Dynamical Algebraic Combinatorics, Asynchronous Cellular Automata, and Toggling Independent Sets

Laurent David
University of Texas, Dallas, TX, USA

Colin Defant
Princeton University, NJ, USA

Michael Joseph
Dalton State College, GA, USA

Matthew Macauley
Clemson University, SC, USA

Alex McDonough
Brown University, Providence, RI, USA

Abstract

Though iterated maps and dynamical systems are not new to combinatorics, they have enjoyed a renewed prominence over the past decade through the elevation of the subfield that has become known as *dynamical algebraic combinatorics*. Some of the problems that have gained popularity can also be cast and analyzed as finite asynchronous cellular automata (CA). However, these two fields are fairly separate, and while there are some individuals who work in both, that is the exception rather than the norm. In this article, we will describe our ongoing work on toggling independent sets on graphs. This will be preceded by an overview of how this project arose from new combinatorial problems involving homomesy, toggling, and resonance. Though the techniques that we explore are directly applicable to ECA rule 1, many of them can be generalized to other cellular automata. Moreover, some of the ideas that we borrow from cellular automata can be adapted to problems in dynamical algebraic combinatorics. It is our hope that this article will inspire new problems in both fields and connections between them.

2012 ACM Subject Classification Mathematics of computing → Combinatoric problems; Theory of computation → Formal languages and automata theory; Hardware → Cellular neural networks

Keywords and phrases Asynchronous cellular automata, Covering space, Coxeter element, Dynamical algebraic combinatorics, Group action, Homomesy, Independent set, Resonance, Toggling, Toric equivalence

Digital Object Identifier 10.4230/OASIcs.AUTOMATA.2021.5

Funding *Colin Defant*: National Science Foundation Graduate Research Fellowship (grant no. DGE-1656466); Fannie and John Hertz Foundation Fellowship.
Matthew Macauley: Simons Foundation Collaboration Grant #358242.

Acknowledgements The authors would like to thank the Banff International Research Station, and the organizers of the Dynamical Algebraic Combinatorics Workshop, held virtually in October 2020.

1 Dynamical algebraic combinatorics

The subfield of dynamical algebraic combinatorics covers a wide swath of problems involving actions on combinatorial objects. Of particular interest is the study of the orbit structure, such as their sizes, and the variation of combinatorial statistics within or across orbits. One such example is the concept of *homomesy*, which is a statistic whose average value is constant over all orbits. Another popular new idea in dynamical algebraic combinatorics involves breaking up global maps into compositions of local involutions called *toggles*. These generate

27th IFIP WG 1.5 International Workshop on Cellular Automata and Discrete Complex Systems (AUTOMATA 2021).
Editors: Alonso Castillo-Ramirez, Pierre Guillon, and Kévin Perrot; Article No. 5; pp. 5:1–5:16
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

so-called *toggle groups*, which can be useful in explaining certain structural properties of the orbits, such as the phenomenon of *resonance*. In this paper, we will show how, in some settings, these toggles can be modeled as an asynchronous cellular automaton (CA). For two fun and easy-to-read articles on dynamical algebraic combinatorics, homomesy, and toggling, containing many examples and colorful pictures, we will refer the reader to surveys by Roby [20] and Striker [24]. We will begin this paper with a crash course on homomesy and toggling – just enough to motivate the idea of why one should be interested in toggling independent sets, and then we will show how this can be realized with ECA rule 1. We will discuss our techniques from combinatorics, some of which can be extended to study other CAs, and conversely, how ideas from CAs can be used to pose and explore new questions in combinatorics. However, before we will do any of that, we will begin with a few concepts from the theory of Coxeter groups, which underlies both subjects.

1.1 Some basic Coxeter theory

The toggling operations that we will introduce in this section are involutions, and so the groups generated by these toggles will always be quotients of Coxeter groups [2]. A *Coxeter system* (W, S) consists of a group W generated by a set $S = \{s_1, \dots, s_n\}$ with presentation

$$W = \left\langle s_1, \dots, s_n \mid s_i^2 = 1,\ (s_i s_j)^{m_{ij}} = 1 \text{ for } i \neq j \right\rangle,$$

where $m_{ij} = |s_i s_j| \in \{2, 3, \dots\} \cup \{\infty\}$. We can encode this by a **Coxeter diagram** $\Gamma(W, S)$, which is a graph with vertex set S and an edge $\{s_i, s_j\}$ for each non-commuting pair, labeled with m_{ij}.[1] We will refer to the unlabeled version as the **Coxeter graph**, Γ. A **Coxeter element** of W is the product of the generators in some order, i.e.,

$$c_\pi := s_{\pi_1} s_{\pi_2} \cdots s_{\pi_n} \in W,$$

for some permutation $\pi = \pi_1 \cdots \pi_n \in S_n$. We denote the set of Coxeter elements by $\mathrm{C}(W, S)$. There is an obvious bijection between Coxeter elements and the set $\mathrm{Acyc}(\Gamma)$ of acyclic orientations, where we orient the edge $\{s_i, s_j\}$ as $s_i \to s_j$ if s_i comes before s_j in c_π. Conversely, every acyclic orientation of Γ defines a partial order on S, and the Coxeter element arises from any of its linear extensions.

It is well known that two Coxeter elements $c, c' \in \mathrm{C}(W, S)$ are conjugate if and only their corresponding acyclic orientations are **torically equivalent**, which means that they differ by a sequence of source-to-sink conversions [8]. In [18], Pretzel showed that a complete invariant for this is the **circulation** ν around each cycle, which is the number of "forward" edges minus the number of "backward" edges.[2]

Figure 1 shows five acyclic orientations of a Coxeter graph. The circulation around the unique cycle of the first three orientations is $\nu = 0$. For the fourth, $\nu = 2$, and for the fifth, $\nu = -2$. This means that the first three orientations represent conjugate Coxeter elements, while the other two fall into two different conjugacy classes.

1.2 Homomesy, rowmotion, promotion, and toggles

Consider a set X of combinatorial objects. There are a surprising number of group actions on X where some statistic $X \to \mathbb{R}$, though not constant, has a constant average of c over all orbits. This is called a **homomesy**, and such a statistic is said to be **homomesic**, or **c-mesic**.

[1] Note that s_i and s_j commute if and only if $m_{ij} = 2$.

[2] If the graph is planar, then we can take "forward" to be clockwise and "backward" to be counterclockwise. It is elementary to extend this notion for non-planar graphs, but we do not need to do that here.

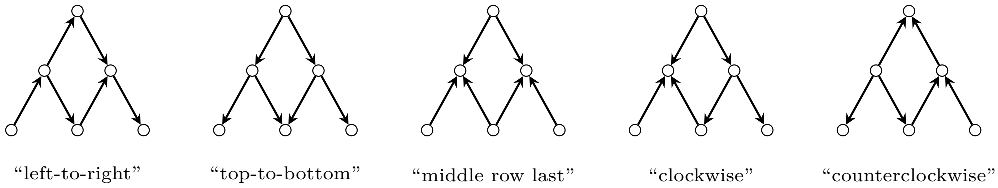

Figure 1 Five acyclic orientations of a Coxeter graph Γ. The first three describe conjugate Coxeter elements, because their circulations around the cycle is $\nu = 0$. The last two have circulations of $\nu = 2$ and $\nu = -2$, respectively.

In many cases, the analysis into why such phenomena arise uncovers the often unexpected fact that the actions can be broken up into a sequence of involutions called *toggles*.

To introduce this idea, let $\mathcal{P}$ be a poset over $[n] = \{1, \dots, n\}$, and let $J(\mathcal{P})$ be its set of **order ideals** (i.e., $I \subseteq \mathcal{P}$ such that if $y \in I$ and $x \leq y$, then $x \in I$). The operation of **rowmotion**, introduced in 1995 by Cameron and Fon-Der-Flaass [3], is a bijection $J(\mathcal{P}) \to J(\mathcal{P})$ defined by sending an ideal I to the ideal generated by the antichain of minimal elements of $\mathcal{P} \setminus I$. An example of rowmotion on the root poset of type A_3 is shown in Figure 2. Elements in the ideals are denoted by solid black dots. Notice that for this example, there are three orbits of sizes 8, 4, and 2.

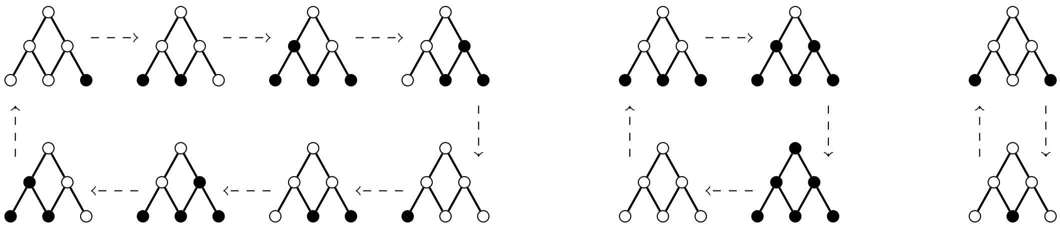

Figure 2 Rowmotion maps an order ideal to the ideal generated by the antichain of minimal elements of its complement. This can also be realized by "toggling" elements in/out (when possible) from top-to-bottom, i.e., along any linear extension of the second acyclic orientation from Figure 1. The number of maximal elements is *homomesic* because its average is 3/2 on all orbits.

A **statistic** on order ideals is a function $J(\mathcal{P}) \to \mathbb{R}$. Three examples, which we will refer back to, include the cardinality of the ideal, the number of minimal elements, and the number of maximal elements. Notice how over the large orbit in Figure 2, the average cardinality is 5/2, the average number of minimal elements is 2, and the average number of maximal elements is 3/2. On the medium orbit, these averages are 7/2, 9/4, and 3/2, respectively. Finally, on the small orbit, they are all 3/2. Consequently, the statistic that counts the number of maximal elements is $\frac{3}{2}$-mesic.

Though it appears infeasible to analyze rowmotion systematically on arbitrary posets, it has some surprising properties on certain families. One such example is the family of *rc-posets* [26], which are characterized by having well-defined notions of "rows" and "columns," such as in Figure 2. For example, if $\mathbf{a}$ and $\mathbf{b}$ are chains of size a and b, respectively, then rowmotion on their product $\mathcal{P} = \mathbf{a} \times \mathbf{b}$ has order $a + b$ and exhibits the cyclic sieving phenomenon [19]. These properties arise because rowmotion can be decomposed as a product of n involutions, called *toggles*, one for each $k \in \mathcal{P}$. Loosely speaking, given an ideal I and an element $k \in \mathcal{P}$ of a poset, toggling at k attempts to add/remove k to/from I. More precisely, **toggling** I **at** k means:

1. if $k \notin I$ and $I \cup \{k\} \in J(\mathcal{P})$, then add it;
2. if $k \in I$ and $I \setminus \{k\} \in J(\mathcal{P})$, then remove it;
3. otherwise, do nothing.

Toggles corresponding to elements k, ℓ commute when neither k nor ℓ covers the other. Thus, in an *rc*-poset such as in Figure 2, the toggles of any two elements in the same horizontal row commute, as do the toggles of any two elements in the same vertical column. Therefore, it is well-defined to speak of toggling elements from top-to-bottom, from left-to-right, etc.

Since each individual toggle is a bijection on $J(\mathcal{P})$, the set of toggles forms a group that we call the **toggle group**, or more specifically, the *order ideal toggle group* $\mathrm{Tog}(J(\mathcal{P})) = \langle t_1, \dots, t_n \rangle$. Clearly, each generating toggle is an involution, and so $\mathrm{Tog}(J(\mathcal{P}))$ is the quotient of a Coxeter group. Following Coxeter theory, we will define a **Coxeter element** to be the product of all toggles of $\mathcal{P}$ in some order. The name "rowmotion" is partially inspired by the fact that it is one of these Coxeter elements, which we will denote as Row.

▶ **Theorem 1.** *If $\mathcal{P}$ is a ranked poset, then rowmotion is the Coxeter element* $\mathsf{Row} \in \mathrm{Tog}(J(\mathcal{P}))$ *defined by toggling across rows, from top-to-bottom.*

One can also ask what is the result of "toggling by columns". Doing so from left-to-right is called *promotion*, denoted $\mathsf{Pro} \in \mathrm{Tog}(J(\mathcal{P}))$, because for certain posets, it agrees with a well-known operation with the same name, first introduced in 1972 by Schützenberger [23]. Promotion was originally defined as an action on the set $\mathcal{L}(\mathcal{P})$ of linear extensions of a poset. Striker and Williams showed that promotion on two-row rectangular tableaux $\mathrm{SYT}(2 \times b)$ is equivalent to promotion on the positive root poset of Coxeter type A_{b-1} [26]. The action of promotion on the root poset of type A_3 is shown in Figure 3.

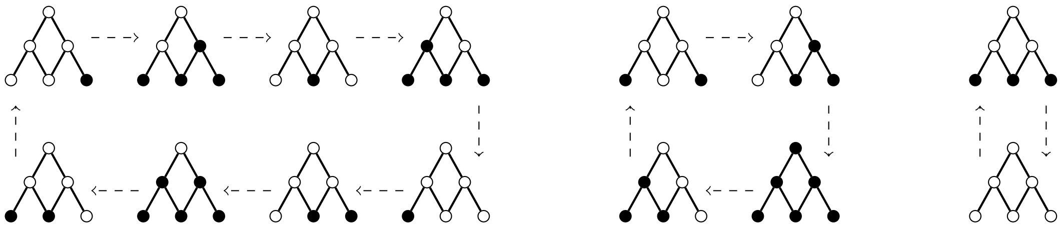

Figure 3 Promotion can be realized by toggling from left-to-right, i.e., any linear extension of the first acyclic orientation in Figure 1. The average cardinality of the ideals, and of the minimal elements, is the same as for rowmotion, but the statistic that counts the number of maximal elements is no longer homomesic. In particular, its average is 13/8, 5/4, and 3/2 across the three orbits.

One immediate observation from Figures 2 and 3 is that promotion and rowmotion have the same orbit structure. This is a simple consequence from Coxeter theory – since the acyclic orientation defined by toggling top-to-bottom has the same circulation as the one defined by toggling left-to-right, they are torically equivalent. Therefore, the corresponding Coxeter elements Row and Pro are conjugate in $\mathrm{Tog}(J(\mathcal{P}))$. However, not all statistics are preserved by conjugation. Observe that the average order ideal cardinality, and the average number of minimal elements, are the same for promotion and rowmotion. However, the number of maximal elements is no longer homomesic: it is 13/8 for the large orbit, 5/4 for the medium orbit, and 3/2 for the small orbit. It is easy to check from Figures 2 and 3 that none of these statistics are preserved elementwise by conjugation from rowmotion to promotion, but the average number of elements in each orbit is preserved.

Another fundamental feature of an action on a set of combinatorial objects is the collection of orbit sizes. Though each of these has to divide the size of the toggle group, in practice, such groups are often large, and this does not provide meaningful information. In a number of cases, computational experiments are used to answer this question for small n. It has often been observed that most, but not necessarily all, of the orbits in an action are divisible by some integer $\omega > 1$. Such actions are said to **resonate** with ω, though for years, people did not know exactly how to precisely define this, because it involved "most," rather than "all" orbits. For example, on the poset $\mathcal{P} = \mathbf{a} \times \mathbf{b} \times \mathbf{c}$, a product of three chains, the sizes of most orbits under rowmotion are multiples of $a + b + c - 1$. The sizes of all 21156 rowmotion orbits of $\mathcal{P} = \mathbf{4} \times \mathbf{4} \times \mathbf{4}$ are multiples of 11: six have size 33 and the others all have size 11. However, for the rowmotion orbits of $\mathcal{P} = \mathbf{4} \times \mathbf{4} \times \mathbf{5}$, most of the orbits have sizes divisible by 12 (136822 of size 12, 4 of size 24, 100 of size 36), but there are also 6 orbits of size 2, 8 of size 3, and 162 of size 6. In 2015, the concept of *resonance* was formally defined, involving commutative diagrams of actions of cyclic groups [6].

1.3 Generalized toggling

The idea of toggling an order ideal I at an element k by attempting to add/remove it is easy to generalize to other combinatorial objects [25]. Consider a finite set X, such as a graph or a poset, and a collection $\mathcal{L} \subseteq 2^X$ of subsets that have a particular property (such as "is an order ideal"). For each $k \in X$ and subset $E \in \mathcal{L}$, we say that **toggling E at k** is the function

$$t_k \colon \mathcal{L} \longrightarrow \mathcal{L}, \qquad t_k(E) = \begin{cases} E \cup \{k\} & k \notin E \text{ and } E \cup \{k\} \in \mathcal{L} \\ E \setminus \{k\} & k \in E \text{ and } E \setminus \{k\} \in \mathcal{L} \\ E & \text{otherwise.} \end{cases}$$

We sometimes speak of the first case above as "*toggling in k*" and the second case as "*toggling out k*." It should be clear that t_k is either the identity or an involution.

For some examples of generalized toggling, we can always toggle out, but not always toggle in. For example, if we toggle the set $\mathcal{L} = \mathcal{A}(\mathcal{P})$ of antichains of a poset $\mathcal{P}$, then $t_k(A) = A \setminus \{k\}$ whenever $k \in A$, because removing any element from an antichain results in another antichain. This is closely related to order ideal toggling, because every order ideal is uniquely determined by an antichain – the minimal elements of its complement. In particular, the toggle groups are isomorphic, despite the individual toggles having fundamentally different properties [9]. In contrast, there are other settings where we can always toggle in, but not necessarily toggle out. Toggling vertex covers, edge covers, or dominating sets of a graph are such examples.

Sometimes, there are sets of objects that do not seem to be "togglable" upon first glance, but admit a nice toggle action if one views them the right way. For example, a **noncrossing partition** of $[n]$ is a partition such that if the numbers $1, \ldots, n$ are arranged cyclically, then the convex hulls of the blocks are disjoint. An alternate way to represent a noncrossing partition is with an **arc diagram**: draw the numbers $1, \ldots n$ in a line, and draw an arc between all consecutive vertices in the same block. By construction, every collection of arcs on $[n]$ represents a noncrossing partition if it does not contain any of the three motifs shown in Figure 4.

Figure 4 Disallowed pairs of arcs in a noncrossing partition: crossing, left-nesting, and right-nesting, respectively.

Let $X = \{\{i, j\} \mid 1 \leq i < j \leq n\}$ be the set of arcs on $[n]$, and $\mathrm{NC}(n) \subseteq 2^X$ the set of noncrossing partitions. It should now be easy to see how to toggle a noncrossing partition $\pi \in \mathrm{NC}(n)$. Specifically, for each arc $\{i, j\} \in X$, the toggle $t_{i,j}$ removes it from π if it already contains it. Otherwise, $t_{i,j}$ adds the arc, as long as $\pi \cup \{\{i,j\}\}$ does not contain one of the forbidden motifs from Figure 4. Notice that arcs can always be toggled out, but not necessarily in. Figure 5 shows an example of toggling a noncrossing partition of size $n = 12$, both as arc diagrams and as convex hulls. Computational experiments originally suggested the surprising fact that under any Coxeter element, the arc count statistic is $\frac{n-1}{2}$-mesic, and this was proven in [7].

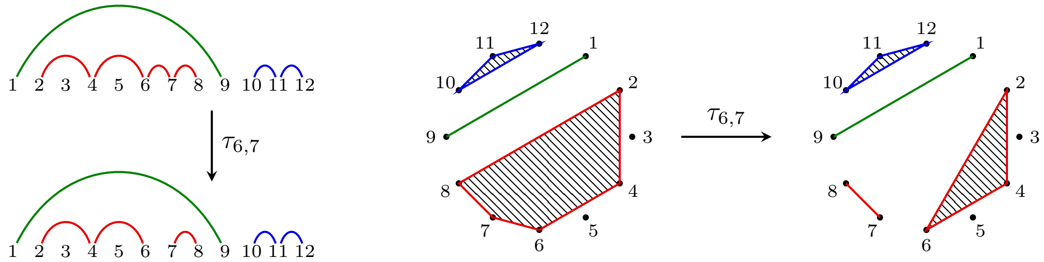

Figure 5 An example of toggling a noncrossing partition of $[n] = [12]$.

The Coxeter graph of $\mathrm{Tog}(\mathrm{NC}(n))$ can be constructed by putting the toggle $t_{i,j}$ in the (i,j) position of a grid, arranged in the upper-triangular entries of a matrix. In doing so, each column and each row is a complete graph (i.e., no two toggles in the same row or column commute), and there are some other edges as well, the details of which are not important here, other than the fact that they all have a "negative slope." An example of this graph for $n = 5$ is shown in Figure 6. One key observation is that the Coxeter element by rows (from top-to-bottom) and the Coxeter element by columns (from left-to-right) both arise from the same acyclic orientation, constructed by orienting all edges in the "south, east, or southeast directions." In other words, toggling by rows is the same element in the toggle group as toggling by columns. It turns out that this element is the inverse of the well-studied *Kreweras complement* κ on noncrossing partitions [11], an operation that has the curious property that κ^2 is a rotation of the "convex hull" diagram by $2\pi/n$. In [1], this map, along with rowmotion, was used to construct a uniform bijection between noncrossing and nonnesting partitions in Weyl groups of all classical types.

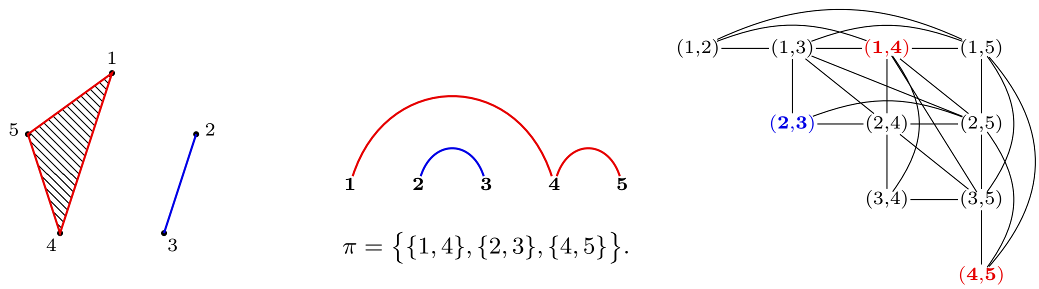

Figure 6 There is a bijection between noncrossing partitions on $[n]$ and independent sets of the Coxeter diagram of the NC toggle group. An example of this for $n = 5$ is shown.

The arcs in a noncrossing partition form an independent set of the Coxeter graph, and every such independent set corresponds to a noncrossing partition. In other words, toggling noncrossing partitions is just a special case of a more general construction: toggling independent sets in a graph. However, the Coxeter graphs of the NC toggle group are quite complicated. The third author and Roby studied toggling independent sets of the path graph in [10], and the toggle groups were recently shown to be symmetric groups [17]. In an ongoing project [4], the authors of this paper are studying toggling independent sets of the cycle graph. Early computational experiments, motivated by the search for examples of homomesy and resonance, brought other curious properties to light. For example, if one records the frequency with which each vertex is toggled in over time, this cyclic vector always has odd period under the update order $\pi = 1, \ldots, n$, but not necessarily under other update orders. There are interesting observations to be made about the period and the sizes of the orbits, as well as how this depends on the update order and on n. In the next section, we will show how this framework can be viewed as an asynchronous CA. We will also describe the analytical tools and techniques we developed, inspired by ideas from dynamical algebraic combinatorics, that we think are mostly new in the field of cellular automata. Additionally, we will discuss ideas from cellular automata that we are bringing back to the field of dynamical algebraic combinatorics.

2 Toggling independents sets in a cellular automata framework

2.1 Asynchronous cellular automata

A *cellular automaton* (CA) is a discrete dynamical system over a regular grid. A one-dimensional CA is either over an infinite path or a cycle graph $\mathcal{C}_n$. We will assume that $\mathcal{C}_n$ has vertex set $v(\mathcal{C}_n) = [n]$, sometimes called **cells**, and take the indices modulo n, when appropriate. Vertex i has a Boolean state $x_i \in \mathbb{F}_2 = \{0, 1\}$ and a function that updates this state based on the states of itself and its neighbors. We encode each function $f_i \colon \mathbb{F}_2^3 \to \mathbb{F}_2$ with a number between 0 and 255, based on the binary string $a_7 a_6 \cdots a_1 a_0$, defined from its truth table, as follows:

$x_{i-1}x_ix_{i+1}$	111	110	101	100	011	010	001	000
$f_i(x_{i-1}, x_i, x_{i+1})$	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

These functions are called **elementary cellular automata (ECA) rules**, and have been well-studied since CAs gained popularity in the 1980s. In a classical CA, the individual functions are updated synchronously to define the dynamical system map $f \colon \mathbb{F}_2^n \to \mathbb{F}_2^n$, which is iterated to generate the global dynamics. In this section, we will consider a common asynchronous framework, where the dynamical system map is defined by updating each function in some predetermined order, sometimes called a *fixed sweep* [22].

Fix an ECA rule, and for each vertex function $f_i \colon \mathbb{F}_2^3 \to \mathbb{F}_2$, define the function

$$F_i \colon \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n, \qquad F_i \colon (x_1, \ldots, x_i, \ldots, x_n) \longmapsto \big(x_1, \ldots, f_i(x_{i-1}, x_i, x_{i+1}), \ldots, x_n\big), \tag{1}$$

which simply updates the global state vector $(x_1, \ldots, x_n)$ at only the i^{th} position. Let $\pi = \pi_1 \cdots \pi_n \in S_n$ be a permutation of the vertices of $\mathcal{C}_n$, written as a word in $v(\mathcal{C}_n)$. The fixed sweep **asynchronous cellular automaton (ACA)** map with respect to π is

$$F_\pi \colon \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n, \qquad F_\pi = F_{\pi_n} \circ \cdots \circ F_{\pi_1}.$$

A point $x \in \mathbb{F}_2^n$ is *periodic* if $F_\pi^k(x) = x$ for some $k \geq 1$, and *transient* otherwise. The fixed points are those for which $F_\pi(x) = x$, and they do not depend on π. The periodic points are partitioned into **periodic orbits**. An ACA is a special case of a *sequential dynamical system* (SDS), which is defined over arbitrary graphs and with update functions that need not be the same at every vertex [16].

The dynamics of an ACA depends on the update order $\pi = \pi_1 \cdots \pi_n$, which we can encode as an acyclic orientation O_π of $\mathcal{C}_n$, just as we did for Coxeter elements. This defines a map

$$S_n \longrightarrow \mathrm{Acyc}(\mathcal{C}_n), \qquad \pi \longmapsto O_\pi,$$

and it is easy to see that if $O_\pi = O_{\pi'}$, then $F_\pi = F_{\pi'}$ as ACA maps, for any fixed ECA rule. Moreover, if O_π and $O_{\pi'}$ are torically equivalent, then the ACA maps F_π and $F_{\pi'}$ are topologically conjugate when restricted to their respective sets of periodic points [14].

2.2 Togglable ACAs

An ACA is **togglable** if its set of periodic points does not depend on the update order $\pi \in S_n$. When this happens, each function F_i restricted to the periodic points is a bijection, and $F_i^2 = 1$, since it either fixes or flips each i^{th} bit. In other words, if ECA rule k is togglable, then we have a natural toggle action on the set $\mathrm{Per}_n^{(k)}$ of periodic points, and we denote the toggle group by

$$\mathrm{Tog}_n^{(k)} = \langle F_1, \ldots, F_n \rangle .$$

It is known that 104 of the 256 ECA rules are togglable [13], and their toggle groups have been classified [12]. Those two papers predate the notion of generalized toggling, and so in them, the property of being togglable is called *order independent*, and the toggle groups are called *dynamics groups*. It is easy to see that $\mathrm{Tog}_n^{(k)}$ is trivial if and only if all periodic points are fixed points.

Every ECA rule is equivalent to the ECA rules formed by “mirror reflection” (swapping x_{i-1} with x_{i+1}), “logical complement” (swapping 0 with 1), and doing both operations. Some rules are their own reflection, own complement, or both. As such, the 256 ECA rules are partitioned into 88 equivalence classes, of sizes 1, 2, and 4. Of these, exactly 41 classes are togglable, but some are for obvious reasons: 26 have trivial toggle groups (i.e., only fixed points), 9 are reversible (i.e., the maps $\mathbb{F}_2^n \to \mathbb{F}_2^n$ in Eq. (1) are invertible), and 1 is both. Some of the invertible rules have interesting toggle groups, and these are described in Section 5 of [12]. The conjectured group for ECA rules 54 and 57 (also conjectured independently in [27]) was recently proven by Salo [21]. Two of the invertible rules: the parity function and its complement (rules 150 and 105), each of which is in a size-1 equivalence class, were recently studied in the context of toggling by the second author [5].

In this paper, we are doing the opposite: taking a problem from combinatorics, toggling independent sets in $\mathcal{C}_n$, and studying it in an ACA framework. Specifically, an **independent set** of $\mathcal{C}_n$ is a length-n binary string with no consecutive 1's, including wrapping around the end. Let $\mathcal{I}_n$ be the set of independent sets of $\mathcal{C}_n$. Toggling in $\mathcal{I}_n$ can be realized by one of several rules, such as ECA rule 1, or the invertible rule 201:

$x_{i-1}x_ix_{i+1}$	111	110	101	100	011	010	001	000
$f_i^{(1)}(x_{i-1}, x_i, x_{i+1})$	0	0	0	0	0	0	0	1
$f_i^{(201)}(x_{i-1}, x_i, x_{i+1})$	1	1	0	0	1	0	0	1

The only differences between these two rules are on the three inputs that do not arise for independent sets: 111, 110, and 011. We are primarily interested in studying the dynamics of ECA rule 1 and $\mathrm{Tog}_n^{(1)}$, because $\mathrm{Per}_n^{(1)} = \mathcal{I}_n$, whereas $\mathrm{Per}_n^{(201)} = \mathbb{F}_2^n$. However, since rule 201 is invertible, we have a surjective homomorphism $\mathrm{Tog}_n^{(201)} \to \mathrm{Tog}_n^{(1)}$ between toggle groups.

The third author and Roby studied toggling independent sets of path graphs [10]. One simplification of this framework is that all acyclic orientations of a tree are torically equivalent, and so all Coxeter elements are conjugate. In contrast, over the cycle graph $\mathcal{C}_n$, there are $n-1$ toric equivalence classes, distinguished by their circulations $\nu = n-2, n-4, \ldots, -(n-4), -(n-2)$. An example of this for $n=6$ is shown in Figure 7. However, since a vertical reflection, which is an automorphism of $\mathcal{C}_n$, reverses the sign of the circulation, there are at most $\lfloor \frac{n}{2} \rfloor$ equivalence classes up to dynamical equivalence.

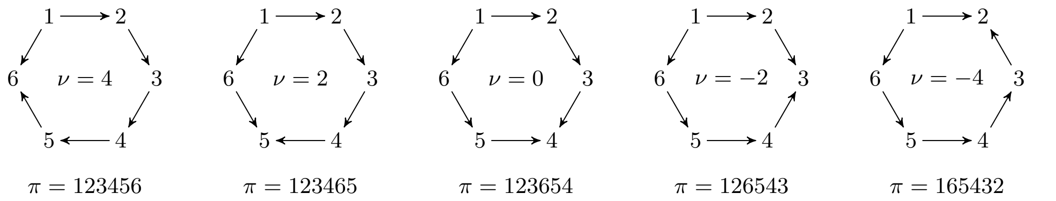

Figure 7 Representatives of the five torically non-equivalent Coxeter elements of $\mathcal{C}_6$ and their circulations ν, which fall into only three classes under the action of $\mathrm{Aut}(\mathcal{C}_6)$.

As a consequence, to understand toggling independent sets over $\mathcal{C}_n$, we only need to consider $\lfloor \frac{n}{2} \rfloor$ update orders. Specifically, we may choose those of the form

$$\pi = \quad 12 \cdots k\, n(n-1) \cdots (k+1), \quad \text{for } k = \left\lceil \frac{n}{2} \right\rceil, \ldots, n-1. \tag{2}$$

Experimental evidence has shown that, for even n, under the **identity update order** $\pi = 12 \cdots n$, the ACA map tends to have larger orbits than over other update orders. We have also observed a peculiar feature about the periodicity of the average number of 1s. For these reasons, we began our analysis using this update order, and that is what we will use for the remainder of this paper. In the next section, we will describe the methods we developed and preliminary results.

Though our focus in the remainder of this paper is on ECA rule 1 under $\pi = 12 \cdots n$, we want our techniques to be adaptable to other update orders, other togglable ECA rules, and more general toggle actions. Since ECA rule 1 only has one 1 in its truth table, the occurrences of 1s in its periodic orbits are sparse. As such, to understand the dynamics, it helps to look at the pattern of the 1s. If we have a rule that has mostly 1s, then its complement is an equivalent rule that has mostly 0s. Thus, we can always assume, without loss of generality, that we are studying a rule that has at least as many 0s as 1s in its periodic orbits. We will refer to such rules as being **sparse**.

2.3 Scrolls and ticker tapes

In this section, we will describe some tools and results for toggling independent sets of $\mathcal{C}_n$. We will view them in two formats: a bi-infinite periodic table of 0s and 1s called the *scroll*, and a bi-infinite sequence called the *ticker tape*. A function called the *successor* on the *live entries* (the set of 1s) defines equivalence classes called *snakes*. A "dual" commuting function called the *shadow* allows us to travel between snakes, and defines less visually obvious equivalence classes called *co-snakes*. These functions generate the abelian *snake group* that acts simply transitively on the set of live entries. If we quotient out by a periodic orbit, sending the infinite scroll to a finite table, then each snake (respectively, co-snake) merges into a finite *ouroboros* (respectively, *co-ouroboros*) on a torus. Topologically, this is a covering map, and it induces a homomorphism from the snake group to the *ouroboros group*, that acts simply transitively on the live entries in the finite orbit table. Studying this action algebraically allows us to deduce properties about the structure of the dynamics. Near the end of this paper, we will formalize an abstraction of this, called *commuting complementary pairs*, which can be adapted to other update orders, other ECA rules, and even arbitrary binary tables and other combinatorial actions.

Fix an initial state $x = x^{(0)} = (x_1, \ldots, x_n) \in \mathcal{I}_n$ and update order $\pi = \pi_1 \cdots \pi_n \in S_n$. Let $x^{(i)}$ be the result of iterating F_π exactly i times from x. Since F_π is bijective on $\mathcal{I}_n$, we can define this for all $i \in \mathbb{Z}$. Consider the bi-infinite table with n columns, indexed by $j = 1, \ldots, n$, and rows indexed downward by $i \in \mathbb{Z}$. The (i,j)-entry $X_{i,j}$ is the state of cell v_{π_j} in $x^{(i)}$. In other words, the i^{th} row consists of the states of the cells at time $t = i$, in the order that they are updated. This infinite table is called the **scroll** of x and π, and denoted $\mathcal{S} = (X_{i,j}) = \mathsf{Scroll}(x, \pi)$.

The global dynamics of F_π is read off the scroll as one reads from a book: left-to-right across rows, and the rows vertically downward. This defines a bi-infinite sequence called the **ticker tape**, denoted $\mathcal{X} = (X_k) = \mathsf{Tape}(x, \pi)$. We will index the entries so $X_0 = X_{0,\pi_1}$, $X_1 = X_{0,\pi_2}$, and so on, so in general, $X_r = X_{\lfloor r/n \rfloor, \pi_{1+(r \bmod n)}}$. It is elementary to translate results in the orbit table framework to ticker tapes or vice-versa, but sometimes, one format is easier to work with that the other. Formally, the **live entries**, in both formats, are the sets

$$\mathsf{Live}(\mathcal{S}) = \big\{(i,j) \in \mathbb{Z} \times \mathbb{Z}_n \mid X_{i,j} = 1\big\}, \qquad \mathsf{Live}(\mathcal{X}) = \big\{k \in \mathbb{Z} \mid X_k = 1\big\}.$$

An example of a portion of a scroll, corresponding to a single periodic orbit, is shown twice in Figure 8. In the top one, the live entries are colored by equivalence classes called *snakes*, and in the bottom one, by *co-snakes*. We will refer back to this example when we formally define these and related terms, such as the successor, shadow, and slither.

x	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}
$x^{(0)}$	0	0	0	0	1	0	1	0	0	0	0
$x^{(1)}$	1	0	1	0	0	0	0	1	0	1	0
$x^{(2)}$	0	0	0	1	0	1	0	0	0	0	1
$x^{(3)}$	0	1	0	0	0	0	1	0	1	0	0
$x^{(4)}$	0	0	1	0	1	0	0	0	0	1	0
$x^{(5)}$	1	0	0	0	0	1	0	1	0	0	0
$x^{(6)}$	0	1	0	1	0	0	0	0	1	0	1

$$\begin{array}{c} \cdots\ 0\ 0\ \cdots \\ \cdots\ 0\ 1\ 0\ \mathbf{a}\ \cdots \\ \cdots\ 0\ 0\ \overline{\mathbf{a}}\ 0\ \cdots \end{array}$$

"*successor of* (i,j)"

x	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}
$x^{(0)}$	0	0	0	0	1	0	1	0	0	0	0
$x^{(1)}$	1	0	1	0	0	0	0	1	0	1	0
$x^{(2)}$	0	0	0	1	0	1	0	0	0	0	1
$x^{(3)}$	0	1	0	0	0	0	1	0	1	0	0
$x^{(4)}$	0	0	1	0	1	0	0	0	0	1	0
$x^{(5)}$	1	0	0	0	0	1	0	1	0	0	0
$x^{(6)}$	0	1	0	1	0	0	0	0	1	0	1

$$\begin{array}{c} \cdots\ 0\ 0\ \cdots \\ \cdots\ 0\ 1\ 0\ \cdots \\ \cdots\ 0\ 0\ \cdots \\ \cdots\ \mathbf{b}\ \overline{\mathbf{b}}\ \cdots \end{array}$$

"*shadow of* (i,j)"

Figure 8 The scroll of ECA rule 1, with $\pi = 12 \cdots n$, consists of the seven rows $x^{(0)}, \ldots, x^{(6)}$, repeated indefinitely. This is shown twice to emphasize the snakes and co-snakes, separately. The top scroll highlights the two snakes, which are generated by the successor function, and have standard slither $(D2)^3$. In the bottom scroll, the colors distinguish the six co-snakes, generated by the shadow function, which have standard co-slither S^2.

2.4 Snakes in scrolls

For the remainder of this section, we will assume that our update order is $\pi = 12\cdots n$, and we will omit this from the notation and write, e.g., $\mathsf{Scroll}(x)$ and $\mathsf{Tape}(x)$. If we take an arbitrary live entry (i,j), we can draw conclusions about its surrounding entries. For example, the entries to the left and right ($X_{i,j-1}$ and $X_{i,j+1}$) are both 0, as are the entries directly above and below ($X_{i-1,j}$ and $X_{i+1,j}$). Since the scroll is just a rendering of the ticker tape, when we get to the end of a row, the next entry is the first entry of the following row. As such, we will assume that $X_{i,n+1} = X_{i+1,1}$.

It is not hard to show that for every live entry (i,j), we must have $X_{i,j+2} + X_{i+1,j+1} = 1$. We will call the coordinates of whichever of these states is live the **successor** of (i,j), denoted $s(i,j)$. Similarly, we also must have $X_{i+2,j-2} + X_{i+2,j-1} = 1$, and we call the coordinates of whichever of these is live the **shadow**, denoted $c(i,j)$. Iterating the successor function defines equivalence classes on $\mathsf{Live}(\mathcal{S})$ that we will call *snakes*, due to their visual appearance in the scroll, as shown in Figure 8. Iterating the shadow function defines equivalence classes that we will call *co-snakes*.

▶ **Definition 2.** *Given a live entry (i,j) in a scroll, the **snake** and the **co-snake** containing it are the sets*

$$\mathsf{Snake}(i,j) = \{s^k(i,j) \mid k \in \mathbb{Z}\}, \qquad \mathsf{CoSnake}(i,j) = \{c^k(i,j) \mid k \in \mathbb{Z}\}.$$

A key property about the successor and shadow functions, illustrated by Figure 9, is that they commute.

▶ **Proposition 3.** *The successor of the shadow is the shadow of the successor. That is, for any live entry (i,j), we have $s(c(i,j)) = c(s(i,j))$.*

$$\begin{array}{r}
\cdots\ 0\ 0\ \cdots\ \ \ \ \ \ \\
\cdots\ 0\ \mathbf{1}\ 0\ \mathbf{a}\ \cdots\ \ \ \\
\cdots\ 0\ 0\ 0\ \bar{\mathbf{a}}\ 0\ \cdots \\
\cdots\ 0\ \mathbf{1}\ 0\ \mathbf{a}\ 0\ \cdots\ \ \ \ \ \ \\
\cdots\ 0\ 0\ \bar{\mathbf{a}}\ 0\ \cdots\ \ \ \ \ \ \ \ \
\end{array}
\qquad\qquad
\begin{array}{r}
\cdots\ 0\ 0\ \cdots\ \ \ \\
\cdots\ 0\ \mathbf{1}\ 0\ \mathbf{a}\ \cdots \\
\cdots\ 0\ 0\ \bar{\mathbf{a}}\ 0\ \cdots \\
\cdots\ 0\ \mathbf{1}\ 0\ \mathbf{a}\ \cdots\ \ \ \\
\cdots\ 0\ 0\ \bar{\mathbf{a}}\ 0\ \cdots\ \ \
\end{array}$$

Figure 9 Given a live entry $X_{i,j} = \mathbf{1}$, the two cases shown here illustrate the property that "*the successor of the shadow is the shadow of the successor*".

We can record the "shape" of a snake by starting at any live entry (i,j), iterating the successor function, and recording a 2 or a D depending on whether the successor is (in ticker tape notation) X_{i+2} or X_{i+n+1}. This defines a periodic bi-infinite sequence, and a subsequence that generates it is called a **slither**. Two slithers are equivalent if they generate the same bi-infinite sequence. We can use exponents for ease of notation, e.g., $(2D)^3 = 2D2D2D \sim D2D2D2 = (D2)^3$. Of course, these are equivalent to $2D$ and $D2$, but as we will see, the most canonical form of a slither is not necessarily a minimal periodic subsequence. Similarly, we can record the shape of a co-snake with a bi-infinite sequence called its **co-slither**. If (i,j) is live, then its shadow is either $(i-1, j+2)$ or $(i-2, j+2)$. We will refer to the former as a *short step* and the latter as a *long step*, and denote these by S and L, respectively.

▶ **Proposition 4.** *In any scroll, all snakes have a common slither, and all co-snakes have a common co-slither.*

If we start at $(i,j) \in \mathsf{Live}(\mathcal{S})$ and iterate the successor function until we return to the same co-snake, then the corresponding sequence of 2s and Ds is called the **standard slither**. Similarly, iterating the shadow function until we return to the same snake, defines the **standard co-slither**. This allows us to prove the following.

▶ **Proposition 5.** *For any live entry (i,j) in $\mathcal{S}$,*

$$s^{\#\text{co-snakes}}\big(i,j\big) = c^{\#\text{snakes}}\big(i,j\big). \tag{3}$$

Returning to our example in Figure 8, it is easy to check that the two snakes have standard slither $(D2)^3$, the six co-snakes have standard slither S^2, and that $s^6 = c^2$.

The successor and shadow functions generate an abelian group, which we write multiplicatively to suggest function composition, and because biologically, not all snakes are adders.

▶ **Definition 6.** *Suppose $\mathcal{S}$ has α snakes and β co-snakes. The **snake group** is defined as*

$$G(\mathcal{S}) = \left\langle s, c \mid sc = cs,\ s^{\beta} = c^{\alpha} \right\rangle.$$

It is not clear *a priori* whether there are other relations between the successor and shadow functions, but the following guarantees that there are none.

▶ **Theorem 7.** *The snake group $G(\mathcal{S})$ acts simply transitively on* $\mathsf{Live}(\mathcal{S})$.

This action endows $\mathsf{Live}(\mathcal{S})$ with the structure of a Cayley diagram for $G(\mathcal{S})$. Moreover, the snakes and co-snakes are cosets of $\langle s \rangle$ and $\langle c \rangle$, respectively. The number of snakes is $[G(\mathcal{S}) : \langle c \rangle]$, the length of the standard co-slither, and the number of co-snakes is $[G(\mathcal{S}) : \langle s \rangle]$, the length of the standard slither.

2.5 Ouroboroi in orbit tables

Though the scroll is infinite, it will often be helpful to restrict our attention to a single orbit, and allow snakes and co-snakes to "wrap around" from bottom-to-top. We will refer to such a "circular snake" as an *ouroboros*, inspired by the ancient symbol of a snake swallowing its tail. Suppose x lies in a periodic orbit $x = x^{(0)}, \ldots, x^{(m-1)}, x^{(m)} = x$ of size m. The $m \times n$ table with top row $x^{(0)}$ and bottom row $x^{(m-1)}$ is the **orbit table**, denoted $\mathcal{T} = \mathsf{Table}(x, \pi)$, or $\mathsf{Table}(x)$ if $\pi = 12 \cdots n$ is understood.

Let $\mathsf{Live}(\mathcal{T}) \subseteq \mathbb{Z}_m \times \mathbb{Z}_n$ be the set of live entries in the orbit table, which is the image of the set of live entries of the scroll under the natural quotient map $p\colon \mathsf{Live}(\mathcal{S}) \to \mathsf{Live}(\mathcal{T})$ that reduces the first coordinate modulo m. The shadow and successor functions descend to bijections on $\mathsf{Live}(\mathcal{T})$ that we will call the **mod m successor function** $\bar{s}$, and **mod m shadow function** $\bar{c}$, respectively. This is illustrated by the following commutative diagram; there is an analogous one for c and $\bar{c}$.

$$\begin{tikzcd} \mathsf{Live}(\mathcal{S}) \arrow[r, "s"] \arrow[d, "p"'] & \mathsf{Live}(\mathcal{S}) \arrow[d, "p"] \\ \mathsf{Live}(\mathcal{T}) \arrow[r, "\bar{s}"] & \mathsf{Live}(\mathcal{T}) \end{tikzcd} \qquad \begin{tikzcd} (a+km, b) \arrow[r, mapsto, "s"] \arrow[d, mapsto, "p"'] & s(a+km, b) \arrow[d, mapsto, "p"] \\ (a, b) \arrow[r, mapsto, "\bar{s}"] & \bar{s}(a, b) \end{tikzcd}$$

The functions $\bar{s}$ and $\bar{c}$ generate a finite abelian group called the **ouroboros group**, denoted $G(\mathcal{T})$. Cosets of $\langle \bar{s} \rangle$ and $\langle \bar{c} \rangle$ are called *ouroboroi* and *co-ouroboroi*, respectively.

▶ **Definition 8.** *Given a live entry (i,j) in an orbit table, the* ***ouroboros*** *and* ***co-ouroboros*** *containing it are the sets*

$$\mathsf{Ouro}(i,j) = \{\bar{s}^k(i,j) \mid k \in \mathbb{Z}\}, \qquad \mathsf{CoOuro}(i,j) = \{\bar{c}^k(i,j) \mid k \in \mathbb{Z}\}.$$

The quotient map $p\colon \mathsf{Live}(\mathcal{S}) \to \mathsf{Live}(\mathcal{T})$ induces a homomorphism $p_*\colon G(\mathcal{S}) \to G(\mathcal{T})$ sending $s \mapsto \bar{s}$ and $c \mapsto \bar{c}$. The **(co-)ouroboros degree** is the number of (co-)snakes in the p-preimage of each (co-)ouroboros. We denote these as

$$\deg(p_*) := \frac{[G(\mathcal{S}) : \langle s\rangle]}{[G(\mathcal{T}) : \langle \bar{s}\rangle]}, \qquad \operatorname{codeg}(p_*) := \frac{[G(\mathcal{S}) : \langle c\rangle]}{[G(\mathcal{T}) : \langle \bar{c}\rangle]}.$$

Returning back to the example from Figure 8, the portions of the scrolls shown can also be considered as orbit tables. It is easy to check by applying the mod m shadow function to any live entry on the bottom row that both snakes merge into a single ouroboros, hence the ouroboros degree is two. Similarly, the co-ouroboros degree is three, because the six co-snakes merge into two co-ouroboroi.

▶ **Theorem 9.** *Suppose $\mathcal{S}$ has α snakes, β co-snakes, $\bar{\alpha}$ ouroboroi, $\bar{\beta}$ co-ouroboroi, and that $\tau = |\mathsf{Live}(\mathcal{T})|$. The ouroboros group has presentation*

$$G(\mathcal{T}) = \Big\langle \bar{s}, \bar{c} \;\Big|\; \overline{sc} = \overline{cs},\ \bar{s}^{\beta} = \bar{c}^{\alpha},\ \bar{s}^{\tau/\bar{\alpha}} = \bar{c}^{\tau/\bar{\beta}} = 1 \Big\rangle,$$

and it acts simply transitively on $\mathsf{Live}(\mathcal{T})$.

Recall that we can endow $\mathsf{Live}(\mathcal{S})$ with a natural Cayley diagram structure of the snake group. The simply transitive action of $G(\mathcal{T})$ on $\mathsf{Live}(\mathcal{T})$ gives it a Cayley diagram structure for the ouroboros group, and the quotient map $p\colon \mathsf{Live}(\mathcal{S}) \to \mathsf{Live}(\mathcal{T})$ is a covering map.

Another advantage of orbit tables is that we can easily define the **sum vector** as the n-tuple that records the number of live entries in each column. Early computational results suggested the curious property that the period of the sum vector, viewed as a cyclic word, is always odd.

▶ **Proposition 10.** *For update order $\pi = 12\cdots n$, the sum vector of any orbit table has odd period.*

x	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}
$x^{(0)}$	1	0	1	0	0	0	0	0	1	0	1	0
$x^{(1)}$	0	0	0	1	0	1	0	0	0	0	0	1
$x^{(2)}$	0	1	0	0	0	0	1	0	1	0	0	0
$x^{(3)}$	0	0	1	0	1	0	0	0	0	1	0	1
$x^{(4)}$	0	0	0	0	0	1	0	1	0	0	0	0
Sum:	**1**	**1**	**2**	**1**	**1**	**2**	**1**	**1**	**2**	**1**	**1**	**2**

Figure 10 An orbit of ECA rule 1 with update order $\pi = 12\cdots n$. Note that the period of the sum vector $(1,1,2,1,1,2,1,1,2,1,1,2)$ is 3, as guaranteed by Proposition 10.

The sum vector of the orbit table in Figure 8 has period 1, as each column contains exactly two live entries. See Figure 10 for an orbit table with a sum vector of period 3. The proof of Proposition 10 follows from translating into ticker tape notation, the fact that the standard slither must have an odd number of Ds. This last fact is due to the relation in Eq. (3) with the observation that the shadow function always skips exactly two rows, but loses one when it wraps around the end of the table. For example, in Figure 10, where $s^6 = c^2$, we can travel from position $(0,1)$ to $(3,10)$ via the standard slither $(D2)^3$ or the standard co-slither SL. Both S and L skip two rows, but we lose one with the S because it wraps back to $(1,12)$.

It is worth noting that Proposition 10 does not necessarily hold for other update orders. For example, if $n = 4$ with update order $\pi = 1243$, there is a size-2 orbit $\{0100, 0001\}$ whose sum vector $(0,1,0,1)$ has period 2. More generally, the methods and theory developed here do not carry over seamlessly to toggling independent sets under other update orders, but we are working on adapting them. It helps that for many problems, we only need to consider the $\lceil \frac{n}{2} \rceil$ update orders described in Eq. (2).

By knowing that the standard slither must contain an odd number of Ds, we can construct and characterize all possible scrolls and orbit tables. Without loss of generality, we can assume that the "first entry," i.e., cell 1 at $t = 0$, is live.

▶ **Theorem 11.** *For a fixed n, we can construct all scrolls having $X_{0,1} = 1$ through the following procedure:*

1. *Take a solution to the equation*

$$2a + 3b + 4c = n + 1 \quad \textit{with} \quad a, b, c \in \mathbb{Z}_{\geq 0} \textit{ and } b + c > 0. \tag{4}$$

2. *Choose any sequence of $2(b + c) - 1$ instances of D and a instances of 2. This gives the standard slither of each snake.*
3. *Choose any sequence of b instances of S, and c instances of L. This gives the standard co-slither of each co-snake.*

Furthermore, each solution descends to a unique orbit table under the quotient map p (though some of these orbit tables may arise from ticker tapes that differ only by translation).

Being able to characterize all possible scrolls for a fixed n by constructing their standard slithers and co-slithers also tells us the possible snake and ouroboros groups. In other words, it gives essential information about the feasible topological structures on the set of live entries. It should be possible to say more for special cases, such as when n is prime, even, odd, etc. This would relate fundamental topological and dynamical properties of ECA rule 1.

3 Ongoing and future work

This paper arose out of our work on toggling independent sets from combinatorics, the connections we saw to cellular automata, and how both fields complement each other. Our original goal was to prove our data-driven conjecture that the period of the sum vector is always odd under the ACA map $F_{12\cdots n}$. We expected this to lead us to explore homomesies and how the dynamics behaves under toric equivalent update orders, because these questions arose in earlier work of toggling independent sets over the path graph [10]. However, once we encountered the algebraic and topological structures that arose from the commuting successor and shadow functions, our research went in a different direction.

Many of the general ideas in this paper, such as studying tables of 0s and 1s in a CA, are obviously not new, and have been around for decades. Nevertheless, there are many novel approaches within our techniques that we think should open up new avenues for research within the fields of both combinatorics and cellular automata. We would still like to return to studying homomesy, resonance, and toric equivalence in ECA rule 1. We would also like to investigate this over other families of graphs, such as the "distance-2 cycle graph". In the literature, some of this is done under the name of "SDSs with logical NOR functions" [15].

However, to our knowledge, these have not been studied using the algebraic framework in this paper. More generally, studying homomesy and resonance in other asynchronous cellular automata beyond rule 1, and relating them to the algebraic features of toggle groups and the combinatorial features of the Coxeter elements, is mostly unexplored. This is an example of theory from dynamical algebraic combinatorics contributing to the field of cellular automata.

For another example, the idea behind the shadow and successor functions can be generalized to other ECA rules. Namely, if $X_{i,j}=1$, then we will call positions (i',j') and (i'',j'') in a scroll a **complementary pair** of (i,j) if $X_{i',j'}+X_{i'',j''}=1$. Given a live position (i,j), every complementary pair defines a function that outputs the position of the unique live entry. By Proposition 3, the successor and shadow functions are **commuting complementary pairs**. This defines the abelian "snake group," and it guarantees that all (co-)snakes have a common (co-)slither, because they arise as cosets. The natural simply transitive actions of these groups define Cayley diagram structures on the live entries of the scrolls and orbit tables, and a topological covering map relating the two. We have a paper in preparation, which also includes the proofs of many results stated here, that uses the theory of covering spaces and deck transformations to deduce properties about the periodicity within the scrolls and ticker tapes. It would be interesting to find examples of complementary pairs arising from other ECA rules, either under a fixed sweep F_π, or synchronous update. If two such pairs commute, then this opens to door to using algebraic and topological tools to analyze those dynamical systems. Even if they do not commute, it is still likely that much of the framework still carries over, but with a nonabelian group.

The complementary pairs that define the successor and shadow functions are characterized by $X_{i,j+2}+X_{i+1,j+1}=1$ and $X_{i+2,j-2}+X_{i+2,j-1}=1$. We can try to directly define other commuting complementary pairs by modifying these indices, and investigate which ones give rise to infinite binary scrolls of width n, that a so-called "generalized snake group" acts on simply transitively. When this happens, we can ask whether or not such a scroll arises from a togglable CA, SDS, or automata network (i.e., possibly over a different graph), how to characterize those that do, and whether all of the orbits can be specified in a number-theoretic manner, such as in Theorem 11.

Of course, these ideas can be extended beyond just pairs – e.g., to a set of 3 entries that must contain exactly 1 (or 2) live entries. Scrolls of ECA rule 1 with $\pi=12\cdots n$ have two commuting complementary pairs, but rules that are less sparse might have scrolls that are characterized by three or more. It seems like if this were to happen, the resulting three-generator abelian group would not act freely on the live entries, because of the two-dimensional structure of a scroll. However, this is all speculative, and deserves investigation.

The previous examples are all instances of ideas from dynamical algebraic combinatorics posing new questions in the field of cellular automata. On the other hand, ideas from CAs can lead to new problems in combinatorics. For example, it is easy to define what it means to toggle other graph-theoretic objects, such as vertex covers or dominating sets. Over the cycle graph $\mathcal{C}_n$, each of these can also be viewed as a CA, albeit a distance-2 one. Though it is more complicated, the idea of complementary pairs should be adaptable for certain toggling problems that do not involve CAs, such as order ideal or antichain toggling. This amounts to finding invariants of an arbitrary live entry at time t, in terms of a pair of entries at, e.g., time t' and time t''. Each such pair will define a bijection on the live entries and thereby generate an equivalence relation. This is just one example of how it might be feasible to adapt tools developed from analyzing tables of 0s and 1s arising from ECA rule 1 to new techniques applicable to generalized toggling problems in combinatorics.

References

1. D. Armstrong, C. Stump, and H. Thomas. A uniform bijection between nonnesting and noncrossing partitions. *Trans. Amer. Math. Soc.*, 365(8):4121–4151, 2013.
2. A. Björner and F. Brenti. *Combinatorics of Coxeter groups.* Springer Verlag, New York, 2005.
3. P. Cameron and D. Fon-Der-Flaass. Orbits of antichains revisited. *European J. Combin.*, 16(6):545–554, 1995.
4. L. David, C. Defant, M. Joseph, M. Macauley, and A. McDonough. Toggling independent sets in cycle graphs, 2021. In preparation.
5. C. Defant. Flexible toggles and symmetric invertible asynchronous elementary cellular automata. *Discrete Math.*, 341(9):2367–2379, 2018.
6. K. Dilks, O. Pechenik, and J. Striker. Resonance in orbits of plane partitions and increasing tableaux. *J. Comb. Theory Ser. A*, 148:244–274, 2017.
7. D. Einstein, M. Farber, E. Gunawan, M. Joseph, M. Macauley, J. Propp, and S. Rubinstein-Salzedo. Noncrossing partitions, toggles, and homomesies. *Electron. J. Combin.*, 23(3), 2016.
8. H. Eriksson and K. Eriksson. Conjugacy of Coxeter elements. *Electron. J. Combin.*, 16(2):#R4, 2009.
9. M. Joseph. Antichain toggling and rowmotion. *Electron. J. Combin.*, 26(1), 2019.
10. M. Joseph and T. Roby. Toggling independent sets of a path graph. *Electron J. Combin.*, 25(1):1–18, 2018.
11. G. Kreweras. Sur les partitions non croisées d'un cycle. *Discrete Math.*, 1(4):333–350, 1972.
12. M. Macauley, J. McCammond, and H. Mortveit. Dynamics groups of asynchronous cellular automata. *J. Algebraic Combin.*, 33(1):11–35, 2011.
13. M. Macauley, J. McCammond, and H.S. Mortveit. Order independence in asynchronous cellular automata. *J. Cell. Autom.*, 3(1):37–56, 2008.
14. M. Macauley and H.S. Mortveit. Cycle equivalence of graph dynamical systems. *Nonlinearity*, 22:421–436, 2009.
15. H.S. Mortveit and C.M. Reidys. Discrete, sequential dynamical systems. *Discrete Math.*, 226(1-3):281–295, 2001.
16. H.S. Mortveit and C.M. Reidys. *An Introduction to Sequential Dynamical Systems.* Universitext. Springer Verlag, 2007.
17. Y. Numata and Y. Yamanouchi. On the action of the toggle group of the Dynkin diagram of type A. *arXiv:2103.16217*, 2021.
18. O. Pretzel. On reorienting graphs by pushing down maximal vertices. *Order*, 3(2):135–153, 1986.
19. J. Propp and T. Roby. Homomesy in products of two chains. *Electron. J. Combin.*, 22(3), 2015.
20. T. Roby. Dynamical algebraic combinatorics and the homomesy phenomenon. In *Recent Trends in Combinatorics*, pages 619–652. Springer, 2016.
21. V. Salo. Universal gates with wires in a row. *arXiv:1809.08050*, 2018.
22. B. Schönfisch and A. de Roos. Synchronous and asynchronous updating in cellular automata. *BioSystems*, 51(3):123–143, 1999.
23. M. Schützenberger. Promotion des morphismes d'ensembles ordonnés. *Discrete Math.*, 2(1):73–94, 1972.
24. J. Striker. Dynamical algebraic combinatorics: promotion, rowmotion, and resonance. *Notices Amer. Math. Soc.*, 64(6), 2017.
25. J. Striker. Rowmotion and generalized toggle groups. *Discrete Math. Theor. Comput. Sci.*, 20, 2018.
26. J. Striker and N. Williams. Promotion and rowmotion. *European J. Combin.*, 33:1919–1942, 2012.
27. M. Vielhaber. Computing by temporal order: asynchronous cellular automata. In *AUTOMATA*, volume 90, pages 166–176. Electron. Proc. Theor. Comput. Sci., 2012.