HAL open science

Operator Representation and Class Transitions in Elementary Cellular Automata

Muhamet Ibrahimi, Arda Güçlü, Naide Jahangirov, Mecit Yaman, Oguz Gülseren, Seymur Jahangirov

▶ **To cite this version:**

Muhamet Ibrahimi, Arda Güçlü, Naide Jahangirov, Mecit Yaman, Oguz Gülseren, et al.. Operator Representation and Class Transitions in Elementary Cellular Automata. Complex Systems , 2022, 31 (4), pp.415-432. 10.25088/ComplexSystems.31.4.415 . hal-04092449

HAL Id: hal-04092449

https://hal.science/hal-04092449v1

Submitted on 9 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Operator Representation and Class Transitions in Elementary Cellular Automata

Muhamet Ibrahimi
Aix-Marseille Université, Université de Toulon, CNRS, CPT (UMR 7332)
Turing Centre for Living Systems, Marseille, France
Arda Güçlü
Department of Electrical and Computer Engineering
University of Illinois Urbana-Champaign, Champaign, IL, USA
Naide Jahangirov
UNAM-Institute of Materials Science and Nanotechnology
Bilkent University, 06800, Ankara, Turkey
Mecit Yaman
Department of Aeronautical Engineering
University of Turkish Aeronautical Association, 06790, Ankara, Turkey
Oguz Gülseren
Department of Physics
UNAM-Institute of Materials Science and Nanotechnology
Bilkent University, Ankara 06800, Turkey
Seymur Jahangirov
UNAM-Institute of Materials Science and Nanotechnology
Interdisciplinary Graduate Program in Neuroscience
Bilkent University, Ankara 06800, Turkey
Corresponding author: seymur@unam.bilkent.edu.tr

We exploit the mirror and complementary symmetries of elementary cellular automata (ECAs) to rewrite their rules in terms of logical operators. The operator representation based on these fundamental symmetries enables us to construct a periodic table of ECAs that maps all unique rules in clusters of similar asymptotic behavior. We also expand the elementary cellular automaton (ECA) dynamics by introducing a parameter that scales the pace with which operators iterate the system. While tuning this parameter continuously, further emergent behavior in ECAs is unveiled as several rules undergo multiple phase transitions between periodic, chaotic and complex (class 4) behavior. This extension provides an environment for studying class transitions and complex behavior in ECAs. Moreover, the emergence of class 4 structures can potentially enlarge the capacity of many ECA rules for universal computation.

Keywords: elementary cellular automata; classes of cellular automata; deterministic transition; logistic map; Cantor set

https://doi.org/10.25088/ComplexSystems.31.4.415

1. Introduction

Elementary cellular automata (ECAs) are computational schemes defined by Wolfram [1] to suggest a broad classification and understanding of dynamical systems. These lattice models operate in discrete domains and are known to generate large-scale behavior only by involving local interactions (rules) between their units [1–4]. Wolfram's apprehension is that the asymptotic behavior of any dynamical system lies in one of four possible classes of behavior (named classes 1 through 4, respectively): homogeneous, periodic, aperiodic and complex [5]. As an archetype, ECAs are a set of 256 different rules with a dynamical variety that comprises all these classes. Among many interesting features that simple models like ECAs can possess, emergence and computability are the most striking ones [6]. Emergence, the semantic gap between behavior and interactions, is the hallmark of complex systems, and Turing computability is a fundamental property in information theoretical systems. Indeed, in ECAs, emergence is typically observed in the form of self-organized patterns, which develop either aperiodically (class 3) or in the form of propagating complex structures (class 4) [5, 6]. The latter type of behavior is also necessary for a cellular automaton (CA) rule to perform universal computation [7, 8].

Remarkable efforts have been made to acquire a general understanding of how several rules generate similar types of asymptotic behavior. Many studies have focused on bridging the observed spatiotemporal patterns in cellular automata (CAs) (phenotype) with their rule space (genotype) [9–17]. Langton proposed a way to map various rules to a single parameter and showed that as the values of this parameter range from zero to one, the resulting behavior tends to follow classes 1, 2, 4 and 3 [13]. This approach suggests that complex behavior lies between periodic and aperiodic classes, although there are many exceptions. Additional approaches have introduced entropies [14, 15], mean-field descriptions [16] or network analyses [17] to capture the relation between patterns and their respective rules. In ECAs, there are 15 unique rules in class 3 and only two unique rules that display class 4 behavior, yet an exact description of the difference between these two particular classes remains a big open challenge. In this paper we try to shed a light on the following related questions: Why do certain rules show similar asymptotic dynamics? Is there more emergent behavior hidden in the symmetries of ECAs? Is it possible to probe class transitions in ECA rules?

Based on an intuition gained by visual inspection of ECA dynamics, we suggest a fundamental approach that helps us understand and expand the dynamics of these systems. We redefine the transition function in a different notation, in the form of logical operators that are

"distilled" from the global dynamics and the symmetries of the system. This approach provides an alternative framework that links the similarities (and differences) observed at the phenotype level to the new representation of the rule space. Moreover, this framework enables us to implement a generalized logistic extension to ECAs [18] where a single parameter scales the pace with which operators update the cellular units. As a result, the binary state space of ECAs transforms to a Cantor set, where in turn we observe additional emergent behavior and transitions between classes [6]. There are also cases where the behavioral difference between some rules sharing similar genetic code is diminished upon the logistic extension.

Furthermore, we reveal several instances of complex (class 4) behavior not present in standard ECA rules, yet hidden in these models. We point out that emergence of complex behavior follows a common theme: as the parameter is tuned, spatiotemporally periodic regions start growing within the chaotic ones to the point where chaotic regions are so constrained that they "shrink" into complex interacting propagators. The emergence of class 4 behavior is important for two main reasons: to shed a light on the mechanisms of class transitions and to reveal further hidden symmetries in ECA models. The emergence of propagators also opens up new potential opportunities that might enlarge the capacity of ECA models for universal computation.

2. Operator Representation

ECAs are time-dependent, one-dimensional, infinite strings of sites $S^t = \{s_n^t\}\big|_{n=-\infty}^{\infty}$ of a binary state space $s_i \in \{0, 1\}$. In ECAs, a rule defines how the value of a certain site is iterated $s_i^{t+1} = f_S s_i^t$ based on its current value and the values of its nearest neighbors, through a transition function $f_S(s_{i-1}^t, s_i^t, s_{i+1}^t)$. Given the binary state space, there are eight possible configurations of a three-site neighborhood, resulting in $2^8 = 256$ possible mappings, that is, rules. Mapping of rule 30 is shown as an example in Figure 1(a). The name "30" of this rule comes from the binary to decimal transformation of the string 00011110 obtained from the particular mapping of the eight configurations listed in the order shown in Figure 1(a).

ECAs possess two important symmetries: complementary and mirror [19]. While mirror means flipping the string, complementary means replacing zeros with ones and vice versa. If rule A and rule B are complementary (mirror) symmetric, running rule A with a certain initial string will give the complementary (mirror) image of running

rule B with the complementary (mirror) version of that string. When these symmetries are taken into account, the number of unique ECA rules reduces to 88. This reduction is explained later by using the operator representation that we introduce in the following paragraphs.

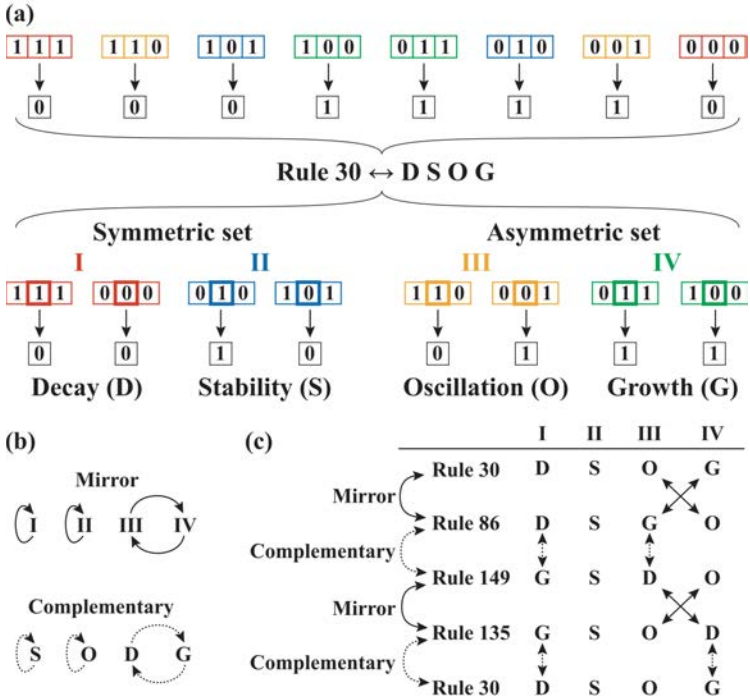

Figure 1. (a) Representation of rule 30 in terms of operators. (b) Transformations needed to switch between mirror and complementary symmetries of a rule. (c) Switching between rule 30 and its symmetries using operator representation.

Exploiting the mirror and complementary symmetries, we suggest a two-step grouping of the nearest-neighbor configurations, $(s_{i-1}^t, s_i^t, s_{i+1}^t)$. The first step separates eight possible configurations by looking at their mirror symmetry. Four of them are mirror invariant (red and blue configurations in Figure 1(a)) and the remaining four (orange and green) are not mirror invariant. We separate these groups into a symmetric and an asymmetric set, respectively. As a second step, within each set we pair the configurations based on their complementary symmetries. For instance, the complementary configuration of 111 is 000 in the symmetric set, and these two make up pair I. Following this step, we end up with four pairs, as shown in the lower part of Figure 1(a).

Simple observations on ECA runs reflect visual structures of uniform, stable, oscillatory or irregular patterns. These structures are prone to a mixture of three types of fundamental iterations $\left[s_i^t \to s_i^{t+1}\right]$, namely: decay $[0\,|\,1 \to 0]$, stability $[0(1) \to 0(1)]$ and growth $[0\,|\,1 \to 1]$. On the other hand, each pair in our grouping has configurations with central sites of values 0 and 1 that can be updated in four possible ways: $[0 \to 0, 1 \to 0]$, $[0 \to 0, 1 \to 1]$, $[0 \to 1, 1 \to 0]$ and $[0 \to 1, 1 \to 1]$. These updates form a complete set of logical operations that can describe any ECA rule. Thus, we define their corresponding operators and conveniently name them as decay (D), stability (S), oscillation (O) and growth (G), respectively. Note that the oscillation operator is a compound of decay and growth iterations. Four possible pairs and four possible operators cover all 256 rules ($4^4 = 2^8$). As an example, the operator representation of rule 30 becomes DSOG (see Figure 1(a)). A full table of operator representation for all ECA rules is given in Figure 6 of Supplemental Material. This table could be used to easily determine the operator representation from the rule number or the other way around.

Symmetric and complementary counterparts of any rule can be easily reached by using the operator representation. As shown in Figure 1(b), to get the mirror symmetry of a rule, we need to switch the operators in pairs III and IV. To get the complement of a rule, we need to replace all D operations (if any) with G and vice versa. For a certain rule, we can obtain all its equivalent rules by carrying out the transformation scheme in Figure 1(c). Note that rules with identical operators acting on pairs III and IV have symmetric invariance, and rules devoid of D and G operators have complementary invariance. This leads to a reduced number of unique rules.

If there is no invariance between a rule and any of its symmetric counterparts, that rule is part of a group of four equivalent rules. An example of this is rule 30 (DSOG), as shown in Figure 1(c). If a rule, for instance, rule 105 (SSOO), is invariant under mirror and complementary transformation, that rule is unique. Using this argument, we can acquire the numbers of 88 unique rules by writing down a combinatorial scheme that counts rules with all possible types of invariance. Note that any transformation based on the mirror and complementary symmetries of ECAs can show the existence of 88 unique rules. An equivalent demonstration is provided in [19].

These combinations are constructed based on Figure 1(b). Let us consider a rule R, its mirror and complementary counterparts M and C, respectively. Out of 256 possible rules, the rules for which $R \equiv M \equiv C$ are $2\times2\times2\times1 = 8$. This because in pairs I, II, III, only operators S and O are allowed, and the pair IV has the same operator with pair III. We can also compute the number of unique rules with $R \equiv M \not\equiv C$, which is 28; with $R \equiv C \not\equiv M$, which is 4; and with

$R \not\equiv M \equiv C$, which is also 4. Adding up the number of unique rules with $R \not\equiv M \not\equiv C$, which is 44, makes a total of 88.

3. Periodic Table of Elementary Cellular Automata

Using our representation, symmetric (I and II) and asymmetric (III and IV) sets of operators are decoupled from each other with respect to both mirror and complementary transformations. Hence, it can be useful to arrange the ECAs in a "periodic table" by placing possible symmetric set operators as abscissa and asymmetric set operators as ordinate. However, using all 16 pairs of operations in both axes leads to the display of equivalent rules. This can be avoided by realizing that, for example, a symmetric set "DO" becomes "GO" under complementary transformation, while remaining the same under mirror transformation. Omitting one of these pairs erases a whole column of repetitions. Continuing in this fashion, we can reach a 10×10 table that has all the 88 unique rules with only 12 repetitions. While constructing this table, we need to decide which repeating columns to erase and how to arrange the rows and the columns that are left at the end. The table that we have constructed, after evaluating numerous options based on mathematical and aesthetic criteria, is presented in Figure 2. The 12 repetitions that appear at the corners of the table are removed for clarity. Note that every adjacent row and column share at least one common operator, which means that all adjacent rules in the table share at least three common operators. This arrangement is not unique; however, it aims to arrange rules with very similar genotypes next to each other.

The periodic table presented in Figure 2 offers a systematic "bird's eye" view of all 88 unique rules of ECAs. Rules dominated by similar simple patterns (homogeneous, vertical lines, diagonal lines, horizontal stripes) tend to appear together. The rules that show rich behavior populate the "fertile crescent" along the diagonal where simple rules with contradicting patterns are expected to overlap. Among these rich rules, the ones that have common features are also brought together. Rule pairs 18, 146 and 122, 126 are striking examples of this. Despite the chaotic nature of these rules, starting a run with one of them and switching to the other rule results in the same pattern that is produced without the switching. This is because rule 18 (122) and rule 146 (126) share the same mapping, except for the configuration 111 (010), which is mapped to 0 in the former and 1 in the latter. This 111 (010) configuration is "washed out" in a few steps and is never visited again. This effect is also present if we start with rule 26 and continue with rule 154, but not the other way around.

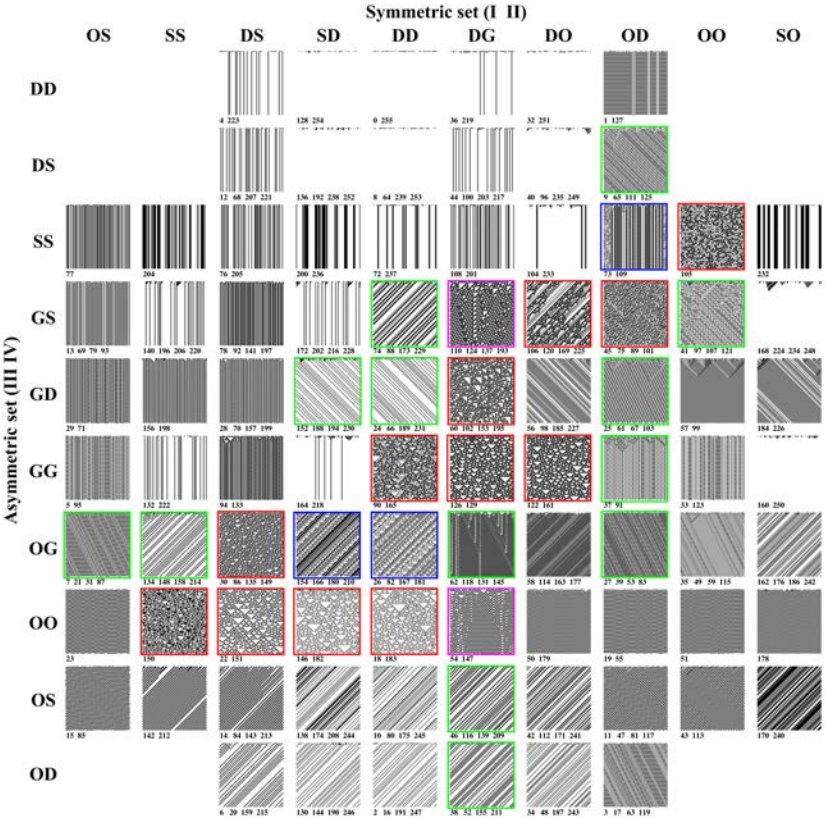

Figure 2. Periodic table of the ECAs. Rules corresponding to operator representations (in the order I, II, III, IV) and their mirror and complementary counterparts (if different) are presented below each box in increasing order. Each box presents a run starting with a random sequence of 100 binary digits evolved for 100 time steps according to the rule that is named by the smallest number. Periodic boundary condition is used. Chaotic, locally chaotic and complex rules are highlighted with red, blue and purple squares, respectively. Rules that acquire aperiodic behavior upon the logistic extension (described in Section 4) are highlighted with green squares.

The periodic table of ECAs also resonates with the findings of Li and Packard [16] in their classic study on the structure of the ECA rule space. They found two clusters of chaotic rules (in this context it includes the complex rules 54 and 110). Chaotic A includes rules 18, 22, 30, 54, 146 and 150, while chaotic B has rules 60, 90, 106, 110, 122 and 126. As seen in Figure 2, they appear as clusters at the bottom left and top right of the "fertile crescent," respectively. The authors found rule 45 to be separated from the clusters, but in our table we find it connected to the cluster B. Furthermore, clusters A

and B are connected over a bridge of locally chaotic rule 26 in the table. There are no other chaotic rules in the row and the column of rule 105, which was also found to be isolated by Li and Packard, but is connected to the cluster B over a bridge of locally chaotic rule 73.

The operator representation can further illuminate the studies on the computational irreducibility of ECAs. In particular, it is interesting to examine the rules that are detached from the coarse-graining network investigated by Israeli and Goldenfeld [20]. They have shown that rule 105 can be coarse-grained by rule 150. In the operator representation, these rules appear as OOSS and SSOO, respectively. Furthermore, both DGDG (rule 60) and DDGG (rule 90) can be coarse-grained by themselves. Finally, the authors were unable to coarse-grain four unique rules: 30, 45, 106 and 154. In the operator representation, they happen to be DSOG, OGDS, OSGD and SDOG. These make up four unique rules that involve all four operators while avoiding two complementary symmetric operators (D and G) in the same mirror symmetric set. In other words, the rules that were found to be irreducible are the ones that appear the most asymmetric in the operator representation. We believe that these observations can guide further studies on this subject.

Coarse-graining of ECAs was taken further by [21], where the emulation of all ECA rules with each other was searched exhaustively, revealing several pairings of different classes. We noticed that when both rules are rather simple, they tend to share three operators. These emulating-emulated pairs include 168 (SODS), 136 (SDDS); 168 (SODS), 170 (SOOS); 164 (SGDD), 128 (SDDD); 108 (DGSS), 76 (DSSS); 94 (DSGG), 90 (DDGG). However, this pattern breaks when the rule pairs (or the compiler) become more complicated. Finally, the authors provide the list of the most-emulated rules as a measure of their complexity (the more emulated, the less complex). The top 10 rules in this list are all composed of rules that are represented by use of only one or two distinct operators.

4. Logistic Extension of Elementary Cellular Automata

Recently, we have introduced the logistic extension of two outer-totalistic CAs: Game of Life and rule 90 [18]. This extension is achieved via introduction of a parameter λ that tunes the dynamics of CAs. $\lambda = 1$ corresponds to the original binary version of the studied systems. As λ is tuned below 1, the logical operations become functional operations and the binary state space thus extends into a Cantor set. As a result, the systems expand their complexity through series of deterministic transitions. In particular, rule 90, which is aperiodic at $\lambda = 1$, shows complex (or class 4) behavior at $\lambda \sim 0.6$.

The operator representation presented here enables us to go beyond the outer-totalistic rules and generalize the logistic extension to all ECAs. We first define four regions of operation for each pair (I, II, III and IV) as shown in Figure 3. The coordinates of a configuration $[L, C, R]$ (denoting left, center and right sites, respectively) defined as the sums $x \equiv L + R + 1/2 \pmod 2$ and $y \equiv L + C + 1/2 \pmod 2$ determine in which operation region the configuration pair falls. As shown in Figure 3, the eight possible binary configurations appear at the centers of the regions that correspond to their pair definitions shown in Figure 1(a). Hence, the configuration $[L, C, R]$ determines the operation region, which in turn determines the corresponding operator based on the rule at hand. Depending on the operator, the value of a site is updated according to one of the following formulas:

$$\text{Decay} \Rightarrow s^{t+1} = (1-\lambda)s^t$$

$$\text{Stability} \Rightarrow s^{t+1} = s^t$$

$$\text{Oscillation} \Rightarrow s^{t+1} = \begin{cases} (1-\lambda)s^t + \lambda, & \text{if } s^t \le \dfrac{1}{2} \\ (1-\lambda)s^t, & \text{if } s^t > \dfrac{1}{2} \end{cases}$$

$$\text{Growth} \Rightarrow s^{t+1} = (1-\lambda)s^t + \lambda.$$

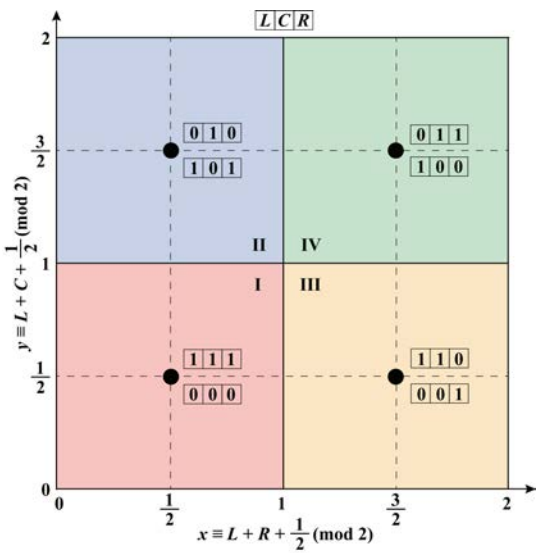

Figure 3. Definition of the operation regions based on a configuration. L, C and R correspond to the values at the left, center and right cells of a configuration.

where s^t and s^{t+1} are the values of the central site at the current and the next time step, respectively. These equations represent the generalized functional operators, which rewrite the transition function in a different fashion. However, the limit $\lambda = 1$ still generates the ECA rules. Note that this generalization is also consistent with the special case of the logistic rule 90 that we have reported earlier [18].

4.1 Further Emergent Behavior

Significant changes in dynamics can occur when x or y passes over from one region to another. This happens when the sum $L + C$ or $L + R$ is equal to the critical thresholds 0.5 or 1.5. The values that L, C and R can take is dictated by the λ-dependent Cantor set. Hence, we can expect these changes at the values of λ that mark the equality of binary sums to the critical thresholds. As λ is tuned below 1, the first time such a transition occurs is when $2\lambda = 1.5$. After this point, some of the rules start behaving differently than their original version because of the changes in the operation regime. This is similar to the noncausal mapping that emerges from sequential operation of two or more rules as discussed in [19], which also leads to complex behavior. However, in our case only one rule is involved, but there are localized operation regime changes that depend on the binary sums.

Accompanied by the collage in Figure 4, in this part we discuss several cases where unprecedented emergent behavior appears from rules of different classes. Some of the rules that are originally periodic can acquire aperiodic behavior. For example, rule 38 becomes locally chaotic at $\lambda \sim 0.69$ and chaotic at $\lambda \sim 0.61$. Periodic rules can also become complex, for example, rules 37 and 46 at $\lambda = 0.72$, also shown in Figure 4. All rules that gain aperiodic behavior upon the logistic extension are highlighted by green squares in Figure 2. Note that these rules are adjacent to the rules that are originally aperiodic.

Rules that originally exhibit chaotic, locally chaotic or complex behavior pass through multiple transitions while going between these regimes. As seen in Figure 4, chaotic rule 18 becomes complex at $\lambda = 0.73$, mimicking (but not exactly copying) the complex patterns seen in the original rule 54, which is (surprisingly) next to it on the periodic table. Rule 82, which is a locally chaotic rule close by in the table, also mimics the original rule 54 behavior at $\lambda = 0.74$.

Logistic extension breaks the symmetry between mirror rules because of the left-right asymmetry in the sum $L + C$. Indeed, using $x = C + R + 1/2 \pmod 2$ would have been more symmetric, but note that $y = L + C + 1/2 \pmod 2$ already enables us to explore the "$R + C$" side of the mirror rules. Hence, setting

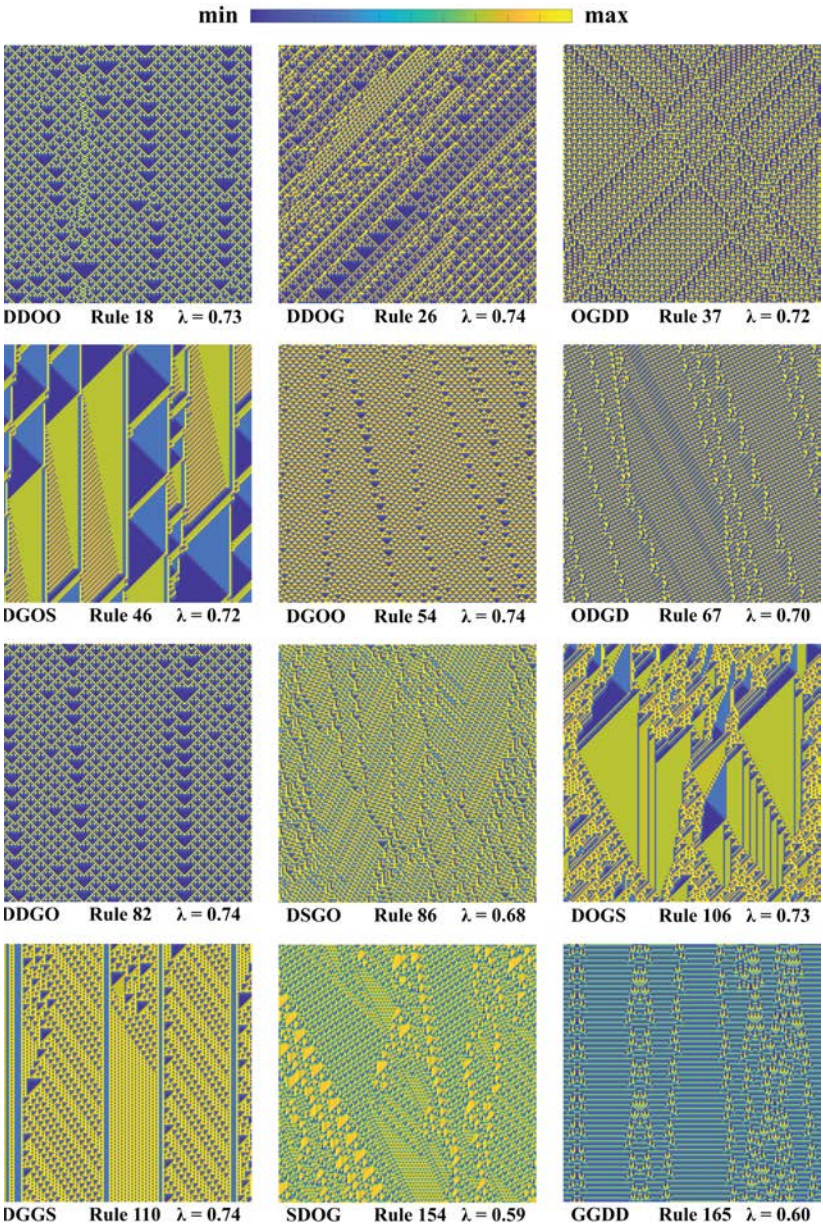

Figure 4. 150×150 cell snapshots at a later stage of a 1000×1000 simulation for various rules with given values of λ. The color bar shown at the top maps the range between the minimum and maximum cell values for each snapshot. Both conventional and operator representations of the rules are given below each panel.

$x = L + R + 1/2 \pmod 2$ reveals the transitions of outer-totalistic nature as well. This is clear in the distinct behavior of rule 26 (the mirror symmetry of rule 82), which has a mixture of chaotic and locally chaotic behavior at $\lambda = 0.74$, as shown in Figure 4. However, complementary rules behave in the same way under the logistic extension. For example, the behavior of complementary rules 90 and 165 is the same at $\lambda = 0.6$ [18].

Logistic extension breaks the symmetry between mirror rules because of the left-right asymmetry in the sum $L + C$. Indeed, using $x = C + R + 1/2 \pmod 2$ would have been more symmetric, but note that $y = L + C + 1/2 \pmod 2$ already enables us to explore the "$R + C$" side of the mirror rules. Hence, setting $x = L + R + 1/2 \pmod 2$ reveals the transitions of outer-totalistic nature as well. This is clear in the distinct behavior of rule 26 (the mirror symmetry of rule 82), which has a mixture of chaotic and locally chaotic behavior at $\lambda = 0.74$, as shown in Figure 4. However, complementary rules behave in the same way under the logistic extension. For example, the behavior of complementary rules 90 and 165 is the same at $\lambda = 0.6$ [18].

Rules that originally have complex behavior may remain complex while having noticeable changes in their dynamics, for example, rule 54 at $\lambda = 0.74$. They also can become locally chaotic like rule 110 at $\lambda = 0.74$ or become chaotic like rule 124 (not shown in Figure 4), which resembles its neighboring chaotic rule 60 at $\lambda = 0.72$. Rules that are chaotic or locally chaotic can behave in a complex fashion, as exemplified by rule 86 (mirror symmetry of the rule 30) at $\lambda = 0.68$ and rule 154 at $\lambda = 0.68$, respectively.

4.2 Class Transitions

A remarkable benefit of our extension is without doubt related to the emergence of complex structures. Originally, ECAs possess a rather small set of class 4 rules. Rules 110 and 54 are well studied for their abilities to perform computation [22–24], yet they are unfortunately limited in providing a framework for studying general properties of class 4 behavior. Making use of our λ parameter, we discuss examples that can help us understand mechanisms of class transitions in one-dimensional CAs.

We start to qualitatively analyze the emergence of class 4 behavior by looking at the dynamics of rules 110 and 54. Among many shared similarities, both rules are characterized by large regions with spatiotemporal periodicity. The periodicity is characterized by oblique lattice units in spacetime, as shown in Figure 5(a, b). We name these uniform lattice regions "fields." Complex structures are correlated

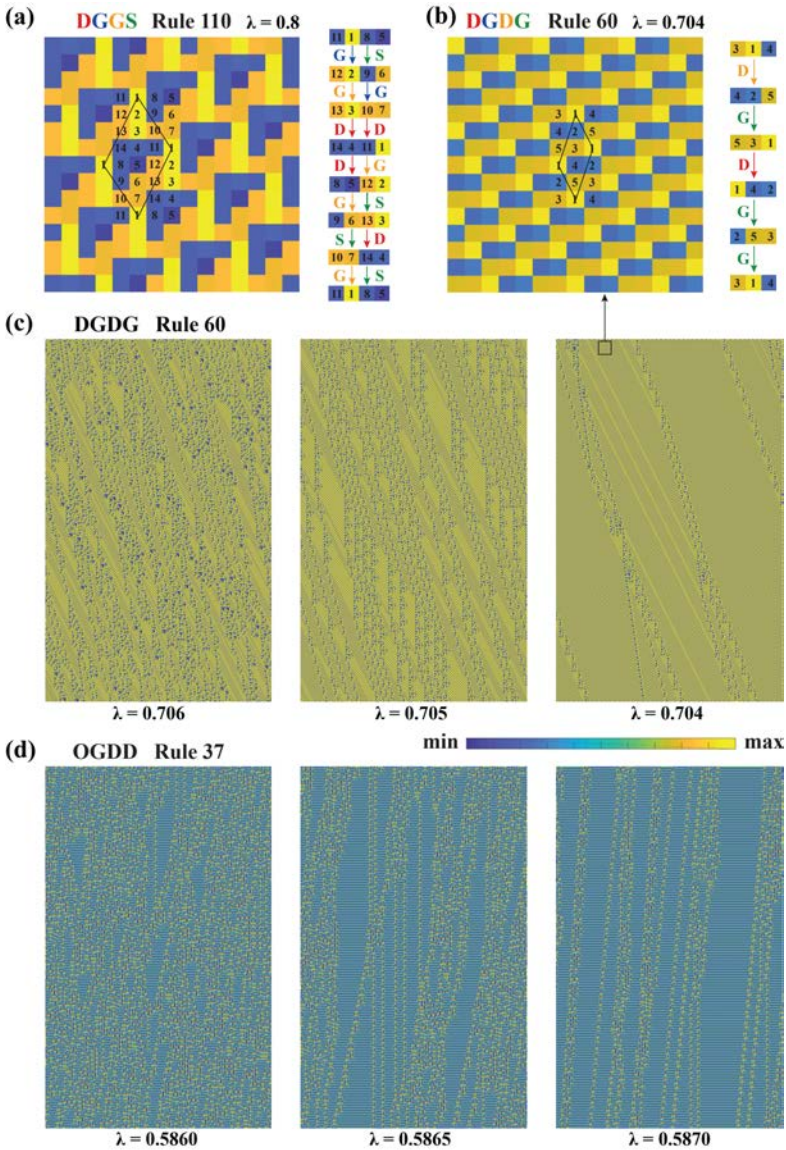

Figure 5. (a) Period-7 field of rule 110 for $\lambda = 0.8$. The operators are colored with respect to their operation region in accordance with Figure 1. The 14 states with distinct values that constitute the period-7 field are denoted with numbers. The repeating unit cell is delineated with solid lines. The area of this unit cell gives the number of distinct states constituting the field. (b) Period-5 field of rule 60 for $\lambda = 0.704$. 400×250 cell snapshots at a later stage of a 2000×2000 simulation of (c) rule 60 and (d) rule 37 with given values of λ. The color bar maps the range between the minimum and maximum cell

values for each snapshot. As the value of λ is decreased from 0.706 to 0.704, the period-5 fields start growing within the chaotic regions. This way the system makes a transition from class 3 to class 4 behavior while a single parameter is tuned. Similar transition is observed for rule 37 when the value of λ is increased from 0.586 to 0.587.

with interfaces between fields that have some offset, or mismatch. In Figure 5(a), we have a closer look at the field that dominates the dynamics of rule 110. At $\lambda = 0.8$ the system follows the same behavior as the original version because $\lambda > 0.75$. However, the logistic extension helps us to discern 14 states with distinct values that constitute the lattice of this field. States 1 through 7 are created by cyclic application of the operator string GGDDGSG. The final operation takes state 7 back to state 1 and the process repeats with temporal periodicity of 7. States 8 through 14 are created by cyclic application of a different string: SGDGSDS. The two sets of seven states arrange themselves in alternating fashion along the space with a phase difference (time shift) and produce each other's neighborhood. Together they form an autocatalytic set of states (and corresponding operations) that sustains the growth of a period-7 field extending both in space and time. We believe that the size and geometry of the lattice units in every field might help us understand the characteristics of their related propagators.

As mentioned earlier, upon the logistic extension we reveal many examples of class 4 behavior in various rules. Similar to rule 110, these systems are also characterized by a dominating field or several fields, like in the case of rule 106 at $\lambda = 0.73$ (see Figure 4). The autocatalytic sets of states and operations can support a certain field only on a limited range of λ. As λ is tuned within this range, dominance of that field can be enhanced or diminished. This can be utilized to make smooth transitions between class 3 and class 4 behavior. Earlier we discussed such a possibility in logistic rule 90 [18]. Here we give two more examples using rule 60 and rule 37. Figure 5(b, c) shows the analysis of rule 60. As λ is decreased from 0.706 to 0.704, we see an increase in the dominance of the period-5 field shown in Figure 5(c). Here a single set of five states created by the string DGDGG is involved. As the fields expand, the complex propagators become more and more clear. Transition in rule 37 is presented in Figure 5(d). Here period-3 stripes take over the system as λ is increased from 0.586 to 0.587. Transition in rule 90 also involves period-3 stripes and occurs in a similar range of λ. Furthermore, the propagators that emerge in these systems resemble each other.

5. Conclusion

In summary, we redefine the transition function of elementary cellular automaton (ECA) rules by introducing a notation based on logical operators. This allows us to organize all unique rules in a periodic table based on their similarities in the operator representation. Rules with similar behavior appear clustered in the table. Hence, similarity in operator representation leads to similarity in global dynamics. We extend the ECA to its logistic counterpart by introducing a single parameter that tunes the iteration rate of operators. This expands the range of behavior that ECA models can offer. In particular, we reveal several instances of complex (class 4) behavior in various rules and corresponding values of a tuning parameter. We also show examples of transition from chaotic to complex behavior while the parameter is tuned continuously. Here, autocatalytic sets of states and operators produce periodic spatiotemporal regions, named fields, that gradually grow within the chaotic regions until the latter become complex propagators.

Alternative approaches can generate emergent behavior using ECA rules. Elementary cellular automata (ECAs) can be decomposed into a smaller set of primary rules that act in conjunction to generate a larger rule space [19]. This set of primary rules comprises all the observed dynamic behavior of ECAs and also serves as an alternative formulation to observe further emergent structures.

In terms of computability in ECAs, there are studies that show that simple ECA rules can generate complex behavior when run with sophisticated initial conditions [21], proposing that class 3 rules can be reprogrammed to perform computation. In contrast, our approach stands closer to discovering systems where computation can emerge from "attractor" properties of the rules.

The class transitions upon tuning λ prove that ECAs are models that embed more complexity in their structures. Moreover, the emergence of class 4 behavior expands the domain of rules that can be studied for Turing computability. We believe that a similar approach may be useful to explore hidden features in other complex systems, such as discrete lattice models [25, 26] and Boolean genetic networks [27].

Acknowledgments

M. Ibrahimi acknowledges support from the Ph.D. fellowship of "Investissements d'Avenir," a French government program managed by the French National Research Agency (ANR-16-CONV-0001), and from A*MIDEX. S. Jahangirov acknowledges support from the

Turkish Academy of Sciences - Outstanding Young Scientists Award Program (TÜBA-GEBIP). We thank Tamer Taskiran for fruitful discussions.

Supplemental Material

In Figure 6, we present all 256 ECA rules as a 16×16 grid with respect to a specific ordering of symmetric and asymmetric operator pairs. The order is arranged in such a way that the rules are grouped in eight 32-rule clusters according to their rule numbers. This could be used to easily derive the operator representation from the rule number or the other way around.

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

Figure 6. All 256 ECA rules are presented as a 16×16 grid with respect to a specific ordering of symmetric and asymmetric operator pairs. The order of the operator pairs is arranged in such a way that the rule numbers are as close to each other as possible. Concomitantly, rules are grouped in eight 32-rule clusters. This figure makes it easy to translate the rule number to operator representation and vice versa.

References

[1] S. Wolfram, "Cellular Automata as Models of Complexity," *Nature*, **311**(5985), 1984 pp. 419–424. doi:10.1038/311419a0.

[2] J. von Neumann, *Theory of Self-Reproducing Automata* (A. W. Burks, ed.), Urbana, IL: University of Illinois Press, 1966.

[3] L. Manukyan, S. A. Montandon, A. Fofonjka, S. Smirnov and M. C. Milinkovitch, "A Living Mesoscopic Cellular Automaton Made of Skin Scales," *Nature*, **544**(7649), 2017 pp. 173–179. doi:10.1038/nature22031.

[4] D. A. Rosenblueth and C. Gershenson "A Model of City Traffic Based on Elementary Cellular Automata," *Complex Systems*, **19**(4), 2011 pp. 305–322. doi:10.25088/ComplexSystems.19.4.305.

[5] S. Wolfram, "Statistical Mechanics of Cellular Automata," *Reviews of Modern Physics*, **55**(3), 1983 601–644. doi:10.1103/RevModPhys.55.601.

[6] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, Inc., 2002.

[7] M. Cook, "Universality in Elementary Cellular Automata," *Complex Systems*, **15**(1), 2004 pp. 1–40. complex-systems.com/pdf/15-1-1.pdf.

[8] G. J. Martinez, A. Adamatzky and H. V. McIntosh,"Phenomenology of Glider Collisions in Cellular Automaton Rule 54 and Associated Logical Gates," *Chaos, Solitons and Fractals*, **28**(1), 2006 pp. 100–111. doi:10.1016/j.chaos.2005.05.013.

[9] A. Wuensche, "Classifying Cellular Automata Automatically: Finding Gliders, Filtering, and Relating Space-Time Patterns, Attractor Basins, and the *Z* Parameter," *Complexity*, **4**(3), 1999 pp. 47–66. dl.acm.org/doi/10.5555/310596.310603.

[10] H. A. Gutowitz, "A Hierarchical Classification of Cellular Automata," *Physica D: Nonlinear Phenomena*, **45**(1–3), 1990 pp. 136–156. doi:10.1016/0167-2789(90)90179-S.

[11] P.-M. Binder, "A Phase Diagram for Elementary Cellular Automata," *Complex Systems*, **7**(3), 1993 pp. 241–247. complex-systems.com/pdf/07-3-4.pdf.

[12] E. Mizraji, "The Emergence of Dynamical Complexity: An Exploration Using Elementary Cellular Automata," *Complexity*, **9**(6), 2004 pp. 33–42. doi:10.1002/cplx.20043.

[13] C. G. Langton, "Computation at the Edge of Chaos: Phase Transitions and Emergent Computation," *Physica D: Nonlinear Phenomena*, **42**(1–3), 1990 pp. 12–37. doi:10.1016/0167-2789(90)90064-V.

[14] R. Badii and A. Politi, "Thermodynamics and Complexity of Cellular Automata," *Physical Review Letters*, **78**(3), 1997 pp. 444–447. doi:10.1103/PhysRevLett.78.444.

[15] E. Borriello and S. I. Walker, "An Information-Based Classification of Elementary Cellular Automata," *Complexity*, 2017 Article ID 1280351. doi:10.1155/2017/1280351.

[16] W. Li and N. H. Packard, "The Structure of the Elementary Cellular Automata Rule Space," *Complex Systems*, **4**(3), 1990 pp. 281–297. complex-systems.com/pdf/04-3-3.pdf.

[17] A. Shreim, P. Grassberger, W. Nadler, B. Samuelsson, J. E. S. Socolar and M. Paczuski, "Network Analysis of the State Space of Discrete Dynamical Systems," *Physical Review Letters*, **98**(19), 2007 198701. doi:10.1103/PhysRevLett.98.198701.

[18] M. Ibrahimi, O. Gülseren and S. Jahangirov, "Deterministic Phase Transitions and Self-Organization in Logistic Cellular Automata," *Physical Review E*, **100**(4), 2019 042216. doi:10.1103/PhysRevE.100.042216.

[19] J. Reidel and H. Zenil, "Rule Primality, Minimal Generating Sets and Turing-Universality in the Causal Decomposition of Elementary Cellular Automata," *Journal of Cellular Automata*, **13**, 2018 pp. 479–497. jca-13-5-6-p-479-497.

[20] N. Israeli and N. Goldenfeld, "Computational Irreducibility and the Predictability of Complex Physical Systems," *Physical Review Letters*, **92**(7), 2004 074105. doi:10.1103/PhysRevLett.92.074105.

[21] J. Reidel and H. Zenil, "Cross-Boundary Behavioural Reprogrammability Reveals Evidence of Pervasive Universality," *International Journal of Unconventional Computing*, **13**, 2018 pp. 309–357. ijuc-13-4-5-p-309-357.

[22] M. Cook, "A Concrete View of Rule 110 Computation," *Electronic Proceedings in Theoretical Computer Science*, **1**, 2009 pp. 31–55. doi:10.4204/EPTCS.1.4.

[23] G. J. Martínez, A. Adamatzky, F. Chen and L. Chua, "On Soliton Collisions between Localizations in Complex Elementary Cellular Automata: Rules 54 and 110 and Beyond," *Complex Systems*, **21**(2), 2012 pp. 117–142. doi:10.25088/ComplexSystems.21.2.117.

[24] S. Ninagawa, "Power Spectral Analysis of Elementary Cellular Automata," *Complex Systems*, **17**(4), 2008 pp. 399–411. complex-systems.com/pdf/17-4-5.pdf.

[25] T. Kawano, "Translating the Conway's Game of Life as a Discrete Logistic Cellular Automata Model with Density Effects," *International Journal of Innovative Computing, Information and Control*, **16**(5), 2020 pp. 1656–1666. doi:10.24507/ijicic.16.05.1655.

[26] K. Kaneko, "Overview of Coupled Map Lattices," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **2**(3), 1992 pp. 279–282. doi:10.1063/1.165869.

[27] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, New York: Oxford University Press, 1993.