The Evolution Homomorphism and Permutation Actions on Group Generated Cellular Automata

Nicole R. Miller[*]
Department of Mathematics,
Virginia Polytechnic Institute and State University,
Blacksburg, VA 24060

Michael J. Bardzell[†]
Department of Mathematics and Computer Science,
Salisbury University,
Salisbury, MD 21801

In this paper cellular automata generated over group alphabets are examined. For abelian groups and numerous local update rules, time evolution is additive and properties such as reversibility of systems can be examined using algebraic techniques. In particular, a necessary and sufficient condition for the reversibility of a finite one-dimensional cellular automata generated over a finite cyclic group using a 2-rule is provided. Finally, evolutions that respect permutations of the cellular configurations are introduced and examined.

1. Introduction

A cellular automaton A is a discrete dynamical system consisting of a lattice of cells, where each cell assumes values from some finite alphabet, and a local update rule that determines the value of each cell at the next time step. The update rule is a function defined in terms of the neighbors of the cell in question. Updates are performed in parallel so that all cells are updated simultaneously. One problem encountered when studying cellular automata (CA) is that, for a fixed alphabet, the number of possible states grows exponentially with the number of cells. This makes understanding global properties of such systems difficult. One of the goals of CA research is to understand the dynamics of large systems without having to perform the evolution computations on every possible initial state. For many systems (e.g., those capable of universal computation) shortcut prediction techniques are not possible. Even for many finite systems, it may be quite cumbersome to describe a system by direct simulation. In this paper we focus on CA where algebraic tech-

[*]Electronic mail address: nimille2@vt.edu.
[†]Electronic mail address: mjbardzell@salisbury.edu.

niques can provide predictive power and avoid the necessity of "brute force" simulation.

Additive CA have been studied extensively in the literature since these systems obey a superposition principle. That is, if a binary operation + is defined on the states, $s_1 \to w_1$ and $s_2 \to w_2$, then $s_1 + s_2 \to w_1 + w_2$. Here s_1, s_2, w_1, and w_2 are states and $\to$ denotes one step of time evolution. The advantage of an additive system is that the evolution of complex states can be derived by decomposing the states into simple "seeds," understanding how the seeds evolve, and then adding the evolved states together using the binary operation. In [1] global properties of finite additive systems are studied *via* their characteristic polynomials and updates are viewed as a dipolynomial multiplication. In this setting, algebraic techniques are used to understand various global properties, including transient and periodic behavior, of numerous systems. In [2] a systematic treatment of linear and additive automata over binary alphabets is provided, including group CA where the evolution transformation T is used to form a cyclic group. In particular, additive CA over finite fields have received much attention, in large part because of the availability of linear algebra. However, much of the analysis also can be applied to CA generated over groups, with some minor modifications.

In this paper we consider CA generated over a group alphabet. Usually, we consider abelian groups and local update rules that are additive. Our motivation is that, in addition to using algebraic techniques to analyze certain systems, these automata often inherit algebraic structure and properties of their own. When viewed as an algebraic structure, these systems can be better understood using techniques from modern algebra. This approach is taken in [3] where certain additive CA are viewed as free R-modules, where R is taken to be a commutative ring, usually finite-dimensional over a field. Evolutions that define linear transformations of the system are considered. The state transition diagram is recovered by examining properties of this map. In particular, the in-degree of the nonGarden of Eden states is shown to be the cardinality of the kernel of evolution.

Next let us fix some notation. Suppose A is a cellular automaton (finite or infinite of any dimension) generated over the group $(G, *)$. Let $a_i^t \in G$ denote the value of cell i at time t. Throughout this paper an *n-rule* will be a local update rule defined by multiplying n different neighboring cell values of cell i using group multiplication. However, these do not necessarily need to be the closest n neighbors to cell i. For instance, $a_i^{t+1} = a_{i-2}^t * a_i^t * a_{i+1}^t$ and $a_i^{t+1} = a_{i+1}^t * a_{i+2}^t * a_{i+4}^t$ are both examples of 3-rules for one-dimensional CA. $a_{i,j}^{t+1} = a_{i-1,j}^t * a_{i,j+2}^t * a_{i,j}^t * a_{i-2,j+1}^t$ is a 4-rule for a two-dimensional cellular automaton on a checkerboard grid. When G is abelian we will write the group operation as +. A *state* of the system is a configuration where each cell has a specific value. CA can be finite or infinite, depending on the number of cells in the

lattice. In the finite case, periodic boundary conditions can be used so that each cell has the same number of neighboring cells. The *null state* (or dead state) is the state where every cell value is the group identity. For example, if we are using the cyclic group $\mathbb{Z}_n$ for an alphabet, each cell value is the equivalence class of 0 for the null state. If $\mathbb{Z}_2 \times \mathbb{Z}_2$ is used, then each cell value is (0,0) for the null state. The update rules in this paper always satisfy the quiescence condition, that is, the rules always map the null state to the null state.

We intend for this paper to be accessible to researchers from a variety of disciplines, including physicists, computer scientists, and mathematicians. While we do assume some familiarity with group theory, the presentation should be at a level appropriate for researchers from numerous backgrounds.

2. The evolution homomorphism

Suppose that A is an n-dimensional cellular automaton, finite or infinite. Let T_0 denote the set of all possible states and, in general, T_i denote the set of all states present after i time steps of evolution. Then $T_0 \setminus T_1$, for example, denotes the Garden of Eden states. Time evolution defines a map $\phi : T_0 \to T_0$. Furthermore, $\phi : T_0 \to T_1$ is onto and the following holds:

$$T_0 \supseteq T_1 \supseteq \cdots \supseteq T_n \supseteq T_{n+1} \supseteq \cdots.$$

If T_0 is finite, there exists a positive integer f such that $|T_i| = |T_f|$ for all $i \geq f$. We call T_f the *set of final states*. Note that T_f consists precisely of the states that lie along oriented cycles in the state transition diagram of the automaton.

Now suppose A is constructed over a group alphabet $(G, *)$. Then T_0 is a group under cell-by-cell multiplication. That is, $T_0 \approx G \times G \times \cdots G$, where the direct product is taken over the number of cells copies of G. For the rest of this section assume the generating alphabet is an abelian group $(G, +)$ and that the update rule is additive, that is, time evolution commutes with the group operation + defined on T_0. In terms of the evolution map, this can be written compactly as $\phi(s_1 + s_2) = \phi(s_1) + \phi(s_2)$ for all $s_1, s_2 \in T_0$. In this case, since T_0 is a group, ϕ is a homomorphism. As a result, the set of states at time t forms an algebraic structure of its own, that is, the set of states at time $t + 1$ is a subgroup of the set of states at time t.

Proposition 1. $T_0 \geq T_1 \geq \cdots \geq T_n \geq T_{n+1} \geq \cdots.$

Now consider a finite cellular automaton over $(G, +)$ and the chain $T_0 \stackrel{\phi}{\to} T_1 \stackrel{\phi}{\to} \cdots \stackrel{\phi}{\to} T_{f-1} \stackrel{\phi}{\to} T_f$. The composition of these maps $\Phi = \phi^f$

defines a homomorphism from the set of all states onto the set of final states. We call this the *evolution homomorphism*. Note that $\ker\Phi$ is the subgroup of all states that eventually "die," that is, evolve to the null state over time. Also, ϕ becomes one-to-one after time $t = f$. So for the higher time steps $t \geq f$, $\phi : T_t \to T_{t+1}$ is an automorphism corresponding to permuting the states of $T_t = T_{t+1} = T_f$ around the oriented cycles of the state transition diagram. By forming the quotient group of T_0 with $\ker\Phi$ we have Proposition 2.

Proposition 2. $T_0/\ker\Phi \simeq T_f$.

In other words, if we take the group of all states and mod out by the subgroup of states that eventually die, the resulting quotient is isomorphic to the group of final states. This provides us with some useful global properties of the system. For example, if two states s_1 and s_2 lie in the same coset of the quotient, then they both evolve to the same final state in T_f after f time steps. That is, if $s_1 = k + s_2$ for some $k \in \ker\Phi$, then $\Phi(s_1) = \Phi(s_2 + k) = \Phi(s_2) + \Phi(k) = \Phi(s_2)$. Since $|\ker\Phi|$ is the number of states that eventually die and $|T_f|$ is the number of states that lie along oriented cycles, we obtain the following:

$$\frac{\text{number of states}}{\text{number of states that die}} = \text{number of states on cycles.}$$

Proposition 2 also allows us to form a "quotient automaton," which is a transition diagram constructed as follows. For each coset, draw a node in the quotient graph. Note that each coset contains exactly one final state. Suppose $s + \ker\Phi$ is such a node. Then $\Phi(s) = u \in T_f$ lies in the coset $u + \ker\Phi$ of the quotient. Draw an arrow from node $s + \ker\Phi$ to node $u + \ker\Phi$. This arrow construction is well defined since each element of the coset $s + \ker\Phi$ gets mapped to u *via* Φ. The new diagram is simply the cycle portion of the original state transition diagram. In other words, when you mod out by the states that die over time, the transient behavior is stripped away and only the long term cyclical (reversible) portion of the system remains.

A similar technique is used to simulate finite state machines. Suppose that the set of states of a finite state machine M_1 can be partitioned so that the blocks of partitioned states form the states for a second finite state machine M_2. If all states in a given block are mapped by the transition function to states that are all in the same block, then M_2 is called a *quotient machine* of M_1. Certainly the more complicated machine M_1 can simulate M_2. But M_2 can also simulate the block behavior, although not the detailed state behavior, of M_1. The same principle holds with the CA and corresponding quotient automata constructed previously. When we mod out the system by the states that eventually die, we obtain a reversible automaton that simulates the original system

(imperfectly) in the sense that the quotient describes the block behavior of the original automaton.

Example 1. Consider a two-dimensional system generated over $\mathbb{Z}_p$, where $p > 2$ is prime, using the 4-rule $a^{t+1}_{i,j} = a^t_{i,j-1} + a^t_{i-1,j} + a^t_{i+1,j} + a^t_{i,j+1}$ mod p with periodic boundary conditions. Note that with periodic boundary conditions, the grid is identified with the surface of a torus. As a small example, apply this to a 2×2 lattice. For an arbitrary initial state using $a, b, c, d \in \mathbb{Z}_p$ we have

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \longrightarrow \begin{bmatrix} 2(b+c) & 2(a+d) \\ 2(a+d) & 2(b+c) \end{bmatrix}.$$

Let

$$H = \left\{ \begin{bmatrix} \alpha & \beta \\ \beta & \alpha \end{bmatrix} : \alpha, \beta \in \mathbb{Z}_p \right\}.$$

Any state evolves to a state in H after one time step, so $T_f \subseteq T_1 \subseteq H$. To compute the kernel of the evolution map we simply solve the equations $a + d = 0$ and $b + c = 0$ over $\mathbb{Z}_p$. This system has p^2 solutions corresponding to states of the form

$$\begin{bmatrix} a & b \\ p-b & p-a \end{bmatrix}.$$

We know $|H| = p^2$ and $|T_0| = p^4$. Since $|\ker \Phi| = p^2$, we have $|T_f| = p^2$. Hence $H = T_f = T_1$.

Now we construct the transition diagram and the corresponding quotient for $p = 5$. There are 625 states. Consider the evolution of an arbitrary initial state:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \longrightarrow \begin{bmatrix} 2(b+c) & 2(a+d) \\ 2(a+d) & 2(b+c) \end{bmatrix} \longrightarrow \begin{bmatrix} 3(a+d) & 3(b+c) \\ 3(b+c) & 3(a+d) \end{bmatrix}$$
$$\longrightarrow \begin{bmatrix} 2(b+c) & 2(a+d) \\ 2(a+d) & 2(b+c) \end{bmatrix}.$$

So all cycles in this system have length at most two. The length one cycles, that is, the fixed points of the system, are the five states of the form

$$\begin{bmatrix} a & 5-a \\ 5-a & a \end{bmatrix}.$$

This leaves 20 final states which must pair off into 10 2-cycles. Hence $H = T_f$, the reversible portion of the system, is as shown in Figure 1.

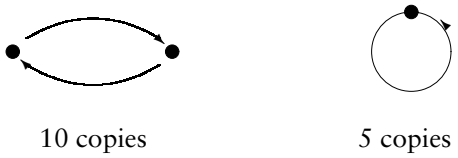

Figure 1. The final states of the 2×2 cellular automaton generated over $\mathbb{Z}_5$. The attractor portion of the system consists of 10 2-cycles and five fixed points. This figure also gives the block decomposition diagram of the original system modulo the 25 states that die.

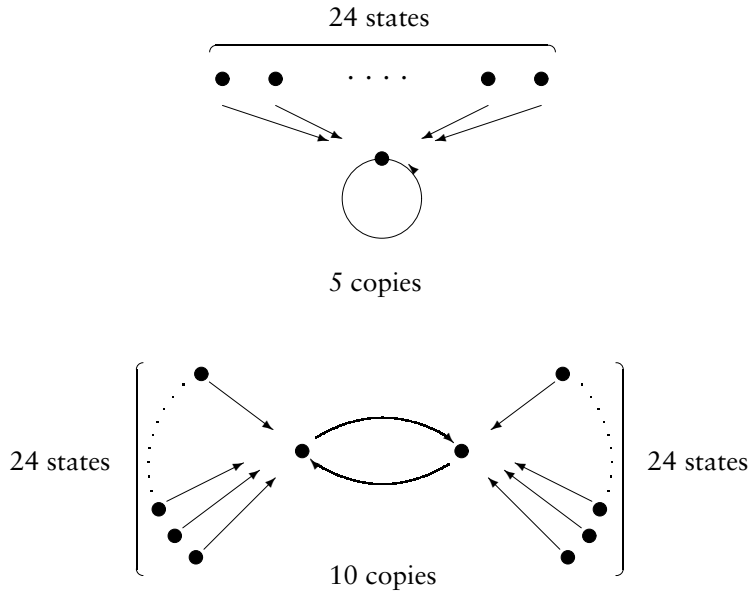

Figure 2. The transition diagram for the 2×2 cellular automaton over $\mathbb{Z}_5$. The graph has 15 connected components.

Now, we know from above that $T_0/\ker\Phi \cong H$. So the block decomposition of the transition diagram must look like the diagram in Figure 1. There is one element of H in each coset of $T_0/\ker\Phi$. Furthermore, elements in the same coset must evolve to the same element of H. Hence the state transition diagram of the original automaton is as shown in Figure 2.

The one major difference between the $p = 5$ case and the arbitrary prime $p > 2$ case is the length of the oriented cycles. Consider the

evolution of an arbitrary initial state:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \longrightarrow \begin{bmatrix} 2(b+c) & 2(a+d) \\ 2(a+d) & 2(b+c) \end{bmatrix} \longrightarrow \begin{bmatrix} 2^3(a+d) & 2^3(b+c) \\ 2^3(b+c) & 2^3(a+d) \end{bmatrix}$$
$$\longrightarrow \cdots \longrightarrow \begin{bmatrix} 2^5(b+c) & 2^5(a+d) \\ 2^5(a+d) & 2^5(b+c) \end{bmatrix}$$
$$\longrightarrow \begin{bmatrix} 2^7(a+d) & 2^7(b+c) \\ 2^7(b+c) & 2^7(a+d) \end{bmatrix} \longrightarrow$$

where each entry is reduced mod p. Since the $t = 1$ state in the sequence above lies along a cycle, eventually the sequence must return to this state. For an arbitrary initial state that does not depend on a particular choice of a, b, c, and d, this must occur at an odd time step. At the times $t = 2n+1$ for $n > 0$, the power of 2 appearing in each cell is 2^{4n+1}. If the $t = 1$ final state is not a fixed point, the cycle returns to this state at time $t = 2n + 1$, where $n > 1$ is the smallest value such that $2^{4n+1} \equiv 2 \bmod p$. This is equivalent to $2^{4n} \equiv 1 \bmod p$. We see then that the maximal cycle length is $2n$. This leads us to Proposition 3.

Proposition 3. The maximal cycle length for any initial state is bounded above by $p - 1$.

Proof. Note that a cycle of length $p - 1$ corresponds to $n = (p - 1)/2$. Furthermore, $2^{4n} = 4^{p-1} \equiv 1 \bmod p$ by Fermat's Little Theorem since $p > 2$. ■

Corollary 1. The maximal cycle length in the automaton is a divisor of $p - 1$.

Example 2. Consider the system generated over $\mathbb{Z}_4$ using a 3×3 grid and the 4-rule from Example 1. There are 2^{18} states in the system. 64 of these states die after the first time step and, by the second time step, a total of 1024 have died. After this no more states die and time evolution becomes one-to-one, that is, the system stabilizes into the reversible portion $T_2 = T_i$ for all $i \geq 2$:

$$T_0 \xrightarrow{\phi_1} T_1 \xrightarrow{\phi_2} T_2.$$

Let $\Phi = \phi_2 \circ \phi_1$. Since $|T_2| = |T_0/\ker\Phi| = 256$, there are 256 cosets in the quotient, one for each final state. Also note that $|T_1| = |T_0/\ker\phi_1| = 2^{12}$. So $|\ker\phi_2| = |T_1|/|T_2| = 16$. There are $1024 - 64 = 15(64)$ states that die after two time steps, but not one. In addition, the in-degree of each vertex at time $t = 1$ must be $|\ker\phi_1| = 64$ since $T_0/\ker\phi_1 \approx T_1$. Hence there must be $64 - 16 = 48$ Garden of Eden states that die at time $t = 1$. From this information we conclude that $\ker\Phi$ is as shown in Figure 3.

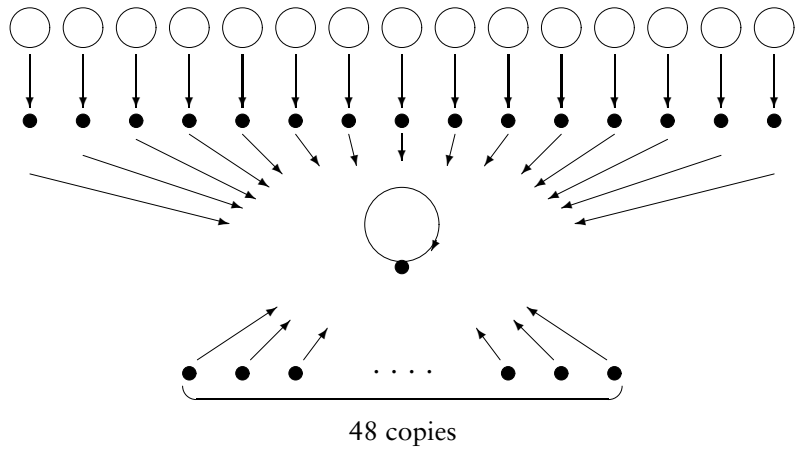

Figure 3. The states that die in the 3×3 system generated over $\mathbb{Z}_4$. The 15 circles at the top each represent a subgraph of the transition diagram containing 64 Garden of Eden states, each of which evolves to the state directly below the circle. The null state is identified as the only vertex with a loop. At the bottom of the figure are the 48 Garden of Eden states that die at time $t = 1$.

Figure 3 contains the states in the zero coset from the quotient $T_0/\ker\Phi$. Using the argument just outlined the other cosets will have exactly the same graph, except possibly for their final states. In general, final states can sometimes be linked together by oriented cycles of length greater than one. However, for this example it is easily checked that each of the 256 final states are fixed points of the system. It follows that the state transition diagram consists of 256 copies of the diagram in Figure 3. The quotient diagram (or equivalently the diagram of T_2) consists of 256 vertices, each containing a loop.

3. Reversible and irreversible systems

In this section we give necessary and sufficient conditions for a finite one-dimensional cellular automaton with periodic boundary conditions generated by a 2-rule over a finite cyclic group to be reversible. Note that the evolution map $\phi : T_0 \to T_1$ tells us when a cellular automaton of this type is reversible. If $\ker\phi$ consists of only the null state 0, then evolution is one-to-one and the automaton is reversible. In this case, all of the states of the system lie on oriented cycles. There are no Garden of Eden states nor any other kind of transient behavior.

In [4], finite one-dimensional CA with m cells over $\mathbb{Z}_n$ using the 2-rule $a_i^{t+1} = a_{i-1}^t + a_{i+1}^t \mod n$ with periodic boundary conditions, are shown to be reversible if and only if n is odd and $m \neq 4q$ for any positive integer

q. In this section we solve the more general problem of reversibility for any 2-rule. First we introduce some new terminology. The *rule distance* d of a 2-rule is the number of cells properly between the two cells being added for the local update rule. Observe that we can choose d so that $d < m$. For example, the one-dimensional automata with update rules $a_i^{t+1} = a_i^t + a_{i+1}^t$, $a_i^{t+1} = a_{i-1}^t + a_{i+1}^t$, and $a_i^{t+1} = a_{i+1}^t + a_{i+5}^t$ have rule distances 0, 1, and 3, respectively. We have the following.

Theorem 1. Let A be a finite one-dimensional cellular automaton with m cells generated over $\mathbb{Z}_n$ using periodic boundary conditions and a 2-rule with rule distance d. Then A is reversible if and only if both n and $m/\gcd(d+1, m)$ are odd.

Proof. Without loss of generality we will assume that a_i^t depends upon a_i^{t-1} and a_{i+d+1}^{t-1}, that is, $a_i^t = a_i^{t-1} + a_{i+d+1}^{t-1} \mod n$. This is not restrictive since the updates are computed in parallel and all the other 2-rules with distance d yield the same set of equations, in a permuted order, as the supposed case. The m cells of the automaton at time t are updated as follows:

$$\begin{array}{llllll} t & a_1 & a_2 & \cdots & a_{m-1} & a_m \\ t+1 & a_1 + a_{d+2} & a_2 + a_{d+3} & \cdots & a_d + a_{m-1} & a_{1+d} + a_m. \end{array} \tag{1}$$

Here $0 \le a_i \le n-1$ for $i = 1, \ldots, m$ and we have dropped the time superscript since it is clear from context. Assume that the state $(a_1, a_2, \ldots, a_m) \to (0, 0, \ldots, 0)$. This produces the following system of m equations over $\mathbb{Z}_n$:

$$\begin{array}{llllllll} a_1 & & +a_{d+2} & & & & & = 0 \\ & a_2 & & +a_{d+3} & & & & = 0 \\ & & a_3 & & +a_{d+4} & & & = 0 \\ & & \vdots & & \vdots & & & \vdots \\ & & a_{m-d-1} & & & & +a_m & = 0 \\ a_1 & & & & +a_{m-d} & & & = 0 \\ & & \vdots & & \vdots & & & \vdots \\ & & & a_{1+d} & & & +a_m & = 0. \end{array} \tag{2}$$

Each cell value a_i appears twice in the systems of equation (2). Let $\gamma = \gcd(d+1, m)$ and $\overline{u}$ denote $u \mod m$. Note that $i + (m/\gamma)(d+1) \equiv i \mod m$ for $1 \le i \le m$. For a given i, consider the variables a_i, $a_{\overline{i+(d+1)}}$, ..., $a_{\overline{i+((m/\gamma)-1)(d+1)}}$, $a_{\overline{i+(m/\gamma)(d+1)}} = a_i$. These variables are linked by one of the subsystems in equation (2). So there are γ independent subsystems, each of which contains m/γ linked variables.

$\Rightarrow$: If n is even, then the m-tuple $(n/2, \ldots, n/2)$ defines a nonnull state that evolves to the null state. If m/γ is even, then for the subsystem

containing a_i, assign cell values as follows:

$$\begin{array}{lcl} a_i & = & 1 \\ a_{\overline{i+(d+1)}} & = & n-1 \\ \vdots & \vdots & \vdots \\ a_{\overline{i+((m/\gamma)-2)(d+1)}} & = & 1 \\ a_{\overline{i+((m/\gamma)-1)(d+1)}} & = & n-1. \end{array} \tag{3}$$

Repeat this for each of the γ subsystems until a complete state α is constructed. Then α is a nonnull state that evolves to the null state.

$\Leftarrow$: Now suppose both n and m/γ are odd. Since each successive pair of variables in the a_i subsystem must add to zero, we have $a_{\overline{i+(d+1)}} = n-a_i$, $a_{\overline{i+2(d+1)}} = a_i$, and so on. Continuing in this fashion and using the fact that m/γ is odd, we arrive at $a_i = n - a_i$. Since n is odd, it follows that $a_i = 0$ for $i = 1, \ldots, m$ and $\ker \phi = \{0\}$. Hence A is reversible. ■

Corollary 2 shows that the test for reversibility is even simpler when the rule distance d is even.

Corollary 2. Let A be a one-dimensional cellular automaton with m cells generated over $\mathbb{Z}_n$ using periodic boundary conditions and a 2-rule with rule distance d. Then if d is even, A is reversible if and only if both m and n are odd.

Proof. By Theorem 1 it suffices to show that m is odd if and only if

$$\frac{m}{\gcd(d+1, m)} = \frac{m}{\gamma}$$

is odd. Clearly if m is odd, then m/γ is odd. To prove the other direction, assume that m is even. Since $d+1$ is odd, we know 2 does not divide γ. Hence $m = 2q\gamma$ for some integer $q \geq 1$. It follows that $m/\gamma = 2q$ is even. ■

For the rest of this section, let A be a one-dimensional finite cellular automaton with m cells generated over $\mathbb{Z}_2$ using a 2-rule with periodic boundary conditions. When $d = 1$ we have Rule 90, which is treated extensively in the literature. Our Theorem 2 generalizes Theorem 3.1 regarding Garden of Eden states from [1].

Theorem 2. Let A be a cellular automaton as defined previously. The fraction of the total number of states which can occur only as initial states, and cannot be reached by evolution, is $1 - (1/2^{\gamma})$, where $\gamma = \gcd(d+1, m)$.

Proof. Consider $\ker \phi$ using the system of equations given in the proof of Theorem 1. Recall that this system has γ independent subsystems, each of which contain m/γ linked variables. Now, since the update is performed using addition mod 2, all m/γ variables of the subsystem

must be 0 or all must be 1. Hence there are 2^γ solutions to the entire system and $|\ker\phi| = 2^\gamma$. Since $T_0/\ker\phi \simeq T_1$ we have $|T_1| = 2^{m-\gamma}$. It follows that the fraction of all initial states which can only occur as initial states is

$$1 - \frac{|T_1|}{|T_0|} = 1 - \frac{2^{m-\gamma}}{2^m} = 1 - \frac{1}{2^\gamma}. \blacksquare$$

Corollary 3 (see [1]) The fraction of the 2^m possible states of a size m one-dimensional cellular automaton using Rule 90 which can occur only as initial states, and cannot be reached by evolution, is 1/2 for m odd and 3/4 for m even.

Proof. For Rule 90 $d = 1$ and $\gamma = \gcd(2, m)$. If m is odd, then $\gamma = 1$. If m is even, then $\gamma = 2$. ■

Since a 2-rule with distance $d = 0$ corresponds to combining adjacent cells, we immediately have Corollary 4.

Corollary 4. If adjacent cells are added, then the fraction of possible states which cannot be reached by time evolution is 1/2.

Finally, we recall Theorem 3.2 from [1]. This theorem says that if a state in a Rule 90 cellular automaton (finite one-dimensional with periodic boundary conditions) has at least one predecessor, then it has exactly two predecessors when m is odd and exactly four when m is even. The next corollary to Theorem 2 generalizes this result.

Corollary 5. Configurations in the cellular automaton which have at least one predecessor have exactly 2^γ predecessors.

Proof. Let $s \in T_1$. Then $|\phi^{-1}(s)| = |\ker\phi| = 2^\gamma$. ■

4. Commuting with permutations

Additive CA are systems where time evolution commutes with the binary operation defined on the set of all states. The symmetry of the 4-rule used in section 2 allows time evolution to commute with rotations of the states of α radians, where α is a multiple of $\pi/2$. So if $s \to t$, then $\alpha(s) \to \alpha(t)$. In this section we introduce the notion of time evolution commuting with permutations of the generating alphabet.

Consider a cellular automaton A over a finite group alphabet whose elements are labeled $1, 2, \ldots, n$. Let S_n denote the symmetric group on n letters. Then S_n acts on T_0 as follows: given $\sigma \in S_n$ and $s \in T_0$, if $a \in \{1, 2, \ldots, n\}$ is the contents of the ith cell of s, then $\sigma(a)$ is the contents of the ith cell of $\sigma(s)$.

Definition 1. Let A be a cellular automaton over an alphabet labeled $\{1, 2, \ldots, n\}$ with time evolution $\phi : T_0 \to T_0$. We say A *respects permutations* if $\phi(\sigma(s)) = \sigma(\phi(s))$ for all $\sigma \in S_n$ and $s \in T_0$.

In other words, A respects permutations if time evolution commutes with the S_n group action for each permutation of $\{1, 2, \ldots, n\}$. In this case, if $s \in T_0$ is a fixed point of the system, then so is $\sigma(s)$ for each $\sigma \in S_n$. Furthermore, q-cycles of final states in the state transition diagram are permuted to q-cycles of final states under this group action. Note that a group alphabet is not required for Definition 1, however, it is assumed throughout this section. Now let us examine cases where systems respect, or do not respect, permutations.

Proposition 4. A cellular automaton, finite or infinite of any dimension, with entries from the group $\mathbb{Z}_3$ using any 4-rule, respects permutations.

Proof. Let $\sigma \in S_n$ and $s \in T_0$. Consider the ith cell of s and suppose that the four neighboring cells used to update the cell contain $a_1, a_2, a_3, a_4 \in \mathbb{Z}_3$. The ith cell of $\sigma(\phi(s))$ is $\sigma(a_1 + a_2 + a_3 + a_4)$ and the ith cell of $\phi(\sigma(s))$ is $\sigma(a_1) + \sigma(a_2) + \sigma(a_3) + \sigma(a_4)$. There are four cases to be checked. First, suppose $a_1 = a_2 = a_3 = a_4$. Then $\sigma(a_1 + a_2 + a_3 + a_4) = \sigma(4a_1) = \sigma(a_1) = 4\sigma(a_1) = \sigma(a_1)+\sigma(a_2)+\sigma(a_3)+\sigma(a_4)$. The case where $a_1 = a_2 = a_3$ but a_4 is different, is similar to the first case. The case where $a_1 = a_2$ and $a_3 = a_4$ but $a_2 \neq a_3$ is easy to check by hand. Finally suppose that $a_1 \neq a_2 \neq a_3$. Then a_4 is necessarily equal to one of the first three values. Without loss of generality, we will let $a_1 = a_4$. Observe that $\{a_1, a_2, a_3\} = \{\sigma(a_1), \sigma(a_2), \sigma(a_3)\} = \{0, 1, 2\}$. Hence $a_1 + a_2 + a_3 = \sigma(a_1) + \sigma(a_2) + \sigma(a_3) = 0$. It follows that $\sigma(a_1 + a_2 + a_3 + a_4) = \sigma(a_4) = \sigma(a_1) + \sigma(a_2) + \sigma(a_3) + \sigma(a_4)$, as desired. ■

Theorem 3. Any cellular automaton A, finite or infinite of any dimension, with entries from the group $\mathbb{Z}_n$ using an m-rule, where $n \geq m > 1$, does not respect permutations.

Proof. Let $\widehat{r}$ denote the state where every cell contains r, for $0 \leq r \leq n-1$. Let $\sigma = (01) \in S_n$. If $m = n$, then $\phi(\sigma(\widehat{0})) = \phi(\widehat{1}) = \widehat{m} = \widehat{0}$. So $\sigma(\phi(\widehat{0})) = \sigma(\widehat{0}) = \widehat{1} \neq \phi(\sigma(\widehat{0}))$. If $n > m$, $\phi(\sigma(\widehat{0})) = \phi(\widehat{1}) = \widehat{m}$, but $\sigma(\phi(\widehat{0})) = \sigma(\widehat{0}) = \widehat{1}$. ■

By Theorem 3 it is clear that all of the Rule 90 CA and all the other systems generated from a 2-rule over $\mathbb{Z}_2$ do not respect permutations. The reader can check that CA generated over $\mathbb{Z}_2$ using a 3-rule respect permutations. However, CA over $\mathbb{Z}_4$ using a 5-rule and over $\mathbb{Z}_5$ using a 6-rule do not respect permutations. Theorem 4 shows that respecting permutations induces a σ-invariance on the subgroup of states T_n at time t.

Theorem 4. If A respects permutations, then $\sigma(T_i) = T_i$ for all $\sigma \in S_n$ and $i \geq 0$.

Proof. The result is clear for $i = 0$. Now assume the result true for i. Let $r \in \sigma(T_{i+1})$. Then $r = \sigma(s)$ for some $s \in T_{i+1}$. Now, $u \to s$ for

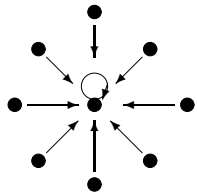

Figure 4. The states that die for a 2×2 system over $\mathbb{Z}_3$. Since there are nine of these states, it follows that there must be exactly nine final states that lie along oriented cycles.

some $u \in T_i$. By the inductive hypothesis, $\sigma(u) \in T_i$. Since A respects permutations, $\sigma(u) \to r \in T_{i+1}$ and so $\sigma(T_{i+1}) \subseteq T_{i+1}$.

If $s \in T_{i+1}$, then $\sigma^{-1}(s) \in \sigma^{-1}(T_{i+1}) \subseteq T_{i+1}$. Thus $\sigma(\sigma^{-1}(s)) = s \in \sigma(T_{i+1})$, as desired. ■

Example 3. Consider the cellular automaton A consisting of a 2×2 grid generated over $\mathbb{Z}_3$ using the 4-rule from Examples 1 and 2. There are 81 initial states. Consider the evolution of an arbitrary initial state:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \longrightarrow \begin{bmatrix} 2(b+c) & 2(a+d) \\ 2(a+d) & 2(b+c) \end{bmatrix}.$$

From Example 1 we know that $T_i = T_1$ for all $i \geq 1$ and there are nine states that evolve to the null state. Thus, there must be nine final states that lie along oriented cycles in the state transition diagram. By inspection, we see that $\begin{bmatrix} 2 & 2 \\ 1 & 1 \end{bmatrix}$ evolves to the null state. Using the symmetry of the 4-rule, we know that the rotates

$$R_1 = \left\{ \begin{bmatrix} 2 & 2 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \right\}$$

must also evolve to the null state. Likewise, $\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$ and the three corresponding rotates

$$R_2 = \left\{ \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix} \right\}$$

must also evolve to the null state. So these states, plus the null state, give the kernel of the system. The transition diagram of the kernel is shown in Figure 4.

By Proposition 4 we know A respects permutations. In fact, the action of S_3 on T_0 induces a well-defined action on $T_0/\ker\Phi$ given by

$\sigma(s + \ker\Phi) = \sigma(s) + \ker\Phi$ for each $s \in T_0$. Letting $K = \ker\Phi$ we have that

$$(01)K = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} + K \qquad (02)K = \begin{bmatrix} 0 & 2 \\ 2 & 1 \end{bmatrix} + K.$$

By applying the remaining three nonidentity permutations to K, the same three cosets are obtained. Hence the action is not faithful. However, since A respects permutations and the null state is a fixed point, so are the states it permutes to. Hence the transition diagram for each of these two new cosets is identical to the one in Figure 4.

To construct the rest of the state transition diagram, consider the orbit of $L = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + K$ under the permutation group action. Applying the 2-cycles produces the following:

$$(01)L = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} + K \quad (02)L = \begin{bmatrix} 1 & 2 \\ 2 & 2 \end{bmatrix} + K \quad (12)L = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} + K.$$

Next apply the 3-cycles:

$$(012)L = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} + K \qquad (021)L = \begin{bmatrix} 0 & 2 \\ 2 & 2 \end{bmatrix} + K.$$

It is easy to see that the orbit of L consists of the remaining states of the system. Each of the six cosets in the orbit contains exactly one final state. The cosets also separate into three pairs, where all the states in one coset evolve to the final state of the partner coset. This means that the two final states of the pair must evolve to each other. Hence the remaining six final states separate into three 2-cycles. The reader can check that the pairs are L with $(02)L$, $(12)L$ with $(012)L$, and $(01)L$ with $(021)L$. It follows that the state transition diagram for the entire system is as shown in Figure 5.

5. Conclusion

Algebraic techniques have been used in other contexts to study cellular automata (CA). For example, algebraic conditions under which two one-dimensional systems commute are presented in [5]. Block algebras are used in [6] to show that certain CA with radius r can be converted into automata with a two-site neighborhood. Efficient time prediction of CA over algebraic structures is examined in [7]. These papers consider group, quasigroup, semigroup, and similar alphabets. Another interesting algebraic approach is given in [8], where eventually periodic behavior in these systems is related to varieties of groupoids.

In this paper abelian group alphabets are used to relate the kernels of the evolution homomorphism to the reversible portion of the automata.

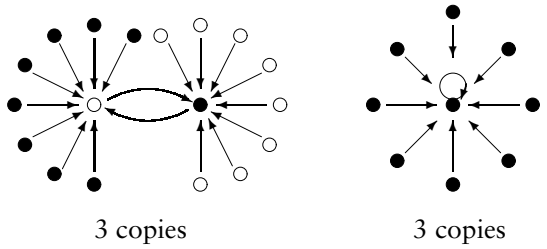

Figure 5. The state transition diagram for the 2×2 system generated over $\mathbb{Z}_3$. The left component of the diagram shows one of the coset pairs. The black vertices denote one coset and the white vertices the other. Notice how time evolution commuting with addition, rotations, and permutations is used in this example to construct the entire transition diagram from just a few states (so it is not necessary to perform the evolution computations on all initial states).

Like the approach in [3] and [9], the linearity of time evolution is used to glean a tremendous amount of information about the automata. Decomposing the systems into blocks modulo the states that die reveals a strong symmetry in many of the global properties. Although the permutation condition introduced in section 4 is a strong requirement, it produces nice predictive power when it is present.

One extension of this work is the study of entropy as defined in information theory. Consider a finite cellular automaton with m cells generated over an alphabet with n elements. Assume at time $t = 0$ that all initial states are equiprobable. For a given time step $t > 0$ we can compute the probability p_s^t of obtaining a given state s from an arbitrary initial condition simply by counting the number of occurrences of s at time t and then dividing by the number of states (including multiplicities) at time t. These probabilities are easily recovered from the state transition diagram. Let $N^t(m)$ be the number of all states generated with nonzero probability at time t. Then the *set entropy* (or *topological entropy*) at time t (see [10]) is

$$E_S^t(m) = \frac{1}{m} \log_n N^t(m). \tag{4}$$

Observe that the weight of a nonzero probability is not taken into consideration for this definition. The weight can be factored in by using a *measure entropy* (or *metric entropy*) at time t:

$$E_M^t(m) = -\frac{1}{m} \sum p_s^t \log_n p_s^t. \tag{5}$$

Here the sum is taken over all states $s \in T_0$. The following holds:

$$0 \le E_M^t(m) \le E_S^t(m) \le 1. \tag{6}$$

The middle inequality is called *saturated* if equality holds. This occurs when all the probabilities p_s^t at time t are equal. In the examples from this paper, the inequality is saturated. This follows from the additivity of the evolution map. All states at time t have the same number of arrows from time $t-1$ states coming into them. Hence the set and measure entropies coincide. Nonabelian groups tend to generate automata where the measure entropy is strictly smaller than the set entropy.

Acknowledgments

Support for this work was provided by NSF award #DUE-0087644.

References

[1] O. Martin, A Odlyzko, and S. Wolfram, "Algebraic Properties of Cellular Automata," *Communications in Mathematical Physics*, **93** (1984) 219–258.

[2] P. P. Chaudhuri, D. R. Chowdhury, S. Nandi, and S. Chattopadhyay, *Additive Cellular Automata: Theory and Applications Volume 1*, (IEEE Computer Society Press, Los Alamitos, CA, 1997).

[3] W. Chin, B. Cortzen, and J. Goldman, "Linear Cellular Automata with Boundary Conditions," *Linear Algebra and Its Applications*, **322** (2001) 193–206.

[4] S. Gymnich, "A Classification of Simple One-Dimensional Automata Generated over Cyclic Groups," Proceedings of NCUR, 2002.

[5] C. Moore and T. Boykett, "Commuting Cellular Automata," *Complex Systems*, **11** (1997) 55–64.

[6] C. Moore and A. Drisko, "Algebraic Properties of the Block Transformation on Cellular Automata," *Complex Systems*, **10**(3) (1997) 185–194.

[7] C. Moore, "Quasi-Linear Cellular Automata," *Physica D*, **103** (1997) 100–132; Proceedings of the International Workshop on Lattice Dynamics.

[8] J. Pederson, "Cellular Automata as Algebraic Systems," *Complex Systems*, **6** (1992) 237–250.

[9] L. LeBruyn and M. Van de Berhg, "Algebraic Properties of Linear Cellular Automata," *Linear Algebra and Its Applications*, **157** (1991) 217–234.

[10] S. Wolfram, "Universality and Complexity in Cellular Automata," *Physica D*, **10** (1984) 1–35.