FINITE-WIDTH ELEMENTARY CELLULAR AUTOMATA

IAN COLEMAN

Abstract. This paper is an empirical study of eight-wide elementary cellular automata motivated by Stephen Wolfram's conjecture about widespread universality in regular elementary cellular automata. Through examples, the concepts of equivalence, reversibility, and additivity in elementary cellular automata are explored. In addition, we will view finite-width cellular automata in the context of finite-size state transition diagrams and develop foundational results about the behavior of finite-width elementary cellular automata.

1. Introduction

Stephen Wolfram's *A New Kind of Science* explores elementary cellular automata and universality in simple computational systems [3]. In 1985, Wolfram conjectured that an elementary cellular automaton could be Turing complete, thus capable of universal computation. At the turn of the century, Matthew Cook published a proof confirming that a particular cellular automaton, known as "Rule 110," was universal [1]. Wolfram currently conjectures that universality in non-trivial cellular automata (and other simple systems) is likely to be extremely common. This paper, in addition to an outline of Wolfram's basic work, is an empirical study seeking to add information and insight to the exploration of elementary cellular automata.

Elementary cellular automata have become relevant given Wolfram's development of the Principle of Computational Equivalence. From Wolfram, the Principle of Computational Equivalence states that "almost all processes that are not obviously simple can be viewed as computations of equivalent sophistication [3, p. 5 , 716-717]." Wolfram's MathWorld explains further that "the principle of computational equivalence says that systems found in the natural world can perform computations up to a maximal ("universal") level of computational power, and that most systems do in fact attain this maximal level of computational power. Consequently, most systems are computationally equivalent. For example, the workings of the human brain or the evolution of weather systems can, in principle, compute the same things as a computer. Computation is therefore simply a question of translating inputs and outputs from one system to another." This principle has powerful implications in a wide range of academic study including mathematics, philosophy, religion, and physics. Much of its weight hinges on the abundance of universality in non-trivial systems. Only one particular elementary CA has been shown to be capable of universal computation ("rule 110"), but showing this property in other cellular automata would add major significance to Wolfram's work. We begin an empirical exploration of finite-width elementary CA below.

Date: May 15, 2011.

After a brief discussion of regular cellular automata, this paper will explore finite-width elementary cellular automata. Section 2 will explain elementary cellular automata (CA) in detail; specifically, we will address how they are constructed, and explain Wolfram's naming conventions used in discussing CA.

Section 3 documents Wolfram's work with general elementary CA equivalence and explores how this might be affected by restricting width. We will discuss mirrored equivalence, complementary equivalence, and the combination of both.

Section 4 addresses the relationship of CA to discrete dynamical systems. An explanation of the state space of finite-width elementary automata leads to a discussion on cycles, transients and properties of finite-width elementary CA.

Section 5 will demonstrate how state transition diagrams assists the study of finite-width elementary CA. Specifically, we explore reversibility and attempt to quantify it through the volume of transient states, "Garden of Eden States", and the number of branches in the state transition diagram.

A system capable of universal computation has the ability to emulate any other system. In the words of Matthew Cook "when we say that some system is universal...we mean that it can run any program, or, in other words, execute any algorithm [1, p. 2]." The most common example of a universal system is the digital computer. This project is focused on the exploration of much more simplistic systems that display universality. Specifically, certain cellular automata are known to be capable of universal computation.

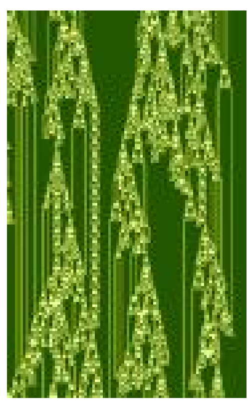

Figure 1. Simple CA Model

2. Cellular Automata

In their most general form, cellular automata are dynamical systems that consist of a regular array of cells. The array can be any finite number of dimensions, and each row of the array takes on a particular element of a finite-set known as the state space. These arrays evolve through time in accordance with some type of rule. Figure 1 is example of a CA.

Wolfram uses elementary cellular automata to emphasize the notion that simple systems following basic rules can acheive highly complex behavior–the first step in recognizing that elementary CA could be capable of universal computation. All elementary CA exhibit the following properties: a nearest neighbor scheme that has a range of 3 (the range will be discussed in Section 2.1), the use of a two color scheme (each cell is either black or white), and its evolution must be one-dimensional. To clarify, the pictorial representation of a one-dimensional CA is a two-dimensional grid, but the evolution of the CA is in only one dimension. Figure 2 shows a typical elementary CA.

Wolfram has organized a comprehensive table of elementary CA in order to analyze typical behavior. Though not rigorously defined, Wolfram categorizes all elementary CA into four classes: Class I CA demonstrate trivial behavior, Class II CA demonstrate behavior that quickly becomes stable or oscillatory, Class III CA demonstrate chaotic behavior, and Class IV CA produce behavior which may eventually become stable but contains nested structures that interact in complex and interesting ways. Along with these classifications, Wolfram has commented on the way elementary CA develop with rough facts, figures and empirical evidence– some of which will be referenced later. It is the goal of this project to highlight and comment further on typical elementary CA behavior. To do so, we further simplify elementary CA by restricting width.

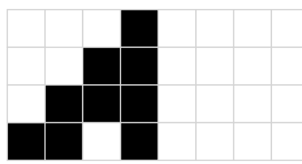

FIGURE 2. One-Dimensional CA

2.1. **Finite-Width One-Dimensional Cellular Automata.** This project restricts its focus to *finite-width* elementary cellular automata, see Figure 2. The grid of finite-width elementary CA is given by a collection of finite-length horizontal arrays, called **states**. The CA of this particular class are still restricted to the values for each cell of an array: "on" or "off", black or white, or 0 or 1.

The way in which a CA evolves is described by a **rule**. Each state is determined in a particular way by the state that precedes it. The rules that govern elementary CA use a "nearest neighbor" scheme. That is, a cell at position p of a state at time step $t + 1$ is given by the values of the cells at positions $p - 1$, p, and $p + 1$ of the state at time step t. The number of nearest neighbor cells used is called the **range**–here, the range is 3. Note that t can also be thought of as the row number of the grid. Rules take into account all possible three-cell combinations, hence there are eight nearest neighbor stencils for each rule. Figure 4 is typically how rules are displayed.

The cells at the beginning and end of each state need a third nearest neighbor. For this reason, we will impose periodic boundaries. The finite-width CA grids can then be visualized as hollow cylinders.

2.2. **Constructing a Finite-Width One-Dimensional CA.** We can now construct a finite-width elementary cellular automata. Assume the width of all finite-width CA discussed in this paper to be 8 unless otherwise stated.

(1) First, select an **initial condition**, the first row of a CA. Here, we use one black cell like so:

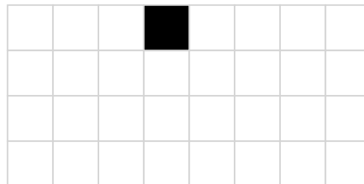

Figure 3. Elementary CA Initial Condition

(2) Next, select a rule. Consider rule 110, given by Figure 4.

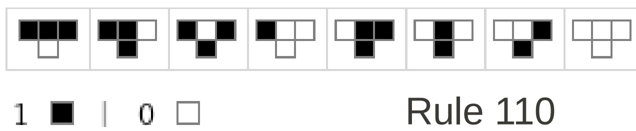

Figure 4. An Elementary CA Rule

(3) Apply the rule to the initial state to obtain the second state. Notice all but three cells will be determined by three white cells. The second state is then

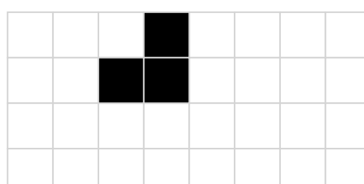

Figure 5. Two Evolutions of a CA Subject to Rule 110

(4) Next, apply the rule a second and third time to obtain

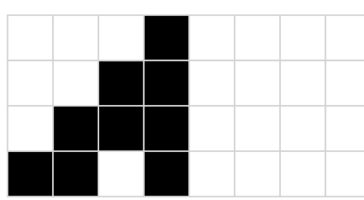

Figure 6. Four Evolutions of a CA Subject to Rule 110

Notice Figure 6 is precisely the CA displayed in Figure 2. A rule may be applied indefinitely to obtain as many evolutions of a particular elementary CA as desired. However, it will be shown later that indefinite extensions of finite-width elementary CA are unnecessary, and there is instead a natural stopping point.

2.3. **Wolfram's Numbering Scheme.** The elementary rules of elementary CA determine functions. Given any state we may apply an elementary rule to obtain the next evolution in the cellular automaton. Each stencil of an elementary rule can be thought of as a function from $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2 \longrightarrow \mathbb{Z}_2$. Perhaps more familiarly, each stencil takes a 3-bit binary number to a single bit binary number. Each of the 8 stencils determines one cell to be either black or white, thus there are a total of $2^8 = 256$ possible configurations for the bottom portion of a rule, or 256 possible elementary rules. The range of the stencil is extremely important: a CA with range r has a total of 2^{2^r} rules. Taking into account all possible functions taking 8-bit numbers to 8-bit numbers would result in over 65,000, $2^8 * 2^8$, rules!

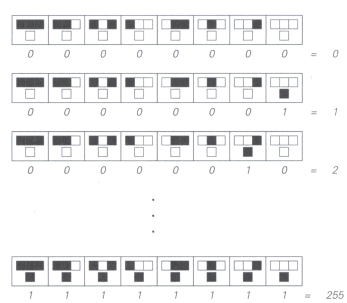

FIGURE 7. Rule Numbering Scheme [3, p. 53]

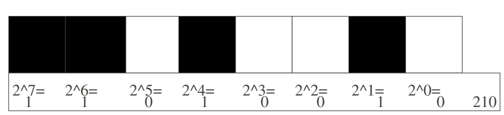

FIGURE 8. State Numbering Scheme

Elementary rules can be labeled from $R = 0 - 255$ in a natural way. Let black be equal to 1 and white be equal to 0. Now, consider the single cells of the bottom portion of the stencils together as an 8-bit binary number. Each rule derives its label from its base ten analog, see Figure 7.

States use a similar naming convention. The number of states depends on the width of the cellular automaton. In this project, the elementary CA have been chosen to be 8-wide, thus there are $2^8 = 256$ states. Though they are the same in number, this is not related to the fact that there are 256 rules. However, the same convention is used in naming. Each state corresponds to an 8-bit binary number where black is 1 and white is 0, see Figure 8.

2.4. **Computation in Elementary CA.** The Principle of Computational Equivalence hinges upon the idea "that all processes, whether they are produced by human effort or occur spontaneously in nature, can be viewed as computation (Wolfram, cite)." Universal computation is achieved through the ability to execute any algorithm, but how can natural processes be computation? It is instructive to examine how CA exhibit computation. Elementary CA can easily count multiples via their center columns: from a single black cell, rule 62 gives the multiples of 3 and rule 190 gives the multiples of four. Rule 129 shows more sophiscated computation. From a single black cell, the center column gives the powers of 2. See Figure 9

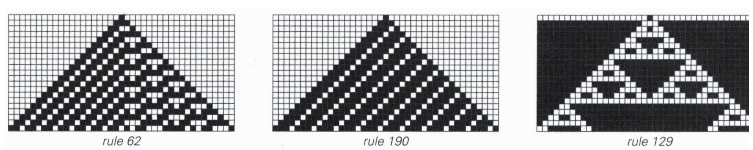

FIGURE 9. From left to right: rule 62, rule 190, rule 129 [3, p. 641]

3. Rule Equivalence

A goal of this paper is to provide empirical insights into the behavior of finite-width elementary CA. In 8-wide elementary CA there are 256 different rules with 256 different states. We can then produce 256 different CA for each rule, thus an examination of all elementary CA involves $256 \times 256 = 65536$ different sets of data. Typically, rules are examined by behavior from simple initial conditions (i.e. let the initial state be equal to 8), or from random initial conditions (i.e. let the initial state be equal to 113). Using the latter method of investigation, Wolfram was able to recognize many common themes and behaviors in elementary CA. Elementary CA are, in many cases, computationally redundant. For example, rule 0 (00000000) and rule 255 (11111111) do the same computation only in the opposite color. Below we discuss the common themes–providing examples of each class–in elementary CA and comment on the effect of imposing a finite-width on elementary CA.

Beyond Wolfram's classifications, elementary CA show three types of basic equivalence to each other: mirrored equivalence (left-right equivalence), complementary equivalence (interchanging black and white), and mirrored complementary equivalence. Wolfram has compiled a chart of equivalence shown in Figure 28 in Appendix A. The second column lists a rule's complementary equivalent, the third column is the mirrored equivalent, and the fourth mirrored complement.

3.1. **Class 1 Rules.** Wolfram characterizes Class 1 rules as those that rapidly evolve into a completely uniform state. Rule 0 is an obvious example. Regardless of the initial configuration, every cell is mapped to an empty cell. Thus, after one evolution, the CA is in a completely uniform state of white cells that will persist after more evolutions. By definition, Class 1 rules can only develop structures with straight-line movement. Clearly, periodic boundaries will only affect structures that move across that boundary, a lateral movement. Thus, the equivalance of Class 1 rules–from random initial conditions–will not be affected by width restrictions.

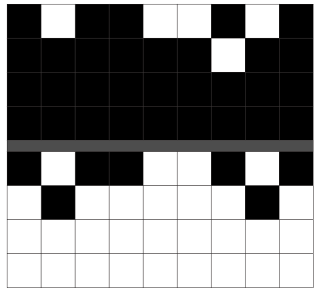

FIGURE 10. Top: rule 250, Bottom: rule 160

In Figure 10, note the maintenance of complementary equivalence for rule 250 and rule 160 (see Table 28), both Class 1 rules. From random initial conditions, rule 250 evolves quickly to a uniform grid of black cells and rule 160 evolves quickly to a uniform grid of white cells. Empirically and intuitively, we note that Class 1 rules will maintain all indefinite-width equivalence patterns: complementary, mirrored, and mirrored complementary.

3.2. **Class 2 Rules.** Class 2 rules are those that rapidly evolve into repetitive or stable states. Rule 4 is an example: every cell with a black cell directly above it remains black. From random initial conditions this produces CA with black and white streaks. Such CA are their own mirrored equivalent, as are many Class 2 rules. In fact, the repetitive and simplistic behavior that defines Class 2 rules can produce CA that are their own complements as well. Rule 170, rule 240, rule 15, rule 85 are examples, refer to Figure 28. Notice also that there are eight elementary rules that are their own mirrored, complementary, and mirrored complementary equivalents. Six of these rules are Class 2 rules from random initial conditions (rule 105 and rule 150 are Class 3 rules). In the case where a rule is equivalent in some way to itself, equivalence is trivially maintained.

In general, Class 2 rules that are not their own equivalents do not maintain equivalence. Consider the following simple example using finite-width CA: rule 170, the shift left rule, and its mirrored equivalent rule 240, the shift right rule are not left-right symmetric, see Figure 11 and Figure 12. The fact that the figures are top-bottom symmetric (as they also are for indefinite-width CA) does give a sense of the effects of periodicity. In particular, periodic boundaries allow for interference at the boundaries. Thus, Wolfram's classifications may not be relied on.

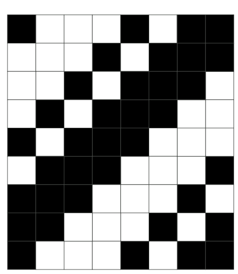

FIGURE 11. A Full Cycle of rule 170, The Shift Left rule

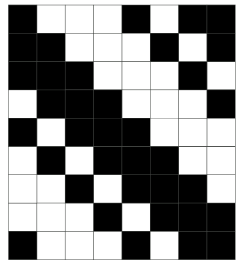

FIGURE 12. A Full Cycle of rule 240, The Shift Right rule

3.3. **Class 3 and Class 4 Rules.** The classifications above make up a majority of elementary CA. Wolfram approximates that only 14% of elementary CA rules produce more complex behavior than what is described above. Furthermore, 24 elementary CA rules produce nested patterns. Of those 24, there are only 3 fundamentally different forms that emerge. To complete the classification, Wolfram asserts that only 10 of the 256 elementary rules produces CA that are "in many respects random." This group of elementary CA with more complex behavior make up Wolfram's Class 3 and Class 4 rules. Specifically, Class 3 rules are those that are essentially random, see rule 30 below in Figure 14. Note that within the pattern there are nested triangles, but of varied size and at no regular intervals. Class 4 rules form areas of repetitive or stable states, but also form from structures that interact with each other in complicated ways, rule 110 is an example, see Figure 13.

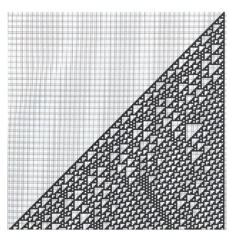

FIGURE 13. Evolutions of rule 110 from simple initial conditions (Wolfram, pg. 32)

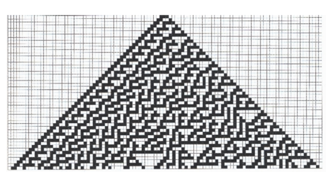

FIGURE 14. Evolutions of rule 30 from simple initial conditions (Wolfram, pg. 27)

In finite-width form, it appears that, in general, class 3 and class 4 rules do not maintain equivalence–complementary or mirrored. From the table in Figure 28, rule 30 has three distinct rules for which it is equivalent to, none of which are itself. This is typical of Class 3 and Class 4 rules. In Figure 15 and Figure 16, we examine rule 30 and rule 86 for mirrored equivalence. Visually we can be sure that neither mirrored nor complementary equivalence is maintained. In Figure 15, there are eleven evolutions and the final step is identical to the first. This is an example of a **cycle (or attractor)** in a cellular automaton. Rule 86 in Figure 16 shows eleven evolutions, but does not repeat any states. Cycles will be discussed in greater length in the following section, but strictly by noting the number of time steps it takes to repeat a state we can conclude that equivalence is not maintained.

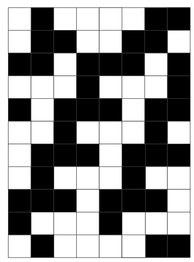

FIGURE 15. Evolutions of rule 30 from random initial conditions

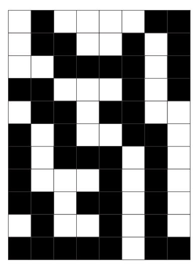

FIGURE 16. Evolutions of rule 86 from random initial conditions

4. Discrete Dynamical Systems

Although Matthew Cook has shown rule 110 to be universal, using this system as a model for other systems (natural or otherwise) is not immediately useful. Elementary CA are non-linear dynamical systems, thus they can be extremely difficult to analyze. Essentially, CA that are not obviously simple–CA without trivial or highly regular behavior–are not computationally reducible. According to Wolfram's MathWorld, "computations that cannot be sped up by means of any shortcut are called computationally irreducible." The answer to computationally irreducible problems comes only from actually performing the computation. CA are often computationally irreducible problems (Class 3 and Class 4 rules for example). However, we can use the notion (and notation) of dynamical systems in an attempt to counter computational irreducibility in finite-width elementary CA.

A dynamical system provides a fixed rule that describes the time dependence of a point in a geometrical space. At any given time a dynamical system has a state (an element of a set of real numbers, vectors, or perhaps a collection of black or white cells) which can be represented by a point in an appropriate **state space**.

For example, Let $\mathbb{C}$ denote the state space. Then $\pi \in \mathbb{C}$ is a state. Now suppose that this particular dynamical system (using state space $\mathbb{C}$), has a time dependence described by the rule $f(x) = e^{ix}$, where x is any state. Using $x_0 = \pi$ as the initial point (time $t = 0$), we note that after one evolution in time $x_1 = f(x_0) = e^{i\pi} = -1$. Thus, -1 is the state at time $t = 1$.

4.1. **State Transition Sets.** In this project, we use the notation of a *discrete* dynamical system to indicate discrete steps in time. A discrete dynamical system presents some rule, or function, f and describes the evolution of a variable x by the formula $x_{n+1} = f(x_n)$ where n is some integer (as in the example above). We will adopt this notation for this project: a rule R acts upon a state s_n such that $s_{n+1} = R(s_n)$. Using the CA defined in Section 2.2 and subject to rule 110 (R), note that $R(8) = 24$, $R(24) = 56$, and so on.

We will call this *ordered* collection of elements from the state space–an initial state, and the resulting states that follow after repeated applications of a rule–a **State Transition Set (STS)**. Let $\mathcal{E}_R$ be a STS which is obtained using rule R from above. Given initial condition $s_0 = 16$, we can write the resulting CA as a STS being careful to preserve order,

$$\mathcal{E}_{110}(s_0) = \{16, 48, 112, 208, 241, 19, 55, 125, 199, 76, 220, 245, 31, 49...\}$$

$\mathcal{E}_{110}(s_0)$ is an ordered set representation of Figure 17.

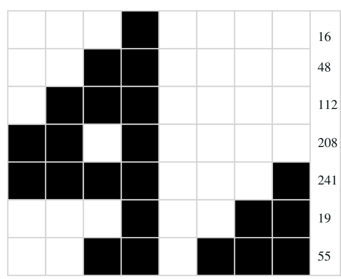

FIGURE 17. Pictorial Representation of $\mathcal{E}(s_0)$ where in the right column we have listed the integer values of the 8-bit binary number represented by the row.

4.2. **Systems of Cellular Automata.** We can assign any of the 256 possible state configurations to s_0 and obtain 256 different CA. Thus, to encompass all information about a particular elementary rule we must consider all initial conditions. A **system of cellular automata** is given by the collection of ordered sets,
$\mathbb{G}(R) = \{\mathcal{E}_R(s) : s \in S\}$, where S is the state space. $\mathbb{G}$ is a collection of all STS for a fixed rule R. A system actually contains less information than it at first appears; CA in the same system often produce STS's with duplicate information. For example, if R is rule 110 consider the following STS

$$\mathcal{E}_{110}(14) = \{14, 26, 62, 98, 230, 175, 248, 137, 155, ...\}$$

and

$$\mathcal{E}_{110}(81) = \{81, 243, 22, 62, 98, 230, 175, 248, 137, 155, ...\}.$$

At four evolutions, $\mathcal{E}_{110}(81)$ performs the same computations as $\mathcal{E}_{110}(14)$ does at three evolutions.

Let R be a fixed elementary rule. Given a state s_0 and its resulting STS, $\mathcal{E}_R(s_0)$, we can find all initial conditions t that produce STS, $\mathcal{E}_R(t)$, that share some state with $\mathcal{E}_R(s_0)$. We will call this set of initial conditions a **lobe set**,
$\mathcal{I}(s_0) = \{t \in S \mid \mathcal{E}_R(s_0) \cap \mathcal{E}_R(t) \neq \emptyset\}$, where S is the state space, and s_0 is an initial conditions.

5. Finite-Size State Transition Diagrams

To organize this information, each STS, $\mathcal{E}_R$, will be interpreted as a directed graph by assigning each element of the ordered set as a node. We then combine STS with duplicate computation into single graphs that we will call lobes. If $\mathcal{E}_R$ is a graph, a lobe in its system is given by

$$\mathcal{L}(s_0) = \bigcup_{t \in \mathcal{I}(s_0)} \mathcal{E}_R(t),$$

where $\mathcal{I}(s_0)$ is the lobe set of s_0 and we use the graph union operation [2]. $\mathcal{L}(s_0)$ represents a union of directed graphs, whereas $\mathcal{I}(s_0)$ is instead a collection of states in no particular order. The collection of all lobes in a system,

$$\mathbb{L} = \bigcup_{s_0 \in S} \mathcal{L}(s_0),$$

is called a finite-size state transition diagram and displays all the computations in a system of CA, see Figure 18. Visually, the finite-size state transition diagram suggests that the lobes of a system partition the state space. This is, in fact, true for all systems of finite-size elementary CA.

Theorem 1. *For any rule R, the lobes of a state transition diagram partition the state space.*

Proof. Let $\mathbb{L}$ be any finite-size state trasition diagram of a system of CA. The lobes of the finite-size state transition diagram will partition the state space if every state is in the system, and every state belongs to exactly one lobe. By definition, every state s is in $\mathbb{L}$. Thus, to show that the lobes of a finite-size state transition diagram partition the state space, we need only to show that states cannot be members of multiple lobes.

Let s, t, and r be states. Suppose $s \in \mathcal{L}(r)$ and $s \in \mathcal{L}(t)$, where $\mathcal{L}(r) \neq \mathcal{L}(t)$. This implies that $\mathcal{I}(r) \neq \mathcal{I}(t)$ and $\mathcal{E}_R(r) \cap \mathcal{E}_R(t) = \emptyset$. But, by definition, $\mathcal{L}(t) = \bigcup_{v \in \mathcal{I}(t)} \mathcal{E}_R(v)$ and since $s \in \mathcal{L}(t)$, $s \in \mathcal{I}(t)$. By the same logic, $s \in \mathcal{I}(r)$. By the definition of the lobe set, then $s \in \mathcal{E}_R(t)$ and $s \in \mathcal{E}_R(r)$. It follows that $\mathcal{E}_R(t) \cap \mathcal{E}_R(r) \neq \emptyset$, a contradiction. Thus, $\mathcal{L}(r) = \mathcal{L}(t)$, and a state cannot exist in distinct lobes.

We conclude that $\mathbb{L} = \bigcup_{s_0 \in S} \mathcal{L}(s_0)$ includes every state in the state space and that no two lobes can share a state in S. □

The finite-size state transition diagram of a system of CA is generally composed of trees that lead to cycles. States that appear as nodes in a tree of a directed graph are known as **transients** and states that appear as nodes in the cycles of a graph are known as **attractors**. The set of all attractors is called the attractor set. The finite-size state transition diagram suggests that, at some point, all finite-width elementary CA will end in an attractor.

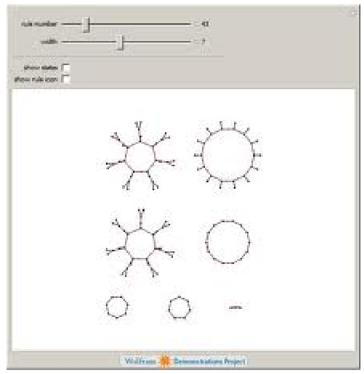

FIGURE 18. Finite-Size State Transition Diagrams in Wolfram's Mathematica

Theorem 2. *All finite-size elementary cellular automata contain an attractor in their evolution.*

Proof. If a state is given by s_i, where i denotes the step in time, then we can write any CA explicitly as the following STS: $\mathcal{E}_R(s) = \{s_0, s_1, s_2, ..., s_n, ...\}$. If $s_i = s_j$ for some nonnegative integers i and j, then the CA contains an attractor. Because the state space is finite, and $R : S \longrightarrow S$, there must be a point at which $s_i = s_j$. □

5.1. **Attractors.** We can formalize the idea of an attractor by considering STS in a given lobe of a system. In particular, the attractor of a lobe can be determined by the intersection of all STS with initial conditions in the same lobe set. Let $\mathcal{I}(q_0) = \{s_0, t_0, ..., q_0\}$ be the lobe set for the lobe $\mathcal{L}(s_0)$. Consider the intersection

$$\bigcap_{r \in \mathcal{I}(q_0)} \mathcal{E}_R(r).$$

This set will be the attractor of the lobe determined by $\mathcal{I}(q_0)$.

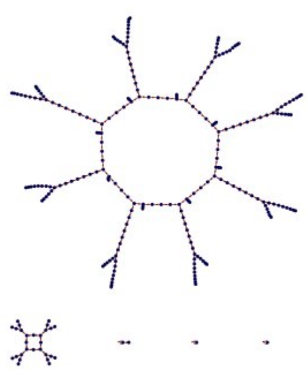

FIGURE 19. Eight-Wide State Transition Diagram for Rule 30

Attractors, in general, do not approach the maximum conceivable length–255 states. For 8-wide elementary CA, the largest attractor of any rule occurs on rule

30 and has size 40. Many initial conditions under rule 30 produce the maximal attractor length of 40. Rule 30 produces CA with an average attractor length of 35.8906; see Appendix A Figure 2 for average attractor lengths of all rules. Attractor length is, in general, a good indication of the Wolfram class in which a rule lies. Those rules with average attractor lengths of 1 are in general class 1 rules. Short attractor lengths indicate CA that rapidly evolve to uniform states. Furthermore, class 3 rules–CA which exhibit behavior that is essentially random–have much higher average attractor lengths. Rule 30 is an example, see Figure 19.

5.2. **Transient States and Reversibility.** The transient states of CA are the states that precede the first cyclic state. The finite-size state transition diagrams for elementary rules are predominately characterized by disjoint directed graphs with trees that evolve into cycles (a reflection of Theorem 2 and Theorem 1). However, there are special cases in which rules produce finite-size state transition diagrams without trees, see Figure 20. These elementary CA are reversibile. In particular, an elementary CA exhibits reversibility when every state has exactly one predecessor, and we can determine that predecessor when provided information about its rule.

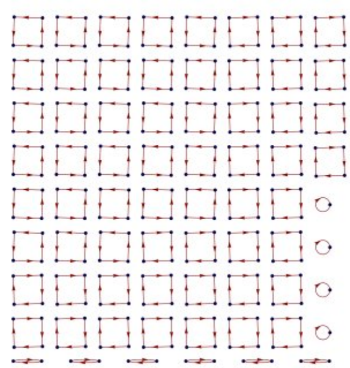

FIGURE 20. Eight-Wide State Transition Diagram for Rule 105

Reversibility is an extremely important topic in CA modeling. Wolfram states that "all current evidence suggests that the underlying laws of physics have [a kind of] reversibility." An effective model not only attempts to predict what will happen in the future, but can tell us something about the past.

In the simple world of elementary CA, Wolfram asserts the existence of six reversible rules: rule 15, rule 51, rule 85, rule 170, rule 204, and rule 240. Each of these exhibit uniform class 2 behavior (see Figure 22, the reader will notice the shift-left (rule 170), shift-right (rule 240), identity (rule 204), and complement (rule 51) rules present in this list. Each of these rules exhibit a dependence on their determining neighborhood. For example, consider $\mathbb{G}(170)$, the system of CA on rule 170. If the cell in position p at time $t+1$ is black, then the cell in position $p+1$ at time t must be black as well. $\mathbb{G}(51)$ is equally simple: the predecessor of any state is equal to its complement.

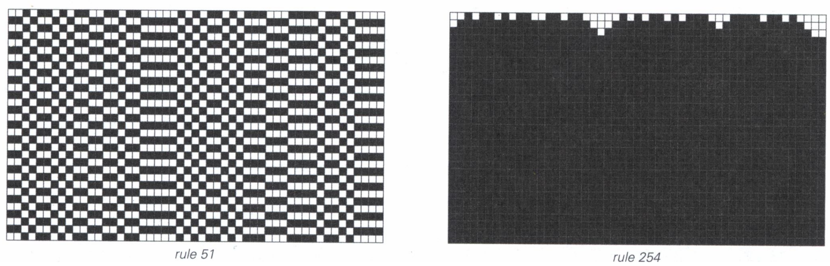

FIGURE 21. Left: Rule 51, a reversible rule. Right: Rule 254 an irreversible rule, note that it evolves to a uniformly black state, and we cannot know the predecessors.

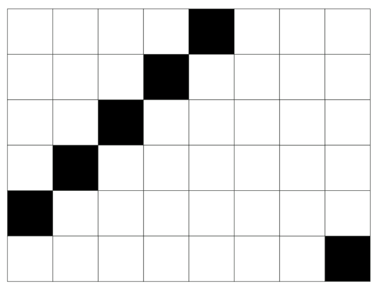

FIGURE 22. Rule 170: The reversible "Shift Left" Rule

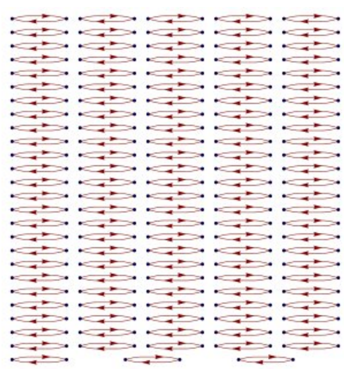

FIGURE 23. Eight-Wide State Transition Diagram for Rule 51

When finite-size elementary CA are considered, the six rules above are, predictably, still reversible. For example, rule 51 produces a finite-size state transition diagram that consists of 128 lobes; each lobe contains no transients and contains attractors of length 2 (See Figure 23). However, the addition of periodic boundaries allows for two more reversible rules, namely rule 105 and rule 150.

5.2.1. *Additivity.* The list of reversible rules (15, 51, 85, 105, 150, 170, 204, and 240) all share the property of additivity. Additive rules are those produced by the addition modulo 2 of part or all of their determining neighborhood. Consider rule

150 shown in figure 24. Note that if we add all three states of each neighborhood modulo two, the result is the bottom portion of the stencil ($1 + 1 + 1 = 1$ mod 2; $1 + 1 + 0 = 0$ mod 2; $1 + 0 + 1 = 0$ mod 2; and so on). Some additive rules add only 2 cells of the determing neighborhood, and others only 1 cell. The list of reversible rules contains only those additive rules which add 1 and 3 cells of their determining neighborhoods. The original list of six reversible rules includes only those which add a single cell (the shift-right, shift left, identity, complement, shift-right complement, and shift-left complement). When we restrict width, however, we gain the ability to reverse the two additive rules which add all three cells in the determining neighborhood, rule 150 and its complement rule 105.

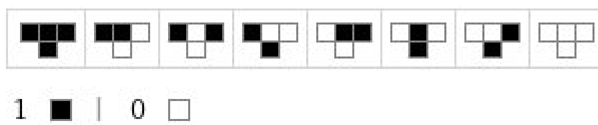

Figure 24. Rule 150: An Additive Rule

5.3. **Quantifying Reversibility.** While determining whether a given CA is reversible is useful, the vast majority of finite-width elementary CA are not. In the Appendix, Table 1 displays the average number of transient states for CA in a given system. This number can be interpreted in two useful ways:

(1) First, the number of transient states provide a measure of how many possible predecessors a state in an attractor has. In modeling, this measurement can be used to construct probability charts describing possible evolutions of particular system. CA with lower average times to attractor indicate a stronger confidence in what has occured previously to the observed state.
(2) Second, the average time to attractor provides a good measure of complexity in finite-width automata. All systems that take over six time steps to enter an attractor are class 3 rules.

Additionally, as we would expect, all reversible rules have zero time steps. Previously we discussed the reptition of computation in elementary CA. In fact, the table in Figure 28 displays Wolfram's observed equivalence explicitly. Note that in most cases, the compliment of a rule does not yield the CA to which it is a complimentary equivalent. For example, CA produced by rule 89 has an observed complimentary equivalence to CA produced by rule 101; however, rule 89 (01011001) is complementary to rule 166 (10100110) in form. This notion is supported by the fact that the average time to attractor follows no such pattern. However, there is a symmetry that may be observed through "Garden of Eden" states.

5.4. **"Garden of Eden" States.** Another way we might quantify reversibility is through **Garden of Eden** states. A state is called a "Garden of Eden" when it has exactly zero predecessors under a given rule. These states are the root nodes in each lobe of a system. The fewer Garden of Eden states a particular system has, the more likely we are able to deduce the predecessor of a given state. Table 3 displays the number of Garden of Eden states for each elementary rule in an eight-wide system. As expected, each reversible rule has zero Garden of Eden states. The more interesting result, however, is the symmetry between rule compliments.

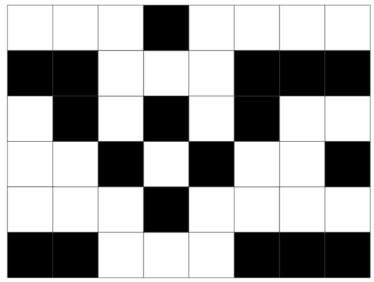

Figure 25. Rule 105: A Reversible Eight-Wide Elementary CA

Rule 0 has the same number of Garden of Eden states as rule 255, as do rule 1 and 254, rule 2 and 253, rule 3 and 252, and so on. Thus, although rules and their compliments have extremely different structure (See rule 1 and rule 254 in Figures 26 and 27), they have an equal number of unreachable states.

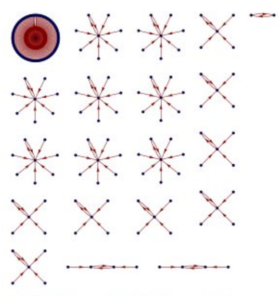

Figure 26. Rule 105: A Reversible Eight-Wide Elementary Rule

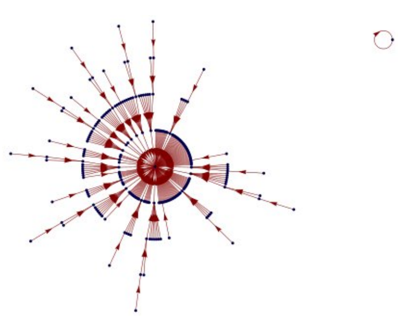

Figure 27. Rule 105: A Reversible Eight-Wide Elementary Rule

5.5. **Open Questions.** The empirical nature of this project often highlighted other areas of interest in elementary CA. In the search for widespread universality in these simple systems, exploration of computation in class 3 and class 4 rules would undoubtedly be productive. In particular, a study of specific computation in rule 30 and additive rule 90 would help build intution about the possibility of their conjectured universality.

The implementation of periodic boundary conditions made much of Wolfram's analysis of indefinite-width elementary CA inapplicable. If we were to model finite-width elementary CA as 3-dimensional objects (hollow cylinders), can we find other duplicate computation or equivalence in these systems?

For modeling purposes, reversibility remains a key topic. Might it be possible to build an algorithm that tests the reversibility of a given state under a given rule? Moreover, can we generalize theory on finite-width elementary CA to make statements about complexity, reversibility, and computation in N-wide elementary CA?

Appendix A. Tables

Table 1. Average Time Steps (k) until Attractor for each 8-wide Elementary Rule

Rule	k	Rule	k	Rule	k	Rule	k	Rule	k	Rule	k	Rule	k	Rule	k
0	0.992188	**32**	1.84375	**64**	1.80469	**96**	2.625	**128**	1.63281	**160**	2.17188	**192**	2.75781	**224**	2.73438
1	0.820312	**33**	1.34375	**65**	2.63281	**97**	2.45312	**129**	2.44141	**161**	2.14453	**193**	2.81641	**225**	0.863281
2	0.914062	**34**	0.8125	**66**	1.30469	**98**	1.45312	**130**	1.38281	**162**	1.125	**194**	2.39844	**226**	1.14062
3	0.648438	**35**	0.984375	**67**	3.11719	**99**	3.67188	**131**	1.59766	**163**	2.05078	**195**	7.00391	**227**	1.45703
4	0.8125	**36**	1.39062	**68**	0.8125	**100**	1.875	**132**	1.125	**164**	2.05469	**196**	1.76562	**228**	2.86719
5	0.609375	**37**	2.63281	**69**	2.51562	**101**	6.36719	**133**	2.61328	**165**	3.66797	**197**	2.78516	**229**	1.77734
6	1.20312	**38**	1.125	**70**	2.46875	**102**	7	**134**	1.54688	**166**	0.382812	**198**	2.23438	**230**	2.39844
7	2.375	**39**	1.05469	**71**	0.484375	**103**	3.11719	**135**	6.20703	**167**	2.27734	**199**	2.47266	**231**	1.30859
8	1.80469	**40**	2.625	**72**	1.39844	**104**	2.85938	**136**	2.75781	**168**	2.73438	**200**	0.648438	**232**	1.125
9	2.63281	**41**	2.45312	**73**	0.851562	**105**	0	**137**	2.81641	**169**	0.863281	**201**	1.01953	**233**	2.86328
10	0.804688	**42**	0.484375	**74**	1.77344	**106**	0.859375	**138**	0.648438	**170**	0	**202**	2.86719	**234**	2.73438
11	1.99219	**43**	1.40625	**75**	6.36719	**107**	2.45312	**139**	1.30078	**171**	0.488281	**203**	1.87891	**235**	2.62891
12	0.8125	**44**	1.875	**76**	0.484375	**108**	1.01562	**140**	1.76562	**172**	2.86719	**204**	0	**236**	0.648438
13	2.51562	**45**	6.36719	**77**	1.125	**109**	0.851562	**141**	2.78516	**173**	1.77734	**205**	0.488281	**237**	1.40234
14	1.85938	**46**	1.29688	**78**	2.78125	**110**	2.8125	**142**	1.40625	**174**	0.648438	**206**	1.76562	**238**	2.75781
15	0	**47**	1.99219	**79**	2.51562	**111**	2.63281	**143**	1.86328	**175**	0.808594	**207**	0.816406	**239**	1.80859
16	0.914062	**48**	0.8125	**80**	0.804688	**112**	0.484375	**144**	1.38281	**176**	1.125	**208**	0.648438	**240**	0
17	0.648438	**49**	0.984375	**81**	1.99219	**113**	1.40625	**145**	1.59766	**177**	2.05078	**209**	1.30078	**241**	0.488281
18	1.89844	**50**	1.57812	**82**	2.27344	**114**	2.04688	**146**	2.83594	**178**	1.125	**210**	0.382812	**242**	1.125
19	1.02344	**51**	0	**83**	1.05469	**115**	0.984375	**147**	1.51953	**179**	1.58203	**211**	1.12891	**243**	0.816406
20	1.20312	**52**	1.125	**84**	1.85938	**116**	1.29688	**148**	1.54688	**180**	0.382812	**212**	1.40625	**244**	0.648438
21	2.375	**53**	1.05469	**85**	0	**117**	1.99219	**149**	6.20703	**181**	2.27734	**213**	1.86328	**245**	0.808594
22	3.04688	**54**	1.51562	**86**	6.20312	**118**	1.59375	**150**	0	**182**	2.83594	**214**	1.54688	**246**	1.38281
23	1.125	**55**	1.02344	**87**	2.375	**119**	0.648438	**151**	3.05078	**183**	1.90234	**215**	1.20703	**247**	0.917969
24	1.30469	**56**	1.45312	**88**	1.77344	**120**	0.859375	**152**	2.39844	**184**	1.14062	**216**	2.86719	**248**	2.73438
25	3.11719	**57**	3.67188	**89**	6.36719	**121**	2.45312	**153**	7.00391	**185**	1.45703	**217**	1.87891	**249**	2.62891
26	2.27344	**58**	2.04688	**90**	3.66406	**122**	2.14062	**154**	0.382812	**186**	1.125	**218**	2.05469	**250**	2.17188
27	1.05469	**59**	0.984375	**91**	2.63281	**123**	1.34375	**155**	1.12891	**187**	0.816406	**219**	1.39453	**251**	1.84766
28	2.46875	**60**	7	**92**	2.78125	**124**	2.8125	**156**	2.23438	**188**	2.39844	**220**	1.76562	**252**	2.75781
29	0.484375	**61**	3.11719	**93**	2.51562	**125**	2.63281	**157**	2.47266	**189**	1.30859	**221**	0.816406	**253**	1.80859
30	6.20312	**62**	1.59375	**94**	2.60938	**126**	2.4375	**158**	1.54688	**190**	1.38281	**222**	1.125	**254**	1.63281
31	2.375	**63**	0.648438	**95**	0.609375	**127**	0.820312	**159**	1.20703	**191**	0.917969	**223**	0.816406	**255**	0.996094

TABLE 2. Average Attractor Length (r) for Each 8-wide Elementary Rule

Rule	r	Rule	r	Rule	r	Rule	r	Rule	r	Rule	r	Rule	r	Rule	r
0	1	**32**	1.00781	**64**	1	**96**	1.22656	**128**	1	**160**	1.11719	**192**	1	**224**	2.14844
1	2	**33**	2	**65**	9.875	**97**	4.8125	**129**	3.21875	**161**	4.28906	**193**	13.9219	**225**	11.0859
2	6.4375	**34**	7.33594	**66**	7.15625	**98**	6.02344	**130**	6.73438	**162**	6.99219	**194**	6.60938	**226**	6.17969
3	13.125	**35**	13.4062	**67**	15.3125	**99**	3.125	**131**	7.20312	**163**	7.99219	**195**	1	**227**	6.02344
4	1	**36**	1	**68**	1	**100**	1	**132**	1	**164**	1	**196**	1	**228**	2.14062
5	1.91406	**37**	2	**69**	1.00781	**101**	10.8125	**133**	1.125	**165**	1	**197**	1	**229**	8.98438
6	6.34375	**38**	7.57812	**70**	1.46875	**102**	1	**134**	6.57812	**166**	7	**198**	1.46875	**230**	6.60938
7	8.55469	**39**	15.1562	**71**	1.94531	**103**	15.3125	**135**	35.8906	**167**	5.875	**199**	1.46875	**231**	7.15625
8	1	**40**	1.22656	**72**	1	**104**	1.02344	**136**	1	**168**	2.14844	**200**	1	**232**	1.00781
9	9.875	**41**	4.8125	**73**	3.14062	**105**	3.85938	**137**	13.9219	**169**	11.0859	**201**	1.39062	**233**	1.02344
10	7.70312	**42**	7.58594	**74**	8.98438	**106**	11.0859	**138**	7.57812	**170**	7.71094	**202**	2.14062	**234**	2.14844
11	5.84375	**43**	5.71875	**75**	10.8125	**107**	4.8125	**139**	7.32812	**171**	7.58594	**203**	1	**235**	1.22656
12	1	**44**	1	**76**	1	**108**	1.39062	**140**	1	**172**	2.14062	**204**	1	**236**	1
13	1.00781	**45**	10.8125	**77**	1.00781	**109**	3.14062	**141**	1	**173**	8.98438	**205**	1	**237**	1
14	5.70312	**46**	7.32812	**78**	1	**110**	13.9219	**142**	5.70312	**174**	7.57812	**206**	1	**238**	1
15	7.71094	**47**	5.84375	**79**	1.00781	**111**	9.875	**143**	5.70312	**175**	7.70312	**207**	1	**239**	1
16	6.4375	**48**	7.33594	**80**	7.70312	**112**	7.58594	**144**	6.73438	**176**	6.99219	**208**	7.57812	**240**	7.71094
17	13.125	**49**	13.4062	**81**	5.84375	**113**	5.71875	**145**	7.20312	**177**	7.99219	**209**	7.32812	**241**	7.58594
18	3.35938	**50**	1.99219	**82**	5.875	**114**	7.99219	**146**	3.20312	**178**	1.99219	**210**	7	**242**	6.99219
19	2	**51**	2	**83**	15.1562	**115**	13.4062	**147**	4.85938	**179**	1.99219	**211**	7.57812	**243**	7.33594
20	6.34375	**52**	7.57812	**84**	5.70312	**116**	7.32812	**148**	6.57812	**180**	7	**212**	5.70312	**244**	7.57812
21	8.55469	**53**	15.1562	**85**	7.71094	**117**	5.84375	**149**	35.8906	**181**	5.875	**213**	5.70312	**245**	7.70312
22	2.57812	**54**	4.85938	**86**	35.8906	**118**	7.20312	**150**	3.85938	**182**	3.20312	**214**	6.57812	**246**	6.73438
23	1.99219	**55**	2	**87**	8.55469	**119**	13.125	**151**	2.57812	**183**	3.35938	**215**	6.34375	**247**	6.4375
24	7.15625	**56**	6.02344	**88**	8.98438	**120**	11.0859	**152**	6.60938	**184**	6.17969	**216**	2.14062	**248**	2.14844
25	15.3125	**57**	3.125	**89**	10.8125	**121**	4.8125	**153**	1	**185**	6.02344	**217**	1	**249**	1.22656
26	5.875	**58**	7.99219	**90**	1	**122**	4.28906	**154**	7	**186**	6.99219	**218**	1	**250**	1.11719
27	15.1562	**59**	13.4062	**91**	2	**123**	2	**155**	7.57812	**187**	7.33594	**219**	1	**251**	1.00781
28	1.46875	**60**	1	**92**	1	**124**	13.9219	**156**	1.46875	**188**	6.60938	**220**	1	**252**	1
29	1.94531	**61**	15.3125	**93**	1.00781	**125**	9.875	**157**	1.46875	**189**	7.15625	**221**	1	**253**	1
30	35.8906	**62**	7.20312	**94**	1.125	**126**	3.21875	**158**	6.57812	**190**	6.73438	**222**	1	**254**	1
31	8.55469	**63**	13.125	**95**	1.91406	**127**	2	**159**	6.34375	**191**	6.4375	**223**	1	**255**	1

Table 3. Garden of Eden States (GOE) for each 8-wide Elementary Rule

Rule	GOE	Rule	GOE	Rule	GOE	Rule	GOE	Rule	GOE	Rule	GOE	Rule	GOE	Rule	GOE
0	255	**32**	209	**64**	235	**96**	157	**128**	210	**160**	156	**192**	166	**224**	108
1	210	**33**	152	**65**	162	**97**	108	**129**	190	**161**	124	**193**	102	**225**	33
2	235	**34**	209	**66**	209	**98**	125	**130**	162	**162**	152	**194**	100	**226**	124
3	166	**35**	108	**67**	100	**99**	88	**131**	102	**163**	109	**195**	128	**227**	125
4	209	**36**	190	**68**	209	**100**	126	**132**	152	**164**	136	**196**	108	**228**	126
5	156	**37**	136	**69**	152	**101**	16	**133**	124	**165**	192	**197**	109	**229**	109
6	157	**38**	126	**70**	125	**102**	128	**134**	108	**166**	16	**198**	88	**230**	100
7	108	**39**	126	**71**	124	**103**	100	**135**	33	**167**	109	**199**	125	**231**	209
8	235	**40**	157	**72**	189	**104**	89	**136**	166	**168**	108	**200**	166	**232**	160
9	162	**41**	108	**73**	116	**105**	0	**137**	102	**169**	33	**201**	106	**233**	89
10	207	**42**	125	**74**	109	**106**	33	**138**	166	**170**	0	**202**	126	**234**	108
11	166	**43**	120	**75**	16	**107**	108	**139**	210	**171**	125	**203**	126	**235**	157
12	209	**44**	126	**76**	125	**108**	106	**140**	108	**172**	126	**204**	0	**236**	166
13	152	**45**	16	**77**	160	**109**	116	**141**	109	**173**	109	**205**	125	**237**	189
14	125	**46**	210	**78**	109	**110**	102	**142**	120	**174**	166	**206**	108	**238**	166
15	0	**47**	166	**79**	152	**111**	162	**143**	125	**175**	207	**207**	209	**239**	235
16	235	**48**	209	**80**	207	**112**	125	**144**	162	**176**	152	**208**	166	**240**	0
17	166	**49**	108	**81**	166	**113**	120	**145**	102	**177**	109	**209**	210	**241**	125
18	189	**50**	125	**82**	109	**114**	109	**146**	116	**178**	160	**210**	16	**242**	152
19	166	**51**	0	**83**	126	**115**	108	**147**	106	**179**	125	**211**	126	**243**	209
20	157	**52**	126	**84**	125	**116**	210	**148**	108	**180**	16	**212**	120	**244**	166
21	108	**53**	126	**85**	0	**117**	166	**149**	33	**181**	109	**213**	125	**245**	207
22	89	**54**	106	**86**	33	**118**	102	**150**	0	**182**	116	**214**	108	**246**	162
23	160	**55**	166	**87**	108	**119**	166	**151**	89	**183**	189	**215**	157	**247**	235
24	209	**56**	125	**88**	109	**120**	33	**152**	100	**184**	124	**216**	126	**248**	108
25	100	**57**	88	**89**	16	**121**	108	**153**	128	**185**	125	**217**	126	**249**	157
26	109	**58**	109	**90**	192	**122**	124	**154**	16	**186**	152	**218**	136	**250**	156
27	126	**59**	108	**91**	136	**123**	152	**155**	126	**187**	209	**219**	190	**251**	209
28	125	**60**	128	**92**	109	**124**	102	**156**	88	**188**	100	**220**	108	**252**	166
29	124	**61**	100	**93**	152	**125**	162	**157**	125	**189**	209	**221**	209	**253**	235
30	33	**62**	102	**94**	124	**126**	190	**158**	108	**190**	162	**222**	152	**254**	210
31	108	**63**	166	**95**	156	**127**	210	**159**	157	**191**	235	**223**	209	**255**	255

0	255	0	255	32	251	32	251	64	253	8	239	96	249	40	235	128	254	128	254	160	250	160	250	192	252	136	238	224	248	168	234
1	127	1	127	33	123	33	123	65	125	9	111	97	121	41	107	129	126	129	126	161	122	161	122	193	124	137	110	225	120	169	106
2	191	16	247	34	187	48	243	66	189	24	231	98	185	56	227	130	190	144	246	162	186	176	242	194	188	152	230	226	184	184	226
3	63	17	119	35	59	49	115	67	61	25	103	99	57	57	99	131	62	145	118	163	58	177	114	195	60	153	102	227	56	185	98
4	223	4	223	36	219	36	219	68	221	12	207	100	217	44	203	132	222	132	222	164	218	164	218	196	220	140	206	228	216	172	202
5	95	5	95	37	91	37	91	69	93	13	79	101	89	45	75	133	94	133	94	165	90	165	90	197	92	141	78	229	88	173	74
6	159	20	215	38	155	52	211	70	157	28	199	102	153	60	195	134	158	148	214	166	154	180	210	198	156	156	198	230	152	188	194
7	31	21	87	39	27	53	83	71	29	29	71	103	25	61	67	135	30	149	86	167	26	181	82	199	28	157	70	231	24	189	66
8	239	64	253	40	235	96	249	72	237	72	237	104	233	104	233	136	238	192	252	168	234	224	248	200	236	200	236	232	232	232	232
9	111	65	125	41	107	97	121	73	109	73	109	105	105	105	105	137	110	193	124	169	106	225	120	201	108	201	108	233	104	233	104
10	175	80	245	42	171	112	241	74	173	88	229	106	169	120	225	138	174	208	244	170	170	240	240	202	172	216	228	234	168	248	224
11	47	81	117	43	43	113	113	75	45	89	101	107	41	121	97	139	46	209	116	171	42	241	112	203	44	217	100	235	40	249	96
12	207	68	221	44	203	100	217	76	205	76	205	108	201	108	201	140	206	196	220	172	202	228	216	204	204	204	204	236	200	236	200
13	79	69	93	45	75	101	89	77	77	77	77	109	73	109	73	141	78	197	92	173	74	229	88	205	76	205	76	237	72	237	72
14	143	84	213	46	139	116	209	78	141	92	197	110	137	124	193	142	142	212	212	174	138	244	208	206	140	220	196	238	136	252	192
15	15	85	85	47	11	117	81	79	13	93	69	111	9	125	65	143	14	213	84	175	10	245	80	207	12	221	68	239	8	253	64
16	247	2	191	48	243	34	187	80	245	10	175	112	241	42	171	144	246	130	190	176	242	162	186	208	244	138	174	240	240	170	170
17	119	3	63	49	115	35	59	81	117	11	47	113	113	43	43	145	118	131	62	177	114	163	58	209	116	139	46	241	112	171	42
18	183	18	183	50	179	50	179	82	181	26	167	114	177	58	163	146	182	146	182	178	178	178	178	210	180	154	166	242	176	186	162
19	55	19	55	51	51	51	51	83	53	27	39	115	49	59	35	147	54	147	54	179	50	179	50	211	52	155	38	243	48	187	34
20	215	6	159	52	211	38	155	84	213	14	143	116	209	46	139	148	214	134	158	180	210	166	154	212	212	142	142	244	208	174	138
21	87	7	31	53	83	39	27	85	85	15	15	117	81	47	11	149	86	135	30	181	82	167	26	213	84	143	14	245	80	175	10
22	151	22	151	54	147	54	147	86	149	30	135	118	145	62	131	150	150	150	150	182	146	182	146	214	148	158	134	246	144	190	130
23	23	23	23	55	19	55	19	87	21	31	7	119	17	63	3	151	22	151	22	183	18	183	18	215	20	159	6	247	16	191	2
24	231	66	189	56	227	98	185	88	229	74	173	120	225	106	169	152	230	194	188	184	226	226	184	216	228	202	172	248	224	234	168
25	103	67	61	57	99	99	57	89	101	75	45	121	97	107	41	153	102	195	60	185	98	227	56	217	100	203	44	249	96	235	40
26	167	82	181	58	163	114	177	90	165	90	165	122	161	122	161	154	166	210	180	186	162	242	176	218	164	218	164	250	160	250	160
27	39	83	53	59	35	115	49	91	37	91	37	123	33	123	33	155	38	211	52	187	34	243	48	219	36	219	36	251	32	251	32
28	199	70	157	60	195	102	153	92	197	78	141	124	193	110	137	156	198	198	156	188	194	230	152	220	196	206	140	252	192	238	136
29	71	71	29	61	67	103	25	93	69	79	13	125	65	111	9	157	70	199	28	189	66	231	24	221	68	207	12	253	64	239	8
30	135	86	149	62	131	118	145	94	133	94	133	126	129	126	129	158	134	214	148	190	130	246	144	222	132	222	132	254	128	254	128
31	7	87	21	63	3	119	17	95	5	95	5	127	1	127	1	159	6	215	20	191	2	247	16	223	4	223	4	255	0	255	0

FIGURE 28. Wolfram's Table of rule Equivalence

REFERENCES

Cook, Matthew, *A Concrete View of Rule 110 Computation*, CSP, 2008, pp. 31-55.
F. Harary, *Graph Theory*, Addison-Wesley, Reading, MA, 1969.
Wolfram, Stephen, *A New Kind of Science*, Wolfram Media, Inc., Champaign, IL, 2002.