arXiv:1812.XXXXX

Spacetime Symmetries, Invariant Sets, and Additive Subdynamics of Cellular Automata

Adam Rupe* and James P. Crutchfield†
Complexity Sciences Center
Physics Department
University of California at Davis,
One Shields Avenue, Davis, CA 95616

(Dated: December 30, 2018)

Cellular automata are fully-discrete, spatially-extended dynamical systems that evolve by simultaneously applying a local update function. Despite their simplicity, the induced global dynamic produces a stunning array of richly-structured, complex behaviors. These behaviors present a challenge to traditional closed-form analytic methods. In certain cases, specifically when the local update is additive, powerful techniques may be brought to bear, including characteristic polynomials, the ergodic theorem with Fourier analysis, and endomorphisms of compact Abelian groups. For general dynamics, though, where such analytics generically do not apply, behavior-driven analysis shows great promise in directly monitoring the emergence of structure and complexity in cellular automata. Here we detail a surprising connection between generalized symmetries in the spacetime fields of configuration orbits as revealed by the behavior-driven local causal states, invariant sets of spatial configurations, and additive subdynamics which allow for closed-form analytic methods.

PACS numbers: 05.45.-a 89.75.Kd 89.70.+c 02.50.Ey
Keywords: coherent structures, spatially extended dynamical systems, emergence, symmetry breaking, cellular automata

I. INTRODUCTION

Early systematic studies of cellular automata (CA) focused largely on phenomenology [1–4] and algebraic properties of *additive* cellular automata [5, 6]. Analysis was possible due to the linear superposition exhibited by additive CAs. One challenge was to extend the analytic techniques appropriate for linear CAs to nonlinear CAs, which produce much richer and more physically-relevant behaviors.

One approach restricted a nonlinear CA to a particular set of spatial configurations that exercised only a linear subset of its behaviors. For example, appropriately restricted the nonlinear CA rule 18 is equivalent to linear CA rule 90 [7]. That is, using one of these special configurations as the initial condition, it does not matter if the evolution follows rule 18 or rule 90, the resulting orbit is the same. This approach was elaborated upon by Refs. [6, 8] and expanded upon by Ref. [9], which proved nonlinear rules 126 and 146 can be reduced to the linear rule 90. The latter also showed, under certain conditions, that more general orbits of the nonlinear rules can be mapped to orbits of rule 90, then mapped back; further extending the use of linear analytics on nonlinear rules. Reference [10] considered additional algebraic structures to generalize additive CAs into a larger set of *quasilinear* CAs that do not obey superposition, but whose spacetime evolutions can still be predicted in less time then general nonlinear CAs. More recently, Ref. [11] defined a special set of *quasiadditive* spatial configurations for the nonlinear rule 22 that emulate rule 146 at all even time steps. (Ref. [9] had already established that rule 146 emulates linear rule 90.)

Parallelling the earlier efforts, Crutchfield and Hanson developed rigorous techniques that applied concepts from dynamical systems theory to cellular automata behaviors [12–16]: invariant sets, stable and unstable manifolds, attractors, and basins. In particular, they introduced techniques that automatically discovered a given CA's invariant sets of spatial configurations [17]. (Reference [18] had also identified invariant sets, but for a special CA class.) A CA's spacetime shift-invariant sets of configurations are its *domains*. They are key to a CA's spatiotemporal organization: spacetime behaviors typically decompose into domains and embedded defects—domain *walls*, *particles*, and other *coherent structures*.

More recently, we employed fully behavior-driven tools known as local causal states to automatically discover (even hidden) spatiotemporal symmetries in CA behaviors [19]. From this, coherent structures were shown to be localized deviations from generalized symmetries of CA spacetime fields.

The following explores a surprising and subtle connection between these three threads of analysis. It considers

* atrupe@ucdavis.edu
† chaos@ucdavis.edu

a CA at three levels of description: (i) its global evolution function that updates spatial configurations, (ii) sets of spatial configuration in the state space of all possible configurations, and (iii) sets of spacetime-field orbits. Building on evidence presented in Ref. [19], we argue here that there is strong evidence that spacetime field orbits resulting from evolution along an invariant set of spatial configurations possess generalized spacetime symmetries that are captured by the local causal states. This means domains can be equivalently described as invariant sets in state space or as sets of behaviors—spacetime field orbits—with generalized symmetries. Do domains similarly have a defining characteristic in terms of the CA global evolution function? In fact, a wealth of examples do link CA domains with additive subdynamics; nonadditive CAs can become additive when evolving certain subsets of spatial configurations—domain invariant subsets. However, this is not always the case. We present a counterexample: a CA that does not become additive when evolving over a domain invariant set.

Defining a CA domain in terms of its equation of motion remains elusive. Despite this, there is an interesting connection between CA domains, given as either a spacetime invariant set or set of symmetric behaviors, and additive subdynamics of the CA. After building up the necessary formalism, we revisit examples given in the earlier literature showing nonlinear CAs displaying linear behaviors to illustrate and explore when these linear behaviors are in fact domain behaviors. The fact that many, *but not all*, domains are linear behaviors of CAs points to a more general theory still waiting to be discovered.

II. BACKGROUND

We first recall CAs and the subclass of additive CAs. We then review how automata-theoretic methods are used to explore the temporal evolution of configuration subsets. The notion of structured behavior that we use is then laid out in our synopsis of spatiotemporal computational mechanics, including how structures are detected by causal filtering and how to reconstruct a CA's local causal states.

A. Cellular automata

A one-dimensional *cellular automaton* or CA $(\mathcal{A}^{\mathcal{L}}, \Phi)$ consists of a spatial lattice $\mathcal{L} = \mathbb{Z}$ whose *sites* take values from a finite *alphabet* $\mathcal{A}$. A CA *state* $x \in \mathcal{A}^{\mathbb{Z}}$ is the configuration of all site values $x^r \in \mathcal{A}$ on the lattice. (For states x, subscripts denote time; superscripts sites.) CA states evolve in discrete time steps according to the *global evolution* $\Phi : \mathcal{A}^{\mathbb{Z}} \to \mathcal{X} \subseteq \mathcal{A}^{\mathbb{Z}}$, where:

$$x_{t+1} = \Phi(x_t) \ .$$

Φ is implemented through parallel, synchronous application of a *local update rule* ϕ that evolves individual sites x^r_t based on their radius R *neighborhoods* $\eta(x^r) = \{x^{r'} \ : \ ||r - r'|| < R\}$:

$$x^r_{t+1} = \phi\big(\ \eta(x^r_t) \ \big) \ .$$

Stacking the states in a CA *orbit* $x_{0:t} = \{x_0, x_1, \ldots, x_{t-1}\}$ in time-order produces a *spacetime field* $\mathbf{x}_{0:t} \in \mathcal{A}^{\mathbb{Z} \otimes \mathbb{Z}}$. Visualizing CA orbits as spacetime fields reveals the fascinating patterns and localized structures that CAs produce and how the patterns and structures evolve and interact over time.

1. *Elementary cellular automata*

The parameters $(\mathcal{A}, R)$ define a CA *class*. One simple but nontrivial class is that of the so-called *elementary* cellular automata (ECAs) [8] with a binary local alphabet $\mathcal{A} = \{0, 1\}$ and radius $R = 1$ local interactions $\eta(x^r_t) = x^{r-1}_t x^r_t x^{r+1}_t$. Due to their definitional simplicity and wide study, we mostly explore ECAs in our examples.

A local update rule ϕ is generally specified through a *lookup table*, which enumerates all possible neighborhood configurations η and their outputs $\phi(\eta)$. The lookup table for ECAs is given as:

η	$O_\eta = \phi(\eta)$
1 1 1	O_7
1 1 0	O_6
1 0 1	O_5
1 0 0	O_4
0 1 1	O_3
0 1 0	O_2
0 0 1	O_1
0 0 0	O_0

,

where each output $O_\eta = \phi(\eta) \in \mathcal{A}$ and the ηs are listed in lexicographical order. There are $2^8 = 256$ possible ECA lookup tables, as specified by the possible strings of output bits: $O_7O_6O_5O_4O_3O_2O_1O_0$. A specific ECA lookup table is often referred to as an ECA *rule* with a *rule number* given as the binary integer $o_7o_6o_5o_4o_3o_2o_1o_0 \in [0, 255]$. For example, ECA 172's lookup table has output bit string 10101100.

In our analysis we examine the powers of the lookup table, also called *higher-order lookup tables*. The n^{th}-order lookup table ϕ^n maps the radius $n \cdot R$ neighborhood of a site to that site's value n time steps in the future. Said another way, a spacetime point x^r_{t+n} is completely determined by the radius $n \cdot R$ neighborhood n time-steps in the past according to:

$$x^r_{t+n} = \phi^n\big(\eta^n(x^r_t)\big) \ .$$

2. *Additive CAs*

The special subclass of additive CAs is of particular interest. A CA is *additive* if the output $\phi(\eta)$ may be written as a linear combination of the neighborhood entries:

$$x^r_{t+1} = \phi\big(\ \eta(x^r_t)\ \big) = \sum_{i=-R}^{R} a_i x^i_t \pmod{|\mathcal{A}|} \ , \qquad (1)$$

where $a_i \in \mathcal{A}$.

Additive CAs are referred to as "linear" because they obey a linear superposition principle. For any N radius-R neighborhood configurations η_i, the local update rule ϕ satisfies:

$$\sum_{i=1}^{N} \phi(\eta_i) = \phi\left(\sum_{i=1}^{N} \eta_i\right) \ , \qquad (2)$$

where $\sum$ denotes addition modulo $|\mathcal{A}|$ over each site in the neighborhoods [5, 9]. For the binary-alphabet CAs considered here additivity and linearity are equivalent (each implies the other) [20]. And so, we use both terms interchangeably.

Linear superposition enables powerful analytic techniques. For example, representing spatial configurations via characteristic polynomials gives a linear time-evolution via multiplication of a second polynomial representing Φ [5]. Or, one can employ the ergodic theorem for commuting transformations together with Fourier analysis [6]. Similarly, ergodic theory interprets additive CAs as endomorphisms of a compact Abelian group, which have been thoroughly investigated [21].

For general one-dimensional CAs with alphabet $\mathcal{A}$ and neighborhood radius R, there are $|\mathcal{A}|^{|\mathcal{A}|^{2R+1}}$ update rules. However, only $|\mathcal{A}|^{2R+1}$ are linear. Though there are cases where nonlinear rules may be mapped onto linear rules [9], the analytic tools just described apply to a vanishingly small subset of CA rules.

B. Automata-theoretic CA evolution

Rather than study how a CA evolves individual configurations, it is particularly informative to investigate how CAs evolve sets of configurations [12, 17]. This allows for discovery of structures in the state space of a CA induced by Φ.

For cellular automata in one spatial dimension, configurations $x \in \mathcal{A}^{\mathbb{Z}}$ are strings over the alphabet $\mathcal{A}$. Sets of strings recognized by finite state machines are called *regular languages*. Any regular language L has a unique minimal finite-state machine $M(L)$ that recognizes or generates it [22]. These automata are useful since they give a finite representation of a typically infinite set of regular-language configurations.

Not all regular languages are appropriate as sets of spatial configurations for cellular automata; we consider only sets that are subword closed and prolongable in the sense that every word in the language can be extended to the left and the right to obtain a longer word in the language. In automata theory these are known as *factorial, prolongable regular languages*. In symbolic dynamics [23] these are the languages of *sofic shifts*—closed, shift-invariant subsets of $\mathcal{A}^{\mathbb{Z}}$ that are described by finite-state machines. The language L of a sofic shift $\mathcal{X} \subseteq \mathcal{A}^{\mathbb{Z}}$, a *sofic language*, is the collection of all words that occur in points $x \in \mathcal{X}$. (See Definition 1.3.1 and Proposition 1.3.4 in Ref. [23] for details of sofic languages; sometimes called *process languages* in the computational mechanics literature.) Every state of the machine $M(L)$ for a sofic language L is both a start and end state.

For the remainder of our development any language L always refers to a sofic language, and the machine $M(L)$ refers to the unique minimal deterministic finite automaton of that language.

To explore how a CA evolves languages we establish a dynamic that evolves machines. This is accomplished via finite-state transducers. Transducers are a particular type of input-output machine that maps strings to strings [24]. This is exactly what a (one-dimensional) CA's global dynamic Φ does [25]. As a mapping from a configuration x_t at time t to one x_{t+1} at time $t+1$, Φ is also a map on a configuration set L_t from one time to the next L_{t+1}:

$$L_{t+1} = \Phi(L_t) \ . \qquad (3)$$

The global dynamic Φ can be represented as a finite-state transducer T_Φ that evolves a set of configurations represented by a finite-state machine. This is the *finite machine evolution* (FME) operator [12]. Its operation composes the CA transducer T_Φ and finite-state machine $M(L_t)$ to get the machine $M_{t+1} = M(L_{t+1})$ describing the set L_{t+1} of spatial configurations at the next time step:

$$M_{t+1} = \min\big(\mathrm{T}_\Phi \circ M(L_t)\big) \ . \qquad (4)$$

Here, $\min(M)$ is the automata-theoretic procedure that minimizes the number of states in machine M. While not entirely necessary for language evolution, the minimization step is helpful when monitoring the complexity of L_t. The net result is that Eq. (4) is the automata-theoretic version of Eq. (3)'s set evolution dynamic. Analyzing how the FME operator evolves configuration sets of different kinds is a key tool in understanding CA emergent patterns.

C. Computational Mechanics

A question central to understanding CA spacetime behaviors concerns structure: What spatial and spatiotemporal structures do CAs generate? How do they do so? How are CA state spaces organized to support structures and their dynamical emergence? Computational mechanics was developed to address the question of structure in dynamical systems and stochastic processes [26]. We first review its pure-temporal version and then its extension to spacetime.

1. Temporal processes and their causal states

To identify and analyze pattern and structure in CA behaviors in a rigorous and principled manner we employ the behavior-driven framework known as *computational mechanics* [26, 27]. Since it was most fully developed in the temporal setting, consider first a stationary bi-infinite stochastic process $\mathcal{P}$—the distribution of all a system's allowed behaviors or *realizations* $\ldots x_{-2}, x_{-1}, x_0, x_1, \ldots$ as specified by their joint probabilities $\Pr(\ldots, X_{-2}, X_{-1}, X_0, X_1, \ldots)$. Here, X_t is the random variable for the outcome of the measurement $x_t \in \mathcal{A}$ at time t, taking values from a finite set $\mathcal{A}$ of all possible events. (Uppercase denotes a random variable; lowercase its value.) We denote a contiguous chain of ℓ random variables as $X_{0:\ell} = X_0 X_1 \cdots X_{\ell-1}$ and their realizations as $x_{0:\ell} = x_0 x_1 \cdots x_{\ell-1}$. (Left indices are inclusive; right, exclusive.) We suppress indices that are infinite. A process is *stationary* when $\Pr\left(X_{t:t+\ell}\right) = \Pr\left(X_{0:\ell}\right)$ for all t and ℓ.

The canonical object of computational mechanics, used to represent pattern and structure in the behavior of a stochastic process, is the ϵ-*machine* [28]. This is a type of stochastic finite-state machine known as a hidden Markov model, which consists of a set $\boldsymbol{\Xi}$ of *causal states* and transitions between them. The causal states are constructed for a given process by calculating the equivalence classes determined by the *causal equivalence relation*:

$$x_{:t} \sim_\epsilon x'_{:t} \iff \Pr(X_{t:}|X_{:t}=x_{:t}) = \Pr(X_{t:}|X_{:t}=x'_{:t}). \quad (5)$$

In words, two pasts $x_{:t}$ and $x'_{:t}$ are *causally equivalent*, i.e., belong to the same causal state, if and only if they make the same prediction for the future $\Pr(X_{t:}|\cdot)$. Equivalent states lead to the same future conditional distribution. Behaviorally, the interpretation is that whenever a process generates the same future (a conditional distribution), it is effectively in the same state.

Each causal state $\xi \in \boldsymbol{\Xi}$ is an element of the coarsest partition of a process' pasts $\{x_{:t} : t \in \mathbb{Z}\}$ such that every $x_{:t} \in \xi$ has the same predictive distribution: $\Pr(X_{t:}|x_{:t}) = \Pr(X_{0:}|\cdot)$. The associated random variable is Ξ. The ϵ-*function* $\epsilon(x_{:t})$ maps a past to its causal state: $\epsilon : x_{:t} \mapsto \xi$. In this way, it generates the partition defined by the causal equivalence relation $\sim_\epsilon$. One can show that the causal states are the unique *minimal sufficient statistic* of the past when predicting the future [28].

2. Spatiotemporal processes, local causal states

A spatiotemporal system, in contrast to a purely temporal one, generates a process $\Pr(\ldots, X_{-1}, X_0, X_1, \ldots)$ consisting of the series of fields X_t over a spatial lattice $\mathcal{L}$. A realization of a spatiotemporal process then is a *spacetime field* $\mathbf{x} \in \mathcal{A}^{\mathcal{L}\otimes\mathbb{Z}}$, consisting of a time series $x_0, x_1, \ldots$ of spatial configurations $x_t \in \mathcal{A}^{\mathcal{L}}$. $\mathcal{A}^{\mathcal{L}\otimes\mathbb{Z}}$ is the orbit space of the process; that is, time is added onto the system's state space. The associated spacetime field random variable is $\mathbf{X}$. A *spacetime point* $\mathbf{x}^r_t \in \mathcal{A}$ is the value of the spacetime field at coordinates (r, t)—that is, at location $r \in \mathcal{L}$ at time t. The associated random variable at that point is $\mathbf{X}^r_t$.

To extract pattern and structure from a system's spacetime field behaviors we employ a spatially local generalization of the causal equivalence relation Eq. (5). For systems that evolve under a homogeneous local dynamic and for which information propagates through the system at a finite speed, it is quite natural to use lightcones as spatially local notions of pasts and futures.

Formally, the *past lightcone* $\mathtt{L}^-$ of a spacetime random variable $\mathbf{X}^r_t$ is the set of all random variables at previous times that could possibly influence it. That is:

$$\mathtt{L}^-(r,t) \equiv \left\{\mathbf{X}^{r'}_{t'} : \ t' \leq t \text{ and } ||r' - r|| \leq c(t - t')\right\} , \quad (6)$$

where c is the finite speed of information propagation in the system. Similarly, the *future lightcone* $\mathtt{L}^+$ is given as all the random variables at subsequent times that could

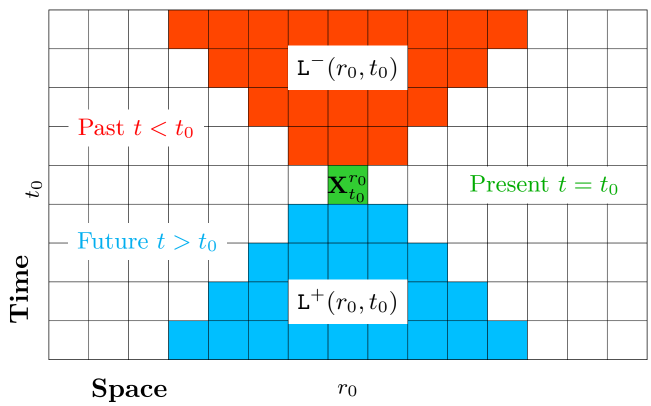


FIG. 1. Lightcone random variable templates: Past lightcone $\mathtt{L}^-(r_0, t_0)$ and future lightcone $\mathtt{L}^+(r_0, t_0)$ for present spacetime point $\mathbf{X}_{t_0}^{r_0}$ in a 1+1 D field with nearest-neighbor (or radius-1) interactions.

possibly be influenced by $\mathbf{X}_t^r$:

$$\mathtt{L}^+(r,t) \equiv \left\{ \mathbf{X}_{t'}^{r'} : t' > t \text{ and } ||r' - r|| \leq c(t' - t) \right\} . \quad (7)$$

We include the *present* random variable $\mathbf{X}_t^r$ in its past lightcone, but not its future lightcone. An illustration for one-space and time $(1 + 1\text{D})$ fields on a lattice with nearest-neighbor (or *radius*-1) interactions is shown in Fig 1. We use $\mathtt{L}^-$ to denote the random variable for past lightcones with realizations ℓ^-; similarly, $\mathtt{L}^+$ those with realizations ℓ^+ for future lightcones.

The choice of lightcone representations for both local pasts and futures is ultimately a weak-causality argument: influence and information propagate locally through a spacetime site from its past lightcone to its future lightcone. As we will see in the rule 60 example in Sec. V A, the utility of the local causal states as a behavior-driven tool is deeply rooted in this notion of weak-causality.

Using lightcones as local pasts and futures, generalizing the causal equivalence relation to spacetime is now straightforward. Two past lightcones are causally equivalent if they have the same distribution over future lightcones:

$$\ell_i^- \sim_\epsilon \ell_j^- \iff \Pr\left(\mathtt{L}^+ | \ell_i^-\right) = \Pr\left(\mathtt{L}^+ | \ell_j^-\right) . \quad (8)$$

This *local causal equivalence relation* over lightcones implements an intuitive notion of *optimal local prediction* [29]. At some point $\mathbf{x}_t^r$ in spacetime, given knowledge of all past spacetime points that could possibly affect $\mathbf{x}_t^r$—i.e., its past lightcone $\ell^-(r,t)$—what might happen at all subsequent spacetime points that could be affected by $\mathbf{x}_t^r$—i.e., its future lightcone $\ell^+(r,t)$?

The equivalence relation induces a set $\boldsymbol{\Xi}$ of *local causal states* ξ. A functional version of the equivalence relation is helpful, as in the temporal setting, as it directly maps a given past lightcone ℓ^- to the equivalence class $[\ell^-]$ of which it is a member:

$$\begin{aligned} \epsilon(\ell^-) &= [\ell^-] \\ &= \{\ell^{-'} : \ell^- \sim_\epsilon \ell^{-'}\} \end{aligned}$$

or, even more directly, to the associated local causal state:

$$\epsilon(\ell^-) = \xi_{\ell^-} .$$

The local causal states are the unique minimal sufficient statistics of past lightcones to predict future lightcones.

3. *Causal filtering*

As in temporal computational mechanics, the local causal equivalence relation Eq. (8) induces a partition over the space of (infinite) past lightcones, with the local causal states being the equivalence classes. We will use the same notation for local causal states as was used for temporal causal states above, as there will be no overlap later: $\boldsymbol{\Xi}$ is the set of local causal states defined by the local causal equivalence partition, Ξ denotes the random variable for a local causal state, and ξ for a specific realized causal state. The ϵ-function $\epsilon(\ell^-)$ maps past lightcones to their local causal states $\epsilon : \ell^- \mapsto \xi$, based on their conditional distribution over future lightcones.

For spatiotemporal systems, a first step to discover emergent patterns applies the local ϵ-function to an entire spacetime field to produce an associated *local causal state field* $\mathcal{S} = \epsilon(\mathbf{x})$. Each point in the local causal state field is a local causal state $\mathcal{S}_t^r = \xi \in \boldsymbol{\Xi}$.

The central strategy here is to extract a spatiotemporal process' pattern and structure from the local causal state field. The transformation $\mathcal{S} = \epsilon(\mathbf{x})$ of a particular spacetime field realization $\mathbf{x}$ is known as *causal state filtering* and is implemented as follows. For every spacetime coordinate (r,t):

1. At $\mathbf{x}_t^r$ determine its past lightcone $\mathtt{L}^-(r,t) = \ell^-$;
2. Identify its local predictive distribution $\Pr(\mathtt{L}^+ | \ell^-)$;
3. Determine the unique local causal state $\xi \in \boldsymbol{\Xi}$ to which it leads; and
4. Label the local causal state field at point (r,t) with ξ: $\mathcal{S}_t^r = \xi$.

Notice the values assigned to $\mathcal{S}$ in step 4 are simply the labels for the corresponding local causal states. Thus, the local causal state field is a *semantic field*, as its values are not measures of any quantity, but rather labels for equivalence classes of local dynamical behaviors as in the *measurement semantics* introduced in Ref. [30].

4. Topological reconstruction

Being a behavior-driven technique, the local causal states are reconstructed from spacetime field realizations, rather than calculated from a system's equations of motion. As the results presented in Sec. V below rely on symmetries in local causal state fields, it is important to know whether one has a faithful local causal state reconstruction or not.

Using the local causal equivalence relation as given in Eq. (8) presents a challenge since the conditional distributions over lightcones must be inferred from finite realizations. To circumvent this, we instead use *topological reconstruction*. This replaces *probabilistic morphs* $\mathrm{morph}_{\mathcal{P}}(\ell_i^-) = \Pr\left(\mathtt{L}^+|\ell_i^-\right)$ with *topological morphs* $\mathrm{morph}_{\mathcal{T}}(\ell_i^-) = \{\text{all } \ell_j^+ \text{ occurring with } \ell_i^-\}$, which are the supports of the probabilistic morphs. Thus, two past lightcones are topologically (causally) equivalent if they lead to the same *set* of future lightcones. Contrast this with being probabilistically (causally) equivalent, if they have the same *distribution* over future lightcones.

Topological reconstruction is particularly convenient for fully-discrete systems such as CAs, since the topological morphs for finite-depth lightcones can be *exactly* reconstructed and the condition for topological equivalence is *exact*. (That is, are the topological morph sets the same or not?) Moreover, the number of unique past lightcone-future lightcone pairs seen in spacetime field data is monotone increasing, providing a measure of convergence for identifying topological morphs. In short, finite-depth approximations to the topological local causal states can be exactly reconstructed with confidence.

For concreteness, Sec. V's topological reconstruction of local causal states uses past and future lightcone depths of 3 for explicit symmetry domains and past lightcone depth 8 and future lightcone depth 3 for hidden symmetry domains. (Domain types are defined shortly.)

III. DOMAINS OF CELLULAR AUTOMATA

With CAs and the necessary analysis tools in hand, we now turn to explore their "structured" behaviors. In particular, we now consider behaviors called *domains*—terminology borrowed from condensed matter physics. Below we give two distinct domain definitions; one in terms of dynamically-invariant sets, the other in terms of local causal-state symmetries. For both, a formal notion of domain is used to discover and describe these and other emergent spacetime structures that form in CA spacetime fields. Reference [19] compared the structural analyses of CAs using both domain definitions and reported on a strong empirical correspondence between them. The results presented here in Sec. V further bolster this correspondence, leading us to conjecture that the two definitions are equivalent.

A. Spacetime invariant sets

The first systematic analysis of these and related structured behaviors of CAs was done by Crutchfield, Hanson, and McTague [12–17, 31]. Using the FME operator, they discovered sets of spatially (and statistically) homogeneous configurations that are invariant under a CA dynamic Φ.

Presently, we find it useful to restate and reinterpret these results using symbolic dynamics [23]. Recall that a *shift space*, or simply a *shift*, $\mathcal{X} \subseteq \mathcal{A}^{\mathbb{Z}}$ is a compact, shift-invariant subset of the full-$\mathcal{A}$ shift $\mathcal{A}^{\mathbb{Z}}$. A *point* $x = \ldots x_{-2}, x_{-1}, x_0, x_1, \ldots$ in a shift space is an indexed bi-infinite string of symbols in $\mathcal{A}$ and the *shift operator* σ increments the indices of points by one; if $y = \sigma(x)$ for $x \in \mathcal{X}$, then $y_i = x_{i+1}$ and by definition $y \in \mathcal{X}$. As the name suggests, a *sliding block code* $\Phi : \mathcal{X} \to \mathcal{Y}$ maps points from one shift space to another using a sliding-window function ϕ: $y_i = \phi(x_{i-m:i+n})$, where $x \in \mathcal{X}$, $y \in \mathcal{Y}$. We are particularly interested in surjective codes, also known as *factor maps*. The notational overlap with CA dynamics is intentional: CAs are uniform sliding block codes ($m = n = R$) that commute with σ [32].

We now give the spacetime invariant set definition of CA domains using this symbolic dynamics formalism. We consider sets of CA configurations given as shift spaces, and these are invariant sets of the CA if the CA dynamic Φ is a factor map from that shift space to itself.

Definition 1. Consider a CA Φ and a set $\Lambda = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_{\widehat{p}}\}$ of shift spaces $\Lambda_i \subseteq \mathcal{A}^{\mathbb{Z}}$. Together, this set of shifts is a *domain* of Φ if the following hold:

1. *Spatial invariance*: Each $\Lambda_i \in \Lambda$ is an *irreducible sofic shift*. That is, the set of strings in each Λ_i is generated by a strongly-connected finite-state machine $M(\Lambda_i)$.
2. *Temporal invariance*: $\Phi : \Lambda_i \to \Lambda_{i+1(\mathrm{mod}\ \widehat{p})}$ is a factor map from Λ_i to $\Lambda_{i+1(\mathrm{mod}\ \widehat{p})}$. Thus, $\Phi^{\widehat{p}} : \Lambda_i \to \Lambda_i$ is a factor from Λ_i to itself, for all Λ_i.

Each distinct Λ_i is a *temporal phase* of the domain and the number $\widehat{p}$ of temporal phases is the *recurrence time* of the domain—the minimum number of time steps required for Φ to map a temporal phase back to itself. The size s of the minimal cycle in $M(\Lambda_i)$ is the *spatial period* of Λ_i.

For all known examples, the spatial period of each Λ_i in a given Λ is the same; thus, making s the spatial period of the domain.

An ambiguity arises here between Λ's recurrence time $\widehat{p}$ and its *temporal period* p. For a certain class of CA domain (those with explicit symmetries, see Sec. III C), the domain states $x \in \Lambda$ are periodic orbits of the CA, with orbit period equal to the domain period: $x = \Phi^p(x)$. It is less clear how to define the temporal period for domains in general using this formalism. The temporal period appears to be related more to the *spacetime shift spaces* of domain orbits that results from evolving domain spatial shift spaces under Φ. The spacetime shift spaces of hidden symmetry domains are more complicated objects than those of explicit symmetry domains. Notably, at present, beyond particular cases it is not known how to generally relate the domain spatial shifts to their resulting spacetime shifts.

Given a CA Φ, there are no general analytic solutions to $\Phi^{\widehat{p}}(\Lambda_i) = \Lambda_i$. However, given a candidate shift $\mathcal{X}$ it is computationally straightforward to find the factor $\mathcal{Y}$ of $\mathcal{X}$ under Φ using the FME operator. That is, we want to restrict the function-domain of Φ to $\mathcal{X}$ and then find the set $\mathcal{Y}$ of images $y = \Phi(x)$ for all pre-images $x \in \mathcal{X}$ so that Φ is a surjective map from $\mathcal{X}$ to $\mathcal{Y}$. This is exactly what the FME operator does. If $\mathcal{X}$ is an irreducible sofic shift and $\mathcal{X} = \Phi^{\widehat{p}}(\mathcal{X})$ for some $\widehat{p}$, then $\mathcal{X}$ is a domain temporal phase of Φ. Since the FME evolves machines, we technically look for $M(\mathcal{X}) = \Phi^{\widehat{p}}\big(M(\mathcal{X})\big)$, where equality here is given by machine isomorphism [33]. If $\widehat{p} > 1$, the other temporal phases can be found using $\Lambda_{i+1 (\mathrm{mod}\ \widehat{p})} = \Phi(\Lambda_i)$.

B. Local causal-state symmetries

More recently, the local causal states were used to identify special CA behaviors directly from spacetime fields [19], rather than from structures in state space that produce the spacetime fields, as with the invariant sets just described. There, the causal filter $\mathcal{S} = \epsilon(\mathbf{x})$ was used to transform the time- and space-shift invariance of a spacetime field $\mathbf{x}$ into explicit time and space translation symmetries.

Consider a spatiotemporal process $\mathbf{X}$, the set $\boldsymbol{\Xi}$ of local causal states induced by the local causal equivalence relation $\sim_\epsilon$ over $\mathbf{X}$, and the local causal state field $\mathcal{S} = \epsilon(\mathbf{x})$ over the spacetime field realization $\mathbf{x}$. Let σ_p denote the *temporal shift operator* that shifts a spacetime field $\mathbf{x}$ by p steps along the time dimension. This translates a point $\mathbf{x}^r_t$ in the spacetime field as: $\sigma_p(\mathbf{x})^r_t = \mathbf{x}^r_{t+p}$. Similarly, let σ^s denote the *spatial shift operator* that shifts a spacetime field $\mathbf{x}$ by s steps along the spatial dimension. This translates a spacetime point $\mathbf{x}^r_t$ via: $\sigma^s(\mathbf{x})^r_t = \mathbf{x}^{r+s}_t$.

Definition 2. A *pure domain field* $\mathbf{x}_\Lambda$ is a realization such that σ_p and σ^s applied to $\mathcal{S}_\Lambda = \epsilon(\mathbf{x}_\Lambda)$ form a symmetry group [34]. The generators of the symmetry group consist of the following translations:

1. *Temporal invariance*: For some finite time shift p the domain causal state field is invariant:

$$\sigma_p(\mathcal{S}_\Lambda) = \mathcal{S}_\Lambda \ , \tag{9}$$

and:

2. *Spatial invariance*: For some finite spatial shift s the domain causal state field is invariant:

$$\sigma^s(\mathcal{S}_\Lambda) = \mathcal{S}_\Lambda \ . \tag{10}$$

The symmetry group is completed by including these translations' inverses, compositions, and the identity null-shift $\sigma_{\mathbf{0}}(\mathbf{x})^r_t = \mathbf{x}^r_t$. The set $\boldsymbol{\Xi}_\Lambda \subseteq \boldsymbol{\Xi}$ is Λ's *domain local causal states*: $\boldsymbol{\Xi}_\Lambda = \{\big(\mathcal{S}_\Lambda\big)^r_t : \ t \in \mathbb{Z}, r \in \mathcal{L}\}$.

A *domain* Λ of $\mathbf{X}$ is the set of all realizations $\mathbf{x}_\Lambda$ such that $\mathcal{S}_\Lambda = \epsilon(\mathbf{x}_\Lambda)$ contains only local causal states from $\boldsymbol{\Xi}_\Lambda$ and has the defining spacetime symmetries. The set Λ is a spacetime shift space—a closed, spacetime-shift invariant subset of $\mathcal{A}^{\mathbb{Z}\otimes\mathbb{Z}}$.

The smallest integer p for which the temporal invariance of Eq. (9) is satisfied is Λ's *temporal period*. The smallest s for which Eq. (10)'s spatial invariance holds is Λ's *spatial period*.

The domain's *recurrence time* $\widehat{p}$ is the smallest time shift that brings $\mathcal{S}_\Lambda$ back to itself when also combined with finite spatial shifts. That is, $\sigma^j \sigma_{\widehat{p}}(\mathcal{S}_\Lambda) = \mathcal{S}_\Lambda$ for some finite space shift σ^j. If $\widehat{p} > 1$, this implies there are distinct tilings of the spatial lattice at intervening times between recurrences. The distinct tilings then correspond to Λ's *temporal phases*: $\Lambda = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_{\widehat{p}}\}$. For systems with a single spatial dimension, like the CAs we consider here, the spatial symmetry tilings are simply $(\mathcal{S}_\Lambda)_t = \cdots w \cdot w \cdot w \cdots = w^\infty$, where $w = (\mathcal{S}_\Lambda)^{i:i+s}_t$. Each domain phase Λ_i corresponds to a unique tiling w_i.

We note that in contrast to the invariant-set approach of Def. 1, the temporal period of a domain is generally well-defined using the local causal-state definition of domain in Def. 2. In fact, it is a defining property of Def. 2, whereas the recurrence time is the defining property of Def. 1. (Though, the recurrence time is still well defined using the local causal-state approach.) This highlights a key distinction between Def. 1 and Def. 2; the invariant-set approach of Def. 1 defines domains in terms of *spatial shift spaces* and their invariance under Φ, whereas the local causal-state approach of Def. 2 defines domains in terms of generalized symmetry properties of *spacetime shift spaces* that are spacetime field orbits of Φ.

While Defs. 1 and 2 are independent definitions of CA domain that even differ in what mathematical objects are identified as domains, we have found a strong empirical correspondence between these two definitions, as shown in Ref. [19] and here in Sec. V. We formalize this correspondence as follows.

Consider a CA Φ and two domain sets Λ^1 and Λ^2. Λ^1 is a set of spatial shifts that satisfy Def. 1 for Φ: each $\Lambda^1_i \in \Lambda^1$ is an irreducible sofic shift such that $\widehat{\Phi^p}(\Lambda^1_i) = \Lambda^1_i$. Since Λ^1 is a set of shift spaces that are themselves sets of spatial configurations x_{Λ^1}, we can simply think of Λ^1 as the set of all configurations in the collection $\{x_{\Lambda^1} \ : \ x_{\Lambda^1} \in \Lambda^1_i \in \Lambda^1\}$ of invariant spatial shifts. Λ^2 is a spacetime shift space that satisfies Def. 2 for Φ: for each spacetime field orbit $\mathbf{x}_{\Lambda^2} \in \Lambda^2$ the local causal-state field $\mathcal{S}_{\Lambda^2} = \epsilon(\mathbf{x}_{\Lambda^2})$ is comprised of states from $\mathbf{\Xi}_{\Lambda^2}$ and is time- and space- translation invariant: $\sigma_p(\mathcal{S}_{\Lambda^2}) = \mathcal{S}_{\Lambda^2}$ and $\sigma^s(\mathcal{S}_{\Lambda^2}) = \mathcal{S}_{\Lambda^2}$. We conjecture the following bijective relationship between Λ^1 and Λ^2.

Conjecture 1. For each configuration $x_{\Lambda^1} \in \Lambda^1$, its orbit under Φ is in Λ^2 and each spacetime field $\mathbf{x}_{\Lambda^2} \in \Lambda^2$ is the orbit, under Φ, of a configuration in Λ^1. That is, first, for all $x_{\Lambda^1} \in \Lambda^1$, there is $\mathbf{x}_{\Lambda^2} \in \Lambda^2$ such that $\mathbf{x}_{\Lambda^2} = \{x_{\Lambda^1}, \Phi(x_{\Lambda^1}), \Phi^2(x_{\Lambda^1}), \Phi^3(x_{\Lambda^1}), \ldots\}$. And, second, for all $\mathbf{x}_{\Lambda^2} \in \Lambda^2$ there is $x_{\Lambda^1} \in \Lambda^1$ such that $\mathbf{x}_{\Lambda^2} = \{x_{\Lambda^1}, \Phi(x_{\Lambda^1}), \Phi^2(x_{\Lambda^1}), \Phi^3(x_{\Lambda^1}), \ldots\}$.

Taking this conjecture to be true, the following uses Λ and "domain" to refer both to sets of invariant spatial shifts (Λ^1) and the set of orbits of those spatial shifts (Λ^2).

C. CA domain classification: Explicit versus hidden symmetry

CA domains fall into one of two classes: explicit symmetry or hidden symmetry. In the local causal state formulation, a domain Λ has *explicit symmetry* if the time and space shift operators σ_p and σ^s—that generate the domain symmetry group over $\mathcal{S}_\Lambda = \epsilon(\mathbf{x}_\Lambda)$—also generate that same symmetry group over $\mathbf{x}_\Lambda$. That is, $\sigma_p(\mathbf{x}_\Lambda) = \mathbf{x}_\Lambda$ and $\sigma^s(\mathbf{x}_\Lambda) = \mathbf{x}_\Lambda$. From this, we see the following.

Lemma 1. Every explicit symmetry domain configuration $x_\Lambda \in \Lambda$ of a CA Φ generates a periodic orbit of that CA, with the orbit period equal to the domain temporal period.

Proof. This follows since time shifts of the spacetime field are essentially equivalent to applying the CA dynamic Φ: $\mathbf{x}_{t+p} = \sigma_p(\mathbf{x})_t$ and $x_{t+p} = \Phi^p(x_t)$. Thus, if x_Λ is any spatial configuration of a domain spacetime field—$x_\Lambda = (\mathbf{x}_\Lambda)_t$, for any t—then $\Phi^p(x_\Lambda) = x_\Lambda$ if and only if $\sigma_p(\mathbf{x}_\Lambda) = \mathbf{x}_\Lambda$.

A *hidden symmetry domain* is one for which the time and space shift operators, which generate the domain symmetry group over $\mathcal{S}_\Lambda$, do not generate a symmetry group over $\mathbf{x}_\Lambda$: $\sigma_p(\mathbf{x}_\Lambda) \neq \mathbf{x}_\Lambda$ or $\sigma^s(\mathbf{x}_\Lambda) \neq \mathbf{x}_\Lambda$ or both.

Domain classification for the invariant-set formulation is similar. A domain Λ has explicit symmetry if its spatial configurations x_Λ are not just shift invariant but also translation invariant, so that $y = \sigma^s(x_\Lambda) = x_\Lambda$, for all $x_\Lambda \in \Lambda$. If Λ has hidden symmetry it is still shift invariant: $y = \sigma^s(x_\Lambda) \in \Lambda$, but it is not translation invariant: $y = \sigma^s(x_\Lambda) \neq x_\Lambda$. From the machine representation, Λ is a hidden symmetry domain if $M(\Lambda)$ has any local branching in transitions between states. That is, if there is any state in $M(\Lambda)$ such that there is more than one transition leaving that state, then Λ has hidden symmetry. Notably, hidden symmetry domains are associated with a level of stochasticity in their raw spacetime fields. We occasionally refer to these as *stochastic domains*.

The distinction between stochastic and explicit symmetry domains is key to the additivity results to follow.

IV. DOMAINS AND ADDITIVE SUBDYNAMICS

The first identification of what we now call a CA domain came from the observation that the *nonlinear* rule 18, when evolving certain configurations, emulates the *linear* rule 90 [7]. As described in more detail below, we know that the set of configurations over which rule 18 is equivalent to rule 90 is in fact the *domain* of rule 18. Thus, in the case of the nonlinear rule 18, its domain represents a subset of behavior that is actually linear. Below we investigate whether there is a more general connection between domains and linear behaviors of CAs, but first we formalize the notion of a nonlinear CA having a subset of linear behaviors.

A. CA subdynamics

Laying the groundwork for a CA's subdynamics requires clarity on what is meant by the CA dynamic itself. The global dynamic Φ is implemented through synchronous, parallel application of the local dynamic ϕ. And so, Φ and

ϕ are closely related, but they should not be conflated. In what follows, *dynamic* refers to ϕ and, more specifically, to its lookup table. However, since ϕ is the building block of Φ, restrictions on ϕ induce restrictions on Φ. Specifically, these constrain the configurations that we consider evolving under Φ.

A subdynamic of ϕ then is determined by a subset of elements from its lookup table. (This need not be a proper subset, we consider the full dynamic ϕ to be a subdynamic.) To formalize this we must recast ϕ's lookup table (LUT) as a set. We do this by considering elements of the set $\mathrm{LUT}(\phi)$ as tuples of neighborhood values and their outputs under ϕ: $\mathrm{LUT}(\phi) = \{\big(\eta, \phi(\eta)\big) : \text{ for all } \eta \in \mathcal{A}^{2R+1}\}$. We also need to consider higher-order lookup tables. The generalization to $\mathrm{LUT}(\phi^n)$ is straightforward:

$$\mathrm{LUT}(\phi^n) = \{\big(\eta^n, \phi^n(\eta^n)\big) : \text{ for all } \eta^n \in \mathcal{A}^{2nR+1}\} ,$$

where the η^n runs through all length $2nR+1$ words in $\mathcal{A}$.

We consider two methods for removing elements from $\mathrm{LUT}(\phi^n)$ to create a subdynamic of ϕ^n. The first is related to additive dynamics.

1. *Lookup table linearizations*

Recall that an additive local dynamic may be written in the form of Eq. (1). For each x^i, $i \in \{-R, -R+1, \ldots, R-1, R\}$, in neighborhood η there is a coefficient $a_i \in \mathcal{A}$ such that the output of ϕ for that neighborhood is given by Eq. (1). Every linear rule then is specified by a length $2R+1$ vector $\mathbf{a}$ of these coefficients. For example, ECA rule 90 is given by $\mathbf{a}_{90} = (1, 0, 1)$, since $x^i_{t+1} = \phi_{90}(x^{i-1}_t x^i_t x^{i+1}_t) = 1x^{i-1}_t + 0x^i + 1x^{i+1}_t \pmod 2$.

For a linear rule given as a coefficient vector, the vector $\mathbf{o} = \mathbf{a}\mathbf{N}^{\mathrm{T}}$ gives the lookup table outputs, where $\mathbf{N}$ is the matrix of neighborhood values, each row of $\mathbf{N}$ is a neighborhood η, and these are given in lexicographical order. Thus, we can refer to linear rules via their coefficient vector $\mathbf{a}$, which will allow us to easily enumerate every additive CA in a given class.

Consider an arbitrary lookup table $\mathrm{LUT}(\phi_\alpha)$ and an additive lookup table $\mathrm{LUT}(\phi_\beta)$ in the same CA class.

Definition 3. Construct the *ϕ_β-linearization of* ϕ_α, denoted $\phi_{\alpha\leftrightarrow\beta}$, by removing elements $(\eta, \phi_\alpha(\eta))$ from $\mathrm{LUT}(\phi_\alpha)$ if and only if $\phi_\alpha(\eta) \neq \phi_\beta(\eta)$. That is, we only keep elements in $\mathrm{LUT}(\phi_\alpha)$ that are also in $\mathrm{LUT}(\phi_\beta)$.

Being linear, ϕ_β has a coefficient vector $\mathbf{a}_\beta$, which is now also associated with $\phi_{\alpha\leftrightarrow\beta}$. For each element $(\eta, \phi_{\alpha\leftrightarrow\beta}(\eta)) \in \mathrm{LUT}(\phi_{\alpha\leftrightarrow\beta})$ the output $\phi_{\alpha\leftrightarrow\beta}(\eta)$ is given by $\mathbf{a}_\beta \cdot \eta$, treating the neighborhood η as a vector and the dot product sum is performed mod $|\mathcal{A}|$. β may be given as a CA rule number or as a coefficient vector; e.g., $\mathrm{LUT}(\phi_{18\leftrightarrow 90})$ is the same as $\mathrm{LUT}(\phi_{18\leftrightarrow(1,0,1)})$.

Generalizing to higher-order lookup tables, $\mathrm{LUT}(\phi^n_{\alpha\leftrightarrow\beta})$ denotes keeping only elements in $\mathrm{LUT}(\phi^n_\alpha)$ if and only if $\phi^n_\alpha(\eta^n) = \phi^n_\beta(\eta^n)$. At higher powers, there may be linearizations that are not powers of a linear rule in the CA class. For example, take ϕ_α to be an ECA—CA class ($\mathcal{A} = \{0, 1\}, R = 1$). At the second power, we may want a linearization with coefficient vector $\mathbf{a} = (0, 1, 1, 1, 1)$, which is not the second power of a linear ECA. However, we still want this as a possible linearization of $\mathrm{LUT}(\phi^2_\alpha)$. In such cases, the coefficient vector will be used for β: $\mathrm{LUT}(\phi^2_{\alpha\leftrightarrow(0,1,1,1,1)})$.

Notice that the construction of $\mathrm{LUT}(\phi_{\alpha\leftrightarrow\beta})$ is symmetric between $\mathrm{LUT}(\phi_\alpha)$ and $\mathrm{LUT}(\phi_\beta)$ (via set intersection), and so $\mathrm{LUT}(\phi_{\alpha\leftrightarrow\beta}) = \mathrm{LUT}(\phi_{\beta\leftrightarrow\alpha})$. That being said, $\mathrm{LUT}(\phi_{\alpha\leftrightarrow\beta})$ *linearizes* ϕ_α to ϕ_β, while the opposite is not true; $\mathrm{LUT}(\phi_{\beta\leftrightarrow\alpha})$ does not nonlinearize ϕ_β. Any subset of an additive lookup table is necessarily also additive.

2. *Language-restricted lookup tables*

Lookup table linearization, as described, is a procedure for defining a particular CA subdynamic by focusing purely on the lookup table for that CA. Such a subdynamic may not be realizable on a nontrivial set of spatial configurations, though. For example, ECA rule 204 is the identity rule, which is linear with coefficient vector $\mathbf{a}_{204} = (0, 1, 0)$. If we reduce rule 18 to rule 204, the only neighborhoods in $\mathrm{LUT}(\phi_{18\leftrightarrow 204})$ are 000 and 101. The only configurations (longer than 3) with only these neighborhoods are all-0 configurations.

The motivation for the second subdynamic-construction method derives directly from the *configurational consistency* lacking in the $\phi_{18\leftrightarrow 204}$ example. While lookup table linearization subdynamics are constructed based by considering the outputs $\phi_\alpha(\eta)$, language-restricted subdynamics are constructed employing the neighborhoods η themselves.

Consider a sofic language L (Sec. II B), its machine $M(L)$, and an arbitrary CA rule ϕ_α.

Definition 4. Construct the *L-restriction of* ϕ_α, denoted $\phi_{\alpha|L}$, by removing elements $(\eta, \phi_\alpha(\eta))$ from $\mathrm{LUT}(\phi_\alpha)$ if and only if $\eta \notin L$.

That is, consider each neighborhood η as a length $2R+1$ word and only keep the neighborhoods that belong to the language L. Operationally, if $\eta \in L$, there exists a path in $M(L)$ such that concatenation of output symbols along that path give the word η.

This easily generalizes to higher powers of $\mathrm{LUT}(\phi_\alpha)$, as we can consider the neighborhoods η^n as words of length $2nR+1$, keeping elements $(\eta^n, \phi_\alpha^n(\eta^n))$ in $\mathrm{LUT}(\phi_\alpha^n)$ if and only if $\eta^n \in L$.

B. Domain-restricted lookup tables and their linearizations

When a CA lookup table is restricted to one of its domain languages, *temporal consistency* is then added to the configurational consistency of the language-restricted subdynamics. The surprising result is that a nonlinear CA can have such a spatiotemporally-consistent subdynamic which is linear, thus embedding realizable linear behavior in a generally nonlinear system. This is true not only for the case of rule 18 and its domain, but we found that every known ECA domain yields an additive subdynamic, as detailed below. This naturally begs the question, are CA domains *always* linear behaviors—are domain-restricted lookup tables always additive? Further, as we show below, linear CAs can produce *only* domain behaviors—the full-$\mathcal{A}$ shift $\mathcal{A}^{\mathbb{Z}}$ is invariant for any CA who's full lookup table is additive. Another natural question then is whether or not additive subdynamics always corresponds to evolution within an invariant set. In short, long experience and much exploration leads one conjecture the general equivalence:

$$\text{Domain invariant set} \quad \overset{?}{\Longleftrightarrow} \quad \text{Additive subdynamics}\ .$$

Perhaps not surprisingly, the connection between CA domains and additive subdynamics turns out to be more subtle and complicated. Though all ECA domains we know of correspond to additive subdynamics, we know of at least one CA in the class $(\mathcal{A}=\{0,1\}, R=2)$ with a nonlinear domain subdynamic. This counterexample shows it cannot be generally true that CA evolution within a domain invariant set must necessarily be linear evolution.

Given that there is no general equivalence, why is that so many domains *do* correspond to linear subdynamics? While we found a counterexample for CA domain $\implies$ additive subdynamics, extensive exploration has yet to find a counterexample for the converse. Thus, it remains an open question whether additive subdynamics $\implies$ CA domain. In what follows we state these questions formally and explore them with many detailed examples and several rigorous results.

V. EXPLORING DOMAINS AND ADDITIVE SUBDYNAMICS

Before beginning, a few comments about implementing the analysis tools are in order. ECA lookup tables and their subdynamics are easily analyzed by hand, but higher powers of the lookup tables quickly become unmanageable. Fortunately, the subdynamic formalism just developed lends itself to automation. In the following, this was implemented in Python, allowing for exact symbolic exploration of higher-order lookup-table linearizations. For example, given the set $\mathrm{LUT}(\phi^n_{\alpha|L})$ we can enumerate all possible $2nR+1$ linearizations $\mathbf{a}$ to check whether $\mathrm{LUT}(\phi^n_{\alpha|L}) \subseteq \mathrm{LUT}(\phi^n_{\alpha\leftrightarrow\mathbf{a}})$. If so, ϕ^n_α linearizes to $\mathbf{a}$ over language L.

Similarly, topological reconstruction of the local causal states for each example was also implemented in Python. Since the local causal state labels in causal filtering are arbitrary, we use a simple, but arbitrary alphabetical labeling. To avoid confusion, we emphasize that for different CAs there is no connection between states labeled the same. For instance, we give causal filterings of ECAs 90 and 150 in Fig. 2, and in each of these there are local causal states labeled A. There is no relation, however, between state A of rule 90 and state A of rule 150.

For simplicity we limit our discussions to one-dimensional CAs with binary alphabet $\mathcal{A}=\{0,1\}$ and finite radius. Examples will mostly focus on elementary cellular automata $(\mathcal{A}=\{0,1\}, R=1)$.

A. Additive CAs produce only domains

We begin our explorations with linear CAs and find that *all* behaviors they produce are domains. From the invariant set perspective of domain, we state this formally.

Theorem 1. Every nonzero linear CA Φ_β is a factor map from the full-$\mathcal{A}$ shift to itself:

$$\Phi_\beta : \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}\ .$$

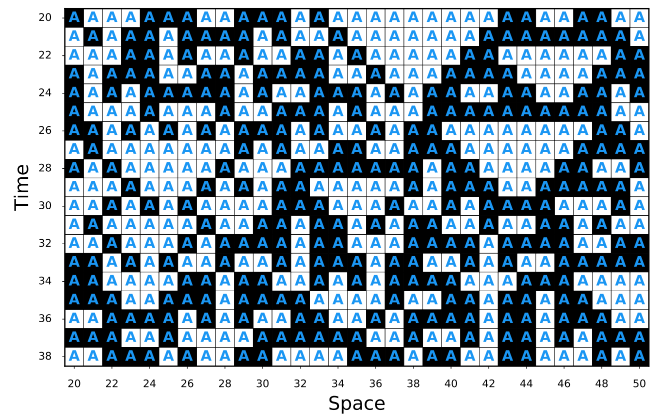
Proof sketch Using linear superposition, Eq. (2), and additivity, Eq. (1), a right inverse $\widetilde{\Phi}_\beta$ can be constructed for any linear Φ_β so that $\widetilde{\Phi}_\beta\big(\Phi_\beta(x)\big) = x$, for all $x \in \mathcal{A}^{\mathbb{Z}}$. See App. A for the detailed proof.

From Thm. 1 we see that every orbit under Φ_β lies in the domain invariant set since the invariant set is the set $\mathcal{A}^{\mathbb{Z}}$ of all spatial configurations. Similarly, this means Φ_β induces no spatial restrictions on its images. Though we must note this result is for bi-infinite spatial configurations

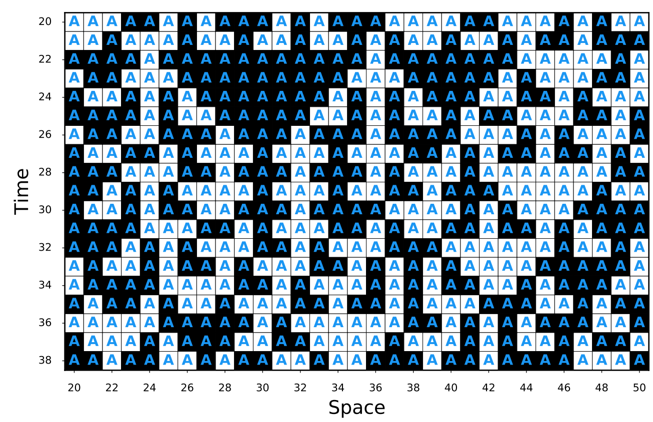
$x \in \mathcal{A}^{\mathbb{Z}}$. If a linear CA evolves configurations using other global topologies (most commonly a finite ring topology) then it may not be surjective. For example, finite spatial configurations with an odd number of sites with value 1 are not reachable by the linear ECA Φ_{90}; there is no y such that $x = \Phi_{90}(y)$ for finite x with an odd number of 1s [5]. Such unreachable states are also known as *Garden of Eden* states.

What is the local causal state signature of additive CAs that only produce domain behaviors? The answer is remarkably simple. When spacetime fields are filtered with the local causal-equivalence relation, there is only a single local causal state. That is, for *every* spacetime field $\mathbf{x}$ produced by Φ_β, the associated local causal state field $\mathcal{S} = \epsilon(\mathbf{x})$ consists of a single state. Thus, the filtered field has trivial time and space translation symmetries. Moreover, since there is only one local causal state, this symmetry can never be broken. And so, all spacetime fields of Φ_β are necessarily pure domain fields.

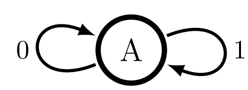
Figure 2 displays spacetime fields $\mathbf{x}$ generated by rule 90, $\mathbf{a}_{90} = (1, 0, 1)$, and rule 150, $\mathbf{a}_{150} = (1, 1, 1)$, with the associated local causal state field $\mathcal{S} = \epsilon(\mathbf{x})$ superimposed on top. The white and black squares are the site values 0 and 1, respectively, of the CA spacetime field. While the blue letters denote the local causal states for each site in the field. Such diagrams, with the local causal state field $\mathcal{S} = \epsilon(\mathbf{x})$ superimposed over its associated CA spacetime field $\mathbf{x}$, are called *local causal state overlay diagrams*.



(a) Rule 90 spacetime field and associated local causal states.



(b) Rule 150 spacetime field and associated local causal states.



(c) Machine $M(\mathcal{A}^{\mathbb{Z}})$ of domain $\Lambda = \mathcal{A}^{\mathbb{Z}}$ for $\phi = 90$ and $\phi = 150$.

FIG. 2. Causally-filtered spacetime fields of (a) rule 90 and (b) rule 150, evolved from random initial conditions. White and black squares represent 0 and 1 CA site values, respectively. While blue letters are the associated local causal state label for that site. (c) Their domains $\Lambda_{90} = \mathcal{A}^{\mathbb{Z}}$ and $\Lambda_{150} = \mathcal{A}^{\mathbb{Z}}$ are described by the single-state machine $M(\mathcal{A}^{\mathbb{Z}})$.

1. *Causal asymmetry of Rule 60*

While $\mathcal{A}^{\mathbb{Z}}$ is invariant under rule 60, the local causal state analysis reveals an interesting subtlety. Using standard lightcones, as depicted in Fig. 1, local causal state inference was run over rule 60 using topological reconstruction. The resulting causal filtering is shown in the overlay diagram of Fig. 3(a). As can be seen, there are multiple causal states (cf. rules 90 and 150 above with one) and there are no obvious spacetime translation symmetries in $\mathcal{S}$.

This occurs since rule 60 breaks the causal symmetry assumption implicit in standard lightcones. Since rule 60, $\mathbf{a}_{60} = (1, 1, 0)$, is the sum mod 2 of the center and left bit in the radius-1 neighborhood the right neighbor is irrelevant to the dynamic. One takes this left-skewed causal asymmetry into account by modifying the lightcone shape used in local causal equivalence. The new, seemingly appropriate lightcone is depicted in Fig. 3(c). Applying local causal state filtering using these half-lightcones yields the overlay diagram of Fig. 3(b): A single local causal state is revealed. Thus, taking into account the causal asymmetry, local causal state analysis in fact demonstrates that rule 60 produces only pure-domain fields. This holds similarly for rule 102, $\mathbf{a}_{102} = (0, 1, 1)$, when the mirror-symmetry right-skewed half-lightcones are used.

Since it demonstrates the consequences (and power) of the weak-causality argument for using lightcones as local pasts and futures, the result here is significant. Tracking weak-causality—how information locally propagates through points in spacetime—is necessary for relating emergent behavior to properties of the system that generated the behavior. Here, we know rule 60 is additive and we know $\mathcal{A}^{\mathbb{Z}}$ is invariant under Φ_{60}, so we know it should have a single local causal state. However, a single local causal state is properly inferred only if we account for rule 60's inherent causal asymmetry.

2. Additive subdynamic generalization

Theorem 1 shows that if $\mathrm{LUT}(\phi_\beta)$ is additive, then Φ_β is a factor from $\mathcal{A}^{\mathbb{Z}}$ to itself. Consider though that $\mathcal{A}^{\mathbb{Z}}$ is in fact a sofic shift; its finite-state machine representation is shown in Fig. 2 (c), and by definition $\mathrm{LUT}(\phi_{\beta|L(\mathcal{A}^{\mathbb{Z}})}) = \mathrm{LUT}(\phi_\beta)$. So, an obvious generalization is to consider a general sofic shift $\mathcal{X}$, rather than $\mathcal{A}^{\mathbb{Z}}$.

Consider an arbitrary CA Φ_α with $\mathrm{LUT}(\phi_{\alpha|L(\mathcal{X})}) \subseteq \mathrm{LUT}(\phi_{\alpha\leftrightarrow\beta})$; that is, $\mathrm{LUT}(\phi_{\alpha|L(\mathcal{X})})$ is additive. Does this imply Φ_α is a factor map from $\mathcal{X}$ to itself: $\Phi_\alpha : \mathcal{X} \to \mathcal{X}$? This cannot be the case, specifically when Φ_α is a linear CA. If ϕ_α is additive, then $\mathrm{LUT}(\phi_{\alpha|L(\mathcal{A}^{\mathbb{Z}})})$ is additive. Thus, any subset of $\mathrm{LUT}(\phi_\alpha)$ will also be additive. In particular, $\mathrm{LUT}(\phi_{\alpha|L(\mathcal{X})})$ will be additive for any sofic shift $\mathcal{X}$. This would mean *every* $\mathcal{X}$ would be an invariant set of Φ_α. This is certainly not the case, however. For example, $\mathcal{X}_{(0,\Sigma)}$ discussed below for rule 90 is not invariant for rule 150.

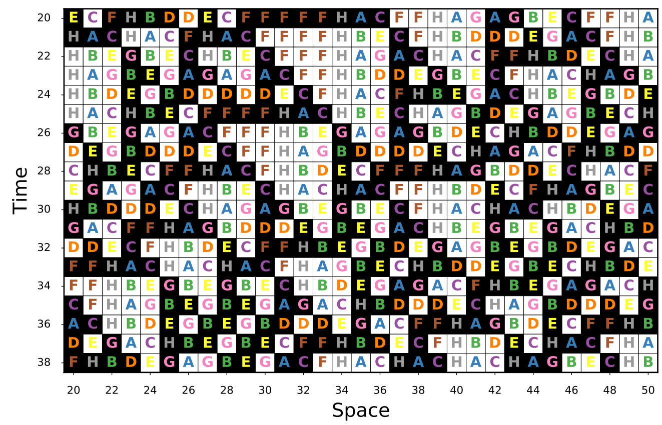
This is not a concern, though, if we consider only nonlinear Φ_α. It may then be the case that for a nonlinear Φ_α and sofic shift $\mathcal{X}$ that if $\mathrm{LUT}(\phi^n_{\alpha|L(\mathcal{X})})$ is additive, then Φ^n_α is a factor map from $\mathcal{X}$ to itself: $\Phi^n_\alpha : \mathcal{X} \to \mathcal{X}$. This though is no longer strictly a generalization of Thm. 1. We note that such a relation between nonlinear CAs Φ_α and sofic shifts $\mathcal{X}$ holds for all known cases, including those presented below.

Conjecture 2. Given an nonadditive Φ_α and a sofic shift $\mathcal{X}$, if $\mathrm{LUT}(\phi^n_{\alpha|L(\mathcal{X})})$ is additive, then Φ^n_α is a factor map from $\mathcal{X}$ to itself:

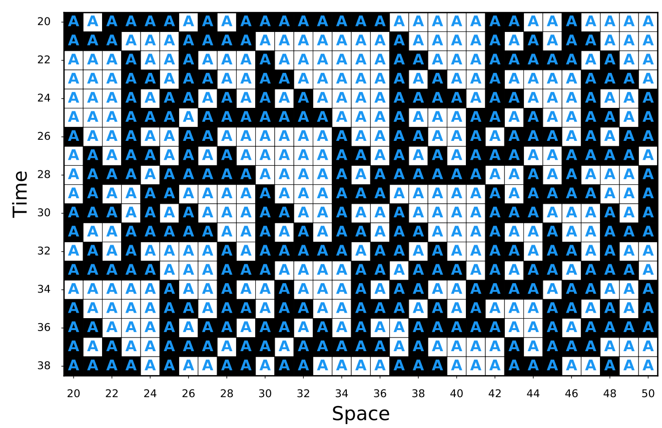
$$\Phi^n_\alpha : \mathcal{X} \to \mathcal{X} \ .$$

The linear CAs, where the full dynamic is additive, produce only domain behavior, as we saw. This leads to the more general question of when a nonlinear CA being additive over a sofic shift $\mathcal{X} \subset \mathcal{A}^{\mathbb{Z}}$ implies the CA is a factor map from $\mathcal{X}$ to itself. This remains an open question, if somewhat more refined by the previous results.

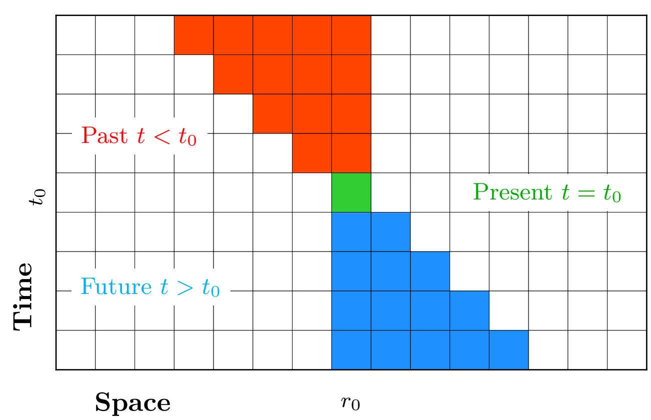
Given this, we now turn to explore in a complementary direction. Does evolution within a domain invariant set always correspond to linear behavior and an additive subdynamic? If not, when does this happen? We start first with explicit symmetry domains where we find that they do indeed always have a corresponding additive subdynamic. In fact, they generally have many additive



(a) Local causal state field of rule 60 using full lightcones.



(b) Local causal state field of rule 60 using half-lightcones.



(c) Half-lightcones for left-skewed causal asymmetry.

FIG. 3. Filtered spacetime field of rule 60, evolved from random initial conditions. White and black squares represent 0 and 1 site values, respectively. Colored letters denote the associated local causal state label for a site. (a) The result of using full lightcones, as depicted in Fig. 1, for local causal state filtering. (b) Local causal state filtering appropriate to the left-skewed causal asymmetry of rule 60, using half-lightcones depicted in (c). This filtering recovers the single-state, trivially symmetric, local causal state field expected for additive CAs.

subdynamics. Afterwards we examine the more complicated case of hidden symmetry domains.

B. Explicit symmetry domains

Lemma 1 established that explicit symmetry domains are periodic orbits of their CA. Due to this, explicit symmetry domains linearize to the identity rule, which for ECAs is rule 204, $\mathbf{a}_{204} = (0, 1, 0)$. For general $\mathcal{A} = \{0, 1\}$ CAs we denote the identity rule as $\mathbf{a}_{\mathrm{I}}$, whose coefficient vector has a single 1 for the center bit and all other elements 0.

Consider a CA Φ_α with an explicit symmetry domain Λ_α with temporal period p.

Theorem 2. The lookup table of ϕ_α, restricted to the domain Λ_α, linearizes to the identity rule at integer multiples of the domain temporal period p; that is:

$$\mathrm{LUT}(\phi^{np}_{\alpha|L(\Lambda_\alpha)}) \subseteq \mathrm{LUT}(\phi^{np}_{\alpha\leftrightarrow\mathbf{a}_{\mathrm{I}}}) \ ,$$

for $n = 1, 2, 3, \ldots$.

Proof. From Lemma 1, any configuration x_{Λ_α} in the domain produces a periodic orbit of ϕ_α, with orbit period p: $\Phi^p_\alpha(x_{\Lambda_\alpha}) = x_{\Lambda_\alpha}$. Since the full configuration returns after p time steps, so do all the individual sites of the configuration: $(x_{\Lambda_\alpha})^{r_0}_{t_0+p} = (x_{\Lambda_\alpha})^{r_0}_{t_0}$. Moreover, $(x_{\Lambda_\alpha})^{r_0}_{t_0+np} = (x_{\Lambda_\alpha})^{r_0}_{t_0}$, for $n = 1, 2, 3, \ldots$. Thus:

$$\begin{aligned}(x_{\Lambda_\alpha})^{r_0}_{t_0+np} &= \phi^{np}_\alpha\big(\eta^{np}((x_{\Lambda_\alpha})^{r_0}_{t_0})\big) \\ &= (x_{\Lambda_\alpha})^{r_0}_{t_0} \\ &= \mathbf{a}_{\mathrm{I}} \cdot \big(\eta^{np}((x_{\Lambda_\alpha})^{r_0}_{t_0})\big) \ .\end{aligned}$$

This is an equivalent statement to $\mathrm{LUT}(\phi^{np}_{\alpha|L(\Lambda_\alpha)}) \subseteq \mathrm{LUT}(\phi^{np}_{\alpha\leftrightarrow\mathbf{a}_{\mathrm{I}}})$.

■

For ϕ_α with an explicit symmetry domain Λ_α, more linearizations are possible, based on the spacetime symmetries of Λ_α. If Λ_α's recurrence time is smaller than its temporal period p, it takes fewer time translations than p to return all sites to themselves if also paired with spatial translations. As the local causal states fully capture the spatiotemporal symmetries of Λ_α, they are particularly convenient for expressing this.

Consider the local causal state field $\mathcal{S} = \epsilon(\mathbf{x})$ of a pure-domain field $\mathbf{x}$. From the definition of domain, $\mathcal{S}^{r_0}_{t_0+p} = \mathcal{S}^{r_0}_{t_0}$. And, for an explicit symmetry domain this means $\mathbf{x}^{r_0}_{t_0+p} = \mathbf{x}^{r_0}_{t_0}$, for all (r_0, t_0), which provides the connection to the identity rule $\mathbf{a}_{\mathrm{I}}$ stated above. From the definition of recurrence time, we have $\mathcal{S}^{r_0+\delta}_{t_0+\widehat{p}} = \mathcal{S}^{r_0}_{t_0}$, for some spatial translation δ. For explicit symmetry domains this again implies the same invariance for $\mathbf{x}$. There may be other spacetime translation symmetry generators such that $\mathcal{S}^{r_0+j}_{t_0+i} = \mathcal{S}^{r_0}_{t_0}$. If $\mathcal{S}^{r_0}_{t_0}$ lies within the past lightcone of $\mathcal{S}^{r_0+j}_{t_0+i}$, there is a linearization of $\mathrm{LUT}(\phi_{\alpha|\Lambda_\alpha})$ at the i^{th} power, with linear coefficient vector $\mathbf{a} = (a_{-iR}, a_{-iR+1}, \ldots, a_{-1}, a_0, a_1, \ldots, a_{iR-1}, a_{iR})$ such that only $a_j = 1$ and all other coefficients are zero.

For example, Λ_{110} has recurrence time $\widehat{p} = 1$, temporal period $p = 7$, and spatial period $s = 14$, as can be seen in Fig. 4(c). Thus, we know $\mathrm{LUT}(\phi^7_{110|L(\Lambda_{110})}) \subseteq \mathrm{LUT}(\phi^7_{110\leftrightarrow 204})$. However, we also see in Fig. 4(c) that $\mathcal{S}^{r_0-2}_{t_0-3} = \mathcal{S}^{r_0}_{t_0}$. If we pick any site in the field, it will have some local causal state label; e.g., A. Shifting up three sites and left two will always take one to same local causal state label. Exact symbolic calculations showed that $\mathrm{LUT}(\phi^3_{110|L(\Lambda_{110})})$ linearizes to $\mathbf{a} = (0, 1, 0, 0, 0, 0, 0)$. Similarly, $\mathcal{S}^{r_0+2}_{t_0-4} = \mathcal{S}^{r_0}_{t_0}$ and $\mathrm{LUT}(\phi^4_{110|L(\Lambda_{110})})$ linearizes to $\mathbf{a} = (0, 0, 0, 0, 0, 0, 1, 0, 0)$.

There may be yet further linearizations. These again derive from Λ_α's symmetries, but the resulting linearizations are not related to the identity rule. Let's start with an example. Consider again rule 110 and its domain, as shown in Fig. 4(c). We find several linearizations of $\mathrm{LUT}(\phi^7_{110|L(\Lambda_{110})})$, including $\mathbf{a}_{\mathrm{I}} = (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0)$, the identity rule discussed at the section's beginning. Another linearization we find at the 7^{th} power is $\mathbf{a} = (0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1)$. To understand this one, and others like it, it is again useful to refer to spacetime diagrams of Λ_α (Λ_{110} in this case).

A linearization $\mathbf{a}$ means a spacetime point $\mathbf{x}^r_t \in \Lambda_\alpha$ is 0 if $\eta^n(\mathbf{x}^r_t) \cdot \mathbf{a}$ is even and $\mathbf{x}^r_t$ is 1 if the dot product is odd, where we treat the neighborhood as a vector. This must be satisfied at *every* spacetime point in a pure domain field $\mathbf{x}_{\Lambda_\alpha}$. If:

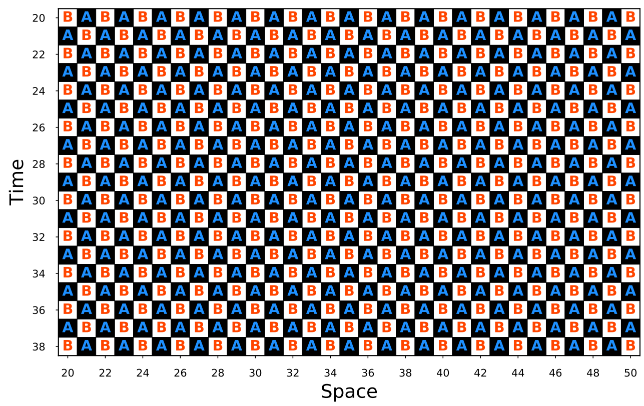
$$\mathbf{x}^{r_0}_{t_0} = \eta^n(\mathbf{x}^{r_0}_{t_0}) \overset{(\mathrm{mod}\ 2)}{\cdot} \mathbf{a} \ , \tag{11}$$

then for an explicit symmetry domain:

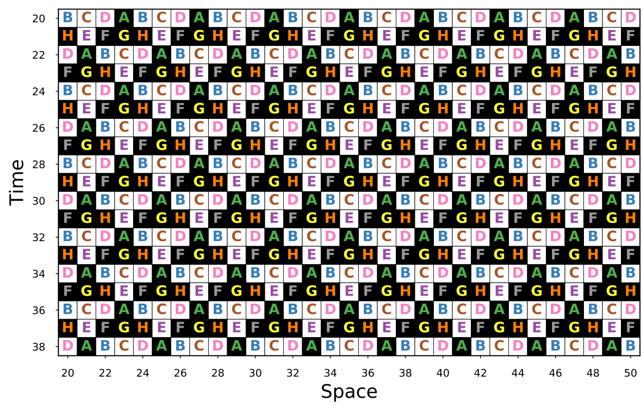
$$\mathbf{x}^{r_0+js}_{t_0+ip} = \eta^n(\mathbf{x}^{r_0+js}_{t_0+ip}) \overset{(\mathrm{mod}\ 2)}{\cdot} \mathbf{a}$$

for $i, j \in \mathbb{Z}$. Thus, there is only a relatively small number of neighborhoods $\eta^n(\mathbf{x}^r_t)$ in spacetime fields of explicit symmetry domains that must satisfy Eq. (11). Said another way, the languages of explicit symmetry domains are very restrictive. This means there are relatively few entries in $\mathrm{LUT}(\phi^n_{\alpha|L(\Lambda_\alpha)})$ that must obey additivity.

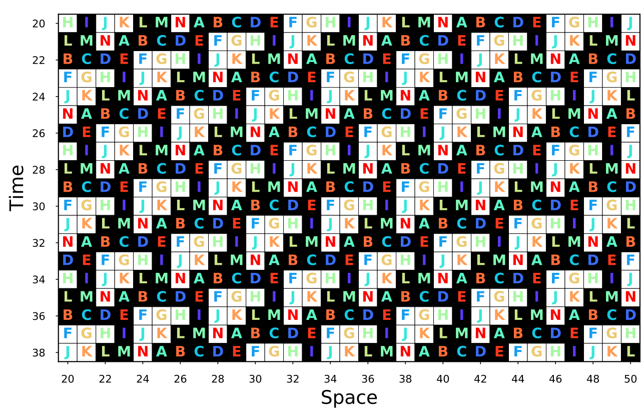
To further illustrate this linearization type, contrast the explicit symmetry domains shown in Fig. 4: (a) the rule 58 domain Λ_{58} with characteristics $\widehat{p} = 1$, $p = 2$, and $s = 2$;



(a) Domain of rule 58 with associated local causal states.



(b) Domain of rule 54 with associated local causal states.



(c) Domain of rule 110 with associated local causal states.

FIG. 4. Filtered spacetime fields from the explicit symmetry domains of rules 58 (a), 54 (b), and 110 (c), with the associated local causal states superimposed on top.

and (b) the rule 54 domain Λ_{54} with characteristics $\widehat{p} = 2$, $p = 4$, and $s = 4$. Now, compare their linearizations at the 4^{th} power, this being a multiple of the temporal period p for both domains.

The 4^{th} power reveals more linearizations for Λ_{58} than for Λ_{54} not related to the identity rule—those with more than one nonzero element in the coefficient vector $\mathbf{a}$. Specifically, there are 29 linearizations for Λ_{54} and 123 for Λ_{58}. This is expected, as Λ_{58} is generated by smaller translations s and p than Λ_{54}. This means at a given power n, there are fewer distinct neighborhoods η^n in Λ_{58} than in Λ_{54}, and so there can be more linearizations $\mathbf{a}$ that can satisfy Eq. (11) everywhere in the field.

Before moving on to hidden symmetry domains, in closing we should highlight the role of spacetime symmetries for linearizations of explicit symmetry domains. Notice that the domain's spatial languages themselves were never directly needed. Rather, the symmetries of the spacetime field orbits and their orbit period (i.e., the domain temporal period) that were key. This reflects the local causal-state perspective of domain, as in Def. 2, coming into play.

C. Hidden symmetry domains

In contrast, as we now show, linearizations of hidden symmetry domains are based on the domain spatial languages and their recurrence times, as in Def. 1. Moreover, unlike explicit symmetry domains, we find that not every hidden symmetry domain has a linearization. After going through detailed examples of ECA domains, we will close with an $R = 2$ CA that has a "nonlinear" domain.

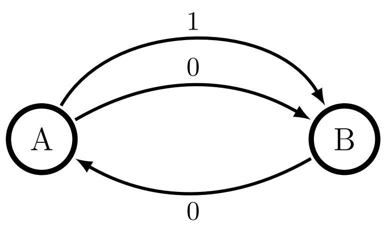
1. *Rule 18*

We start by examining in detail the original observations [7, 9, 12] concerning the domain in the nonadditive CA rule 18 and the domain's linearization to rule 90, $\mathbf{a}_{90} = (1, 0, 1)$. In words, ϕ_{90} updates lattice sites according to the sum mod 2 of that site's left and right neighbors:

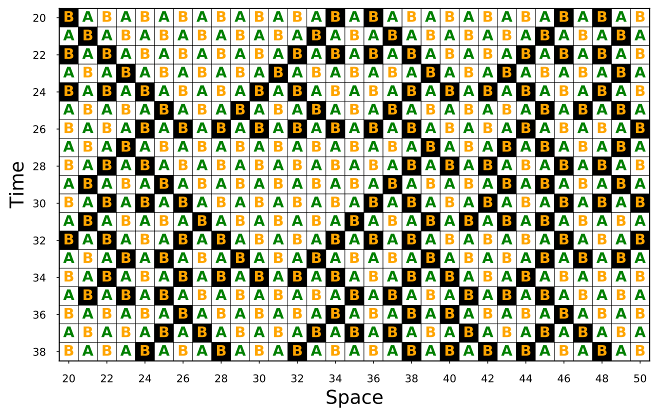
$$\begin{aligned} \mathbf{x}^r_{t+1} &= \phi_{90}\big(\eta(\mathbf{x}^r_t)\big) \\ &= \mathbf{x}^{r-1}_t + \mathbf{x}^{r+1}_t \pmod 2 . \end{aligned}$$

Table I gives rule 90's lookup table.

Rule 18 is not additive, as can be seen from its lookup table also given in Table I. However, there are special behaviors produced by rule 18 that were originally noted due to the equivalence between these behaviors of rule 18 and rule 90. More importantly, they suggested that a *nonlinear rule is capable of producing linear behaviors*. From Refs. [12–16], we know the special behaviors of rule 18 that emulate rule 90 are, in fact, rule 18's domain behaviors.



(a) Finite-state machine $M(\Lambda_{18})$ of invariant set language Λ_{18}.



(b) Sample spacetime field $\mathbf{x}_{\Lambda_{18}}$ of Λ_{18} and associated local causal state field $\mathcal{S}_{\Lambda_{18}} = \epsilon(\mathbf{x}_{\Lambda_{18}})$.

FIG. 5. (a) Finite-state machine $M(\Lambda_{18})$ for the invariant set language $L(\Lambda_{18})$ of the rule 18 domain. (b) Filtered spacetime field $\mathbf{x}_{\Lambda_{18}}$ (white and black squares) of the rule 18 domain Λ_{18} with the associated local causal state field $\mathcal{S}_{\Lambda_{18}} = \epsilon(\mathbf{x}_{\Lambda_{18}})$ (green and orange letters) superimposed.

Rule 18's domain is the set of spatial configurations that is invariant under Φ_{18} and their spacetime field orbits. This invariant set is the single sofic shift $\Lambda_{18} = \{\mathcal{X}_{(0,\Sigma)}\}$, where Σ represents wildcard-sites that can be either 0 or 1. Its domain language is $L(\Lambda_{18}) = (0\Sigma)^* + (\Sigma 0)^*$. The set's finite-state machine $M(\Lambda_{18})$ is shown in Fig. 5(a). Since the machine states lie in a single recurrent component, i.e., it has a single temporal phase, the recurrence time of Λ_{18} is $\widehat{p} = 1$. Its spatial period is $s = 2$, since this is the size of the minimal cycle of $M(\Lambda_{18})$.

Evolving spatial configurations $x \in \Lambda_{18}$ creates spacetime fields—their orbits $\mathbf{x}_{\Lambda_{18}}$. Applying the causal equivalence relation over these fields yields two local causal states, corresponding to the fixed-0 and wildcard sites. A sample spacetime field $\mathbf{x}_{\Lambda_{18}}$ of Λ_{18} and its causal filtering $\mathcal{S}_{\Lambda_{18}} = \epsilon(\mathbf{x}_{\Lambda_{18}})$ are shown as a local causal state overlay diagram in Fig. 5 (b). State A corresponds to the fixed-0 sites and state B the wildcard states. These states appear in a checkerboard tiling in the field, displaying the defining spacetime symmetry of Λ_{18}. At each time step the same two states tile the spatial lattice, giving the recurrence time $\widehat{p} = 1$. The spatial period $s = 2$ and

η	$\phi_{90}(\eta)$	$\phi_{18}(\eta)$	$\phi_{18\leftrightarrow 90}(\eta)$	$\phi_{18\|L(\Lambda_{18})}(\eta)$
1 1 1	0	0	0	−
1 1 0	1	0	−	−
1 0 1	0	0	0	0
1 0 0	1	1	1	1
0 1 1	1	0	−	−
0 1 0	0	0	0	0
0 0 1	1	1	1	1
0 0 0	0	0	0	0

TABLE I. Lookup tables for rule 90 (ϕ_{90}) and rule 18 (ϕ_{18}) as well as for rule 18 linearized to rule 90 ($\phi_{18\leftrightarrow 90}$) and rule 18 restricted to its domain ($\phi_{18|L(\Lambda_{18})}$). The leftmost column gives all ECA neighborhood values in lexicographical order, and each subsequent column is the output of the neighborhoods for the specified dynamic or subdynamic. Symbol − indicates a lookup table element excluded from the respective subdynamic.

temporal period $p = 2$ are found from $\mathcal{S}_{\Lambda_{18}}$'s space and time translation invariance.

Having defined and described rule 18's domain Λ_{18} allows us to explain its linearization to rule 90: Keeping only elements of $\text{LUT}(\phi_{18})$ that are also in $\text{LUT}(\phi_{90})$ gives the linearization $\text{LUT}(\phi_{18\leftrightarrow 90})$ of rule 18 to rule 90. This is shown in Table I. In contrast, keeping only elements of $\text{LUT}(\phi_{18})$ if the neighborhood η_i of that element belongs to the 0-Σ language of Λ_{18} gives the restriction $\text{LUT}(\phi_{18|L(\Lambda_{18})})$ of ϕ_{18} to its domain. As shown in Table I, this subdynamic excludes the neighborhoods $\eta \in \{111, 110, 011\}$. Table I shows that the only two elements that differ between $\text{LUT}(\phi_{90})$ and $\text{LUT}(\phi_{18})$ have neighborhoods $\eta \in \{110, 011\}$. Thus, $\text{LUT}(\phi_{18|L(\Lambda_{18})}) \subset \text{LUT}(\phi_{18\leftrightarrow 90})$. Moreover, as Λ_{18} is a stochastic domain with recurrence time $\widehat{p} = 1$, we find that rule 18 restricted to its domain linearizes to rule 90 at all powers of the lookup tables:

$$\text{LUT}(\phi^n_{18|L(\Lambda_{18})}) \subseteq \text{LUT}(\phi^n_{18\leftrightarrow 90}) ,$$

for $n = 1, 2, 3, \ldots$.

2. *Invariant sets of rule 90*

Historically the 0-Σ domain was of interest because the nonlinear rule 18 exhibits linear behavior over $\Lambda_{0,\Sigma}$, since it emulates the linear rule 90 over $\Lambda_{0,\Sigma}$. However, we now know that $\Lambda_{0,\Sigma}$ is an invariant set of rule 18 and $\text{LUT}(\phi_{18|L(\Lambda_{0,\Sigma})}) \subset \text{LUT}(\phi_{18\leftrightarrow 90})$. This means $\Lambda_{0,\Sigma}$ is also an invariant set of rule 90.

Since $\text{LUT}(\phi_{90})$ is additive, so is $\text{LUT}(\phi_{90|L(\Lambda_{0,\Sigma})})$. And, as described above, there are three neighborhoods $\eta \in \{111, 110, 011\}$ in $\text{LUT}(\phi_{90})$ that are not in $\text{LUT}(\phi_{90|L(\Lambda_{0,\Sigma})})$. So, starting from $\text{LUT}(\phi_{90|L(\Lambda_{0,\Sigma})})$,

η	ϕ_{90}	ϕ_{18}	ϕ_{26}	ϕ_{82}	ϕ_{146}	ϕ_{154}	ϕ_{210}	ϕ_{218}
1 0 1	0	0	0	0	0	0	0	0
1 0 0	1	1	1	1	1	1	1	1
0 1 0	0	0	0	0	0	0	0	0
0 0 1	1	1	1	1	1	1	1	1
0 0 0	0	0	0	0	0	0	0	0
1 1 1	0	0	0	0	1	1	1	1
1 1 0	1	0	0	1	0	0	1	1
0 1 1	1	0	1	0	0	1	0	1

TABLE II. Lookup tables for rule 90 (ϕ_{90}) and the seven nonlinear rules, ϕ_α, $\alpha \in \{18, 26, 82, 146, 154, 210, 218\}$, that also have the $\Lambda_{0,\Sigma}$ domain invariant set. The first five rows correspond to the neighborhoods that belong to the domain language $L(\Lambda_{0,\Sigma})$. Since all eight rules have $\Lambda_{0,\Sigma}$ as a domain, the output for these five neighborhoods in $L(\Lambda_{0,\Sigma})$ are the same. The bottom three rows are the neighborhoods not in $L(\Lambda_{0,\Sigma})$. The eight rules in this table are all possible 2^3 output assignments for these three remaining neighborhoods.

which is additive, there are three unconstrained outputs, one for each $\eta \in \{111, 110, 011\}$, to fill in to create an ECA lookup table that has $\Lambda_{0,\Sigma}$ as a linear invariant set. This is shown graphically in Table II. The neighborhoods are ordered there so that the top five neighborhoods are those in $L(\Lambda_{0,\Sigma})$ and the bottom three are those that are not.

Rule 18 is just one of seven nonlinear rules that have $\Lambda_{0,\Sigma}$ as an invariant set that linearizes to rule 90: $\mathrm{LUT}(\phi^n_{\alpha|L(\Lambda_{0,\Sigma})}) \subseteq \mathrm{LUT}(\phi^n_{\alpha\leftrightarrow 90})$, for $n = 1, 2, 3, \ldots$ and $\alpha \in \{18, 26, 82, 146, 154, 210, 218\}$. Though Rule 146 emulating rule 90 over $\Lambda_{0,\Sigma}$ was pointed out in Ref. [9], to our knowledge the analysis here is the first identification of the linear $\Lambda_{0,\Sigma}$ domain in the nonlinear ECAs 26, 82, 154, 210, and 218. This is likely because $\Lambda_{0,\Sigma}$ does not appear to be a dominant behavior of rules 26, 82, 154, 210, and 218 from random initial conditions. In fact, simple stationary or oscillatory behaviors seem to be the dominant attractors for these rules.

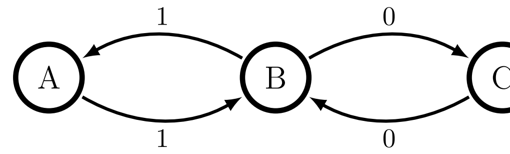
Reference [9] also reported on the nonlinear rule 126 emulating rule 90. This was not over $\Lambda_{0,\Sigma}$ though. Instead it was over a different invariant set, that we call the *even* domain Λ_{even}. This domain also consists of a single temporal phase, which is the sofic shift that contains only even blocks of 1s and 0s. The machine $M(\Lambda_{\mathrm{even}})$ for this domain is shown in Fig. 6. A sample domain spacetime field $\mathbf{x}_{\Lambda_{\mathrm{even}}}$, evolved from a domain configuration initial condition, is shown in Fig. 6 with the associated local causal state field $\mathcal{S}_{\Lambda_{\mathrm{even}}} = \epsilon(\mathbf{x}_{\Lambda_{\mathrm{even}}})$ superimposed. Interestingly, though Λ_{even} and $\Lambda_{0,\Sigma}$ have different invariant spatial shifts, the resulting spacetime shifts of their orbits have the same generalized symmetries, as captured by the local causal states.

The lookup table for rule 126 is compared with that of rule 90 in Table III, as well as its linearization to rule 90 and its restriction to Λ_{even}. From this we can see that $\mathrm{LUT}(\phi_{126|L(\Lambda_{\mathrm{even}})}) = \mathrm{LUT}(\phi_{126\leftrightarrow 90})$. As with $\Lambda_{0,\Sigma}$ and rule 18, this means Λ_{even} is also an invariant set of rule 90. Thus, rule 126 linearizes to rule 90 over Λ_{even} at all powers:

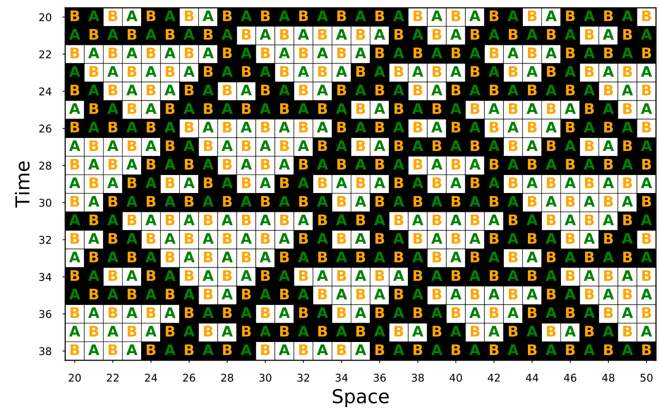
$$\mathrm{LUT}(\phi^n_{126|L(\Lambda_{\mathrm{even}})}) \subseteq \mathrm{LUT}(\phi^n_{126\leftrightarrow 90}) ,$$

for $n = 1, 2, 3, \ldots$.

Following the same combinatorics as with $\Lambda_{0,\Sigma}$, we see



(a) Machine $M(\Lambda_{\mathrm{even}})$ of invariant language $L(\Lambda_{\mathrm{even}})$.



(b) A spacetime field $\mathbf{x}_{\Lambda_{126}}$ of Λ_{126} and associated filtered local causal state field $\mathcal{S}_{\Lambda_{126}} = \epsilon(\mathbf{x}_{\Lambda_{126}})$.

FIG. 6. (a) Machine $M(\Lambda_{\mathrm{even}})$ for the invariant language $L(\Lambda_{\mathrm{even}})$ of the rule 126 even domain. (b) Sample spacetime field $\mathbf{x}_{\Lambda_{126}}$ (black and white squares) of the even domain Λ_{even} of rule 126 with the associated local causal state field $\mathcal{S}_{\Lambda_{126}} = \epsilon(\mathbf{x}_{\Lambda_{126}})$ (green and orange letters) superimposed.

η	$\phi_{90}(\eta)$	$\phi_{126}(\eta)$	$\phi_{126\leftrightarrow 90}(\eta)$	$\phi_{126\|L(\Lambda_{\mathrm{even}})}(\eta)$
1 1 1	0	0	0	0
1 1 0	1	1	1	1
1 0 1	0	1	–	–
1 0 0	1	1	1	1
0 1 1	1	1	1	1
0 1 0	0	1	–	–
0 0 1	1	1	1	1
0 0 0	0	0	0	0

TABLE III. Lookup tables for rule 90 (ϕ_{90}) and rule 126 (ϕ_{126}) as well as for rule 126 linearized to rule 90 ($\phi_{126\leftrightarrow 90}$) and rule 126 restricted to its domain ($\phi_{126|L(\Lambda_{\mathrm{even}})}$). Same format as in Table I.

from Table III that there are only two neighborhoods not in $L(\Lambda_{\text{even}})$. And so, there are 2^2 ECA rules with Λ_{even} as a domain invariant set. Two of these are rule 90 and rule 126; the other two are rule 94 and rule 122. Rule 122 is qualitatively similar to rule 126 over random initial conditions, while rule 94 generically settles into a fixed-point orbit.

Before moving on, clarification is in order. We said that $\Lambda_{0,\Sigma}$ and Λ_{even} are domain invariant sets of rule 90. Formally, from Def. 1, Φ_{90} is a factor map from Λ to Λ for both of these domains. More specifically, this is true for *every* power of Φ_{90}: $\Phi_{90}^n : \Lambda \to \Lambda$ for all $n = 1, 2, 3, \ldots$. This is also holds for all the nonlinear rules just discussed that also have one of these domain invariant sets. Thus, these rules *emulate* rule 90 over their domain. That is, for all of these nonlinear ϕ_α, any orbit starting from an initial configuration $\widehat{x}$ in the appropriate domain Λ is the same if evolved under Φ_α or Φ_{90}:

$$\begin{aligned}&\{\widehat{x}, \Phi_\alpha(\widehat{x}), \Phi_\alpha^2(\widehat{x}), \Phi_\alpha^3(\widehat{x}), \ldots\} \\ &\qquad = \{\widehat{x}, \Phi_{90}(\widehat{x}), \Phi_{90}^2(\widehat{x}), \Phi_{90}^3(\widehat{x}), \ldots\} \ .\end{aligned}$$

3. *Rule 22*

The next example we explore is the enigmatic rule 22 [35], which exhibits a more general notion of linearization. Using symbolic manipulation methods, Crutchfield and McTague used the FME analysis to discover this ECA's domain [36].

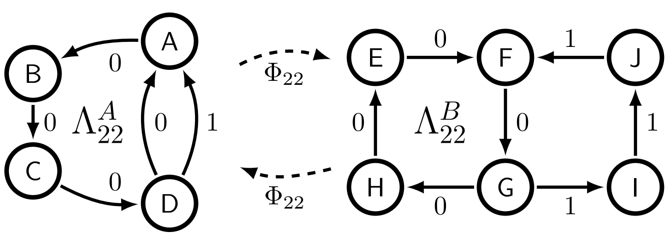


FIG. 7. Machine $M(\Lambda_{22})$ for rule 22's domain Λ_{22}, which has two distinct temporal phases: $\Lambda_{22}^A = \Phi_{22}(\Lambda_{22}^B)$ and $\Lambda_{22}^B = \Phi_{22}(\Lambda_{22}^A)$.

Rules dominated by a stochastic symmetry domain, such as rule 22, are sometimes referred to as "chaotic" CAs. As such, it is typically challenging to extract meaningful structures purely from visually inspecting spacetime fields. So, while not visually apparent, the domain underlying rule 22 is rather more complex than the previous ones. Much of rule 22's mystery stems from its complex domain.

The domain of rule 22 is comprised of two temporal phases, $\Lambda_{22} = \{\Lambda_{22}^A, \Lambda_{22}^B\}$. The machine representation $M(\Lambda_{22})$ of Λ_{22} is shown in Fig. 7. The two components correspond to the irreducible sofic shifts Λ_{22}^A and Λ_{22}^B. A sample spacetime field $\mathbf{x}_{\Lambda_{22}}$ of Λ_{22} is shown in Fig. 8, with the associated local causal state field $\mathcal{S}_{\Lambda_{22}} = \epsilon(\mathbf{x}_{\Lambda_{22}})$ superimposed on top. There are two distinct spatial tilings of the local causal states, ABCD and EFGH, associated with Λ_{22}^A and Λ_{22}^B, respectively, giving a recurrence time $\widehat{p} = 2$. The spacetime translation invariance of $\mathcal{S}_{\Lambda_{22}}$ gives a spatial period of $s = 4$ and temporal period $p = 4$.

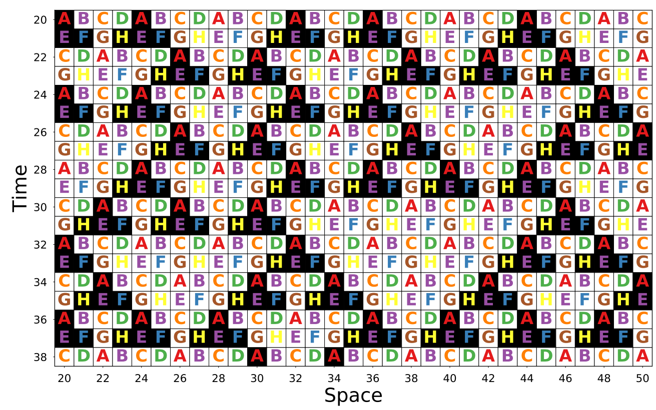


FIG. 8. Filtered spacetime field $\mathbf{x}_{\Lambda_{22}}$ (white and black squares) of the rule 22 domain Λ_{22} with the associated local causal state field $\mathcal{S}_{\Lambda_{22}} = \epsilon(\mathbf{x}_{\Lambda_{22}})$ (colored letters) superimposed.

Since Λ_{22} has two temporal phases, care must be taken when discussing its invariance and linearization. Reference [36]'s FME analysis established that $\Lambda_{22}^A = \Phi_{22}(\Lambda_{22}^B)$ and $\Lambda_{22}^B = \Phi_{22}(\Lambda_{22}^A)$. Thus, Φ_{22} is a factor map from each phase to itself only at the second power: $\Phi_{22}^2 : \Lambda_{22}^A \to \Lambda_{22}^A$ and $\Phi_{22}^2 : \Lambda_{22}^B \to \Lambda_{22}^B$. It is not surprising then that $\mathrm{LUT}(\phi_{22|L(\Lambda_{22})})$ is not additive at the first power, but at the second power. Specifically, $\mathrm{LUT}(\phi^2_{22|L(\Lambda_{22})})$ linearizes again to rule 90. In fact, exact symbolic calculation finds this linearization occurs at all even powers:

$$\mathrm{LUT}(\phi^{2n}_{22|L(\Lambda_{22})}) \subseteq \mathrm{LUT}(\phi^{2n}_{22\leftrightarrow 90}) \ ,$$

for $n = 1, 2, 3, \ldots$.

To be clear, $\mathrm{LUT}(\phi_{22|L(\Lambda_{22})})$ is constructed by keeping only neighborhoods that are in the language $L(\Lambda_{22})$, which is the union $L(\Lambda_{22}) = L(\Lambda_{22}^A) \cup L(\Lambda_{22}^B)$ of the temporal phase languages. Since $\mathrm{LUT}(\phi^{2n}_{22|L(\Lambda_{22})})$ is additive, then its subsets $\mathrm{LUT}(\phi^{2n}_{22|L(\Lambda_{22}^A)})$ and $\mathrm{LUT}(\phi^{2n}_{22|L(\Lambda_{22}^B)})$ are also additive. That is, $\mathrm{LUT}(\phi^{2n}_{22|L(\Lambda_{22}^A)}) \subseteq \mathrm{LUT}(\phi^{2n}_{22\leftrightarrow 90})$ and similarly for phase B. Therefore Λ_{22}^A and Λ_{22}^B are invariant sets of Φ_{90}^2. However, we cannot say Λ_{22} is a domain of rule 90 because $\Lambda_{22}^A \neq \Phi_{90}(\Lambda_{22}^B)$ and $\Lambda_{22}^B \neq \Phi_{90}(\Lambda_{22}^A)$. Thus, rule 22 over its domain does not fully emulate rule 90, they only agree every other time step. Given similar terminology used elsewhere, we may call Λ_{22} a *quasidomain* of rule 90.

Table IV gives the 2nd power of the rule 22 lookup table as well as that for rule 90. It also gives the linearization to rule 90 as well as the restriction of rule 22 to Λ_{22} at the 2nd power, where the first linearization of this subdynamic occurs.

As with $\Lambda_{0,\Sigma}$ and Λ_{even}, rule 22's domain linearizes to the additive rule 90. Rule 90 produces the mod-2 Pascal triangle spacetime patterns characteristic of many chaotic CAs. It is well known that the sum-mod-2 of the neighborhood outer bits is the mechanism that generates these patterns. Since $\Lambda_{0,\Sigma}$ and Λ_{even} are a subset of rule 90's behaviors they also exhibit the mod-2 Pascal triangles, as can be seen in Figs. 5 and 6. Since the discovery of rule 22's domain, it was known that it produces similar Pascal triangle patterns. Though these are not exactly the same as rule 90's, since $\phi_{22|L(\Lambda_{22})}$ only emulates rule 90 every other time step. However, as far as we are aware, rule 22's domain linearization to rule 90 at every even power here is the first report of such a mechanism.

4. *A nonelementary CA*

Section V A showed that additive dynamics always implies a domain invariant set and conjectured that additive subdynamics over a language $L(\mathcal{X})$ potentially also implies $\mathcal{X}$ is an invariant set. Section V B showed that explicit symmetry domains always imply an additive subdynamic, connecting the periodic orbits of these domains with the additive identity rule. So far in Sec. V C we have seen many examples of ECA hidden symmetry domains that similarly are associated with an additive subdynamic, in this case connected to the additive rule 90. However, now we come to an example in the class $(\mathcal{A} = \{0, 1\}, R = 2)$ that breaks the connection between domains and additive subdynamics. It possesses a domain that admits no linearizations.

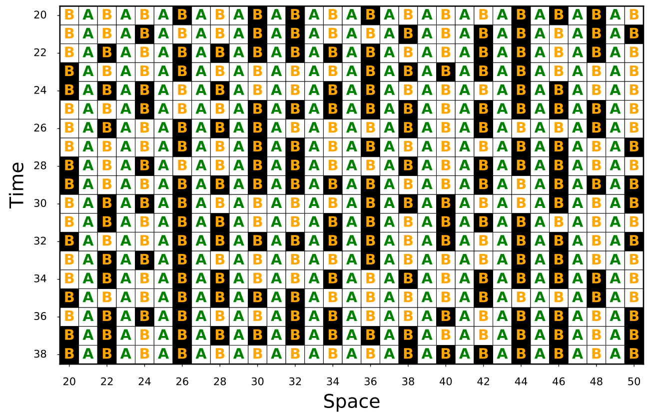
The CA in question is radius-2 rule 2614700074, named according to the same numbering scheme used for ECAs. This was previously studied by Crutchfield and Hanson [15]. They designed it to have the $\Lambda_{0,\Sigma}$ domain along with another structurally distinct domain—the $\Lambda_{1,1,0,\Sigma}$ domain. This domain has a single temporal phase consisting of the sofic shift $\mathcal{X}_{1,1,0,\Sigma}$ with strings of the form $\cdots 110\Sigma110\Sigma110\Sigma\cdots$, where Σ is a wildcard that can be either 1 or 0. The machine for $\Lambda_{1,1,0,\Sigma}$ is shown in Fig. 9(c).

Reference [15] showed that $\Lambda_{0,\Sigma}$ and $\Lambda_{1,1,0,\Sigma}$ have distinct statistical signatures. Here, we investigate these differences via local causal states. Filtered spacetime fields for the $\Lambda_{0,\Sigma}$ and $\Lambda_{1,1,0,\Sigma}$ domains are shown in Fig. 9(a) and (b), respectively. In each, as above, the colored letters represent the local causal state label at each site. These local causal state overlay diagrams clearly demonstrate that the domains have different spacetime symmetry groups. $\Lambda_{1,1,0,\Sigma}$ has recurrence time $\widehat{p} = 1$, temporal period $p = 2$, and spatial period 4, while $\Lambda_{0,\Sigma}$ has recurrence time $\widehat{p} = 1$, temporal period $p = 1$, and spatial period $s = 2$ for rule 2614700074.

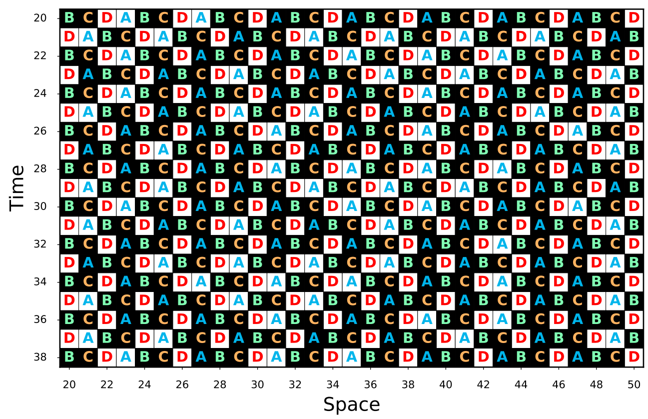
η^2	$\phi^2_{90}(\eta^2)$	$\phi^2_{22}(\eta^2)$	$\phi^2_{22\leftrightarrow 90}(\eta^2)$	$\phi^2_{22\|L(\Lambda_{22})}(\eta^2)$
1 1 1 1 1	0	0	0	−
1 1 1 1 0	1	0	−	−
1 1 1 0 1	0	0	0	0
1 1 1 0 0	1	1	1	1
1 1 0 1 1	0	0	0	0
1 1 0 1 0	1	1	1	−
1 1 0 0 1	0	0	0	−
1 1 0 0 0	1	1	1	1
1 0 1 1 1	0	0	0	0
1 0 1 1 0	1	0	−	−
1 0 1 0 1	0	1	−	−
1 0 1 0 0	1	0	−	−
1 0 0 1 1	0	0	0	−
1 0 0 1 0	1	0	−	−
1 0 0 0 1	0	0	0	0
1 0 0 0 0	1	1	1	1
0 1 1 1 1	1	0	−	−
0 1 1 1 0	0	0	0	0
0 1 1 0 1	1	0	−	−
0 1 1 0 0	0	1	−	−
0 1 0 1 1	1	1	1	−
0 1 0 1 0	0	0	0	−
0 1 0 0 1	1	0	−	−
0 1 0 0 0	0	0	0	0
0 0 1 1 1	1	1	1	1
0 0 1 1 0	0	1	−	−
0 0 1 0 1	1	0	−	−
0 0 1 0 0	0	0	0	0
0 0 0 1 1	1	1	1	1
0 0 0 1 0	0	0	0	0
0 0 0 0 1	1	1	1	1
0 0 0 0 0	0	0	0	0

TABLE IV. Second-order lookup tables for rule 90 (ϕ^2_{90}) and rule 22 (ϕ^2_{22}), as well as for ϕ^2_{22} linearized to ϕ^2_{90} ($\phi^2_{22\leftrightarrow 90}$) and ϕ^2_{22} restricted to its domain ($\phi^2_{22|L(\Lambda_{22})}$). The leftmost column gives all second-order ECA neighborhood values (that is, all radius-2 neighborhood values) in lexicographical order. Each subsequent column is the output of the neighborhoods for the specified dynamic or subdynamic. The symbol − indicates that lookup table element is excluded from the respective subdynamic.

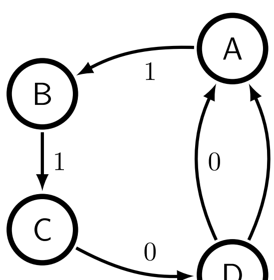
Recall that $\Lambda_{0,\Sigma}$ is a domain invariant set of ECA rule 90. It is thus also a domain invariant set of Φ^2_{90}, which is itself a CA in the class $(\mathcal{A} = \{0, 1\}, R = 2)$: $\mathbf{a} = (1, 0, 0, 0, 1)$. From this we can understand the $\Lambda_{0,\Sigma}$ domain of rule 2614700074 from the same combinatorial perspective as $\Lambda_{0,\Sigma}$ with rule 18. The restric-



(a) Filtered $\Lambda_{0,\Sigma}$ domain with associated local causal states.



(b) Filtered $\Lambda_{1,1,0,\Sigma}$ domain with associated local causal states.



(c) Machine for the domain invariant language $L(\Lambda_{1,1,0,\Sigma})$.

FIG. 9. Filtered spacetime fields of the two domains of the $(\mathcal{A} = \{0, 1\}, R = 2)$ CA rule 2614700074. Format and notation similar to previous such diagrams. (a) The $\Lambda_{0,\Sigma}$ domain has the same invariant set language of ECA rule 90. Its machine is shown in Fig. 5(a)). (b) The $\Lambda_{1,1,0,\Sigma}$ domain. (c) Machine for $L(\Lambda_{1,1,0,\Sigma})$.

tion $\mathrm{LUT}(\phi^2_{90|L(\Lambda_{0,\Sigma})})$ leaves several output values unconstrained for assignment to construct a full CA lookup table. One such assignment gives rule 2614700074. As such, we find that rule 2614700074 linearizes to $\mathbf{a} = (1, 0, 0, 0, 1)$ at all powers:

$$\mathrm{LUT}(\phi^n_{2614700074|L(\Lambda_{0,\Sigma})}) \subset \mathrm{LUT}(\phi^n_{2614700074\leftrightarrow\mathbf{a}=(1,0,0,0,1)}),$$

for $n = 1, 2, 3, \ldots$.

This connection to rule 90 also explains why $\Lambda_{0,\Sigma}$ has temporal period $p = 2$ for rule 18, but temporal period $p = 1$ for CA 2614700074. The local causal state field of $\Lambda_{0,\Sigma}$ for rule 18 has a checkerboard symmetry. If one starts in state A at some spacetime point and moves forward one time step (i.e., applying Φ_{90}) one arrives at state B. However, starting in state A and moving forward two time steps (i.e., applying Φ^2_{90}) one ends in state A again.

The $\Lambda_{1,1,0,\Sigma}$ domain of rule 2614700074, in contrast to all other examples we have seen, does not appear to have any linearizations. From all examples we have seen and know of so far, we would expect rule 2614700074 to linearize over $\Lambda_{1,1,0,\Sigma}$ at its first power since it is a hidden symmetry domain with recurrence time $\widehat{p} = 1$: $\Phi_{2614700074} : \mathcal{X}_{1,1,0,\Sigma} \to \mathcal{X}_{1,1,0,\Sigma}$. Appendix B demonstrates that $\mathrm{LUT}(\phi_{2614700074|L(\Lambda_{1,1,0,\Sigma})})$ cannot be additive. We also algorithmically checked for all possible linearizations of $\mathrm{LUT}(\phi^n_{2614700074|L(\Lambda_{1,1,0,\Sigma})})$ for $n = 1, 2, 3, 4$, finding none. For each $n \in \{1, 2, 3, 4\}$ we constructed the linearization $\mathrm{LUT}(\phi^n_{2614700074\leftrightarrow\mathbf{a}})$ for all possible length $2nR + 1$ coefficient vectors $\mathbf{a}$ and found $\mathrm{LUT}(\phi^n_{2614700074|L(\Lambda_{1,1,0,\Sigma})})$ to be a subset of none of the linearizations.

VI. CONCLUSION

The most basic ingredient of a cellular automaton, its lookup table, could not be simpler—a finite number of possible inputs are enumerated with their outputs explicitly specified. However, the overlapping interactions that occur when applying this simple lookup table synchronously for simultaneous global update of spatial configurations conspires to produce arbitrarily complex behaviors. The emergent complexity enshrouds a cellular automaton's simplicity, making it difficult to answer seemingly basic questions. Specifying a lookup table ϕ determines the global update Φ. Given a lookup table for ϕ, and hence Φ, what invariant sets are induced by Φ in the state space $\mathcal{A}^{\mathbb{Z}}$? In contrast with low-dimensional dynamical systems, the states $x \in \mathcal{A}^{\mathbb{Z}}$ of spatially-extended dynamical systems like CAs possess internal structure and live in infinite dimensions. For a given invariant set $\Lambda \subseteq \mathcal{A}^{\mathbb{Z}}$, is there a unifying structure in the states $x \in \Lambda$? Moreover, is there spacetime structure in the orbits of sequential states in Λ?

Our investigations provide several inroads to these questions, but they also highlight the challenge presented by complex spatially-extended dynamical systems and their emergent behaviors. While domain invariant sets appear to strongly correspond to the spacetime symmetries revealed by the local causal states in the orbit flows along the invariant sets, relating the invariant spatial shift spaces of Def. 1 with the resulting spacetime shift spaces of Def. 2 and their generalized symmetries remains unsolved. Since Φ is deterministic, the spacetime shift space that results from a given spatial shift space $\mathcal{X}$ is uniquely determined by Φ. This is not to say, though, as is often assumed, that the spacetime shift space trivially follows from Φ applied to $\mathcal{X}$. In fact, we still do not know how to fully characterize the spacetime shift space of orbits that follow from hidden symmetry domain invariant spatial shift spaces. In particular, we do not know how to properly define the domain temporal period for these spaces from their invariant spatial shift spaces. Such difficulties in understanding the spacetime shift spaces of hidden symmetry domains is perhaps one of the clearest examples of the fallacy of the "constructionist" hypothesis that often accompanies reductionism [37]. Tackling this will be a focus of subsequent investigations.

Beyond characterizing the spacetime shift spaces of domains, there remains the challenge of connecting domains, both the invariant spatial shift spaces and their resulting spacetime shift space of orbits, to the equation of motion Φ and the lookup table ϕ that generates it. Why does the particular assignment of lookup table outputs that form $\mathrm{LUT}(\phi_{18})$ generate the invariant set $\Lambda_{0,\Sigma}$? Why should $\mathrm{LUT}(\phi_{18|L(\Lambda_{0,\Sigma})})$ be additive when $\mathrm{LUT}(\phi_{18})$ is not? We have shown that this linear behavior of the nonlinear rule 18 actually follows from the combinatorics of $\Lambda_{0,\Sigma}$ being an invariant set of the additive rule 90. In fact, this is the mechanism behind every known stochastic ECA domain, including the enigmatic rule 22. Is this due to historical focus on rule 90? Or, is rule 90 particularly special? Beyond ECAs, we know from the $\Lambda_{1,1,0,\Sigma}$ domain of the $R = 2$ rule 2614700074 that this is not the only mechanism for generating hidden symmetry domains. Hidden symmetry domains need not be associated with an additive subdynamic. One possible path forward could be through the *partial permutivity* outlined in Ref. [38]; the permutive subalphabets of examples 1.2 and 1.3 there correspond to the domains of ECAs 18 and 22, respectively.

Perhaps in sixteenth and seventeenth centuries—the burgeoning days of celestial mechanics—one could take for granted that knowing the equations of motion was tantamount to knowing the system and its behavior. Those days have long passed. Today, we appreciate that having the Navier-Stokes equations of hydrodynamics in hand does not translate into understanding emergent coherent structures, such as fluid vortices. The preceding recounted this lesson yet again. Even systems as "simple" as elementary cellular automata continue to hold surprises.

ACKNOWLEDGMENTS

The authors thank Greg Wimsatt for helpful discussions and feedback. As a faculty member, JPC thanks the Santa Fe Institute and the Telluride Science Research Center for their hospitality during visits. This material is based upon work supported by, or in part by, the U.S. Army Research Laboratory and the U. S. Army Research Office under contract W911NF-13-1-0390 and grant W911NF-18-1-0028 and via Intel Corporation support of CSC as an Intel Parallel Computing Center.

---

[1] J. D. Farmer, T. Toffoli, and S. Wolfram, editors. *Cellular Automata, Proceedings of an Interdisciplinary Workshop*, Amsterdam, 1984. North-Holland Publishing Company.

[2] N. H. Packard and S. Wolfram. Two-dimensional cellular automata. *J. Stat. Physics*, 38(5–6):901–946, 1985.

[3] T. Toffoli and N. Margolis. *Cellular Automata Machines: A New Environment for Modeling*. MIT Press, Cambridge, Massachusetts, 1987.

[4] H. Gutowitz. *Cellular Automata: Theory and Experiment*. Special Issues of Physica D. Bradford Books, Cambridge, Massachusetts, 1991.

[5] O. Martin, A. Odlyzko, and S. Wolfram. Algebraic properties of cellular automata. *Commun. Math. Phys.*, 93:219, 1984.

[6] D. A. Lind. Applications of ergodic theory and sofic systems to cellular automata. *Physica*, 10D:36, 1984.

[7] P. Grassberger. New mechanism for deterministic diffusion. *Phys. Rev. A*, 28:3666, 1983.

[8] S. Wolfram. Statistical mechanics of cellular automata. *Rev. Mod. Phys.*, 55:601, 1983.

[9] E. Jen. Exact solvability and quasiperiodicity of one-dimensional cellular automata. *Nonlinearity*, 4:251, 1990.

[10] C. Moore. Quasilinear cellular automata. *Physica D: Nonlinear Phenomena*, 103(1-4):100–132, 1997.

[11] J. Gravner and D. Griffeath. The one-dimensional Exactly 1 cellular automaton: Replication, periodicity, and chaos from finite seeds. *J. Stat. Physics*, 142(1):168–200, 2011.

[12] J. E. Hanson and J. P. Crutchfield. The attractor-basin portrait of a cellular automaton. *J. Stat. Phys.*, 66:1415 –

1462, 1992.
[13] J. P. Crutchfield. Discovering coherent structures in nonlinear spatial systems. In A. Brandt, S. Ramberg, and M. Shlesinger, editors, *Nonlinear Ocean Waves*, pages 190–216, Singapore, 1992. World Scientific. also appears in Complexity in Physics and Technology, R. Vilela-Mendes, editor, World Scientific, Singapore (1992).
[14] J. P. Crutchfield and J. E. Hanson. Attractor vicinity decay for a cellular automaton. *CHAOS*, 3(2):215–224, 1993.
[15] J. P. Crutchfield and J. E. Hanson. Turbulent pattern bases for cellular automata. *Physica D*, 69:279 – 301, 1993.
[16] J. E. Hanson and J. P. Crutchfield. Computational mechanics of cellular automata: An example. *Physica D*, 103:169–189, 1997.
[17] C. S. McTague and J. P. Crutchfield. Automated pattern discovery—An algorithm for constructing optimally synchronizing multi-regular language filters. *Theo. Comp. Sci.*, 359(1-3):306–328, 2006.
[18] M. G. Nordahl. Limit sets of class two cellular automata. 1988. unpublished.
[19] A. Rupe and J. P. Crutchfield. Local causal states and discrete coherent structures. *Chaos*, 28(7):1–22, 2018.
[20] L. Le Bruyn and M. Van den Bergh. Algebraic properties of linear cellular automata. *Linear Algebra and its Applications*, 157:217–234, 1991.
[21] D. A. Lind. The structure of skew products with ergodic group automorphisms. *Israel J. Math.*, 28(3):205–248, 1977.
[22] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Prentice-Hall, New York, third edition, 2006.
[23] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, New York, 1995.
[24] J. G. Brookshear. *Theory of computation: Formal languages, automata, and complexity*. Benjamin/Cummings, Redwood City, California, 1989.
[25] S. Wolfram. Computation theory of cellular automata. *Comm. Math. Phys.*, 96:15, 1984.
[26] J. P. Crutchfield. Between order and chaos. *Nature Physics*, 8(January):17–24, 2012.
[27] J. P. Crutchfield and K. Young. Inferring statistical complexity. *Phys. Rev. Let.*, 63:105–108, 1989.
[28] C. R. Shalizi and J. P. Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *J. Stat. Phys.*, 104:817–879, 2001.
[29] C.R. Shalizi. Optimal nonlinear prediction of random fields on networks. *Discrete Mathematics & Theoretical Computer Science*, 2003.
[30] J. P. Crutchfield. Semantics and thermodynamics. In M. Casdagli and S. Eubank, editors, *Nonlinear Modeling and Forecasting*, volume XII of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 317 – 359, Reading, Massachusetts, 1992. Addison-Wesley.
[31] B. D. Johnson, J. P. Crutchfield, C. J. Ellison, and C. S. McTague. Enumerating finitary processes. arxiv.org:1011.0036.
[32] G. A. Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Theory of Computing Systems*, 3(4):320–375, 1969.
[33] Crutchfield and McTague implemented an efficient, but exhaustive search algorithm to solve the invariant equation using the enumerated library of machines of Ref. [31]. Reference [36] analyzed ECA 22 using the approach.
[34] W. M. L. Holcombe. *Algebraic Automata Theory*. Cambridge University Press, Cambridge, 1982.
[35] P. Grassberger. Long-range effects in an elementary cellular automaton. *J. Stat. Physics*, 45(1-2):27–39, 1986.
[36] J. P. Crutchfield and C. S. McTague. Unveiling an enigma: Patterns in elementary cellular automaton 22 and how to discover them. 2002. Santa Fe Institute Technical Report.
[37] P. W. Anderson. More is different. *Science*, 177(4047):393–396, 1972.
[38] K. Eloranta. Partially permutive cellular automata. *Nonlinearity*, 6(6):1009, 1993.

Appendix A: Proof of Thm. 1

Theorem 1. Every nonzero linear CA Φ_β is a factor map from the full-$\mathcal{A}$ shift to itself: $\Phi_\beta : \mathcal{A}^\mathbb{Z} \to \mathcal{A}^\mathbb{Z}$.

Proof. For any $x \in \mathcal{A}^\mathbb{Z}$ we want to show that there exists a $y \in \mathcal{A}^\mathbb{Z}$ such that $x = \Phi_\beta(y)$. In other words, for a linear Φ_β there is always a right inverse $\widetilde{\Phi}_\beta$ such that $\Phi_\beta\big(\widetilde{\Phi}_\beta\,(x)\big) = x$.

First, decompose the string x into several 'basis' strings as follows; for each index i in x such that $x_i = 1$, create a basis string x^i for which $x^i_i = 1$ and $x^i_j = 0$ for all $j \neq i$. With this we have $x = \sum_i x^i$, which is shorthand for performing $x_j = \sum_i x^i_j \pmod 2$ at each index j of x.

If we can show the basis strings x^i always have a pre-image y^i, $x^i = \Phi_\beta(y^i)$, then it follows from linear superposition that $y = \sum_i y^i$ is the pre-image of x. If $x = \sum_i x^i = \sum_i \Phi_\beta(y^i)$, then using superposition, Eq. (2), we have $\sum_i \Phi_\beta(y^i) = \Phi_\beta(\sum_i y^i)$. Thus, for any x there is a y such that $\Phi_\beta(y) = \Phi(\sum_i y^i) = \sum_i \Phi_\beta(y^i) = \sum_i x^i = x$.

We now need to show that for any basis string x^i with a single 1 there is always a pre-image y^i. This follows from additivity of ϕ_β. Since we are considering nonzero linear CAs there is at least one coefficient $a_i = 1$ in the additivity coefficient vector $\mathbf{a}$. In fact, it simplifies things to only consider CAs that have one or both of the outer bits of the neighborhood with a nonzero coefficient: $a_{-R} = 1$ or $a_R = 1$ or both. Any rule where this is not the case is equivalent to one that is; e.g., the radius $R = 2$ CA $\mathbf{a} = (0, 1, 0, 1, 0)$ is equivalent to the radius $R = 1$ CA $\mathbf{a} = (1, 0, 1)$. (The only exception is the identity rule with $a_0 = 1$ and $a_i = 0$ for all $i \in \{-R, \ldots, 0, \ldots, R\} \backslash \{0\}$, but clearly every image is its own pre-image for the identity rule, so it is surjective).

Chose i to be either $-R$ or R such that $a_i = 1$. The neighborhood that has $x_i = 1$ and $x_j = 0$ for $j \in \{-R, \ldots, 0, \ldots, R\}$ and $j \neq i$ will necessarily output 1. To find the string that outputs the 3-block 010 there are three neighborhoods to consider, η_{-1}, η_0, and η_1. For the central neighborhood η_0, chose the same as above to still output 1. Neighborhood η_{-1} is to the left of η_0 and η_1 to the right. The outer neighborhoods overlap with η_0 and thus share all but one entry: $\eta_{-1}[-(R-1), \ldots, R] = \eta_0[-R, \ldots, R-1]$ and $\eta_1[-R, \ldots, R-1] = \eta_0[-(R-1), \ldots, R]$. Thus, we need only fill in $\eta_{-1}[-R]$ and $\eta_1[R]$. If $i = -R$, then $\eta_0[-R] = 1$ and $\eta_{-1}[-(R-1)] = 1$. And so, for η_{-1} to output a 0 we set $\eta_{-1}[-R] = 1$ if $a_{-R+1} = 1$ or $\eta_{-1}[-R] = 0$ if $a_{-R+1} = 0$. Meanwhile $\eta_1[R-1] = 0$, so for η_1 to output a 0 we must set $\eta_1[R] = 0$. If $i = R$ perform the symmetric operation: set $\eta_{-1}[-R] = 0$ and $\eta_1[R] = 1$.

To output the 5-block 00100 we similarly have two new neighborhoods to consider, η_{-2} and η_2. As above, we only need to fill in $\eta_{-2}[-R]$ and $\eta_2[R]$. If $i = -R$ we again just set $\eta_2[R] = 0$. Now, simply set $\eta_{-2}[-R]$ to either 0 or 1 so that $\mathbf{a} \cdot \eta_{-2} = 0$. Perform the similar symmetric construction if $i = R$. We can continue in this way to extend out to output arbitrary blocks $\cdots 000 \cdots 1 \cdots 000 \cdots$ with a single 1. Thus, we showed how to construct a pre-image y^i for any basis string $x^i = \cdots 000 \cdots 1 \cdots 000 \cdots$.

All that remains is the case of the all-0 string as an image. Clearly, though, from additivity, Eq. (1), the all-0 string is its own pre-image for all linear CAs.

If Φ_β is surjective over all finite blocks, as we just showed, then Φ_β is necessarily surjective over $\mathcal{A}^{\mathbb{Z}}$. Thus, $y = \sum_i y^i = \widetilde{\Phi}_\beta(x)$ for all $x \in \mathcal{A}^{\mathbb{Z}}$.

■

Appendix B: Domain $\Lambda_{1,1,0,\Sigma}$ of rule 2614700074 is not linear

Reference [15] showed, using the FME, that $\Lambda_{1,1,0,\Sigma}$ is a domain invariant set of rule 2614700074. First, we check that all 5-block words, i.e., all $R = 2$ neighborhoods η, are in the language $L(\Lambda_{1,1,0,\Sigma})$. To do so, consider the string $\cdots 110\Sigma110\Sigma110\Sigma \cdots$ and a sliding 5-block window over this string. This yields the 5-blocks 110Σ1, 10Σ11, 0Σ110, and Σ110Σ. Replacing each Σ with realizations 0 and 1 gives the neighborhoods in $L(\Lambda_{1,1,0,\Sigma})$, from which we can create $\mathrm{LUT}(\phi_{2614700074|L(\Lambda_{1,1,0,\Sigma})})$, shown in Table V.

Now we show that there is no additivity assignment $\mathbf{a} = (a_{-2}, a_{-1}, a_0, a_1, a_2)$ such that $\phi_{2614700074|L(\Lambda_{1,1,0,\Sigma})}(\eta)$ is given by $\mathbf{a} \overset{(\mathrm{mod}\ 2)}{\cdot} \eta$. From the first two rows of Table V we see that $\phi_{2614700074|L(\Lambda_{1,1,0,\Sigma})}(11101) = 0$ and $\phi_{2614700074|L(\Lambda_{1,1,0,\Sigma})}(11100) = 1$. If $\phi_{2614700074|L(\Lambda_{1,1,0,\Sigma})}(\eta)$ is additive, this shows the right-most entry of η must contribute to the additivity sum, and so we must have $a_2 = 1$. Similarly, from $\phi_{2614700074|L(\Lambda_{1,1,0,\Sigma})}(01100) = 0$ and $\phi_{2614700074|L(\Lambda_{1,1,0,\Sigma})}(11100) = 0$ we would need $a_{-2} = 1$. The neighborhoods 11001 and 11011 have the same output, giving $a_1 = 0$. Similarly, 10011 and 10111 have the same output, giving $a_0 = 0$, and 00110 and 01110 have the same output giving $a_{-1} = 0$.

η	$\phi_{2614700074\|L(\Lambda_{1,1,0,\Sigma})}(\eta)$
1 1 1 0 1	0
1 1 1 0 0	1
1 1 0 1 1	1
1 1 0 0 1	1
1 0 1 1 1	1
1 0 0 1 1	1
0 1 1 1 0	0
0 1 1 0 1	1
0 1 1 0 0	0
0 0 1 1 0	0

TABLE V. First-order lookup table of $(\mathcal{A} = \{0,1\}, R = 2)$ CA 2614700074 restricted to its domain $\Lambda_{1,1,0,\Sigma}$. For simplicity, all elements of $\mathrm{LUT}(\phi_{2614700074})$ not in $\mathrm{LUT}(\phi_{2614700074|L(\Lambda_{1,1,0,\Sigma})})$ are not shown.

Therefore, if there is an additivity assignment $\mathbf{a}$, it must be $\mathbf{a} = (1,0,0,0,1)$. However, we can see that $\phi_{2614700074|L(\Lambda_{1,1,0,\Sigma})}(11011) = 1$, for example, and $(1,0,0,0,1) \overset{(\mathrm{mod}\ 2)}{\cdot} (1,1,0,1,1) \neq 1$. And so, $\phi_{2614700074|L(\Lambda_{1,1,0,\Sigma})}(\eta)$ cannot be additive.

■