown that as a function of increasing N, the upper bound on the fraction of states in the attractor Λ that map to 0^* decays with a power law. The same is true of the vicinity V. This indicates that $S(\mathcal{B}(0^*) \cap \Lambda|\Lambda) = 0$ and $S(\mathcal{B}(0^*) \cap V|V) = 0$ on all but the sparse set R_d of lattices.

On infinite lattices Λ^0 and $\Lambda^{0,0}$ coexist. To keep the attractors distinct, it is necessary to subtract from each language those states that evolve into 0^*. To this end we define the languages

$$\tilde{\Lambda}^0 \equiv \Lambda^0 - \mathcal{B}(0^*)$$
$$\tilde{\Lambda}^{0,0} \equiv \Lambda^{0,0} - \mathcal{B}(0^*)$$

where the set-theoretic difference $P - Q$ denotes the set of states in P that are not in Q. Note that $\tilde{\Lambda}^0$ and $\tilde{\Lambda}^{0,0}$ are disjoint and time-invariant.

Proposition. On an infinite lattice with finite initial conditions, $\tilde{\Lambda}^0$ and $\tilde{\Lambda}^{0,0}$ are fixed point regular attractors.

Proof: In this case, 0^* is a Gardern-of-Eden state, so $\mathcal{B}(0^*) = 0^*$. $\mathrm{M}\left(\tilde{\Lambda}^0\right)$ and $\mathrm{M}\left(\tilde{\Lambda}^{0,0}\right)$ are easily constructed from $\mathrm{M}\left(\Lambda^0\right)$ and $\mathrm{M}\left(\Lambda^{0,0}\right)$, shown in Fig. 4, by making the start states nonaccepting. $\tilde{\Lambda}^0$ and $\tilde{\Lambda}^{0,0}$ are therefore both regular languages. We use the regular vicinities $\tilde{V}^0_{\mathrm{supp}} \equiv V_{H_{restricted}}\left(\Lambda^0\right) - \mathcal{B}(0^*)$ and $\tilde{V}^{0,0}_{\mathrm{supp}} \equiv V_{H_{restricted}}\left(\Lambda^{0,0}\right) - \mathcal{B}(0^*)$. To prove attraction, we quote the following results of ref. [36]: (i) all states, apart from 0^*, in which the initial number n_d of dislocations is even, fall into $\tilde{\Lambda}^0$; and (ii) if n_d is odd, then the state will fall into $\tilde{\Lambda}^{0,0}$. If we denote the set of finite initial conditions by F, this means in particular that all states in $\tilde{V}^0_{\mathrm{supp}} \cap F$ fall into $\tilde{\Lambda}^0$. More formally,

$$\forall\, \mathbf{s} \in \tilde{V}^0_{\mathrm{supp}} \cap F, \exists t < \infty \text{ such that } \Phi_t \mathbf{s} \in \tilde{\Lambda}^0$$

Similarly, all states in $\tilde{V}^{0,0}_{\mathrm{supp}} \cap F$ fall into $\tilde{\Lambda}^{0,0}$. ∎

Conjecture. On an infinite lattice with infinite initial conditions, $\Lambda_\infty \equiv \Lambda^0 \cup \Lambda^{0,0}$ is a fixed point regular attractor. Specifically,

$$\Lambda_\infty = \tilde{\Lambda}^0 \cup \tilde{\Lambda}^{0,0} \cup [\mathcal{B}(0^*) \cap \Lambda_\infty]$$

where $\tilde{\Lambda}^0$ and $\tilde{\Lambda}^{0,0}$, as defined above, are disjoint, possibly nonregular, attracting invariant sets, and $S(\mathcal{B}(0^*) \cap \Lambda_\infty|\Lambda_\infty) = 0$.

Discussion: In the case of infinite initial conditions, $\mathcal{B}(0^*)$ may or may not be regular. If $\mathcal{B}(0^*)$ is nonregular, then $\tilde{\Lambda}^0$ and $\tilde{\Lambda}^{0,0}$ can be nonregular as well. Nevertheless, they may still be attracting. The language Λ_∞ is clearly regular and time-invariant, since it is the union of two invariant regular languages. $\mathrm{M}(\Lambda_\infty)$ is readily formed from $\mathrm{M}\left(\Lambda^{0,0}\right)$ shown in Fig. 4(b),

by making all machine states accepting. All configurations in the regular Hamming vicinity $V_\infty \equiv V^0 \cup V^{0,0}$ remain in V_∞ for all time. To argue for attraction, we partition V_∞ into the disjoint, possibly nonregular, sets $\tilde{V}^0 \equiv V^0 - \mathcal{B}(0^*)$, $\tilde{V}^{0,0} \equiv V^{0,0} - \mathcal{B}(0^*)$, and $\mathcal{B}(0^*)$. The preceding observations indicate that almost all states in $\tilde{V}^0$ and $\tilde{V}^{0,0}$ eventually fall into $\tilde{\Lambda}^0$ and $\tilde{\Lambda}^{0,0}$, respectively. By definition, all states in $\mathcal{B}(0^*)$ evolve to the invariant state 0^*. We consider only $\tilde{\Lambda}^0$ in detail, since the discussion for $\tilde{\Lambda}^{0,0}$ is quite similar. The third observation above indicates that in the limit of infinite N, $\mathcal{B}(0^*)$ has zero size relative to Λ^0 and V^0. Equivalently, $S\Big(\tilde{\Lambda}^0 | \Lambda^0\Big) = \mathcal{O}(1)$ and $S\Big(\tilde{V}^0 | V^0\Big) = \mathcal{O}(1)$. So, we can concentrate on V^0 and Λ^0 since they differ from $\tilde{V}^0$ and $\tilde{\Lambda}^0$ by a set of zero size. To show that almost all states in $\tilde{V}^0$ eventually fall into $\tilde{\Lambda}^0$, it is sufficient to show that almost all states in V^0 fall into Λ^0.

The set V^0 breaks up into three disjoint subsets:

i. $\big\{\mathrm{s} : \Phi_t \mathrm{s} \in \Lambda^0 \text{ for some } t < \infty\big\}$. This is by far the largest set. Indeed, half of the states in V^0 have 0 dislocations, and hence are already in Λ^0 at $t = 0$.

ii. $\Big\{\mathrm{s} : \Phi_t \mathrm{s} \underset{t\to\infty}{\to} \text{temporally periodic state not in } \Lambda^0\Big\}$. The temporal periodicity implies that s is spatially periodic. This is because of the expanding nature of rule 18: local site information is transmitted along the lattice in both directions at the speed of light. If the state as a whole is to be temporally periodic, it must be due to an underlying spatial periodicity. Since a spatially periodic state of period p evolving on an infinite lattice is equivalent to a primitive period of that state evolving on a finite periodic lattice of p cells, this set falls into the case of the first observation above, which indicates that it is of vanishing size.

iii. $\Big\{\mathrm{s} : \Phi_t \mathrm{s} \underset{t\to\infty}{\to} \text{temporally aperiodic state not in } \Lambda^0\Big\}$. This is the "quasiperiodic" case in which the two dislocations either diverge forever from each other or march along the lattice roughly in parallel, their velocities having the same sign. It is not yet clear whether such states even exist. At the very least, if they do, they are extremely rare, and we ignore them.

The net result is that at $t = \infty$ at most a set of vanishing size remains outside of Λ^0. Almost all states in V^0 eventually fall into Λ^0 and so $\tilde{\Lambda}^0$ is attracting. As we said, similar arguments show that $\tilde{\Lambda}^{0,0}$ is attracting and that $S\big(\mathcal{B}(0^*) \cap \Lambda^{0,0} | \Lambda^{0,0}\big) = 0$. Putting the two cases together immediately gives that Λ_∞ is a regular attractor, and that $S(\mathcal{B}(0^*) \cap \Lambda_\infty | \Lambda_\infty) = 0$.

Remark. We expect to be able to use the language-theoretic apparatus we have set up to demonstrate rigorously the attracting nature of Λ^0 and $\Lambda^{0,0}$. This will be left for presentation elsewhere. The strategy to be used is as follows. Define $\widehat{V}^0 = V^0_{\text{supp}} - \Lambda^0$, and evolve this language forward in time using the FME algorithm, at each time step excising those states which have fallen into Λ^0. In this way, examine the time series of languages

$$\widehat{V}^0_t = \mathbf{\Phi}\Big(\widehat{V}^0_{t-1}\Big) - \Lambda^0, \text{ with } \widehat{V}^0_0 = V^0_{\text{supp}} - \Lambda^0$$

This is the set of vicinity patterns that have not fallen onto the attractor. Our goal is to find sufficient regularity in the time series to make it possible to bound the decay in the size of $\widehat{V}_t^0$ as $t \to \infty$. If the asymptotic size is zero, then attraction is proved. The conjecture is that there is a monotonic decrease in the upper bound on the size of the nonattracted set and that this decay rate can be estimated by a functional renormalization equation determined by the CA rule acting on the defined ensemble. The spatial rescaling should reveal with increased lattice size a "fractal" structure. The critical exponent associated with the self-similarity will describe that set and determine the size decay rate. All we require is monotonic decrease, a fairly lenient property compared to those used in the existing algebraic and combinatorial analyses.

The arguments in this section employed both proof and numerical observation. The remaining steps necessary to turn them into rigorous propositions can be laid out in detail:

i. Prove that the sets of nonattracted states in V^0 and $V^{0,0}$ have zero size relative to Λ^0 and $\Lambda^{0,0}$ for typical N.
ii. Prove that $\mathcal{B}(0^*)$ has zero size relative to Λ^0 and $\Lambda^{0,0}$ on all but a sparse set of lattices.
iii. Prove that the set of aperiodic nonattracted states in V_∞ on an infinite lattice with infinite initial conditions has zero size relative to Λ_∞.

In this way the formalism we have introduced has reduced the overall problem to these three cases. Each requires that a given set have vanishing size. The numerical investigations indicate that the sizes of the above three sets rapidly decay with increasing lattice size.

A simple summary follows from accepting the numerical evidence: the regular attracting domain Λ^0 organizes the state space. Disregarding for the moment $\mathcal{B}(0^*)$, we can consider $\Lambda^{0,0}$ as the lowest "energy" fluctuation above the vacuum state Λ^0. Like analogous situations with topological solitons the dislocation in $\Lambda^{0,0}$ is stabilized by the global topology of either finite odd-size or infinite lattices. And so $\Lambda^{0,0}$ is an attractor, although not a domain.

3.5 Other Structures

Finally, we note that we can use the above definition of regular attractor as a basis for defining other structures in language space, namely basins and separatrices. These definitions mirror those found in standard dynamical systems theory.

Definition. The set of all preimages of a regular attractor Λ is its **basin**: $\mathcal{B}(\Lambda) = \bigcup_{t \geq 0} \Phi_{-t}(\Lambda)$.

The set of states that are not in basins, but still in the accessible region of configuration space, make up the **separatrices**: $\mathcal{S} = \mathcal{A}^N - \bigcup_{\Lambda \in \{\Lambda^i\}} \mathcal{B}(\Lambda)$, where $\{\Lambda^i\}$ is the set of all attractors of the system.

Both CA basins and separatrices may contain states with no predecessors. These so-called Garden-of-Eden configurations occur only as initial configurations in $\mathcal{A}^N$. The list of attractors, basins, and separatrices of a CA is its **attractor-basin portrait**. The attractor basin portrait of rule 18 is the subject of the next section.

4 Global Structure of Rule 18

Having defined some of the fundamental structural objects in language space, we can now put them together with a discussion of certain behavioral properties determined by the global organization of the state space, to come up with a fairly complete picture of the overall behavior of rule 18. The main question is the following: What usually happens to a typical initial configuration as it evolves in time? By "typical", we mean in particular "almost all configurations on almost all lattices". Atypical behavior is best treated as a refinement of the overall picture we present here, to be addressed only *after* the typical behavior is understood. In discussing the attractors, therefore, we will use Λ^0 and $\Lambda^{0,0}$ for both finite and infinite lattices, with the understanding that we are disregarding the sparse set of finite lattices for which a significant fraction of states evolve to 0^*, and on the remaining lattices, the nonattracted sets of zero size in the vicinity of each attractor. For the infinite case, this last includes $\mathcal{B}(0^*)$ as well.

As a first step, we note again that we can treat any initial state $\mathtt{s}$ as a concatenation of domains of various lengths, separated by dislocations. Then part of the question is to decide what happens to these dislocations as the state evolves in time.

Lemma.[37] The number of dislocations in an arbitrary CA state evolving under rule 18 decreases or remains constant in time. Furthermore, for all states except those that evolve immediately to the null state, dislocations always annihilate in pairs.

Proof: (Outline) From the CA rule table. There are three cases, each concerning a spatially-local configuration: (i) no dislocations, (ii) an isolated dislocation, and (iii) two or more nearby dislocations that may or may not collide during the next time step. The first case has been addressed by showing that Λ^0 is an invariant language: no dislocations are created in time. The single-dislocation case is proven by enumerating all local neighborhoods that contain one dislocation. The result is that the dislocation moves to the left or right, but neither disappears nor multiplies. Thus, when Ω_0 consists of configurations in Λ^0 or $\Lambda^{0,0}$, no dislocations are created in time. Case (iii) is proven as follows. The local action of the update rule ensures that unless the dislocations' closest boundaries are separated by less than $2r + 1 = 5$ cells, the dislocations do not interact, and can be treated under case (ii) above. The sole exception to this is when two dislocations and the cells between them are mapped to a region of consecutive 0's, in which case they annihilate and the number of dislocations decreases by two. Enumeration

of all 5-cell neighborhoods shows that the number of dislocations either remains the same or decreases by an even number. ▌

In addition, the domain walls appear to execute a random walk on the lattice, as we describe in detail in the next section. This suggests a stochastic model of dislocation motion, in which they perform independent random walks on the lattice until they collide, at which time they annihilate. The stochastic model is further motivated by a proof that the motion of a single dislocation in rule 18 performs a random walk.[38] The classical model of diffusing annihilating particles was studied by Griffeath,[24] who proved that with probability 1 the density of dislocations asymptotically falls to zero. In essence, diffusive motion ensures that almost every dislocation eventually will collide with another and the pair will annihilate. On finite lattices, the dislocations annihilate as far as possible. For infinite lattices the random walk implies that the limiting set of states with a positive dislocation density has vanishing size relative to the full state space.

The consequence of diffusing dislocations for rule 18, given sufficiently large and typical N, is that almost every initial configuration in $\mathcal{A}^N$ eventually goes to a regular attractor: either to Λ^0, if N is even, or to $\Lambda^{0,0}$, if N is odd. Requiring that N be typical excludes the sparse set of lattices for which 0^* is an important attracting invariant set. There is an even number of dislocations in any initial condition on an even lattice. And so with probability 1, the state will eventually evolve into Λ^0 on an even lattice. If N is odd, the number of dislocations is always odd due to pairwise annihilation. And so, almost all initial conditions on an odd lattice eventually fall into $\Lambda^{0,0}$. The same argument holds for the case when N is infinite, but the initial condition has a finite number of dislocations. On infinite lattices with an infinite number of dislocations the dislocation density asymptotically falls to zero, and hence any finite subsequence in the state falls into Λ^0 with probability 1. Under the assumption of diffusive motion, then, Λ^0 and $\Lambda^{0,0}$ are the only two statistically significant attractors.

The central conclusion to be drawn from this discussion is that dislocations annihilate, leading eventually to simpler, *not* more complex, spatial patterns. The regular language complexity in the conventional "global" analysis[4] blows up due to the complexity of the motion of all possible dislocations implicit in the initial ensemble of all possible configurations. There are many routes of evolution onto the attractors, and thus very large machines will persist for arbitrarily long times. The exact topological analysis necessarily includes these, but is unfortunately also dominated by them.† Consequently, these machines obscure important features such as the *typicalness* of the patterns observed and the simple decomposition into domains and walls. The simple, observed fact is that the number of dislocations is almost always decreasing. This means that ultimately the language, for all initial conditions but a vanishingly small set, asymptotically becomes quite simple.

†We note that the direct probabilistic generalization of the FME algorithm also obscures this structure, since it carries along the full topology in the support and also is sensitive to all variations in the initial configuration distribution.[39]

There is undoubtedly a deep connection between the apparent diffusivity of the dislocations and the structure of the attractors Λ^0 and $\Lambda^{0,0}$. The exact nature of that connection, and the particular mechanism by which the randomness of the initial ensemble is transformed into space-time random walks, remain open questions. Nevertheless, the greater part of the behavior of rule 18 is described by the picture of computationally simple domains bounded by dislocations that wander about at random, annihilating as they collide.

4.1 Transducers for Global Structure Analysis

Given this identification of the gross behavior of rule 18, we can organize our discussion around the observed domains and the dislocations between them. In doing so, we will find use for three different transducers, all based on the domain language Λ^0: the machine metric $\|\mathrm{s}\|_{\mathrm{M}(\Lambda^0)}$ that counts the number of dislocations in a spatial state s, an intercalation operator $\mathrm{T}_{\mathrm{intercalate}}(\Lambda^0)$, and a domain isomorphism $\mathrm{T}_{\mathrm{decimate}}(\Lambda^0)$. The machine metric was introduced above in a more general context. The remaining two transducers are defined here.

The site-intercalation transducer $\mathrm{T}_{\mathrm{intercalate}}(\Lambda^0)$, shown in Fig. 9, is a machine for removing dislocations by adding extra sites to the lattice. Within a domain it is the identity on all transitions, i.e. $0|0$ and $0|1$, but when a wall is detected, we put $1|01$. The result is a many-to-one mapping from arbitrary patterns to patterns in Λ^0. By inserting the additional 0, the wall is effectively removed. The mapping $f_{\mathrm{T}_{\mathrm{intercalate}}(\Lambda^0)} : \mathcal{A}^* \to \Lambda^0$ is equivalent to the global linearization operator $G(\mathrm{s})$ of ref. [36]. Since the evolution of Λ^0 is identical in rule 18 and rule 90, this operator linearizes rule 18 in the sense that the rule becomes equivalent to the linear, superposition-preserving dynamics of rule 90.

This type of global linearization applies only to patterns consisting of domains of the same type, i.e., the same regular attractor. For CA with more than one domain type $\{\Lambda^i\}$, such global linearization does not exist over the entire state space. Rather it appears only on subspaces composed of configurations with domains of the same type.

The domain isomorphism transducer $\mathrm{T}_{\mathrm{decimate}}(\Lambda^0)$ (Fig. 10(a)) demonstrates in a transparent manner that rule 18 acting on Λ^0 is equivalent under decimation to rule 90 acting on $\mathcal{A}^*$. Its effect is to drop the (predictable) 0 transition from the input pattern. This can also be considered an encoding and is, in fact, a compression of the input pattern. The corresponding function $f_{\mathrm{T}_{\mathrm{decimate}}(\Lambda^0)} : \Lambda^0 \to \mathcal{A}^*$ maps the domain patterns of rule 18 onto the full extended state space. It is important to point out that, in general, decimation of space-time patterns requires both a spacelike machine to decimate spatial configurations and a timelike machine to decimate in time. The proper framework for this requires introducing space-time machines. We defer this discussion to later,[35] since pursuing it here would take us too far afield. In this particular example the issue can be effectively skirted because the space, time, and space-time machines are identical.

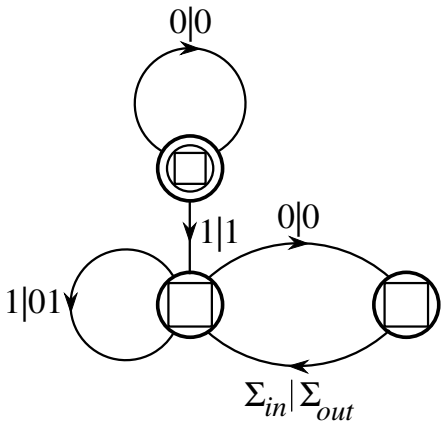

Fig. 9. Transducer $\mathrm{T}_{\mathrm{intercalate}}(\Lambda^0)$ for inserting new sites at dislocations. The result of such an operation is a global linearization of rule 18. Note that when a dislocation is detected two symbols, rather than one, are emitted. The transducer takes length-N strings to length-$(N+d)$ strings, where d is the number of dislocations in the input string.

One must also look at the effect of $f_{\mathrm{T}_{\mathrm{decimate}}(\Lambda^0)}$ on the rule table. From this we find the machine commutation relation

$$\mathrm{T}_{90} \circ \mathrm{T}_{\mathrm{decimate}(\Lambda^0)} = \mathrm{T}_{\mathrm{decimate}(\Lambda^0)} \circ \mathrm{T}_{18}^2$$

In terms of languages, direct comparison of the space-time patterns shows that the action of rule 18 on neighborhoods of size five after two time steps is identical to rule 90 acting on neighborhood of size three after one time step, when one drops intervening zeros:

$$\begin{array}{ccccccccc} \cdots & 0 & s_t^{i-2} & 0 & s_t^i & 0 & s_t^{i+2} & 0 & \cdots \\ \cdots & s_{t+1}^{i-3} & 0 & s_{t+1}^{i-1} & 0 & s_{t+1}^{i+1} & 0 & s_{t+1}^{i+3} & \cdots \\ \cdots & 0 & s_{t+2}^{i-2} & 0 & s_{t+2}^i & 0 & s_{t+2}^{i+2} & 0 & \cdots \end{array} \quad \begin{array}{c} \overrightarrow{\mathrm{T}_{\mathrm{decimate}}} \\ \\ \overrightarrow{\mathrm{T}_{\mathrm{decimate}}} \end{array} \quad \begin{array}{ccccc} \cdots & s_t^{i-2} & s_t^i & s_t^{i+2} & \cdots \\ \cdots & s_{t+2}^{i-2} & s_{t+2}^i & s_{t+2}^{i+2} & \cdots \end{array}$$

where $s \in \mathcal{A}$. In other words, we have the language commutation relation

$$\Phi_{90} \circ f_{\mathrm{T}_{\mathrm{decimate}(\Lambda^0)}}(\Lambda^0) = f_{\mathrm{T}_{\mathrm{decimate}(\Lambda^0)}} \circ \Phi_{18}^2(\Lambda^0)$$

Thus, not only is there an equivalence on patterns, but also on the action of the rules. The transducer implements this directly, being the projection operator for the associated commuting diagram

$$\begin{array}{ccc} \Lambda^0 & \overset{\Phi_{18}^2}{\rightarrow} & \Lambda^0 \\ f_{\mathrm{T}_{\mathrm{decimate}}(\Lambda^0)} \downarrow & & \downarrow f_{\mathrm{T}_{\mathrm{decimate}}(\Lambda^0)} \\ \mathcal{A}^* & \underset{\Phi_{90}}{\rightarrow} & \mathcal{A}^* \end{array}$$

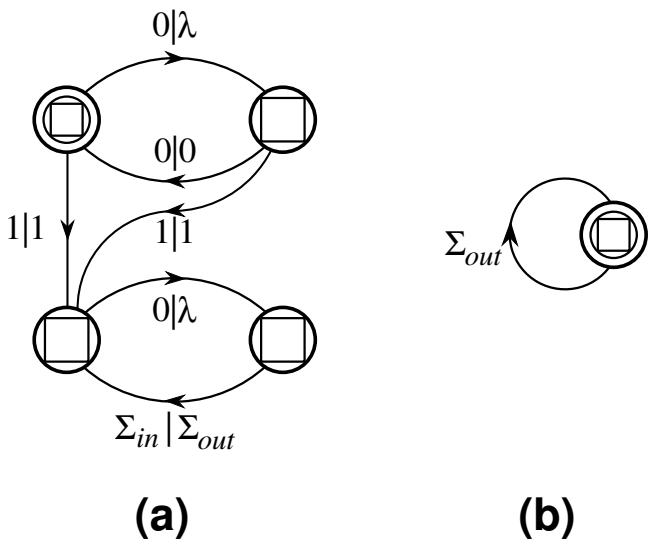

Fig. 10. (a) Transducer $\mathrm{T}_{\mathrm{decimate}}(\Lambda^0)$ for removing, in fact compressing or encoding, the patterns in Λ^0 into $\mathcal{A}^*$. The two transient states ensure that the machine synchronizes properly to strings beginning with any number of consecutive 0's. (b) The transducer has been minimized with respect to its output. The resulting automaton accepts $\mathcal{A}^*$.

If we minimize it with respect to the output alphabet, then we obtain $\mathcal{A}^*$, which is invariant under rule 90. One consequence is that rule 18 is linear on the domain in the sense that $\mathbf{\Phi}(\Lambda^0)$ obeys a linear superposition in space-time. (See also ref. [36].)

We can use the inverse decimation operator $f^{-1}_{\mathrm{T}_{\mathrm{decimate}}(\Lambda^0)}$ to enumerate a cascade of unstable periodic sublanguages embedded in Λ^0. If we think of the latter as defined by the rule "every other site is arbitrary and the rest are zero", then these sublanguages are "every (2^k)th site is arbitrary and the rest are zero", assuming, of course, that $k > 0$. These are regular languages, which are denoted $\Lambda^0_k,\ k > 0$. In this notation $\Lambda^0 = \Lambda^0_1$. Clearly $\Lambda^0_{k+1} \subset \Lambda^0_k$. They are the successive preimages of Λ^0 under $f_{\mathrm{T}_{\mathrm{decimate}}(\Lambda^0)}$; that is,

$$\Lambda^0_{k+1} = f^{-1}_{\mathrm{T}_{\mathrm{decimate}}}(\Lambda^0_k),\ k > 0$$

For all positive k, Λ^0_k is a periodic language with temporal period 2^{k-1}. Each language orbit

$$\left\{\mathbf{\Phi}_t(\Lambda^0_k)\ :\ 0 \le t < 2^{k-1},\ k > 0\right\}$$

is easily shown to be a domain. When the orbit recurs the origin is shifted by 2^{k-1} sites. Finally, $\Lambda^0_k, k > 1$, are unstable to unit Hamming perturbations and decay to Λ^0.

4.2 Basins and Separatrix Structure

The basin of Λ^0 consists of those configurations, aside from Λ^0 itself, that have an even number of dislocations, but do not converge to temporally periodic or quasiperiodic states

containing dislocations. That is,

$$\mathcal{B}\left(\Lambda^0\right) = \Lambda^0 \bigcup \left\{ \mathbf{s} \in \mathcal{A}^N : \|\mathbf{s}\|_{\mathrm{M}(\Lambda^0)} = 2n, n = 1, 2, \ldots, \text{ and } \lim_{t \to \infty} \|\Phi_t \mathbf{s}\|_{\mathrm{M}(\Lambda^0)} = 0 \right\}$$

Similarly, for $\Lambda^{0,0}$ we have

$$\mathcal{B}\left(\Lambda^{0,0}\right) = \Lambda^{0,0} \bigcup \left\{ \mathbf{s} \in \mathcal{A}^N : \|\mathbf{s}\|_{\mathrm{M}(\Lambda^0)} = 2n+1, n = 1, 2, \ldots, \text{ and } \lim_{t \to \infty} \|\Phi_t \mathbf{s}\|_{\mathrm{M}(\Lambda^0)} = 1 \right\}$$

The preceding analysis indicated that Λ^0 and $\Lambda^{0,0}$ are the only two important attractors. As we are concerned with typical behavior, we collect into the global separatrix $\mathcal{S}$ all spatial states that do not evolve into Λ^0 or $\Lambda^{0,0}$. For example, we include in $\mathcal{S}$ any other attractors since, if they exist, the numerical results indicate that they have vanishing size relative to Λ^0 and $\Lambda^{0,0}$. We can formally write this in set-theoretic notation as

$$\mathcal{S} = \mathcal{A}^N - \mathcal{B}\left(\Lambda^0\right) - \mathcal{B}\left(\Lambda^{0,0}\right)$$

It is also the set of states for which the number of dislocations remains greater than 1 forever. As we noted above, one set for which this is true is the set of temporally periodic states with more than one dislocation. Examples of this are readily obtained by looking at periodic lattices with only a few cells. For instance, the spatial configuration $(1000010000100010)^*$, with two dislocations in the basic pattern, when iterated on periodic lattices with $N = 16m$, $m \geq 1$, or N infinite, has temporal period $p = 14$. However, all such languages are also *spatially* periodic, and hence have zero size. For an infinite lattice there are also quasiperiodic orbits, in which, for example, two dislocations (i) travel away from each other forever or (ii) translate in parallel with a roughly constant or periodic separating distance.

We identify three disjoint subsets of the separatrix $\mathcal{S} = \mathcal{S}_{\mathrm{even}} \bigcup \mathcal{S}_{\mathrm{odd}} \bigcup \mathcal{S}_{\infty}$. $\mathcal{S}_{\mathrm{even}}$ contains those periodic orbits, and their preimages, with an even number of dislocations greater than one; $\mathcal{S}_{\mathrm{odd}}$ has an odd number greater than one. $\mathcal{S}_{\infty}$ contains all the aperiodic, quasiperiodic orbits and their preimages.

The foregoing analysis specifies the attractor-basin portrait. The remaining problems include identifying the complexity class of the basins and separatrices and giving their finite descriptions, if such exist. We leave these to the future.

4.3 The Attractor-Basin Portrait

We gather together the preceding results for an essentially graphical display of the global view of the attractor basin (AB) portrait of rule 18. We present two views. The first is somewhat skeletal and is the picture provided by regular language space. The second is schematic and gives a cartoon of the state space of arbitrary but high dimension.

From the viewpoint of regular language space the AB portrait is relatively simple. We can identify the basic sets in a skeletal AB portrait in which the vertices of the graph are various languages. A skeletal picture of rule 18's attractor-basin portrait as seen in regular language space is given in Fig. 11. The various symbols indicate the formal languages identified in the computation theoretic analysis: attractors $\{\Lambda^0, \Lambda^{0,0}\}$, basins $B^0 = \mathcal{B}(\Lambda^0)$ and $B^{0,0} = \mathcal{B}(\Lambda^{0,0})$, and separatrices $\mathcal{S} = \{\mathcal{S}_{\rm even}, \mathcal{S}_\infty, \mathcal{S}_{\rm odd}\}$. The temporal evolution from one language to another goes down the figure. The context of this diagram is a lattice of arbitrary typical size N, either infinite or finite with periodic boundary conditions. If N is odd, then only the portion to the right of the right vertical dashed line is observed. If N is even, then only that to the left of the left dashed line is relevant. On an infinite lattice the entire diagram is present. The vertical dashed lines also denote the extent to which small perturbations can affect states in the attracting languages. A unit Hamming distance perturbation $\delta_r \Lambda$ from the attractor drives the state to its basin or, at worst, to a periodic orbit in the associated subset of the separatrix. In fact, topological constraints prohibit bit-flip perturbations within the nonzero support from driving the configuration to the other attractor. If the lattice is infinite and the initial patterns are restricted to a finite support of nonzero site values, then there are no separatrices, since all states eventually collapse onto one or the other attractor in finite time.[36] Unit Hamming distance perturbations outside the region of nonzero sites then can introduce single dislocations and so move from B^0 to $B^{0,0}$, and visa versa. The subseparatrices $\mathcal{S} = \{\mathcal{S}_{\rm even}, \mathcal{S}_\infty, \mathcal{S}_{\rm odd}\}$ are disjoint. $\mathcal{S}_{\rm even}$ and $\mathcal{S}_{\rm odd}$ consist of temporally periodic patterns with an even and odd number of dislocations greater than one asymptotically in time, respectively. $\mathcal{S}_\infty$ contains all aperiodic patterns with more than one dislocation, and all patterns with an infinite number of dislocations, in the limit $t \to \infty$. As noted at the outset of this section, we omit from the diagram (i) behavior on the sparse set of atypical lattices for which 0^* is an important attracting invariant set and (ii) on the remaining lattices, the zero-sized set $\mathcal{B}(0^*)$. The nonattracted sets of zero size are collected in the separatrices.

But there is much more that can be said about the configuration space itself. Generally, local particle annihilation dynamics induces a hierarchical structure of subbasins and portals.[40] The subbasins are subspaces in which the number of dislocations is constant. Thus, they are level sets of the machine metric: $b_d = \left\{ {\rm s} : \|{\rm s}\|_{{\rm M}(\Lambda^0)} = d \right\}$. They are connected by portals which are defined solely in terms of the particle annihilation geometry. Since the number of dislocations cannot increase, there is a flow directed from higher- to lower-dimensional subspaces. Here by "dimension" we refer to the number of "active" degrees of freedom, which we take to be the number of dislocations. In the transition from a d-subbasin b_d, caused by two dislocations annihilating, the number of dislocations decreases by two. And so the state moves from a d-subbasin to a $(d-2)$-subbasin. It is also possible for more than one pair of dislocations to annihilate in one time step, in which case the state falls into a $(d-2m)$-subbasin, where m is the number of annihilating pairs. The subbasin-portal structure is shown schematically in Fig. 12.

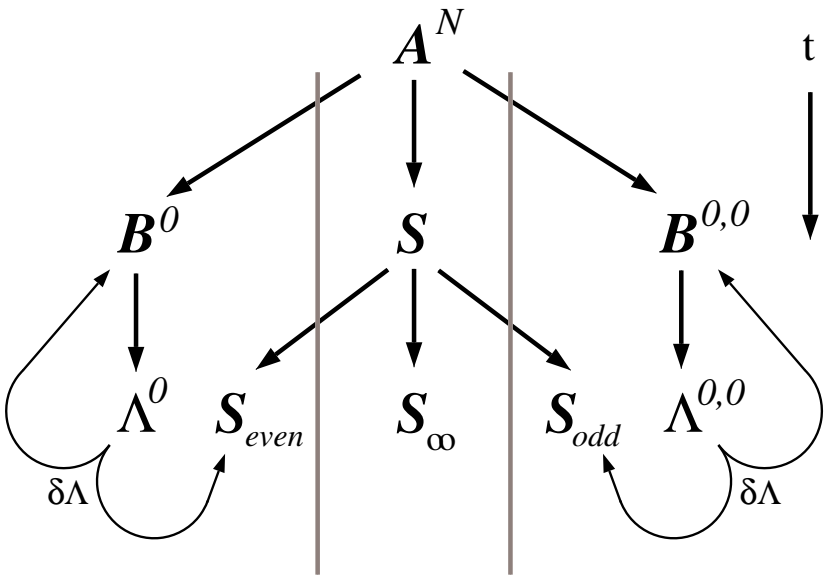

Fig. 11. A skeletal picture of rule 18's attractor-basin portrait as seen in language space. The various symbols indicate the formal languages identified in the computation-theoretic analysis. The temporal evolution from one language to another goes down the page. The thin lines with arrows indicate the effect of perturbations $\delta\Lambda$ away from the attractors.

The intercalation operator $f_{\mathrm{T}_{\mathrm{intercalate}}(\Lambda^0)}$ maps the configuration space of an N-site lattice containing d dislocations to the domain subspace of an $(N+d)$ lattice on which the rule is linear.[36] This gives an additional hierarchical constraint to the state space.

It is important to point out that the state space structures implied by the portal concept and the intercalation operator are not the same. Portals, on the one hand, are local in the state space and their geometric structure is entirely determined by the local "real" space constraints on particle annihilation. The intercalation operator, on the other hand, is global in both the state and "real" spaces. The result is that the portals still organize the state space of multiple distinct-domain dynamics, a situation in which the global intercalation operator does not exist. Indeed, the underlying views behind each are complementary. Understanding their relationship is certain to lead to an even more detailed picture of the state space geometry of domain-wall-dominated CA than that presented in this section.

5 Dislocation Trajectories

This section presents a concrete application of the preceding ideas. We give a construction procedure that builds detectors for propagating space-time structures in domain-wall-dominated CA. The application uses these transducers to find dislocations in rule 18 in order to investigate their statistical behavior in detail. We are interested specifically in the diffusive behavior of an isolated dislocation and of a dislocation gas. The results strongly suggest that rule 18 is a nearly

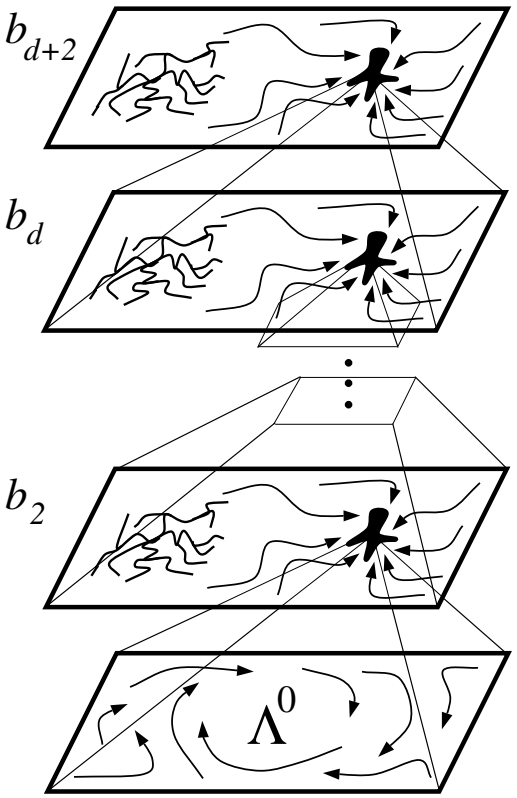

Fig. 12. The state-space view of the attractor-basin portrait of rule 18 for an even length lattice or, equivalently, the left-most portion of Fig. 11. The diagram illustrates the subbasin-portal structure of the state space. One subbasin corresponds to each horizontal plane. Within each subbasin b_d, the number of dislocations d is constant. The portal is indicated by the solid dark blob. When pairs of particles annihilate, the state passes through the portal. Its shape is determined by the geometric constraints of annihilation. That portion of the separatrix consisting of configurations with the subbasin's number of dislocations is shown as a filamentary structure to denote its vanishing relative size. The general flow of time is down the figure. Asymptotically in time and lattice size, the overwhelming fraction of initial patterns in V^0 end up on the attractor Λ^0, all of the dislocations having annihilated.

ideal diffusion process with annihilating particles. This study of the statistical properties along with the dynamical systems analysis gives a very complete picture of rule 18.

5.1 Detection

We can recode space-time data to show the dislocation trajectories generated by rule 18 by taking the domain-machine $\mathrm{M}(\Lambda^0)$ and defining a transducer $\mathrm{T}_{\mathrm{domain}}(\Lambda^0)$ that emits a (domain) symbol D on allowed transitions of Λ^0, and a (wall) symbol W on disallowed transitions. This dislocation recognizer is shown in Fig. 13. In general, for an arbitrary set of domain types $\{\Lambda^i\}$, it is straightforward to construct a transducer that will encode the data to show each of the domain and wall structures.

For rule 18 running $\mathrm{T}_{\mathrm{domain}}(\Lambda^0)$ over the space-time data gives an encoded picture that shows only the dislocation trajectories. Figures 14—16 are typical of simulations starting with a

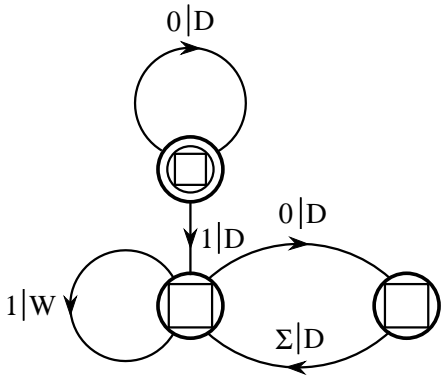

Fig. 13. Transducer $\mathrm{T}_{\mathrm{domain}}(\Lambda^0)$ for recognizing dislocations in rule 18 domain structure Λ^0.

random initial condition. Each of these pictures shows a different space-time region of the same simulation. Note the progressively reduced scale in Figs. 14 and 15 used to make the diffusive behavior more apparent. The thickness of the lines is due to the figure's finite resolution. In Fig. 16, blackened squares correspond to dislocations and hollow squares show the nonzero cells of the original pattern.

The transducer is run over the spatial data in a particular direction: in our case, from left to right. This introduces a spatial asymmetry into the picture. For example, the visible asymmetry in the trajectory at the top of a triangle of white space (Fig. 16) is due to the fact that as $\mathrm{T}_{\mathrm{domain}}(\Lambda^0)$ is moved to the right over the data, it identifies a dislocation only by reading a 1 in the wrong place. Thus, a 1 followed by an even number of consecutive 0's shows a dislocation at the right side, when the next 1 is encountered. This has the effect of a constraint on the motion of the observed dislocation: all moves to the left are a single cell, but moves to the right may be any odd-valued distance. Further, a move left may be followed by a move either to right or left, but a move right must be followed by one to the left. More complicated transducers can be defined to evade this directionality. For example, the transducer could label all blocks $1(00)^n1, \;\; n = 0, 1, 2, \ldots$, as walls. With this detector, the dislocation becomes delocalized as it enters a triangle of 0's. The associated detector for bounding the delocalization region is more complex, requiring an infinite queue (FIFO) memory, if it is to not only recognize dislocations but also to label all participating sites appropriately.

Given a machine that will encode the space-time data to recognize dislocations, it is possible to examine the encoded data for other, higher-order regularities. By designing the transducer properly, one can use this process to identify structures of arbitrary regular language complexity. The result is the original language "modulo" the transducer machine. In this all the encoded structures are excised, giving a clearer picture of long-range structures. This technique is applicable to a large class of systems, including those which possess regular domains. It also generalizes to structures of higher complexity, using recognizers based (say) on stack transducers that implement context-free or context-sensitive grammars for recognition.