

Emergence of Self-Replicating Hierarchical Structures in a Binary Cellular Automaton

Bo Yang
Adjacent Lab
bo@adjacentlab.com

Abstract We have discovered a novel transition rule for binary cellular automata (CAs) that yields self-replicating structures across two spatial and temporal scales from sparse random initial conditions. Lower-level, shape-shifting clusters frequently follow a transient attractor trajectory, generating new clusters, some of which periodically self-duplicate. When the initial distribution of live cells is sufficiently sparse, these clusters coalesce into larger formations that also self-replicate. These formations may further form the boundaries of an expanding complex on an even larger scale. This rule, dubbed “Outlier,” is rotationally symmetric and applies to 2-D Moore neighborhoods. It was evolved through genetic programming during an extensive search for rules that foster open-ended evolution in CAs. While self-replicating structures, both crafted and emergent, have been created in CAs with state sets intentionally designed for this purpose, the Outlier may be the first known rule to facilitate nontrivial emergent self-replication across two spatial scales in binary CAs.

Keywords

Cellular automata, CA, replication, complexity, genetic programming, open-ended evolution, OEE

I Background and Introduction

Traditionally, the “building blocks” one level below self-replicating structures in cellular automata (CAs) are cells with multiple possible states. Notable examples include 29 states in the original universal construction machine by von Neumann (1966), 8 states in the self-replicating loops by Langton (1984), and 9 states in the evoloop by Sayama (1999). They also require carefully designed initial configurations. Emergent self-replicating structures from random initial conditions have previously been achieved using CAs with 8-bit state sets (Chou & Reggia, 1997). A comprehensive review of self-replicating and self-reproducing structures in CAs can be found in Sayama and Nehaniv (2024).

In all these cases, each state or its subcomponent generally assumes a specific “role,” such as signal passage, replication trigger, or structural protection, all of which are vital elements of the intended replication mechanisms. In contrast, a cell in a binary CA carries minimal information and is unlikely to perform any specific role. Consequently, if self-replicating structures were to emerge from randomness in a binary CA, the “building blocks” must be clusters of cells, emerging solely from the rule itself. This could be of interest to studies of emergence and self-organization.

In this article, we report the discovery of a novel two-state CA rule that enables the spontaneous assembly of larger self-replicating “formations” from smaller, shape-shifting “clusters” that themselves emerge from random initial conditions. A growing number of these replicating

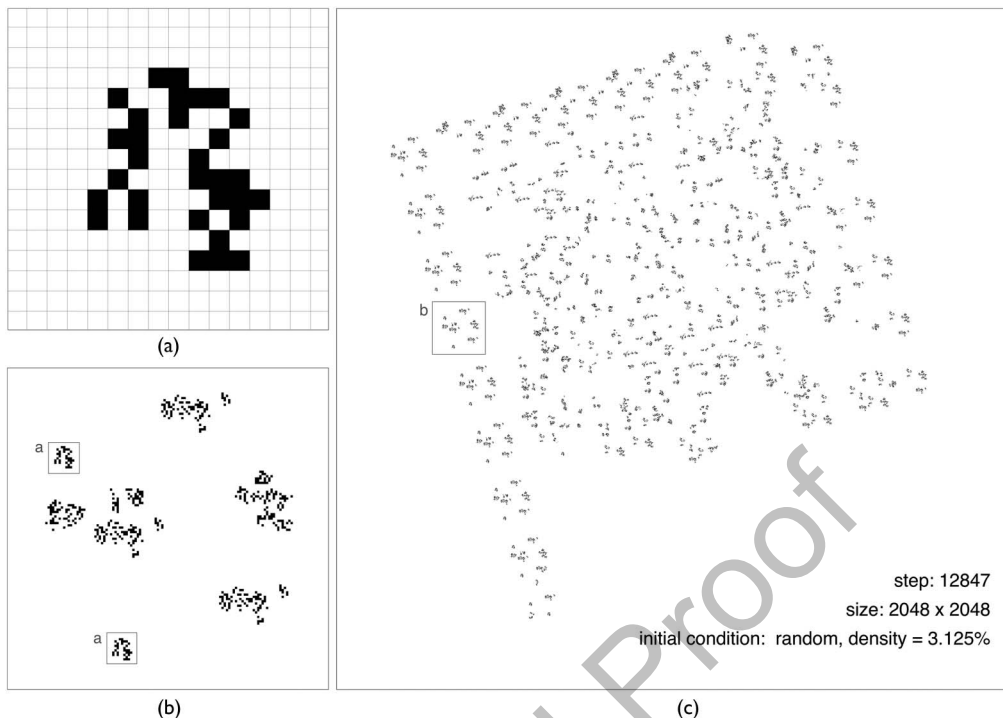


Figure 1. Sample outcome from the Outlier rule starting with a sparse random initial condition. (a) Two clusters on the smallest scale. (b) A self-replicating formation, assembled from a few clusters. (c) On the largest scale, an expanding complex with a semichaotic interior, bordered by replicating formations.

formations often subsequently form an expanding superstructure, or “complex,” on an even larger scale. Figure 1 illustrates the hierarchical arrangement of these structures. We have named this rule the “Outlier,” as it generates the most seemingly complex behaviors among all the interesting rules we have encountered.

2 Discovery of the Outlier Rule

The Outlier rule was serendipitously discovered during an extensive automated search for CA rules that would support open-ended evolution (OEE), defined as the continuous emergence of novel and increasingly complex behaviors (Bedau et al., 2000). Although often associated with OEE, self-replication was never an explicit goal of this search project. Though many method details were identified as crucial through trial and error, here we focus on two aspects relevant to the characteristics of the rules selected for evaluation.

The specific search runs that led to the Outlier were conducted within the space \mathcal{R} of 2^{140} rotationally symmetric rules on Moore neighborhoods of 2-D binary CAs. Mirror parity was not required. To keep the search tractable, we used genetic programming (GP) and various forms of bit representation of the rules as genotypes in several phases of the project.

In general, the details of genotype representation in genetic algorithm (GA) and GP searches modulate the probability distribution of random sampling in the parameter space, thereby shaping the evolutionary search. This fact is particularly significant in our project, as the entirety of all rules ever evaluated will compose a vanishingly small fraction of \mathcal{R} .

The Outlier rule was discovered during a GP search run in which each genotype, or rule, was represented as a tree structure of bitwise logic operations (G_1, \dots, G_L). Specifically, each node

G_i in the tree of length L is a tuple of three integers representing one logical operator and two operands:

$$G_i = (f_i, i_1, i_2) \quad (1)$$

where $f_i = 0$ or 1 and $0 \leq i_1 \leq i_2 < i \leq L$. For each cell in a Moore neighborhood, we then compute its new state by traversing the entire tree, starting from its neighborhood states as initial inputs:

$$N_i = S_i, \text{ when } 1 \leq i \leq 9 \quad (2)$$

Here S_1 through S_9 represent the current states of the cells in one Moore neighborhood, with S_1 at the center and S_2 at the top-left corner, followed by S_3 through S_9 in clockwise order. We also introduce a constant input $N_0 = \text{TRUE}$ to facilitate negation operations. Then,

$$N_i = \begin{cases} N_{i_1} \oplus N_{i_2}, & \text{if } f_i = 1 \\ N_{i_1} \wedge N_{i_2}, & \text{if } f_i = 0 \end{cases} \text{ when } i > 9 \quad (3)$$

where \oplus and \wedge denote Boolean logical XOR and AND, respectively. The center cell S_1 is then updated to the value of the last node N_L for the next step. All other cells in the automaton are synchronously updated in the same manner.

The traditional lookup table representation of each rule can be mapped to single or multiple trees expressed this way, and they are computationally equivalent. Mathematically, any binary CA rule can be represented as a Boolean function of its inputs, and it is known that any Boolean function can be constructed with only the NAND operator, $\text{NAND}(A, B) = \text{TRUE} \oplus (A \wedge B)$, and N_0 is initialized with TRUE; therefore our logical tree G_i can represent any binary CA rule. In practice, a shorter length of tree L is much preferred for performance reasons. In our implementations, while functional equivalence is observed, we introduce constraints and fixed handcrafted substructures into G_i , as well as auxiliary procedures during tree traversals to enforce rotational symmetry. In the specific search runs in which the Outlier rule was found, we were able to keep $L \leq 280$ thanks to these optimizations.

The choice of this representation was initially motivated by computational efficiency, crucial to CA rule search, as a single fitness evaluation often necessitates the computation of hundreds of billions of cell updates. With one bit per cell memory representation, many adjacent cells can be loaded into long registers in processors and updated in parallel via consecutive bitwise logical operations specified by the aforementioned trees. Modern CPUs and GPUs can update hundreds to billions of bits concurrently in this manner, with excellent memory locality. For instance, the GP search that led to the Outlier rule ran on a 14-core Xeon CPU capable of updating thousands of cells concurrently with AVX-512 support in each core. In later iterations on GPUs, bitwise logical operation trees were tweaked to keep most, if not all, operations among the GPU register files, thanks to optimizing kernel compilers. This often resulted in a speedup by one to two orders of magnitude.

The second implementation detail pertinent to our findings is the fitness function, which ideally would measure the complexity or “open-endedness” of the phenotypes, which in our case are the CA bitmaps generated by each rule. As often happens in GA and GP searches, fitness functions derived directly from spatial and temporal analysis are prone to “cheating,” whereby rules maximize the fitness score with surprisingly novel yet unwanted simplistic behaviors. In the later stages of the project, we adopted “novelty search” as first developed by Lehman and Stanley (2008). This approach rewards new phenotypes that bring “novelty” to the pool of all previously evaluated phenotypes. In our implementation, we extract a feature vector, \mathbf{F} , for each rule from the complexity profile (Bar-Yam, 2003) of CA bitmaps in the later stages of convergence. For each new rule, a

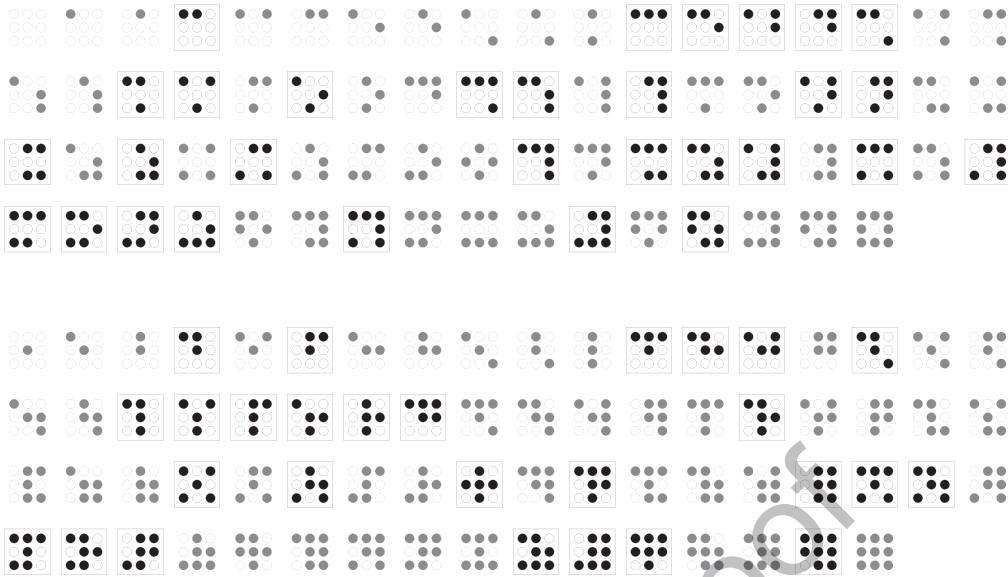


Figure 2. The Outlier rule. The center cell in each of the boxed neighborhoods and their three quarter-turn rotations becomes or stays alive. Filled and empty circles stand for live (TRUE) and dead (FALSE) cells, respectively.

novelty score, used as fitness, is calculated from the distances from \mathbf{F} to its k nearest neighbors in the space of all (or a large sample of) previously computed \mathbf{F} .

This implementation of novelty search was somewhat successful, yielding a few rules with intriguing behaviors not seen when other fitness functions were used. These rules included the Outlier, which was algorithmically tagged as sufficiently “novel” and thus worthy of visual inspection. The search took about a month of aggregated machine run time during which hundreds of thousands of rules were evaluated. We ended each search run when the novelty score plateaued and it appeared that neither new combinations of parameter settings nor additional computing resources would help. Thus far, nothing more complex than the Outlier has been observed.

3 The Outlier Rule

As shown in Figure 2, the Outlier rule observes rotational symmetry but lacks mirror symmetry. Similar to many solutions produced by GP searches, it does not possess a clearly recognizable structure or definable formulation. Notably, its rule table representation has 220 live entries out of 512, which is denser than Conway’s Game of Life, which has 140.

The Outlier rule differs from its mirror twin, defined by replacing each neighborhood in Figure 2 with its mirror image. Nevertheless, all the essential properties observed in this article should also apply to the Outlier’s mirror twin.

A run length–encoded (RLE) representation of this rule is included in Appendix 1, along with a method to reproduce most results in this article without coding.

4 Cluster, Formation, and Complex

Under the Outlier rule, three categories of trajectories typically follow from random initial configurations. Although each individual outcome varies when the initial configuration is chosen randomly, the statistical likelihoods are highly dependent on the initial density of live cells D_0 and the grid size. A $1,024 \times 1,024$ grid is more likely to become completely empty when $D_0 < 0.02$, to

become semichaotic when $D_0 > 0.15$, and to support replicating formations when D_0 falls between these values. We will refer to these three types of outcomes as “barren,” “dense,” and “sparse,” respectively. These cutoff values for D_0 are grid size dependent. For grids smaller than 512×512 , replicating formations do not occur at all. We explain the dependency of the likelihoods on D_0 in the next section.

Regardless of D_0 , shape-shifting clusters, each composed of a few dozen live cells at most, form in fewer than a hundred steps. A *cluster* is formally characterized as a temporally evolving shape composed of live cells that are topologically connected, whereby two live cells are considered adjacent if they are in each other’s Moore neighborhood. Each cluster continuously shape-shifts, sometimes splitting into two or interacting with another cluster through collision or merging. When isolated due to proximity, most of these clusters would disappear within a hundred steps.

However, on a “sparse” grid that is sufficiently large, a small fraction of the clusters can survive and grow into larger, self-replicating formations by spawning new clusters. Each of these formations consists of a group of clusters, with the number of clusters fluctuating around 10. A replicating formation expands its territory by creating copies of itself while slowly moving, until it collides with another formation or cluster outside its territory. Collisions break down a formation back to clusters, which then change shape and interact among themselves continuously and in a chaotic manner, eventually occupying the entire grid. As shown in Figure 3, a dense grid will transition directly into this semichaotic phase before any formation has the opportunity to emerge.

When a single replicating formation survives in the middle of a sufficiently large empty area, it periodically generates new formations that form an even larger structure, or a “complex,” at a still higher scale. Each boundary region of a complex’s four edges consists mostly of replicating formations. These are mostly identical to each other and shape-shift synchronously. The initial expansion of the boundaries appears to be driven by a formation protruding out of the rectangular boundary, as shown in Figure 1. Given that all replicating formations are spaced $(52, 172)$ or $(172, 52)$ cells

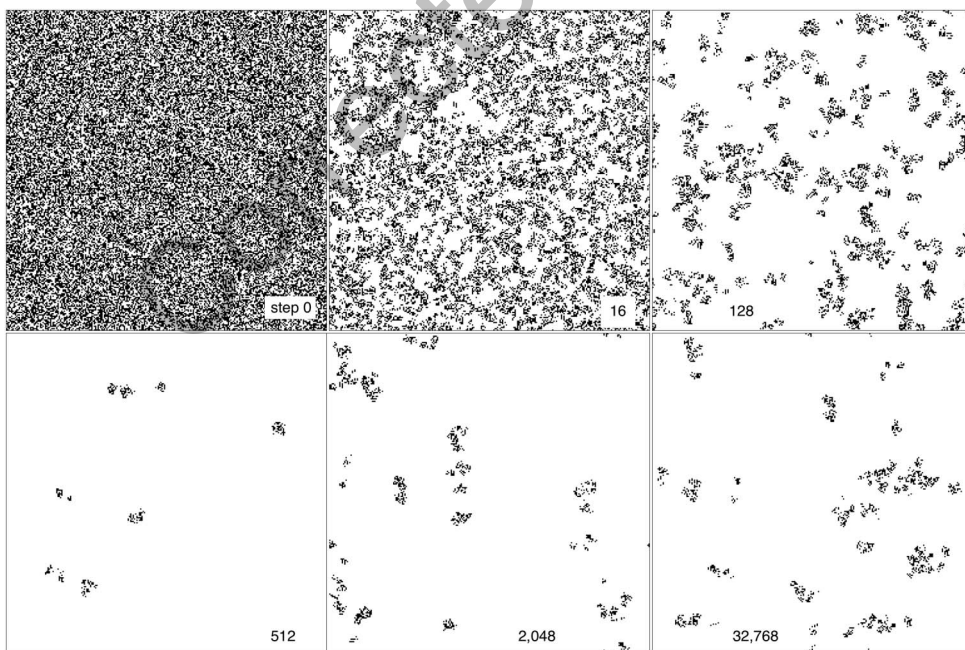


Figure 3. Transitions from a densely populated random grid. The grid is 256×256 cells with periodic boundary conditions and an initial density of 50%. The step numbers are shown as labeled. At around Step 128, the grid enters a “semichaotic” phase, with clusters shape-shifting in random manners but within a much lower upper bound of their sizes. The density of the clusters also stabilizes at around Step 2,048.

apart from neighboring formations on each side of the complex, the edges of a complex form a rectangle that is tilted counterclockwise from the axes by $\arctan(13/43)$, or approximately 16.8214° .

The interior of a complex is occupied by surplus clusters, or “debris,” that are generated by the replication process but are not part of the replicating formation themselves. These evolve in time in the same semichaotic manner as on a dense grid and occasionally interact with the bordering formations without affecting their integrity.

A complex continuously expands until it occupies all available space or collides with other structures outside its territory. Upon such collision, formations break down, and their clusters continue their dynamic transformations on a lower scale, similar to a dense grid.

5 Temporal Loops and Transient Attractors

To understand the dynamics underlying the replicating formations, we conducted experiments by initializing an empty grid with a single isolated 3×3 cluster in the center. Results are displayed in Figure 4. Out of the 140 possible initial configurations, rotational symmetry considered, two (c_0 and

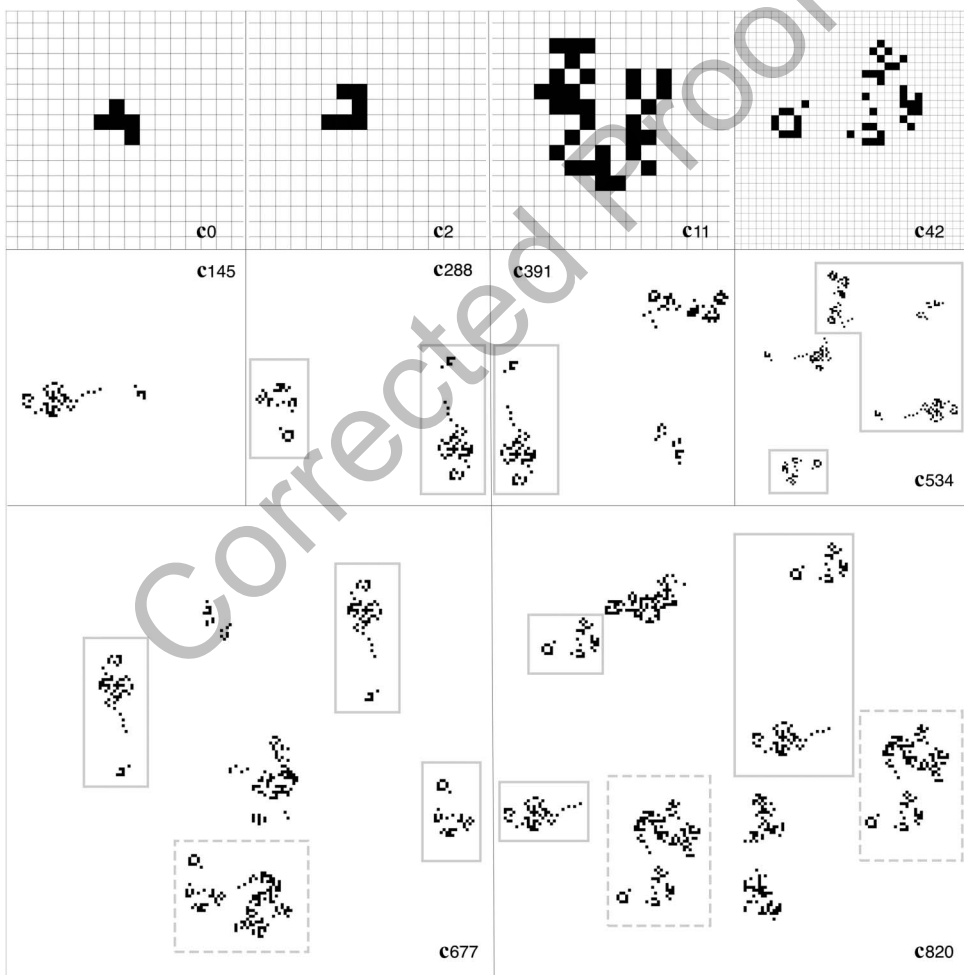


Figure 4. Seed trajectory \mathcal{A}_s . Configurations are numbered with step counts. Configuration c_2 reappears periodically every 143 steps in c_2 – c_{145} – c_{288} and c_{391} – c_{534} – c_{677} . Formations in solid and dashed boxes have appeared previously in whole or in part, respectively. Configuration c_{391} reappears in c_{534} (rotated) and later in Figure 5. Note that rotations of c_{11} appear in Figure 1(a).

\mathbf{c}_2) develop into replicating formations, while all others die out. Configuration \mathbf{c}_0 updates into \mathbf{c}_2 in two steps (with a configuration \mathbf{c}_1 in between that is larger than 3×3), thus following the same trajectory thereafter. We refer to them as “seed” clusters and their trajectory of expansion as \mathcal{A}_s . In short, any isolated 3×3 initial cluster either disappears or follows \mathcal{A}_s .

A detailed examination of \mathcal{A}_s reveals that \mathbf{c}_2 and some of its follow-up clusters reappear periodically, rotated 90° counterclockwise each time, with a period of 143 steps. The first period begins when \mathbf{c}_2 is initialized and ends when it reappears in \mathbf{c}_{145} , rotated and translated, among several clusters spun off during the period. This happens again after another 143 steps, and the formation grows larger. Another two-period run starts at \mathbf{c}_{391} , with two, then three rotated copies of \mathbf{c}_2 .

Each new reappearance of the rotated \mathbf{c}_2 introduces a new subtrajectory into \mathcal{A}_s if the new cluster is sufficiently isolated from the rest of the formation and can thus seed its own trajectory. Because the rule is deterministic and rotationally symmetric, all the structures appearing in the first period, such as \mathbf{c}_{11} and \mathbf{c}_{42} , reappear and sometimes self-replicate in the same 143-step period. We also identify these as “seed clusters,” and each time one materializes outside the existing trajectory, it adds a new branch of subtrajectory onto \mathcal{A}_s .

These subtrajectories are only partially self-similar to the original \mathcal{A}_s , as collisions restrict their growth when the vicinity becomes crowded. Hence fourth (rotated) reappearances rarely occur. For example, \mathbf{c}_2 is absent in \mathbf{c}_{820} , which is 143 steps after \mathbf{c}_{677} , but four copies of \mathbf{c}_{42} appear.

Configuration \mathbf{c}_{391} appears to be around the time when the accumulation of new clusters suppresses dynamics of 143-step periods by crowding the empty space, and intercluster interactions form parallel dynamics on a longer timescale, embodied in larger structures emerging at the formation scale, some of which can self-replicate and thus be identified visually.

The formations in the third 143-step period, for example, \mathbf{c}_{391} in Figure 4, both as a whole and as part of a higher-level structure, start to reappear with a period of 1,556 steps. In fact, the formation develops into the “protruding arm” of the larger complex, as illustrated in Figure 5. Visually, it appears to be shape-shifting while slowly moving away from its original position, producing a new replicating formation “behind” itself in each period. A closer inspection reveals that the “protruding arm” shares many clusters with the adjacent replicating formation that is being formed, and the boundary between the formations shifts constantly and lacks a clear definition. Many identical clusters are components of both the protruding arm and the replicating formations, and they shape-shift in sync; most of these clusters and formations reappear every 1,556 steps.

A similar replicating process later starts at the diagonally opposite corner of the complex, with the corner formation appearing to be “caved in” rather than protruding. In Figure 5, the replicating process first starts at the left edge of the complex, then proceeds at each following edge in a clockwise order, spaced by a time lag characterized by the same 1,556-step period. The bottom edge forms last and is least defined. As the complex expands, one new replicating formation is added to each edge every 1,556 steps. Most of each edge’s formations shape-shift perfectly in sync and repeat with the same 1,556-step period.

Under sparse initial random conditions, our close examination of the updates revealed a clear pattern: The majority of arbitrarily formed clusters eventually vanish. The rare survivors consistently enter the same trajectory \mathcal{A}_s at various entry points. (The only exception observed is one type of small period 4 “spinner” that rotates 90° per step, as illustrated in Appendix 2.) For example, the automaton in Figure 1 has one surviving cluster that enters \mathcal{A}_s as a seed cluster at \mathbf{c}_2 .

When more than one cluster survives, their individual temporal trajectories along \mathcal{A}_s derail when there is a collision between clusters originating from different seeds. Consequently, to develop into replicating formations, surviving clusters need to maintain sufficient spatial separation. This explains how the density of the initial random configuration determines the existential likelihood of replicating formations.

Neither random initial conditions nor 3×3 initial seeds cleanly generate “pure” replicating formations, as they always produce additional clusters, or “debris,” in their vicinities. Out of curiosity, we initialized a grid with nothing but an isolated replicating formation without the debris, for instance, as Figure 1(b) or the dashed boxes in Figure 5, and found the subsequent behavior to be



Figure 5. Formations form a larger complex by self-replicating every 1,556 steps. Step counts are (a) 391, (b) 1,947, (c) 3,503, (d) 5,059, and (e) 6,615. The formation in dashed boxes, referred to throughout this article as a “replicating formation,” first appears in (b) as a part of the whole, is replicated in largest quantities later after (e), and becomes the key building block of the edges of the expanding complex.

similar, as it self-replicates and then grows into a complex. Additionally, we isolated each individual component cluster in the same replicating formation in Figure 5(b) and successively used each as an individual seed for initialization, finding that about half disappeared, while the others developed into full formations. In short, \mathcal{A}_1 appeared to be dominating, even though it is not robust.

When strictly observing Moore’s criterion (Sayama & Nehaniv, 2024), we can assert that many structures at multiple levels of hierarchical structures self-replicate. For example, in Figure 4, \mathbf{c}_2 is a component of \mathbf{c}_{145} , and \mathbf{c}_{145} in turn is a component of the dashed boxes in Figure 5, which self-replicate both as part of a complex and when initialized in isolation. On the other hand, not all components of a self-replicating formation would satisfy Moore’s criterion. As mentioned, many do not survive when isolated as initial seed structures.

In summary, a CA operating under the Outlier rule transitions into one of three phases: empty, semichaotic, or replication at the formation level. The latter phase is characterized by a trajectory that resembles an expanding transient attractor. Reappearances of both clusters and formations attach to the trajectory new temporally evolving spatial patterns as subbranches, each with distinct

characteristic period lengths: 143 steps for clusters and 1,556 steps for formations. However, this pattern of expansion, characterized by temporally shifted and imperfect “self-similarity,” is transient, as the complex eventually either exhausts the available space for expansion or collides with other structures, prompting the semichaotic phase to take over.

6 Discussion

As discussed at the beginning, each state in most many-state human-designed CAs generally assumes a specific role (or its subcomponent) that is vital to mechanisms of replication. In binary CAs, in the case of the Outlier rule, it is not clear if each of the clusters carries a role that is specific to the assembly of a replicating formation. The clusters seem to be different, but among equals, and perhaps lack such specificity because of the constraints imposed by the shared cell-level updating rule. Instead, processes on a higher-scale level emerge from interactions among clusters in proximity and in turn support the continued existence and temporal evolution of the clusters. The Outlier rule is unusual in that some of its emergent processes self-repeat, embodied in self-replicating formations.

Interestingly, the larger complex generated by the Outlier rule, for example, in Figure 1(c), presents a boundary shape that superficially resembles the “loop”-shaped self-replicators, specifically, a rectangle with a protruding arm, as in Langton’s loops and evoloops. Whether this resemblance is coincidental or substantial warrants further investigation. We suspect that the similarities are related to shared topological constraints intrinsic to 2-D rectangular grids. However, the complex generated by the Outlier rule is not self-replicating as a whole. Furthermore, self-replicating structures on the scale of complexes or higher look unlikely to occur, even on an infinite grid, as the complex expands monotonically on all four sides and the bounded region left behind looks semichaotic.

The Outlier is the only rule that can generate replicating formations among the few hundred thousand rules we examined. Yet its composition looks irregular and arbitrary, which begs the question of how common similarly capable rules are in \mathcal{R} , the space of all rotationally symmetric 2-D binary CA rules. We performed single-point mutations on its genotype, by flipping the live/dead rule result of one neighborhood in Figure 2 at a time, and found no such capability in any of the mutations. It appears that the Outlier is unique, at least in its immediate adjacent rule space. This, of course, helps very little in answering the question. Nevertheless, the fact that nontrivial emergent behavior occurs on multiple scales in a simple binary CA is intriguing, and the author hopes it will contribute to our understanding of self-replicating structures in CAs.

Acknowledgments

The author thanks Bert Chan and Hiroki Sayama for their encouragement to write this down before venturing further, William P. Cavendish for spotting the period 4 spinner, the reviewer for being remarkably thorough and constructive, and all of them for their valuable comments and suggestions.

References

- Bar-Yam, Y. (2003). *Dynamics of complex systems*. Westview Press.
- Bedau, M. A., McCaskill, J. S., Packard, N. H., Rasmussen, S., Adami, C., Green, D. G., Ikegami, T., Kaneko, K., & Ray, T. S. (2000). Open problems in Artificial Life. *Artificial Life*, 6(4), 363–376. <https://doi.org/10.1162/106454600300103683>, PubMed: 11348587
- Chou, H. H., & Reggia, J. A. (1997). Emergence of self-replicating structures in a cellular automata space. *Physica D: Nonlinear Phenomena*, 110(3–4), 252–276. [https://doi.org/10.1016/S0167-2789\(97\)00132-2](https://doi.org/10.1016/S0167-2789(97)00132-2)
- Langton, C. G. (1984). Self-reproduction in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1–2), 135–144. [https://doi.org/10.1016/0167-2789\(84\)90256-2](https://doi.org/10.1016/0167-2789(84)90256-2)
- Lehman, J., & Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In S. Bullock, J. Noble, R. Watson, & M. A. Bedau (Eds.), *Artificial Life XI: Proceedings of the eleventh international conference on the Synthesis and simulation of living systems* (pp. 329–336). MIT Press.

Sayama, H. (1999). A new structurally dissolvable self-reproducing loop evolving in a simple cellular automata space. *Artificial Life*, 5(4), 343–365. <https://doi.org/10.1162/106454699568818>, PubMed: 10829086

Sayama, H., & Nehaniv, C. L. (2024). *Self-reproduction and evolution in cellular automata: 25 years after evolooops*. ArXiv. <https://arxiv.org/abs/2402.03961>

von Neumann, J. (1966). *Theory of self-reproducing automata* (A. W. Burks, Ed.). University of Illinois Press.

Appendices

Appendix 1: Reproducible Programming

The author thanks Keith Y. Patarroyo for independently reproducing the seed run in Figure 4. His procedures are detailed here. The Outlier rule can be encoded in the RLE format, listed here in Base64 as

```
ERETQB4eHWkQ7xD4eYZosBQZFixOBHmtFeehExrKVhURLRAqGxeIISO1JYZP6DRi69rop7TQckvWTIag7kAS8g
```

To seed the reproduced run, paste the following data into the public LifeViewer at <https://lazyslug.com/lifeviewer/>:

```
x=3, y=3, rule=MAPERETQB4eHWkQ7xD4eYZosBQZFixOBHmtFeehExrKVhURLRAqGxeIISO1JYZP6DRi69rop7TQckvWTIag7kAS8g$bo$3o$2bo!$
```

It is recommended to turn on “Auto.” The seed configuration on the second line can be modified for further experimentation.

Appendix 2: Period 4 Spinner

The only pure (free from production of extraneous “debris”) periodically occurring cluster observed thus far is the period 4 spinner shown in Figure A1. From random or seed initial conditions, collisions of shape-shifting clusters occasionally result in this spinner, which is usually destroyed after a few periods by collisions with other nearby clusters.

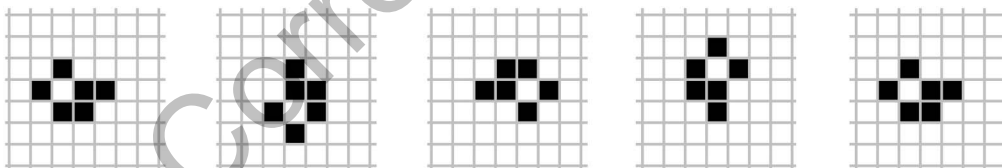


Figure A1. Consecutive updates of a period 4 spinner.