Proposition. For any regular language $\mathcal{L}$, the Hamming vicinity $V_H(\mathcal{L})$ and the restricted Hamming vicinity $V_r(\mathcal{L})$ are both regular vicinities; i.e., $V_H(\mathcal{L}) \in \mathcal{R}$ and $V_r(\mathcal{L}) \in \mathcal{R}$ for all $\mathcal{L} \in \mathcal{R}$.

Proof: $V_H(\mathcal{L})$ is generated by machine composition of $\mathrm{M}(\mathcal{L})$ with T_{V_H} shown in fig. 8.1(a), followed by minimizing with respect to the output: $\mathrm{M}(V_H(\mathcal{L})) = [\mathrm{T}_{V_H} \circ \mathrm{M}(\mathcal{L})]_{\mathrm{out}}$. By construction, $\mathrm{M}(V_H(\mathcal{L}))$ is a finite automaton, and so $V_H(\mathcal{L})$ is a regular language. The proof for $V_r(\mathcal{L})$ is identical. ▌

Note that T_{V_H} and T_{V_r} are irreducibly nondeterministic finite state transducers. This does not pose any problem in the stability analysis of a language $\mathcal{L}$, since the Hamming vicinity transducers are not used to generate individual states, but to generate the vicinity as a whole via the construction $\mathrm{M}(V(\mathcal{L})) = [\mathrm{T}_V \circ \mathrm{M}(\mathcal{L})]_{\mathrm{out}}$. Both machine composition and minimization with respect to the output can be carried out regardless of whether the transducer involved is deterministic or nondeterministic.

8.2 Regular Attractor: Definitions

Having set up the necessary geometrical prerequisites, it is now possible to define a regular attractor of a CA. The intuitive notion of a regular attractor is of a regular language A (such as, for example, a domain) that is stable under small perturbations—i.e., states in some vicinity $V(A)$ eventually fall back into A. The original definition of regular attractor appeared in ref. [47], and is reproduced as definition (1) below. Since that time, further reflection has shown the desirability of improvements on the original definition. One proposed modification follows as definition (2).

It bears repeating here that the notion of attractor defined in this section is of an attracting *language*—i.e., a pattern. This is fundamentally different from the definition of an attractor as a cycle on a finite lattice in the "transients and cycles" analysis (section 2.3), or as a set of points in the infinite-dimensional state space (section 3.7). Neither of the latter possess any notion of spatial structure shared by configurations on an attractor.

Definition (1). An invariant language A is a **fixed-point regular attractor** if it is a regular language and there exists a regular vicinity $V(A)$ such that

i. All states in V remain in V forever: $\mathbf{\Phi}^t(V) \subseteq V$, $\forall\, t \geq 0$; and
ii. All states in V, except possibly for a set R of zero size, i.e., $S(R|V) = 0$, eventually fall back into A. Thus for almost all $\mathrm{s} \in V$, there exists a $t \geq 0$ such that $\Phi^t(\mathrm{s}) \in A$.

Letting a set of zero size remain outside the attractor indefinitely is absolutely essential to domain-wall dominated CA, as we will see. This is analogous to the existence in continuous-state dynamical systems of unstable or marginally stable fixed points and periodic orbits near 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.

On further examination, it became evident that definition (1) was unnecessarily restrictive. For example, although the Hamming vicinity $V_H(A)$ provides the most compelling notion of "small" perturbation, it does not necessarily satisfy condition (i), even in cases where perturbed states eventually fall back into A. The iterates $\mathbf{\Phi}^t(V_H(A))$ need not remain in $V_H(A)$, but instead may evolve into some larger vicinity that might or might not be regular. This is not unlike the situation in continuous-state dynamical systems, in which a small perturbation of magnitude δ away from a fixed point x_0 remains within some larger region of size $\varepsilon \geq \delta$ about x_0 (see section 3.3). We therefore relax condition (i) slightly to arrive at an alternative definition:

Definition (2). An invariant language A is a **fixed-point regular attractor** if it is a regular language and there exist regular vicinities $V_1(A)$ and $V_2(A)$ such that

i. $V_1(A) \subseteq V_2(A)$;
ii. $V_2(A)$ is a trapping region: $\mathbf{\Phi}(V_2) \subseteq V_2$; and
iii. All states in $V_1(A)$, except possibly for a set R such that $S(R|V_1) = 0$, eventually collapse down onto A. Thus for almost all $\mathrm{s} \in V_1$, there exists a $t \geq 0$ such that $\Phi^t(\mathrm{s}) \in A$.

Conditions (i) and (ii) imply that iterates of $V_1(A)$ remain in $V_2(A)$ for all time: $\mathbf{\Phi}^t(V_1) \subseteq V_2, \ \forall \, t \geq 0$. This is the direct analog of Lyapunov stability, with V_1 being the set of states perturbed away from the invariant set A, and V_2 serving as the larger vicinity containing all the iterates of V_1. Condition (iii) is then the analog of asymptotic stability, with the difference that a set of zero size is allowed to remain outside the attractor.

Even this definition must be regarded as imperfect, however. Although it captures the notion of stability to the extent allowed by the discrete CA state space, the use of two nested vicinities is problematic. This is made evident by the realization that the entire extended state space $\mathcal{A}^*$ is a regular vicinity of *any* regular language. Since $\mathbf{\Phi}(\mathcal{A}^*) \subseteq \mathcal{A}^*$ for any CA rule $\mathbf{\Phi}$, conditions (i) and (ii) can always be trivially satisfied by taking $V_2 = \mathcal{A}^*$ regardless of $\mathbf{\Phi}$, A and V_1. The content of definition (2) is then effectively reduced to the single condition (iii): that almost all states in V_1 must fall back into A. In practice, we will choose as small a vicinity as possible for V_2, though again, at the root of this problem is the lack of a good definition of "small".

The definition may prove to be more useful in CA with more than one attractor. In such cases we could require that the two vicinities be disjoint, i.e. that two invariant languages A_0 and A_1 are *resolvable* regular attractors if there exist vicinities $V_2(A_0)$ and $V_2(A_1)$ such that $V_2(A_1) \cap V_2(A_2) = \emptyset$. Further development and testing against many more example systems are clearly necessary.

Also conspicuously missing from both definitions (1) and (2) is any requirement that an attractor be *transitive*—i.e., that it cannot be partitioned up into disjoint subsets that are themselves attractors. Again, when the state space is discrete, it is hard to even formulate a satisfactory notion of transitivity. On a finite lattice, the "transients and cycles" analysis of section 2.3 indicates that all asymptotic states lie on temporally periodic cycles, which (if there is more than one) are disjoint. This might suggest that only the cycles can be considered as attractors, as, in fact, they have often been defined.[46,66,108] Yet the number and periodicity of the cycles changes drastically with lattice size; furthermore, there is no notion of *pattern* in that view. All hope of describing the generic behavior of a given CA on an arbitrary lattice is lost, thus abandoning the very essence of the analysis of CA pattern dynamics. We regard this as unsatisfactory.

It was originally supposed that the discreteness of the state space would simplify the analysis considerably. Instead we find that it has made formulating satisfactory definitions of certain standard notions of dynamical systems theory—i.e., infinitesimal perturbation and transitivity—more difficult, especially when coupled with the absence of a preferred spatial origin.

On the other hand, in the CA which we have analyzed (see chapter 12, for example), our definition of regular attractor correctly identifies the system's asymptotic behavior, and provides a compact and powerful characterization of the system's qualitative dynamics. At this point, we are therefore left with the realization that definition (2), though incomplete, is demonstrably useful—and that it can only become more so if a fully appropriate way of completing it is found.

8.3 Other Structures

Finally, we note that the above definition of regular attractor, provisional as it is, can be used as a foundation 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**:

$$\mathcal{B}(\Lambda) = \bigcup_{t \geq 0} \mathbf{\Phi}_{-t}(\Lambda). \tag{8.3}$$

The set of states that are not in basins, but still in the accessible region of configuration space make up the **separatrices**:

$$\mathcal{S} = \mathcal{A}^N - \bigcup_{\Lambda \in \{\Lambda^i\}} \mathcal{B}(\Lambda) \tag{8.4}$$

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 ECA 18 is the subject of chapter 12.

9 Pattern Quantifiers

Once the domains of a given CA have been identified using machine reconstruction and the FME algorithm, the next logical step is to describe them. Because domains are basic structures organizing the state space, statistics that are misleading or meaningless when taken over the entire state space (such as, typically, the relative frequency of individual symbols or the entropy rate) are well-defined on domains or other invariant sets. This chapter introduces a number of such *pattern quantifiers*, as well as some simple numerical and graphical measures of the effect of perturbations on the domains, and of the interfaces between domains.

9.1 Entropy and Complexity

Two of the most important properties of a domain, the spatial entropy rate and the complexity, can be directly calculated from its finite automaton representation. From the structure of the process graph, we can obtain exact values of the topological entropy $h_0(\Lambda)$ and the topological complexity $C_0(\Lambda)$ of the (spatial) domain language Λ. If we further estimate the machine state transition probabilities by (for example) performing machine reconstruction on a large sample of domain configurations, we can calculate the metric entropy $h_1(\Lambda)$ and the statistical complexity $C_1(\Lambda)$. Definitions and methods for calculating these four quantities were given in section 4.5.

The entropy h_0 measures the exponential growth rate in the number of words of a given length in a language; h_1 further takes the word probabilities into account. For a domain language Λ, therefore, they measure the degree of spatial disorder. Spatially periodic domains have $h_0 = h_1 = 0$, since the number of words grows linearly with length. At the other end of the spectrum, a fully random domain $\Lambda = \mathcal{A}^*$ with uniform branching probabilities on the stochastic machine $\mathrm{M}(\mathcal{A}^*)$, has maximal entropy: $h_0 = h_1 = \log k$, where $k = \|\mathcal{A}\|$.

The complexities C_0 and C_1 measure a complementary property of the domain. $C_0(\Lambda)$, given by the logarithm of the number of states in the process graph of Λ, measures the amount of spatial memory in the domain—in other words, the number of bits of information required to specify the phase of a cell. C_1 augments this by weighting machine states according to their probability.

9.2 Single-site Statistics

The next pair of quantities we define are related to the single-site statistics of states in the ensemble. They provide a contrast with the entropies defined in the previous section, in that the latter characterize the statistics of *long* blocks of symbols, while the former describe the statistics of *short* blocks; i.e., blocks of length one, or single sites. The quantifiers here are defined with respect to a particular "quiescent" symbol $\hat{\sigma} \in \mathcal{A}$ (see section 2.1). Again, there are topological and metric versions.

The metric quantity, $\rho_1(\mathcal{L})$, is the fraction of cells with nonquiescent values, measured over all states in the ensemble. For a finite language ($\|\mathcal{L}\| < \infty$) such as an ensemble of CA states in a numerical experiment, this is given by

$$\rho_1(\mathcal{L}) = \frac{\sum\limits_{\omega \in \mathcal{L}} \sum\limits_{i=0}^{|\omega|-1} \Theta(\omega_i)}{\sum\limits_{\omega \in \mathcal{L}} |\omega|} \tag{9.1}$$

where ω_i is the $i\mathrm{th}$ symbol of ω, $\Theta(\hat{\sigma}) = 0$, and $\Theta(\sigma) = 1$ for all $\sigma \neq \hat{\sigma}$. When $\mathcal{L}$ is infinite, the average is first made over all words $\omega \in \mathcal{L}$ up to finite length $|\omega| = L$, then the limit $L \to \infty$ is taken. In keeping with the common use of these statistics in numerical experiments, we will use the form for finite $\mathcal{L}$ throughout, with the understanding that the appropriate modifications will be made for $\mathcal{L}$ infinite. Note that writing $\rho_1(\mathcal{L})$ in terms of the indicator function $\Theta(\sigma)$ makes the generalization to other "quiescent" symbols besides $\hat{\sigma} = 0$ straightforward.

The corresponding topological quantity, $\rho_0(\mathcal{L})$, is the fraction of nonquiescent cells in the CA lookup table outputs of parent neighborhoods occurring in $\mathcal{L}$. The set $\Pi_{\mathcal{L}}$ of parent neighborhoods in $\mathcal{L}$ is identical to the set of words in $\mathcal{L}$ of length $2r + 1$, where r is the radius of the CA rule:

$$\Pi_{\mathcal{L}} = \{\omega :\ \omega \in \mathcal{L},\ |\omega| = 2r + 1\} \tag{9.2}$$

Then $\rho_0(\mathcal{L})$ is the just fraction of the $\Pi_{\mathcal{L}}$ that map to nonquiescent values under the rule:

$$\rho_0(\mathcal{L}) = \frac{\sum\limits_{\substack{\sigma \in \mathcal{A} \\ \sigma \neq \hat{\sigma}}} \eta_{\mathcal{L}}(\sigma)}{\sum\limits_{\sigma \in \mathcal{A}} \eta_{\mathcal{L}}(\sigma)} \tag{9.3}$$

where $\eta_{\mathcal{L}}(\sigma)$ is the number of parent neighborhoods in $\Pi_{\mathcal{L}}$ that map to the value σ under the CA rule ϕ. If all possible parent neighborhoods occur in the ensemble $\mathcal{L}$—i.e., $\mathcal{A}^{2r+1} \subset \mathcal{L}$—then $\rho_0(\mathcal{L})$ is equivalent to the "lambda parameter" defined in ref. [69]. Our choice of notation for these quantities emphasizes the ensemble-dependence of their definitions and differentiates them from the "Lyapunov exponents" defined below. In measuring the single-site statistics of some pattern $\mathcal{L}$, it is important to keep in mind that $\rho_0(\mathcal{L})$ does not reflect the pattern's evolution, and so may or may not be a relevant quantity, and that $\rho_1(\mathcal{L})$ is not indicative of the pattern's structural regularity.

9.3 Difference Patterns

One graphical measure of the structure of information flow in a CA is the *difference pattern*.[105] Here we introduce two important modifications to this technique. We refer to the resulting method as domain-dependent difference patterns. The consequences are striking and demonstrate the important structural role played by regular domains, as we will see in chapter 11.

To generate a difference pattern, we start with a given "fiducial" state $\mathrm{s}_0 = \ldots\sigma_0^{i-1}\sigma_0^{i}\sigma_0^{i+1}\ldots$ in some pattern ensemble $\mathcal{L}$ and create a new state $\widetilde{\mathrm{s}_0} = \ldots\tilde{\sigma}_0^{i-1}\tilde{\sigma}_0^{i}\tilde{\sigma}_0^{i+1}\ldots$ that differs from s_0 in a single cell. This is done by adding to s_0 the perturbation δ^j consisting of a single nonzero cell at site j: $\sigma^j = 1$, $\sigma^{i \neq j} = 0$; this is the smallest possible variation in the local state. The result is

$$\widetilde{\mathrm{s}_0} = \mathrm{s}_0 \dot{+} \delta^j \tag{9.4}$$

where $\dot{+}$ denotes cell-by-cell addition modulo $k = \|\mathcal{A}\|$, the alphabet size. In addition, we require that cell j be chosen so that the perturbed state $\widetilde{\mathrm{s}_0}$ is still in $\mathcal{L}$; this is equivalent to requiring that cell j be a "wild-card". Both the fiducial and perturbed states are iterated independently using the CA rule, and at each time step we plot the difference pattern

$$\mathbf{\Delta}_t = \mathrm{s}_t \dot{-} \widetilde{\mathrm{s}_t} \tag{9.5}$$

where $\dot{-}$ is subtraction modulo k. The resulting picture shows the space-time cells whose values are dependent on the value of the perturbed cell σ_0^j. Not surprisingly, the difference pattern is dependent on the particular choice of initial condition s_0. The maximum instantaneous rate at which the perturbation may spread in either direction is equal to the "speed of light" of the rule, $c = 2r\ \mathrm{cells/iteration}$.

9.4 Difference Plumes

To investigate the statistical information transmission properties of an entire ensemble, we use the *difference plume*, or averaged difference pattern. For each state s_0 in an ensemble $\mathcal{L}$, we form the difference pattern as above, creating an ensemble of "difference pairs" for a given perturbation δ^j: $\left\{(\mathrm{s}_0, \widetilde{\mathrm{s}_0})\ :\ \widetilde{\mathrm{s}_0} = \mathrm{s}_0 \dot{+} \delta^j,\ \mathrm{s}_0, \widetilde{\mathrm{s}_0} \in \mathcal{L}\right\}$. As above, both the original and the perturbed state must be in the ensemble; i.e. the $\sigma^j = 1$ in δ^j appears only at wild card sites. Then at each point (i, t) in space-time, we average $\Delta_t^i = \sigma_t^i \dot{-} \tilde{\sigma}_t^i$ over the ensemble:

$$\left\langle \Delta_t^i \right\rangle_{\mathcal{L}} = \|\mathcal{L}\|^{-1} \sum_{\mathrm{s}_t \in \mathcal{L}} \Delta_t^i \tag{9.6}$$

Thus $\left\langle \Delta_t^i \right\rangle_{\mathcal{L}}$ is the fraction of difference pairs for which $\sigma_t^i \neq \tilde{\sigma}_t^i$. At $t = 0$, for example, $\left\langle \Delta_0^i \right\rangle_{\mathcal{L}} = 1$ for $i = j$ and $\left\langle \Delta_0^i \right\rangle_{\mathcal{L}} = 0$ for all $i \neq j$.

Three quantitative measures of the plume help to characterize its shape. The first two describe the average rate at which the difference plume spreads in each spatial direction. They are the two quantities $\lambda_{\mathrm{L}}(\mathcal{L})$ and $\lambda_{\mathrm{R}}(\mathcal{L})$, which are the linearized average velocities of the left and right edges of the plume, respectively. These have been called the left and right "Lyapunov exponents,"[83,91] restricted here to the ensemble $\mathcal{L}$. They are defined as follows.

For each difference pattern Δ_t in the ensemble, denote the leftmost and rightmost cells σ_t^i for which $\Delta_t^i \neq 0$ by $\mathrm{Left}(\Delta_t)$ and $\mathrm{Right}(\Delta_t)$. That is,

$$\begin{aligned} \mathrm{Left}(\Delta_t) &= i \ : \ \Delta_t^i \neq 0, \ \Delta_t^j = 0 \, \forall \, j < i \\ \mathrm{Right}(\Delta_t) &= i \ : \ \Delta_t^i \neq 0, \ \Delta_t^j = 0 \, \forall \, j > i \end{aligned} \tag{9.7}$$

Then the left and right edges of the plume are defined by

$$\begin{aligned} \mu_{\mathrm{L}}(t) = \langle \mathrm{Left}(\Delta_t) \rangle_{\mathcal{L}} = \frac{1}{\|\mathcal{L}\|} \sum_{\mathrm{s}_0 \in \mathcal{L}} \mathrm{Left}(\Delta_t) \\ \mu_{\mathrm{R}}(t) = \langle \mathrm{Right}(\Delta_t) \rangle_{\mathcal{L}} = \frac{1}{\|\mathcal{L}\|} \sum_{\mathrm{s}_0 \in \mathcal{L}} \mathrm{Right}(\Delta_t) \end{aligned} \tag{9.8}$$

The "Lyapunov exponents" λ_{L} and λ_{R} are the linearized growth rates of $\mu_{\mathrm{L}}(t)$ and $\mu_{\mathrm{R}}(t)$ with time:

$$\begin{aligned} \mu_{\mathrm{L}}(t) = j + \lambda_{\mathrm{L}} t + \mathcal{O}(t^2) \\ \mu_{\mathrm{R}}(t) = j + \lambda_{\mathrm{R}} t + \mathcal{O}(t^2) \end{aligned} \tag{9.9}$$

The third quantity associated with the difference plume is its total volume $\zeta_{\mathcal{L}}(t)$, defined as the average number of cells in which the perturbed and unperturbed states differ at time t:

$$\zeta_{\mathcal{L}}(t) = \sum_i \langle \Delta_t^i \rangle_{\mathcal{L}} \tag{9.10}$$

Note that at $t = 0$, $\zeta_{\mathcal{L}}(0) = 1$ for all $\mathcal{L}$.

9.5 Domain Interfaces

Having identified the regular domains $\{\Lambda^i\}$ of a CA and measured their properties, it is natural to turn next to the study of the interface between two adjacent domains. This section gives a brief outline of such a study. We make no attempt at a complete characterization, but merely indicate the starting point. Our purpose is to illustrate the utility of analyzing the state space in terms of the pattern bases: i.e., to show how knowledge of the domains gives a new systematic method of identifying the variety of the space-time structures—walls, dislocations, and (possibly) "particles"—that a spatial system can generate.

We are concerned with the behavior of states initially consisting of two concatenated domains Λ^i and Λ^j, with the wall between the two at a given cell m. Each domain configuration has a specific phase, but is otherwise chosen at random from the domain language. As in section 7.3, we denote ensembles of such "two-domain" states $\Lambda^{i,j}$. We then allow the states to evolve under the CA rule, and analyze the structure of the interface between the domains using the domain transducer to detect the walls and identify the domains between them.

The relative phase between the domains (see section 7.3) is an important quantity for differentiating among different types of interface. The range of allowed values of the relative phase depends on the number of states in the process graphs of the domains on either side. In chapter 11 we will examine a pair of domains for which some values of relative phase are equivalent, due to the particular connectivity of the domain machines. Such degeneracies depend on the details of the particular domains, and are left out of the definition of relative phase.

9.6 Interface Plumes

The space-time plot of a domain interface shows only a single state in the ensemble of interface states. It is possible, however, that any given plot shows behavior that is atypical of that ensemble. To characterize the ensembles as a whole, it is necessary to gather statistics over all the states, or at least a Monte Carlo sample of them. We do this by plotting the *interface plumes*, that is, the sum of the interfaces of the states in each two-domain ensemble.

The structure of the plumes is made clearer by a few simple statistics, defined below. These are all measures of the positions of the walls in the interface, and are closely related to the difference plume statistics defined above.

Let θ_t^i be the sign function of the output of the domain filter T_Λ upon reading the symbol σ_t^i. That is, let $\theta_t^i = 1$ whenever σ_t^i is a wall, and $\theta_t^i = 0$ otherwise. The ensemble average of θ_t^i gives the fraction of states in the ensemble $\mathcal{L}$ for which there is a wall at cell (i, t):

$$\langle\theta_t^i\rangle_{\mathcal{L}} = \frac{1}{\|\mathcal{L}\|} \sum_{\mathrm{s}_0 \in \mathcal{L}} \theta_t^i \tag{9.11}$$

The roughest measure of interface evolution is just its **mean position** or **center** μ_C, defined as the average position of the walls in all states in the ensemble. This is given by the formula

$$\mu_{\mathrm{C}}(t) = \sum_{i=a}^{b} i \langle\theta_t^i\rangle_{\mathcal{L}} \tag{9.12}$$

with a and b chosen outside the interfacial region. The mean position will capture the average drift of the walls, but it is insensitive to the width of the interface. For this reason, we also look at the edges of the plume. For any iterate s_t of a two-domain initial condition $\mathrm{s}_0 \in \Lambda^{i,j}$ there will be a left-most and right-most wall cell. Denote these by $\mathrm{Left}(\theta_t)$ and $\mathrm{Right}(\theta_t)$, respectively. The **mean left edge** μ_{L} of the interface is given as before by the ensemble average

$$\mu_{\mathrm{L}}(t) = \langle \mathrm{Left}(\theta_t) \rangle_{\mathcal{L}} = \frac{1}{\|\mathcal{L}\|} \sum_{\mathrm{s}_0 \in \mathcal{L}} \mathrm{Left}(\theta_t) \tag{9.13}$$

where in this case $\mathcal{L} = \Lambda^{i,j}$ with the chosen relative phase. Similarly, the **mean right edge** of the interface μ_{R} is given by

$$\mu_{\mathrm{R}}(t) = \langle \mathrm{Right}(\theta_t) \rangle_{\mathcal{L}} = \frac{1}{\|\mathcal{L}\|} \sum_{\mathrm{s}_0 \in \mathcal{L}} \mathrm{Right}(\theta_t) \tag{9.14}$$

The time-evolution of μ_{L} and μ_{R} measure the spreading of the interface to the left and right, respectively. The left and right spreading rates are clearly analogous to the "Lyapunov exponents" defined above.

A quantitative measure of the degree of localization of the interface is provided by the distance

$$\mu_{\mathrm{R-L}} = \mu_{\mathrm{R}} - \mu_{\mathrm{L}} \tag{9.15}$$

between the mean left edge and mean right edge of the plume. If the interface remains localized, then $\mathrm{Left}(\theta_t) \approx \mathrm{Right}(\theta_t)$ for each state in $\mathcal{L}$, and $\mu_{\mathrm{L}} \approx \mu_{\mathrm{R}}$.

Another statistic that differentiates among plumes is the **interface plume volume** $\xi_{\mathcal{L}}(t)$. This is the total number of walls in the interface, given by the sum over space of $\langle \theta_t^i \rangle_{\mathcal{L}}$:

$$\xi_{\mathcal{L}}(t) = \sum_{i=a}^{b} \langle \theta_t^i \rangle_{\mathcal{L}} \tag{9.16}$$

where the limits a and b are chosen to lie outside the envelope of the interface. The plume volume provides another test of whether the interface is behaving as a particle or whether it is unstable. Since a particle is by definition localized in space, it necessarily must have a roughly constant—or at least bounded—number of walls. We might consider the plume volume to be roughly analogous the "mass" of a quasiparticle.

10 Domains in Elementary Cellular Automata

Regular domains in elementary CA show the same wide variety of behavior that occurs in CA generally. This chapter does not attempt to make an exhaustive list of the domains found in all elementary CA. Rather, it presents a survey with representative examples chosen to exhibit the range of behavior of ECA. We will see examples of domains that are uniform or periodic in space and time (spatio-temporal order), disordered in space but uniform or periodic in time ("spatial chaos"), and disordered in space and time ("spatio-temporal chaos"). The discrete site alphabet prevents the remaining type of behavior, purely temporal chaos, from occurring in CA.

Note the emphasis here. We are not attempting to classify CA *rules*, except insofar as they make particular domain languages invariant and possibly stable. Rather, we are directly classifying the *patterns* that show up in the evolution. This is an important distinction since, as we will see, it is not uncommon for a single rule to have a number of different domains with very different patterns—and so any attempt to put the rule itself into a single behavior class can be extremely misleading. (For a striking example of this, see also chapter 11.) This fact is one of the reasons that a simple global classification of spatial systems, such as the four "Wolfram classes" or refinements of them, must be approximate at the very best.

Throughout, we will be looking for the most typical domains present in the evolution of a given CA. It is quite often the case a rule has nested domains, so that one domain is a subset of another. For example, in the so-called "legal" rules[103] for which a string of 0s is invariant, the language 0^* is a domain. Nevertheless, except for the "class 1" rules, of which we will give one example, this pattern is not typical of evolution from arbitrary initial conditions on systems of arbitrary lattice size. It is therefore properly treated as a special case, say as part of a larger domain.

Which of the domains in a CA are considered to be the "essential" ones is determined by the purpose for which the CA is to be used, if indeed such a designation is even necessary. For each domain language, a domain filter can be constructed as described in chapter 7. If the question is, *what sort of patterns are possible?* then a complete answer would require finding *all* of the domains. If, on the other hand, we are trying to characterize the *typically occurring* patterns, it suffices to find just those domains that dominate in the state space. In terms of pattern bases, the latter is the better course. The dominant patterns are the ones that organize the evolution of the system at large, the ones in terms of which the system is to be analyzed.

Consistent with the notation of chapter 7 the domains of a given CA with rule index m will be denoted by Λ_m^i, and two domains separated by a single wall by $\Lambda_m^{i,j}$. The domain transducer that filters out the set of domains $\Lambda_m = \left\{\Lambda_m^i\right\}$ will be denoted by T_{Λ_m}, or, if it filters out a single domain Λ_m^i, $\mathrm{T}_{\Lambda_m^i}$.

The series of rules examined in this chapter begins with ECA 32, the "class 1" rule mentioned above, for which almost all initial conditions evolve to a spatio-temporally uniform field of 0s. Rules with spatially disordered, temporally invariant or periodic domains include ECA 44 and ECA 55. In these rules, domain walls are frozen in place, their positions dependent ultimately on the details of the initial condition. More interesting is the spatio-temporally periodic domain in ECA 54, which has domain walls that move and interact in a nontrivial fashion. Domains that are both spatially and temporally disordered are found in ECA 18, 122, and 90. This last is the prototypical "linear" rule described in section 2.4; it is interesting to note again that nonlinearity in the sense defined in that section, is not required for chaotic (disordered) behavior.

Finally, at the end of the chapter, we return to ECA 18 for a more detailed examination of the behavior of dislocations. This will include statistical analysis of the interactions between nearby dislocations, the statistics of the motion of a single dislocation, and a discussion of the potential for harnessing dislocations to perform computational tasks.

10.1 ECA 32

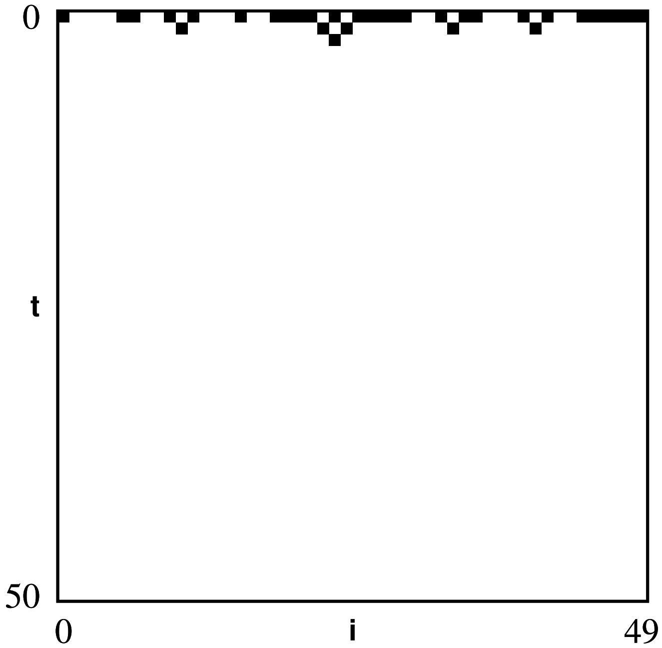

Fig. 10.1 Space-time plot of ECA 32, starting from a random initial condition on a lattice of size $N = 50$.

ECA 32 is the "class 1" CA whose space-time plot was shown in fig. 2.1(a). Fig. 10.1 shows another space-time plot of ECA 32, this time with a random initial condition on a lattice of size $N = 50$ evolving to $t = 50$. As is evident from the figure, the domain here is the language 0^* containing all strings of contiguous 0s.

Checking this by the FME algorithm, we find that the language $\Lambda^0_{32} = 0^*$ is indeed invariant under the ECA 32 update rule $\mathbf{\Phi}_{32}$. The domain transducer $\mathrm{T}_{\Lambda_{32}}$ is just the

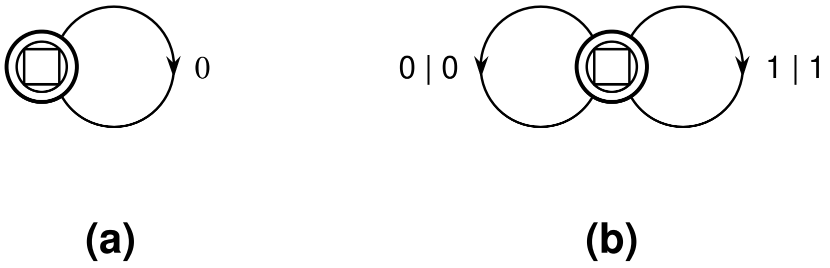

Fig. 10.2 (a) Minimal DFA for domain Λ^0_{32} and (b) domain transducer $\mathrm{T}_{\Lambda_{32}}$ for ECA 32. These are the machines for the language 0^* and the identity mapping, respectively.

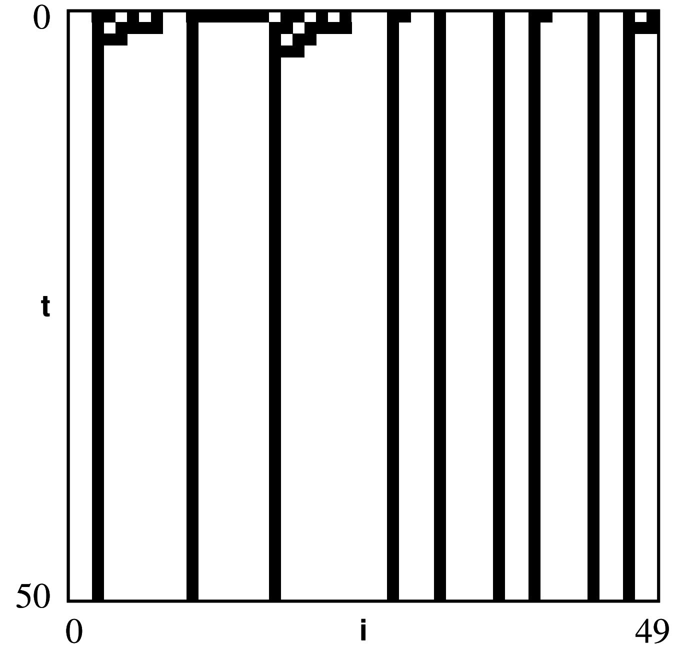

Fig. 10.3 Space-time plot of ECA 44, starting from a random initial condition on a lattice of size $N = 50$.

identity, since any nonzero cell is a wall. The domain language Λ^0_{32} and the domain transducer $\mathrm{T}_{\Lambda_{32}}$ are shown in fig. 10.2(a) and (b).

Clearly, all of the other "class 1" rules that evolve to 0^* have this same domain. For binary site alphabets, the other type of "class 1" rule evolves to uniform field of 1s. For these rules, the domain is 1^*, and the corresponding domain transducer is the "bit-flip" machine that simply reverses the value of each symbol in the string. Each cell with $\sigma = 0$ is therefore a domain wall.

10.2 ECA 44

ECA 44 is the "class 2" CA whose space-time plot was shown in fig. 2.1(b). Fig. 10.3 shows another space-time plot of ECA 44, this time with a random initial condition on a lattice of size $N = 50$ evolving up to $t = 50$. The most obvious domain is again $\Lambda^0_{44} = 0^*$, as can be seen from the space-time plot. Another domain Λ^1_{44}, one that captures all of the asymptotic structure, is shown in fig. 10.4. Both of these languages are invariant by the FME algorithm, and their process graphs are strongly connected—so they are regular domains. Note that $\Lambda^0_{44} \subset \Lambda^1_{44}$.

Λ^1_{44} is also the limit set of this rule.[81] As fig. 10.3 shows, a random initial condition quickly settles down onto an asymptotic (long-time) invariant configuration in the domain language Λ^1_{44}. The transducer $\mathrm{T}_{\Lambda^1_{44}}$ that filters out Λ^1_{44} maps all asymptotic states to 0^*, so the filtered space-time pattern would show the approach to the asymptotic state as a pattern of quickly disappearing nonzero cells, somewhat like the unfiltered plot of ECA 32 above (fig. 10.1). The transducer for the other domain Λ^0_{44} is, as for Λ^0_{32}, just the identity.

As we noted in section 2.5 (see fig. 2.3), the evolution of the global language $\Omega_0 = \mathcal{A}^*$ is not so simple. We can now state more precisely what is plotted in that figure: it shows the number of states $V_t = \|\mathrm{M}(\Omega_t)\|$ in the minimal DFA $\mathrm{M}(\Omega_t)$ vs. t. The number of states grows approximately linearly as $V_t \approx 3t$. This is essentially because of the different possible approaches to the asymptotic language Λ^1_{44}, as indicated by the small "shrinking checkerboard" patterns at the top of fig. 10.3. A few initial conditions start out with large checkerboard patterns, which take a long time to reach the asymptotic pattern. It is the "working out" of these atypical transient patterns that leads to the growth in V_t.

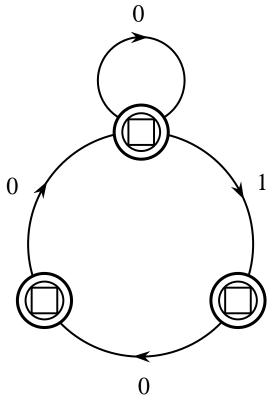

Fig. 10.4 Process graph of the nontrivial domain Λ^1_{44} of ECA 44.

10.3 ECA 55

ECA 55 is another rule in which the domains are obvious to the eye. A space-time plot of the evolution of a random initial condition under ECA 55 is shown below in fig. 10.7(top). As for ECA 44, the dislocations are frozen where the initial condition placed them. After a few iterations, the evolution settles down to a simple period-two oscillation in which the value of each cell is the complement of its value at the previous time step.

As for ECA 44, there are two domains of interest. The first, Λ^0_{55}, consists of contiguous blocks of cells of a single value, 0s at one time step, 1s the next.

This domain has temporal period 2, and is shown in fig. 10.5(a). Λ^0_{55} therefore consists of two disconnected components that are mapped into one another by the rule. This was verified by direct calculation using the FME algorithm. The process graph of each component is strongly connected, so the patterns at a given time are spatially homogeneous. Note that the entire language Λ^0_{55} (i.e., the union of the two components) has temporal period 1, but it is not spatially homogeneous.

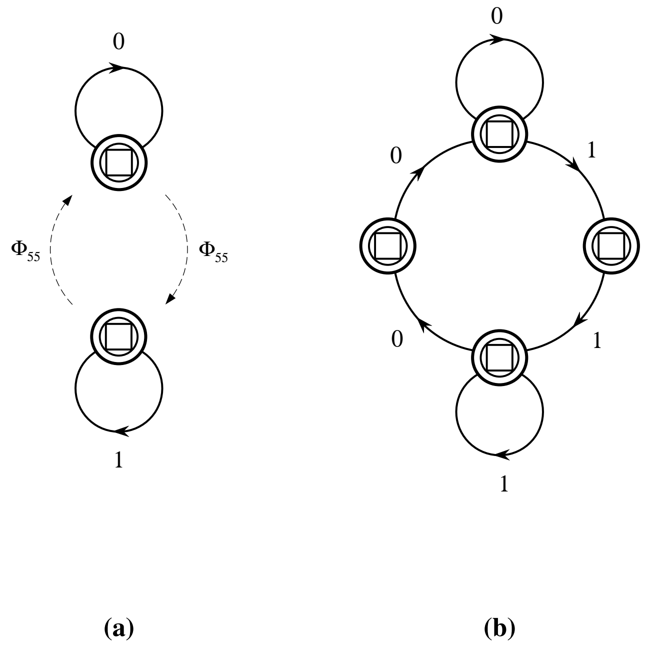

Fig. 10.5 Process graphs of two different domain languages in ECA 55. (a) Λ^0_{55}, a temporally periodic domain consisting of two disconnected pieces mapped into each other by the rule. (b) Λ^1_{55}, a single temporally invariant language.

The second domain, Λ^1_{55}, shown in fig. 10.5(b), contains all blocks of 0s or 1s of length 2 or more; i.e., there are no isolated 0s or 1s. The FME algorithm establishes that this language has temporal period 1, and the figure shows that the process graph is strongly connected; this proves that Λ^1_{55} is indeed a domain. It is also the limit set of this rule.[81] As in ECA 44, the domains are nested: $\Lambda^0_{55} \subset \Lambda^1_{55}$. The role of Λ^1_{55} in ECA 55 is exactly analogous to that of Λ^1_{44} in ECA 44.

Unlike ECA 44, however, ECA 55 is sufficiently simple that its global evolution admits a closed-form expression. After just three iterations, the image $\Omega_t = \mathbf{\Phi}_t(\mathcal{A}^*)$ of the entire state space falls onto its asymptotic invariant set, i.e.,

$$\Omega_t = \Omega_\infty, \quad \forall\, t \geq 3. \tag{10.1}$$

The global invariant set is identical to the domain Λ^1_{55} shown in fig. 10.5(b).

Fig. 10.6 shows the domain transducer $\mathrm{T}_{\Lambda^0_{55}}$ for the domain Λ^0_{55} of fig. 10.5(a). Its structure is simple, being in effect a one-dimensional edge detector: a wall consists of any cell not identical to its immediate predecessor. The filtered pattern generated by $\mathrm{T}_{\Lambda^0_{55}}$ is shown in fig. 10.7 along with the unfiltered data.

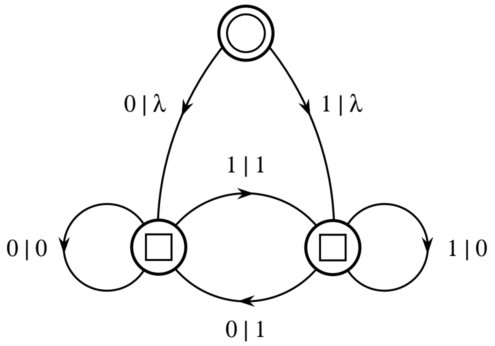

Fig. 10.6 Transducer $\mathrm{T}_{\Lambda^0_{55}}$ for domain Λ^0_{55} in ECA 55. This is nothing more than a one–dimensional edge detector.

10.4 ECA 54

ECA 54 shows a considerably more interesting behavior than the previous examples. The domain is periodic both in space and time, but the structure of the walls is much richer. The phenomenology of the walls (or “particles”) has been studied in some detail elsewhere.[8]

The domain Λ^0_{54} is shown in fig. 10.8. Like Λ^0_{55}, is has temporal period 2, as is shown by the two disconnected components, representing the two temporal phases of the domain. The dotted lines indicate that each component maps into the other, as verified by the FME algorithm. The spatial periodicity is also evident, in that each component is a cyclic machine.

The domain filter, shown in fig. 10.9, is at first glance considerably more complex than in the previous examples. Much of the structure is devoted to synchronizing the transducer to the underlying pattern. Since the domains have longer spatial period, there are correspondingly more synchronization states. Nevertheless, after synchronization, the image of the original domain machine can be seen in the bottom portion of the transducer.

On applying $\mathrm{T}_{\Lambda_{54}}$ to the raw space-time plot shown in fig. 10.11(top), the filtered pattern of fig. 10.11(bottom) was obtained. A cursory visual inspection indicates that there are three basic types of wall structure in the filtered data, shown in fig. 10.10: (I) an isolated 1 surrounded by 0s, moving to the left or right with speed $c = 1\ \mathrm{cell/iteration}$, representing a simple phase shift in the domain; (II) a stationary pattern with temporal period 4 and width fluctuating from 1 to 3 to 5 cells; and (III) a stationary pattern with temporal period 4 and width 2 or 4 cells. The direction of further investigation is clear: having identified the different types of wall, the next step is to investigate the various interactions among walls. For example, in fig. 10.11(bottom) one sees that the collision of a right-moving Type I wall and a left-moving Type I wall creates a single Type III wall. Also note that when a Type I wall encounters a Type II wall, the Type II wall is shifted to the side and two Type I walls going in opposite directions are created.

Fig. 10.12 shows a much larger lattice, evolving for a much longer time. The figures are rasterfiles taken directly from the computer screen, with an image resolution

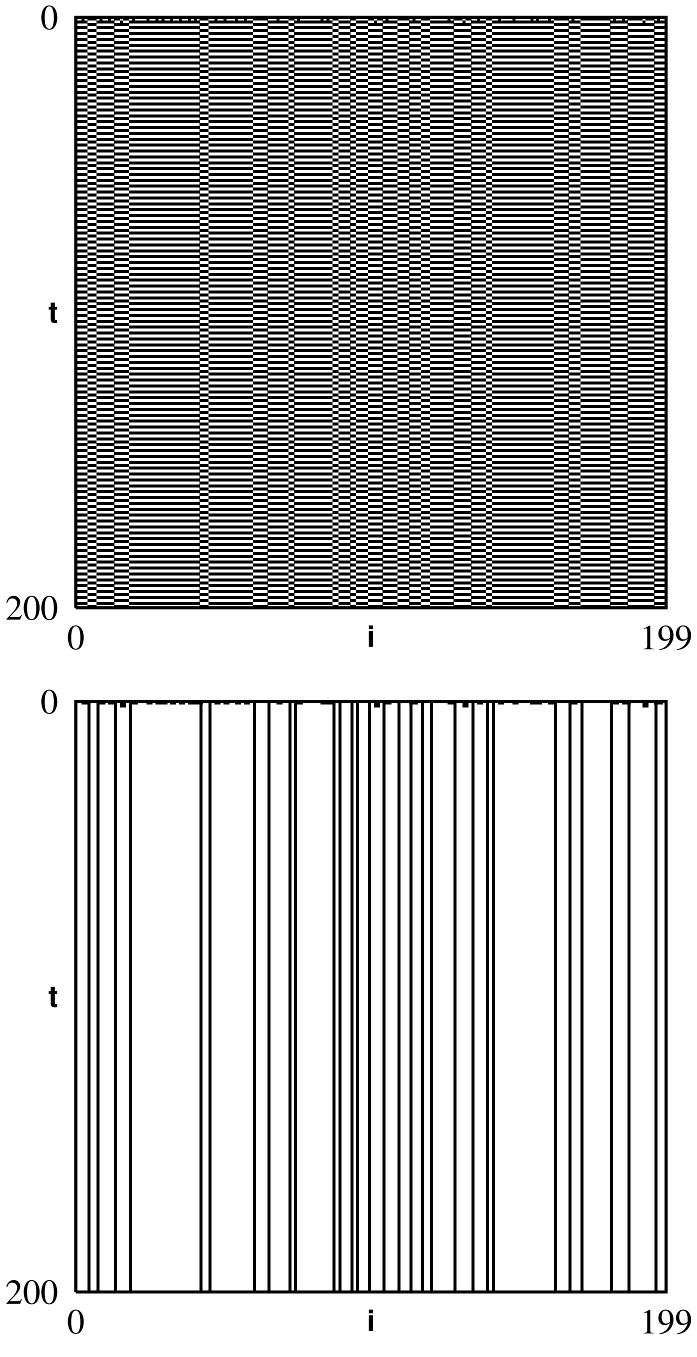

Fig. 10.7 Raw and filtered space-time plots of ECA 55, starting from a random initial condition and filtering with transducer $\mathrm{T}_{\Lambda^0_{55}}$.

considerably coarser than the cell size: a single pixel covers approximately 3 spatial cells and 33 time steps. Because of this, the unfiltered data appears to be a wash of randomly distributed pixels. The filtered plot, on the other hand, clearly shows long-lived structures evolving on a white background which is peppered with occasional black pixels. The peppering in the background is the image of the thin Type I walls, which are usually invisible at the figure's resolution, showing up only occasionally as

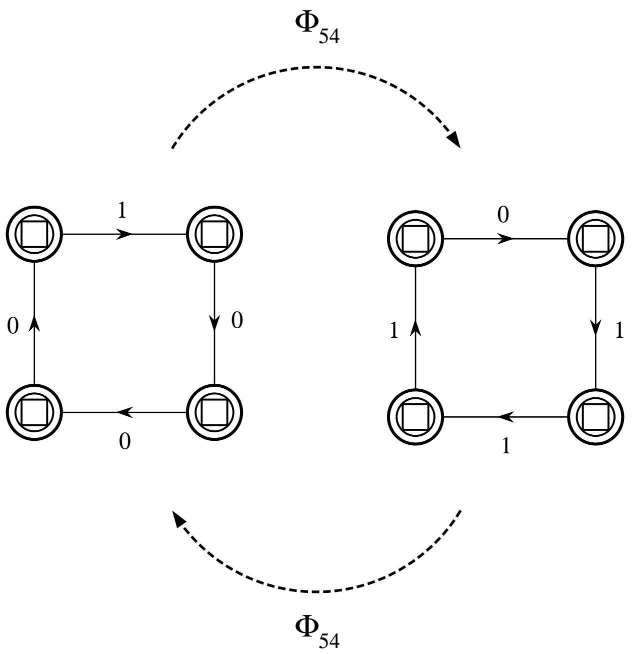

Fig. 10.8 Process graph of ECA 54's domain Λ^0_{54}.

isolated dots. The thicker Type II and Type III walls appear as roughly vertical lines, moving slowly back and forth in time. The effective dynamics at this level is of stable thick walls (mostly Type II) apparently undergoing a sort of statistical "repulsion" mediated by the fast-moving Type I walls. After a transient period, the number of thick walls remains constant and they remain roughly equidistant, but wander about within a close neighborhood.

10.5 ECA 18

We will delve a little more deepy into the properties of chaotic ECA 18, in anticipation of the attractor-basin analysis of chapters 12 and 13.* This also gives us the opportunity to show the FME algorithm in action. ECA 18 is the first of our examples in which the full apparatus of the FME algorithm is necessary. This is because the ECA 18's domain language Λ^0_{18} has positive topological entropy, meaning that the number of words of a given length grows exponentially with word length. Without some method of explicitly calculating the iterate of a positive-entropy language, the temporal invariance of Λ^0_{18} could not be established.

The basic phenomenology ECA 18 has been known for some time. Empirical studies[40,41] have shown that 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

---

*Much of the material in this section, and in section 10.8 below, originally appeared in ref. [47]

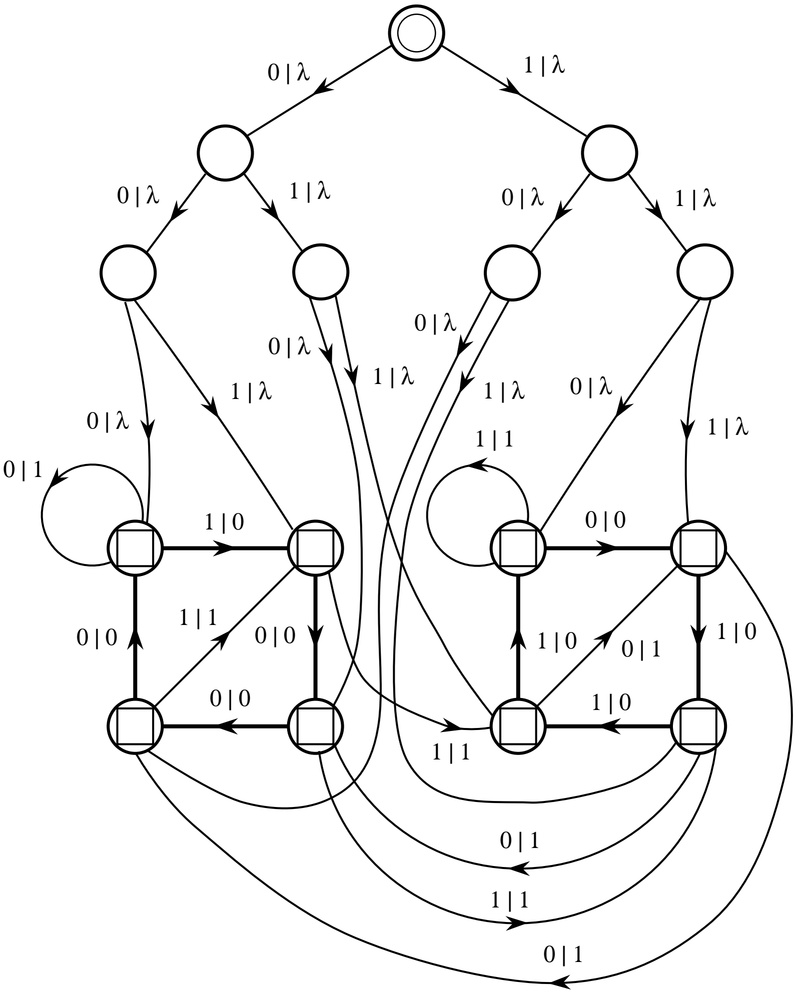

Fig. 10.9 ECA 54's domain filter.

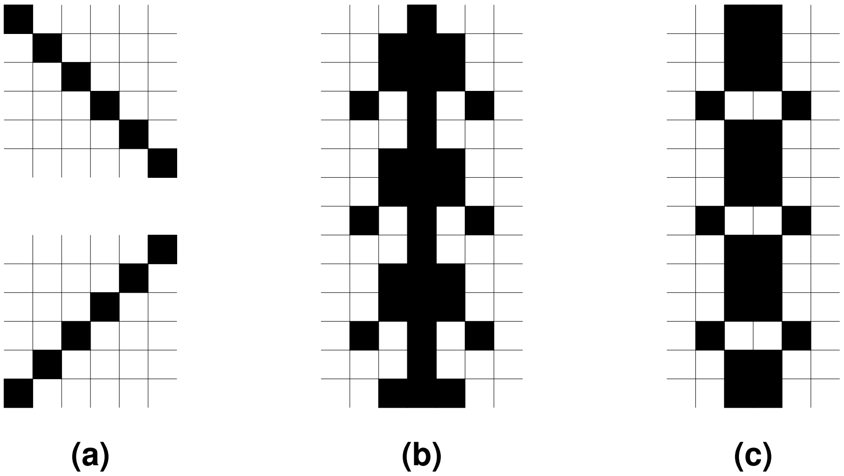

Fig. 10.10 Basic wall structures ("fundamental particles") in the filtered space-time patterns of ECA 54. (a) Right- and left-moving versions of Type I. (b) Type II. (c) Type III.

decays in time as $d \approx (8\pi D t)^{-\frac{1}{2}}$, with $D \approx \frac{1}{2}$. Furthermore, the excursion x of a set