ISSN: 3067-2317
Research Article
World Journal of Artificial Intelligence and Robotics Research

# Three-Colors (BGW or Black-Gray-White) Cellular Automata and Free Will for Artificial Machines

**Luis F. Copertari * & Gloria V. Reyna-Barajas**

[1]*Computer Engineering. Autonomous University of Zacatecas (UAZ). Zacatecas, México*
[2]*Psychology Department. Autonomous University of Zacatecas (UAZ). Zacatecas, México*

***Corresponding author:** Luis F. Copertari, Computer Engineering. Autonomous University of Zacatecas (UAZ). Zacatecas, México.

**Submitted:** 23 January 2026 **Accepted:** 02 February 2026 **Published:** 07 February 2026

***Citation:*** *Copertari, L. F., & Reyna-Barajas, G. V. (2026). Three-Colors (Bgw or Black-Gray-White) Cellular Automata and Free Will for Artificial Machines. Wor Jour of Arti inte and Rob Res, 3(1), 01-11.* https://doi.org/10.63620/MKWJAIRR.2026.1026

***Abstract***

*Instead of the usual 256 rules Cellular Automata (CA) using two bits (black and white), a different type of CA having three symbols ("trits") and 19,683 possible rules using three different colors (black or 0, gray or 1, and white or 2) is proposed. A score system (with four scores) is discussed, which allows to classify each universe for each different rule into "Blank", "Pattern", "One Cycle", "Complex" or "Unclassified". This is important because otherwise the overwhelming number of possible universes would make the results intractable. Also, a brief but wide literature review on CA is conducted. The constraint for "allowing" machines to have "free will", consisting in giving them or denying them stochastic behavior is proposed. Finally, considering the recent advancements concerning Large Language Models (LLMs), the conclusion derived from CA regarding their "free will" is discussed for the case of LLMs and the achievement of Artificial General Intelligence (AGI) in the case of LLMs.*

**Keywords:** Cellular Automata, Score System, Trits, Free will, Large Language Models.

## Introduction

Interacting with cellular automata is like playing God. The world in which the cellular automata exist may be very simple, but that is due to the limited nature of human consciousness and existence. Consider there to exist a world of ten spaces for a given color of either black, gray, or white (a "trit" instead of a "bit"). Also, for this world to be finite yet unlimited, let us have the rightmost position linked to the leftmost position and vice versa. One would think individual "trits" would not notice such universe not to be infinite in nature due to their very limited existence. As time goes by, the colors in the grid change according to the rules set by the immediate past behavior. Only in a God's like eye view of the grids as they change while time passes by gives an idea of how dull or beautifully complex any given universe is.

Also, a given and finite set of rules for the evolution of the alternative universes that would rise make it clear God's work is not necessarily impossible. There would be 39 = 19,683 possible universes or realities to be created. Finally, a scoring model to automatically classify these universes makes this BGW environment treatable. Although Wolfram thoroughly discusses cellular automata, he does not consider the BGW variant we are working with here and, more importantly, the scoring model used. Nevertheless, the cellular automata discussed in Wolfram's work allows for considerable number, possibilities, uses and interpretations of cellular automata [1].

This paper presents a novel way to classify the behavior of cellular automata to find useful rules that can be applied in several scientific and engineering endeavors. Also, instead of the usual two-bit cellular automata, the three-bits ("trits") cellular automata variation is introduced and the exponential growth in its possible rules explored. Cellular automata can be used in different applications. A brief but wide literature review was conducted on the creation, uses and applications of cellular automata.

Typically, cellular automata are used as special algorithms and models in a wide set of applications. Cellular automata serve as powerful algorithms and models because their core principles (discreteness, locality, and parallelism) perfectly mimic the dy-

namics found in many complex natural and social systems [2, 3]. As models, they distill phenomena like fluid flow, forest fire spread, or chemical reaction-diffusion into simple rules governing local cell interactions [4-7]. This method allows researchers to study emergent, macroscopic patterns (like turbulence or a wildfire's perimeter) that arise from tiny, microscopic decisions, often revealing insights that are difficult to capture with continuous mathematics like differential equations [8-10]. Their simplicity makes them computationally efficient and highly intuitive for visualization, making them excellent tools for simulating and predicting system behavior [11-15].

The utility of CA as special algorithms stems directly from their inherent parallelism. Since the transition rule for every cell is applied simultaneously and depends only on its local neighborhood, a CA naturally maps to parallel hardware architectures [16, 17]. This characteristic is exploited in areas like image processing, where local pixel filters can be modeled as CA rules for rapid tasks like noise reduction or edge detection. Furthermore, in computer science theory, the proof that certain CA (like Conway's Game of Life) are Turing Complete establishes them as fundamental paradigms for computation itself, demonstrating that universal computing power can emerge from minimal, localized rules [18].

In essence, CA bridge the gap between simple rules and complex outcomes. Their ability to generate intricate patterns, from the self-replicating structures in biology [19, 20] to the formation of traffic jams on a highway, makes them indispensable across disciplines. They provide a non-linear, bottom-up modeling approach, contrasting with traditional top-down methods. Whether employed to design efficient encryption algorithms in cryptography or to explore the theoretical boundaries of complexity in theoretical physics, CA remain a remarkably versatile tool for understanding the universe through computation [21-25].

The combination of cellular automata and simulation is also another popular application. The marriage of cellular automata and simulation is a particularly potent application because CA are, by their very definition, discrete simulation engines. They provide a powerful framework for modeling systems where local interactions drive global, emergent behavior, which is difficult to capture with continuous mathematics [26-29]. For instance, simulating traffic flow involves simple rules for each car's movement, and the CA framework naturally reveals complex phenomena like the spontaneous formation and dissolution of traffic jams. Similarly, in environmental modeling, simulating the spread of a forest fire or an epidemic relies on the CA grid to track the state of individual cells (tree/fire, healthy/infected) and apply transition rules over discrete time steps, offering a computationally efficient and visually intuitive method for predicting system evolution under varying initial conditions [30-33].

Some artificial intelligence techniques are sometimes combined with cellular automata. The combination of CA and Artificial Intelligence (AI), particularly Machine Learning (ML), creates powerful hybrid systems. CA rules, which dictate the complex evolution of the system, can be too intricate to design manually. AI techniques, such as Genetic Algorithms or Deep Reinforcement Learning (DRL), are employed to discover or optimize these transition rules to achieve a desired global outcome. For example, a DRL agent might be tasked with finding the optimal CA rules to simulate a perfect self-healing material or to generate a specific complex pattern efficiently. Conversely, CA themselves can serve as efficient parallel computational substrates for AI tasks, providing a high-speed, localized processing architecture for tasks like image recognition or simulating physical environments for training AI agents [34, 35].

Cellular automata sometimes are modified to include probabilistic or stochastic behaviors. The inclusion of probabilistic or stochastic behaviors represents a crucial modification to the standard, purely deterministic cellular automata framework, significantly expanding their modeling capabilities [36]. In a stochastic CA, the transition rule is not fixed but involves a probability distribution; for example, a cell might have a 90% chance of adopting a new state and a 10% chance of remaining in its current state, even if the deterministic rule suggests otherwise. This modification is vital for accurately simulating systems where randomness and noise play a significant role, such as the spread of a disease where transmission might depend on random encounters, or complex physical processes like phase transitions, where individual particle decisions are probabilistic. By introducing chance, stochastic CA can generate more realistic and less predictable patterns, making them superior models for natural phenomena with inherent uncertainty [37, 38].

Finally, cellular automata in one, two or three dimensions are created and used. The use of cellular automata across one, two, and three dimensions highlights their versatility and scalability as modeling tools, with the choice of dimensionality directly corresponding to the complexity of the system being simulated. One-dimensional CA (e.g., Wolfram's Rule 30 –which I call rule 31 in a 1 to 256 set of rules; it is rule 30 in a 0 to 255 scale–, or traffic flow models) are the simplest, often used for theoretical studies of complexity, pseudorandom number generation, and linear processes. Two-dimensional CA (e.g., Conway's Game of Life, forest fire models) are the most common, ideal for simulating phenomena that evolve on a planar surface, such as pattern formation, fluid dynamics on a grid, and image processing. Finally, three-dimensional CA extend these capabilities into volumes, allowing for the realistic modeling of complex spatial phenomena, including crystal growth, biological tissue formation, and complex diffusion processes within materials, requiring significantly more computational power but providing richer, volumetric simulation data [39-46].

**Theory**

Typical cellular automata are drawn using two colors: black and white. That means using a binary numerical system with two numerical symbols: 0 and 1. In the case of three-colors based cellular automata, we require a trinary numerical system, that is, a numerical system with three numerical symbols. In this case, these symbols are: 0, 1 and 2. Therefore, the first number in such numerical system is 0 = 0 the second number is 1 = 1, then follows 2 = 2. After that we have 10 = 3, then 11 = 4, followed by 12 = 5. Following we would have 20 = 6, then 21 = 7, then 22 = 8. Then 100 = 9, 101 = 10, 102 = 11, 110 = 12, 111 = 13, followed by 112 = 14. We then would continue counting using 120 = 15, then 121 = 16, then 122 = 17, then 200 = 18, and so on.

We are going to use the previous and right-to-the-previous cell

with these three possible colors (BGW, that is black, gray and white, where 0 = black, 1 = gray and 2 = white), so that there are 32 = 9 combinations for a rule (00, 01, 02, 10, 11, 12, 20, 21, 22) and 39 = 19,683 possible universes (possible number of rules).

Figure 1 shows the arrangement for our BGW cellular automata mechanism. Notice that now instead of having "bits" we have "trits": T0, T1, …, T8. These "trits" can take values of either 0, 1 or 2.

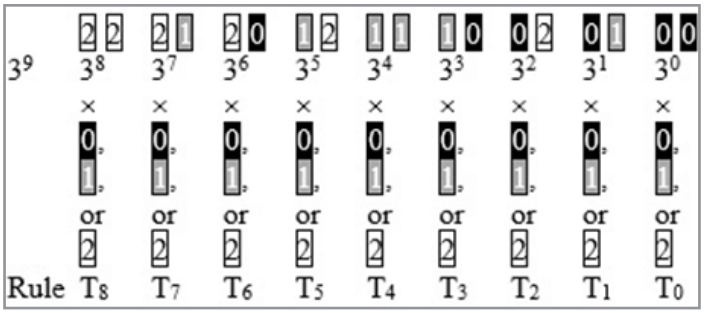


**Figure 1:** BGW cellular automata "trit" (trinary) arrangement.

To find out the rule number, we should do the sum indicated in equation (1).

$$R = T_0\times3^0+T_1\times3^1+T_2\times3^2+T_3\times3^3+T_4\times3^4+T_5\times3^5+T_6\times3^6+T_7\times3^7+T_8\times3^8 \quad (1)$$

The maximum value for R is given as Max(R) = 39-1 = 19,682, where the minimum value is Min(R) = 0. The rule number goes from 1 to R+1, that is, it goes from 1 to 39 = 19,683.

For a given rule number, to find out the components, where R = Rule-1, we can use the following sequence of equations: T0 = R mod 3, T1 = (R div 3) mod 3, then we add another division resulting in T2 = ((R div 3) div 3) mod 3, and so on, where mod indicates obtaining the residual (module) of dividing by three and div means obtaining the integer part of a division by three. The use of the mod operator at the end makes sense because in that way the only possible results for a single "trit" are 0, 1 and 2 (instead of the "bit" used in the usual cellular automata black and white arrangement where the numerical symbols are 0 and 1).

**Testing**

Each rule leads to a given screen result. Each population of cellular automata is arranged in a one-dimensional universe with 256 possible pixels, where the pixel to the right of the rightmost position is the pixel at the leftmost position. It makes no sense to consider the pixel at the left of the leftmost position. This constitutes the horizontal dimension. The vertical dimension is the time dimension. We have possible values for time going from 0 to 511, that is, 512 pixels. For time 0, the initial population of cellular automata is having all black pixels from position 0 to position 253, a gray pixel at position 254 and a white pixel at position 255.

The problem with BGW cellular automata is the huge number of possible universes that require analysis (39 = 19,683). It is possible to have a blank screen as a result, a one cycle screen (a screen with a complex pattern that occurs only once and then vanishes), a pattern screen (the equivalent of a chessboard screen for the usual two-bit cellular automata), a complex screen (a screen showing inherent complexity) or an unclassified screen (a screen that has not fallen in any of the previous four categories). That means we need to use an automatic classification system to filter out the blank, one cycle, or pattern results from the results considered to show inherent complexity.

But how can we classify the results? In a two-bit cellular automata arrangement, that can easily be achieved using a score-based classification system. How can we calculate such score? Assigning a black pixel to the number zero and a white pixel to the number one, we can calculate the percentage of black pixels (B) as indicated in equation (2) and the percentage of white pixels (W) as indicated in equation (3), where bj = 1 if there is a black pixel at position j and bj = 0 if there is a white pixel at position j for a given time t (where there are values for t = 0, …, 511). The squaring in equations (2) and (3) is simply for removing the sign in the expressions given.

$$B = \frac{1}{256}\sum_{j=0}^{255}(b_j - 0)^2 \quad (2)$$

$$W = \frac{1}{256}\sum_{j=0}^{255}(b_j - 1)^2 \quad (3)$$

Notice that equation (4) must always be satisfied.

$$W = 1-B \quad (4)$$

The combined percentage (C) is given as indicated in equation (5).

$$C = BW \quad (5)$$

Substituting W from equation (4) into equation (5) results in a function based on B as indicated by equation (6).

$$f(B) = B(1-B) = B - B^2 \quad (6)$$

Taking the derivative for B from equation (6) and equating to zero leads us to the maximum possible value for B, as shown in equations (7) and (8).

$$f'(B) = 1 - 2B = 0 \quad (7)$$

$$B = ½ = 0.5 \quad (8)$$

Since the maximum value for B is 0.5, from equation (4) we have that the maximum value for W is also 0.5. Thus, the maximum value for the combined score (C), that is, function f(B) is given by equation (9).

$$Max(C) = (½)\times(½) = ¼ = 0.25 \quad (9)$$

Thus, a numerical score (S) given as a value going from 0 to 10 can be calculated according to equation (10).

$$S = 10\times\left(\frac{C}{0.25}\right) = 10\times\left(\frac{BW}{0.25}\right) = 10\times\left(\frac{B(1-B)}{0.25}\right) \quad (10)$$

Extrapolating into our "trit" numerical system, we have that the maximum number for a combined score (C) composed by the percentage of black pixels multiplied by the percentage of gray pixels multiplied by the percentage of white pixels is given as indicated in equation (11).

$$Max(C) = (1/3)\times(1/3)\times(1/3) = (1/27) \quad (11)$$

Let PB be the percentage of black pixels, PG be the percentage of gray pixels and PW be the percentage of white pixels. Then, the score (S) for our "trit" numerical system is given according to equation (12).

$$S = 10 \times \left(\frac{C}{1/27}\right) = 10 \times \left(\frac{P_B P_G P_W}{1/27}\right) \quad (12)$$

However, if PB = 0, then the score would be given by equation (13), if PG = 0 the score would be given by equation (14) and if PW = 0 the score would be given according to equation (15).

$$S = 10 \times \left(\frac{P_G P_W}{1/4}\right), \text{if } P_B = 0 \quad (13)$$

$$S = 10 \times \left(\frac{P_B P_W}{1/4}\right), \text{if } P_G = 0 \quad (14)$$

$$S = 10 \times \left(\frac{P_B P_G}{1/4}\right), \text{if } P_W = 0 \quad (15)$$

A total of four scores is used. The first score (Score 1) is at one quarter of the time dimension going from the top of the screen to the bottom. The second score (Score 2) is at one half the time dimension. The third such score (Score 3) is at three quarters of the time dimension. Finally, the fourth score (Score 4) is at the bottom of the time dimension. According to my experience with two bits cellular automata, a score greater than 6 could be indicative of a complex arrangement [47].

I used the experience obtained with my work on cellular automata of two bits, as well as simply viewing the first results of the arrangement for my universe of "trits". If all four scores are equal to zero, I label the screen as "Blank". Otherwise, if the last score (Score 4) is greater than 9.99, I label the screen as "Pattern". If not, if the first score is greater than zero and either of all three remaining scores are equal to zero, I label the screen as "One Cycle". Otherwise, if Score 1 > 3, Score 2 > 5, Score 3 > 6 and Score 4 > 6, then I label the screen as "Complex". Else, the screen is labeled as "Unclassified".

## Results

The 39 = 19,683 alternative universes are sorted based on the scoring system previously described. Each universe can be classified as: 1) Blank, 2) Pattern, 3) One Cycle, 4) Complex, and 5) Unclassified. Table 1 shows the frequency arrangement for this classification.

**Table 1:** Frequency arrangement for the three-colors (BGW) cellular automata classification system.

| | Frequency | Percentage | Actual Results |
|---|---|---|---|
| Blank | 2451 | 12.45% | 35.98% |
| Pattern | 936 | 4.76% | 13.74% |
| One Cycle | 2127 | 10.81% | 31.22% |
| Complex | 1298 | 6.59% | 19.05% |
| Unclassified | 12871 | 65.39% | 100.00% |
| | 19683 | 100.00% | 6812 |

The column corresponding to Frequency in Table 1 lists the corresponding number of universes fitting any given classification with a total of 19,683 = 39 universes. The relative percentages for these five classification alternatives are shown in the following column. Finally, the percentages for the universes that were classified (without counting the 12,871 unclassified universes) is shown in the Actual Results column of Table 1.

The given frequencies are plotted in Figure 2. Notice that most cases are "Unclassified" (12,871). Of the classified universes, most of them (2,451) are "Blank", and there are 936 universes classified as "Pattern". The interesting ones are "One Cycle" (2,127) and "Complex" (1,298).

Figure 3 shows a pie chart with the relative percentages of the universes that were classified. From Table 1 we can see that most universes are "Blank" or "One Cycle" (35.98% and 31.22%, respectively). The "Pattern" universes (13.74%) are of no interest. The interesting ones should be the ones classified as "Complex" (19.05%). That is, about one every five universes should be of interest ("Complex").

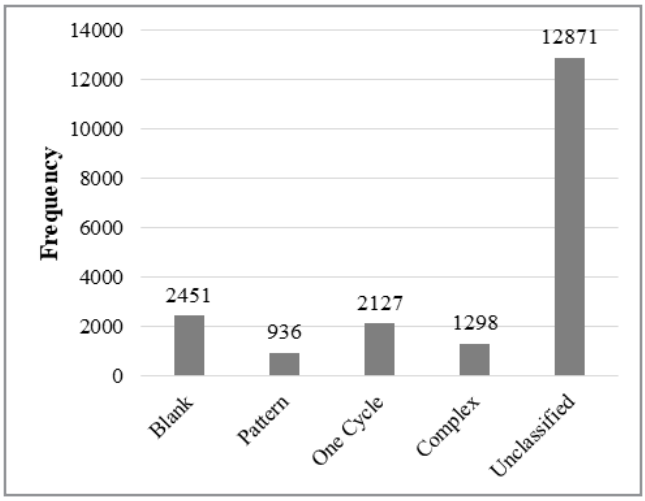


**Figure 2:** Plotting the frequencies in a column chart.

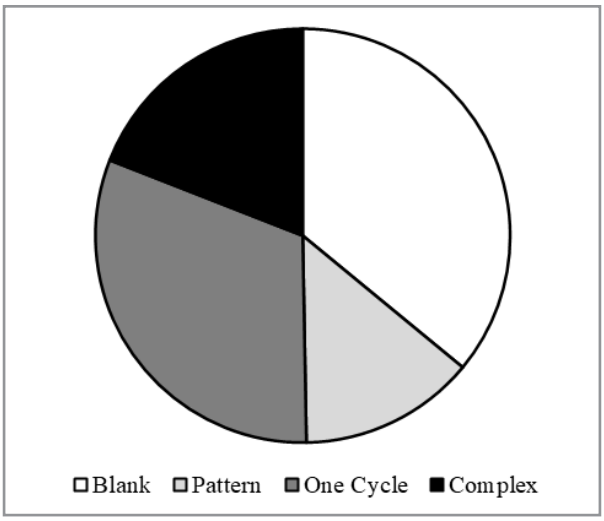


**Figure 3:** Relative percentages for the actual results obtained.

## Discussion and Conclusion

The question, of course, is whether the classification scheme based on four scores works. To get an intuitive hint about this problem we should look at the universes sorted out this way. However, even if we only consider the 6,812 classified universes, that is still too much to explore in detail. Thus, I am going to consider a subset of universes from the total of 39 = 19,683 possible ones: the universes going between rules 1 and 256.

In that case, "Complex" universes are: 89, 137, 146, 155, 194, 202 and 251. Not any one of these were wrongly classified. For "One Cycle" universes, these were: 31, 40, 43, 49, 51, 83, 92, 101, 112, 121, 124, 130, and 205. As we can see, only 4 "One Cycle" universes (the ones striped) were not properly classified. What about "Blank" universes? Out of the total of 49 "Blank" universes classified, only universes 164 and 230 were not actually "Blank" universes (they look more like a set of horizontal lines of different colors). Finally, for "Pattern" universes between rules 1 and 256, there were no universes having a patterned (chessboard like) look. Consequently, the scheme based on my four-score system seems to work properly, at least when considering the universes that can be classified.

What is the total number of different arrangements for our 256 "trits" as time goes by? There is a maximum of 3256 possible combinations, after which, if we are following deterministic rules, the universe must start repeating itself. However, we can consider a smaller case. How about universes with only 10 "trits"? In that case, for a deterministic set of rules, after a maximum of 310 = 59,049 time-steps, the universe must start repeating itself.

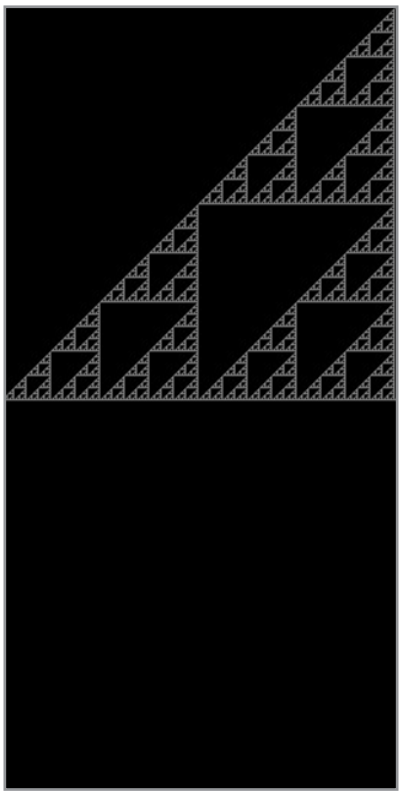

**Figure 4:** Rule 31 universe having 256 "trits".

Is the set of rules shown in Figure 1 that are possible all that varying? To answer that question we are going to consider "One Cycle" rule 31, "Complex" rule 89 and "Complex" rule 146 (remember that in a "bit" based universe, rule 146 is always the one mentioned as showing incredible complexity and beauty) [48].

Figure 4 shows rule 31. We can see it is a triangular fractal, which clearly is a case of a "One Cycle" universe in which there is symmetrical complexity only at the beginning of such universe.

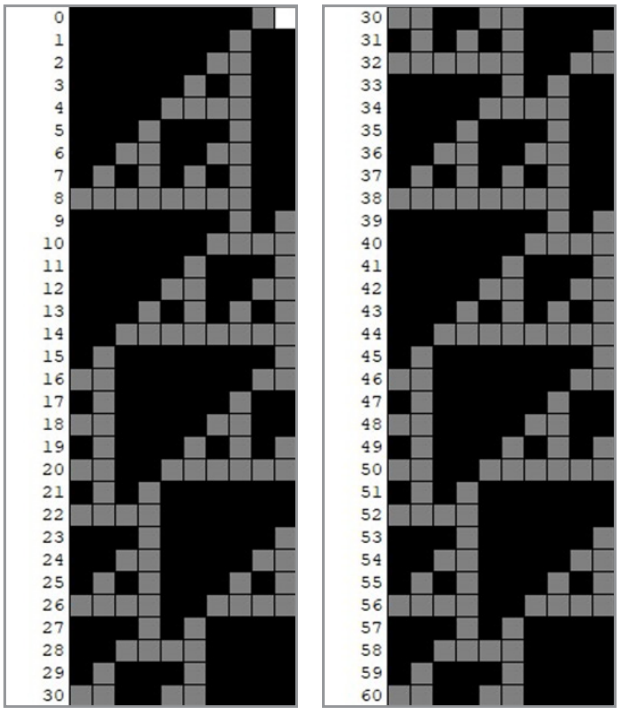


**Figure 5:** Rule 31 (000001010) universe having 10 "trits" for times between 0 and 30 and for times between 30 and 60.

To further explore this universe, we are going to consider the same rule (31), but for a 10 "trits" universe, as shown in Figure 5. We can see in Figure 5 that time step 33 is equal to time step 3, so that after time step 33, the arrangement starts repeating itself. Certainly, it only lasted 33 steps, not the maximum possible of $3^{10} = 59{,}049$ time steps!

What about a "Complex" arrangement. We are going to consider rule 89, which can be seen in Figure 6. The arrangement shown in Figure 6 has complexity, but is somewhat boring, since it consists of the collection of two patterned arrangements that evolve through time.

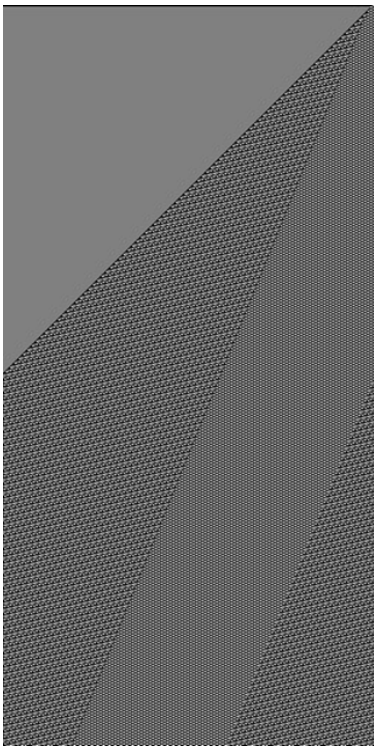

**Figure 6:** Rule 89 universe having 256 "trits".

Figure 7 shows the same rule when applied to a universe of 10 “trits”. Time step 33 is equal to time step 8, so after step 33 the “trits” start repeating themselves.

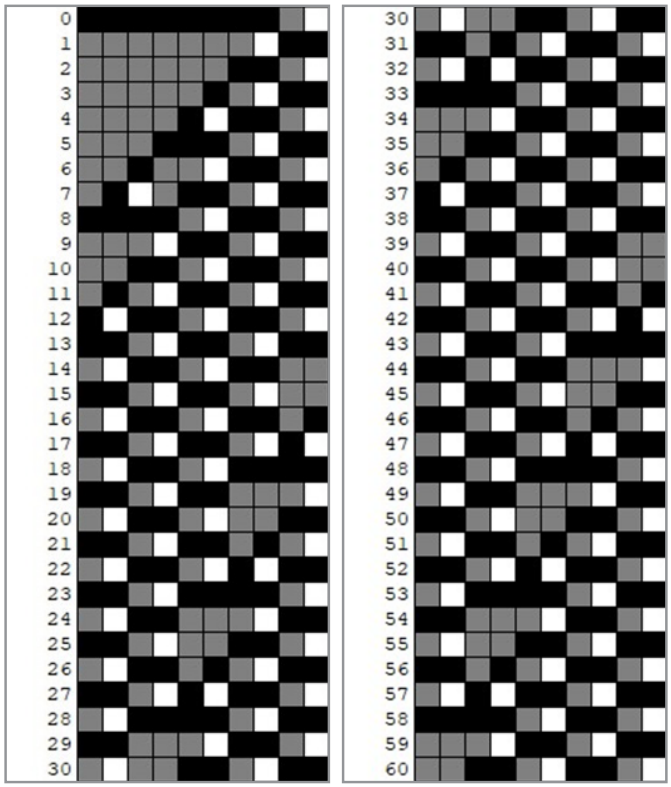

**Figure 7:** Rule 89 (000010021) universe having 10 “trits” for times between 0 and 30 and for times between 30 and 60.

Figure 8 shows rule 146 (000012101). The arrangement for this rule exhibits true beauty. However, rule 146 in a “trits” universe is not the same one as rule 146 in a “bits” universe. We can see in Figure 8 that the complexity shown continues for several time steps as the arrangement develops. Also, notice that in a “trits” universe, there are three colors present: black, gray, and white, and not just black and white.

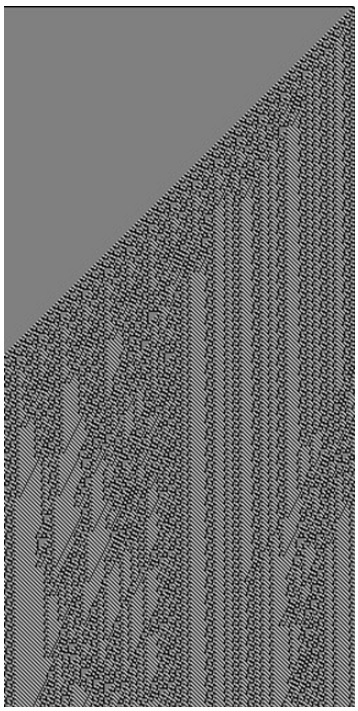

**Figure 8:** Rule 146 universe having 256 “trits”.

Figure 9 shows the time steps in a 10 “trits” arrangement for rule 146.

Rule 146 is certainly more complex, because in a 10 “trits” universe following rule 146, it is only at time step 55 that step 25 starts repeating itself. Compared to the previous two cases, rule 146 lasts longer without repetitions.

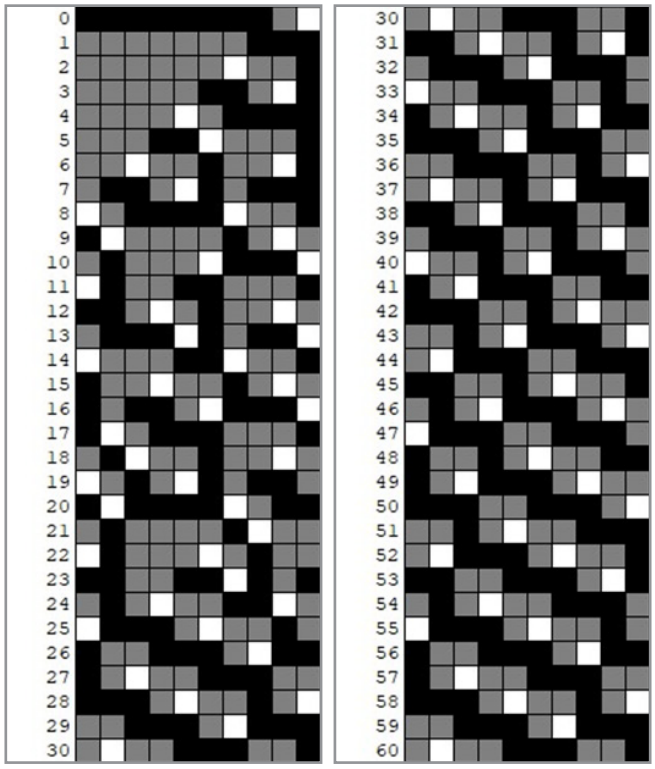

**Figure 9:** Rule 146 (000012101) universe having 10 "trits" for times between 0 and 30 and for times between 30 and 60.

What does free will have to do with cellular automata? How the findings related to cellular automata can be generalized to artificial machines with natural intelligence (that is, showing Artificial General Intelligence or AGI)?

In the CA world, if the rules are deterministic, then it is not possible for the system to be anything more than a mindless machine. However, if we add a probabilistic or stochastic behavior to the CA system (as it would be the case for stochastic CA), then the possibility for "free will" to emerge arises.

Large Language Models (LLMs) have impressed both experts and common people. They seem to exhibit some intelligent behavior. So, it is important in our CA discussion to consider LLMs.

The relationship between Large Language Models and the pursuit of Artificial General Intelligence (AGI) has become one of the most significant scientific debates of the mid-2020s. At its core, the discussion centers on whether the current trajectory of scaling up transformer-based architectures can eventually result in a machine that possesses the same broad, adaptable, and autonomous cognitive abilities as a human [49, 50]. While the achievements of these models in creative writing, coding, and complex reasoning are undeniable, critics and proponents remain divided on whether these systems are merely sophisticated "statistical parrots" or the first genuine sparks of a general intelligence [51].

Large Language Models operate primarily through the prediction of sequential data, a process that allows them to internalize vast amounts of human knowledge encoded in text [52, 53]. This has led to the emergence of capabilities that were not explicitly programmed, such as the ability to translate languages or solve logic puzzles. Some researchers argue that by training on multimodal data including video, audio, and sensorimotor inputs, these models are developing a "world model" that transcends simple word associations. They suggest that AGI is an emergent property of sufficient scale and data diversity, where the model eventually grasps the underlying causal structures of reality rather than just the linguistic patterns used to describe them.

However, a significant portion of the academic community maintains that Large Language Models face fundamental architectural barriers that prevent them from reaching true AGI. One primary concern is the "symbol grounding problem", which refers to the fact that these models manipulate abstract symbols without having a physical or experiential connection to what those symbols represent. A model can describe the taste of a peach or the physics of a falling glass perfectly, yet it has never tasted anything nor interacted with gravity. This lack of embodiment leads to "brittleness", where a system might solve a Ph.D.-level physics problem but fail at a task requiring basic common sense or real-world intuition.

What do CA teach us about LLMs? Clearly, the trick in making LLMs to exhibit "free will" lies in allowing them to behave probabilistically instead of deterministically. But a stochastic behavior in LLMs beyond the probabilistic nature of the way in which response words are connected would mean destroying their deterministic rule-based "reasoning" process [54]. Thus, the only way to make LLMs exhibit "free will" is by connecting them to their human users [55]. However, that is already happening, since LLMs answer to human queries. That is the reason why LLMs seem intelligent. It is because we provide the intelligent input to the system.

Furthermore, the quest for AGI requires more than just high-level pattern recognition; it necessitates autonomous goal-setting, long-term memory, and the ability to learn from a single expe-

rience [56]. Current models are largely static after their training phase and lack a persistent, evolving identity. While "agentic" frameworks have begun to wrap language models in loops that allow them to use tools and plan multi-step tasks, these are often seen as external scaffolds rather than internal cognitive shifts. The debate in 2026 has shifted toward hybrid systems that might combine the intuitive, associative power of neural networks with the rigorous, logical frameworks of symbolic AI. Whether the path to AGI is a straight line of scaling or a mountain range requiring entirely new architectural "base camps", the journey continues to redefine our understanding of both artificial and human intelligence.

## References


1. Wolfram, S. (2002). A new kind of science. Wolfram Media.
2. Malarz, K., Wołoszyn, M., & Kułakowski, K. (2020). Towards the Heider balance with a cellular automaton. Physica D: Nonlinear Phenomena, 411, 132556. https://doi.org/10.1016/j.physd.2020.132556
3. Padovani, D., Neto, J. J., & Massa Cereda, P. R. (2018). Modeling pedestrian dynamics with adaptive cellular automata. Procedia Computer Science, 130, 1120–1127. https://doi.org/10.1016/j.procs.2018.04.161
4. González Tejada, I., Monteiro-Alves, R., Morán, R., & Toledo, M. Á. (2021). Cellular automata modeling of rockfill dam failure caused by overtopping or extreme throughflow. Engineering Structures, 245, 112933. https://doi.org/10.1016/j.engstruct.2021.112933
5. Sahila, A., Fiorese, A., Perello, N., Trucchia, A., & Pagnini, G. (2025). Patterns of burned area by a cellular-automata fire simulator: The role of microscale wind field. Communications in Nonlinear Science and Numerical Simulation, 151, 107112.
6. Bailey, K., Sunny, S., Mathews, R., & Malik, A. (2025). Impact of dynamic recrystallization in laser shock peening predicted via a coupled cellular automaton–finite element model. Manufacturing Letters, 44, 364–375. https://doi.org/10.1016/j.mfglet.2025.01.012
7. Traka, K., Sedighiani, K., Bos, C., Galan Lopez, J., Angenendt, K., Raabe, D., & Sietsma, J. (2021). Topological aspects responsible for recrystallization evolution in an IF-steel sheet: Investigation with cellular automaton simulation. Computational Materials Science, 198, 110643. https://doi.org/10.1016/j.commatsci.2021.110643
8. Luo, L., Liu, J., Fei, J., Xia, H., & Xie, W. (2021). Cellular automata ray tracing in flow field near the optical window of the optical dome. Results in Physics, 25, 104319. https://doi.org/10.1016/j.rinp.2021.104319
9. Jiang, W., Wang, F., Fang, L., Zheng, X., Qiao, X., Li, Z., & Meng, Q. (2021). Modelling of wildland–urban interface fire spread with the heterogeneous cellular automata model. Environmental Modelling & Software, 135, 104895. https://doi.org/10.1016/j.envsoft.2020.104895
10. Zeng, Y., Jiang, T., Tang, J., Zhu, J., Shen, X., & Ling, Y. (2025). Grain refinement mechanism of nitrogen ion implantation in 7075-T651 aluminum alloy: Insights from the cellular automaton method. Journal of Materials Research and Technology, 38, 3218–3329.
11. Bozsóki, I., Illés, B., & Géczy, A. (2025). Numerical simulation of Sn grain growth in composite solder joints using a modified cellular automaton model. Results in Engineering, 26, 100985.
12. Viore, V., & Istiono, W. (2025). Combining the cellular automata and marching square to generate maps. Jurnal RESTI, 9(2), 416–424. https://doi.org/10.29207/resti.v9i2.XXXX
13. Gui, B., Bhardwaj, A., & Sam, L. (2025). Cellular automata models for simulation and prediction of urban land use change: Development and prospects. Artificial Intelligence in Geosciences, 6, 1–16.
14. Liu, H., Kitamura, M., Briffod, F., & Shiraiwa, T. (2025). A cellular automata-based crystal plasticity analysis of slip activity in additively manufactured Ti-6Al-4V during dwell fatigue. Journal of Alloys and Compounds, 1041, 183867.
15. Dang, P., Zhu, J., Pirasteh, S., Li, W., You, J., Xu, B., & Liang, C. (2021). A chain navigation grid based on cellular automata for large-scale crowd evacuation in virtual reality. International Journal of Applied Earth Observation and Geoinformation, 103, 102507. https://doi.org/10.1016/j.jag.2021.102507
16. Vivas, A., & Sanabria, J. (2020). A microservice approach for a cellular automata parallel programming environment. Electronic Notes in Theoretical Computer Science, 349, 119–134. https://doi.org/10.1016/j.entcs.2020.09.010
17. Bakhteri, R., Cheng, J., & Semmelhack, A. (2020). Design and implementation of cellular automata on FPGA for hardware acceleration. Procedia Computer Science, 171, 1999–2007. https://doi.org/10.1016/j.procs.2020.04.214
18. Yamashita, T., Isokawa, T., Peper, F., Kawamata, I., & Hagiya, M. (2020). Turing-completeness of asynchronous non-camouflage cellular automata. Information and Computation, 274, 104539. https://doi.org/10.1016/j.ic.2020.104539
19. Ortigoza, G., Brauer, F., & Neri, I. (2020). Modelling and simulating Chikungunya spread with an unstructured triangular cellular automaton. Infectious Disease Modelling, 5, 197–220. https://doi.org/10.1016/j.idm.2020.02.001
20. Hillmann, A., Crane, M., & Ruskin, H. J. (2020). Assessing the impact of HIV treatment interruptions using stochastic cellular automata. Journal of Theoretical Biology, 502, 110376. https://doi.org/10.1016/j.jtbi.2020.110376
21. Chamoli, A., Ahmed, J., Alam, M. A., & Alankar, B. (2024). Image encryption using cellular automata. Nanotechnology Perceptions, 20(S14), 4158–4164.
22. Stănică, G. C., & Anghelescu, P. (2025). Multi-layer cryptosystem using reversible cellular automata. Electronics, 14, 2627–2648. https://doi.org/10.3390/electronics14132627
23. Vijitha, P. H., & Phamila, Y. A. V. (2025). A novel image encryption and decryption scheme integrating two-way chaotic maps, iterative cellular automata, and online tessellation automata. Results in Engineering, 27, 105673–105687.
24. Bhardwaj, R., & Bhagat, D. (2018). Two-level encryption of gray scale image through 2D cellular automata. Procedia Computer Science, 125, 855–861. https://doi.org/10.1016/j.procs.2017.12.110
25. Yogi, B., & Khan, A. K. (2025). CAChaIoT: Hybrid lightweight image encryption for IoT using cellular automata and chaotic maps. Franklin Open, 12, 100361–100372.
26. Gui, B., Bhardwaj, A., & Sam, L. (2025). Simulating urban expansion as a nonlinear constrained evolution process: A hybrid logistic–Monte Carlo cellular automata framework. Chaos, Solitons & Fractals, 199, 116938–116955.

27. Romitti, G. S., Liberos, A., Termenón-Rivas, M., Barrios-Álvarez de Arcaya, J., Serra, D., Romero, P., Calvo, D., Lozano, M., García-Fernández, I., Sebastian, R., & Rodrigo, M. (2025). Implementation of a cellular automaton for efficient simulations of atrial arrhythmias. Medical Image Analysis, 101, 103484–103495.
28. Chai, Q., Fang, C., Hu, J., Xing, Y., & Huang, D. (2020). Cellular automaton model for the simulation of laser cladding profile of metal alloys. Materials & Design, 195, 109033–109042.
29. Owusu, P. A., Leonenko, V. N., Mamchik, N. A., & Skorb, E. V. (2019). Modeling the growth of dendritic electroless silver colonies using hexagonal cellular automata. Procedia Computer Science, 156, 43–48.
30. Sonnenschein, T. S., Yuan, Z., Khan, J., Kerckhoffs, J., Vermeulen, R. C. H., & Scheider, S. (2025). Hybrid cellular automata-based air pollution model for traffic scenario microsimulations. Environmental Modelling & Software, 186, 106356–106368.
31. Purnomo, D. M. J., Bonner, M., Moafi, S., & Rein, G. (2021). Using cellular automata to simulate field-scale flaming and smouldering wildfires in tropical peatlands. Proceedings of the Combustion Institute, 38, 5119–5127.
32. Brasil Cavalcante, A. L., Parreira de Faria Borges, L., da Costa Lemos, M. A., Muniz de Farias, M., & Sampaio Carvalho, H. (2021). Modelling the spread of COVID-19 in the capital of Brazil using numerical solution and cellular automata. Computational Biology and Chemistry, 94, 107554–107562.
33. Yu, J., Hagen-Zanker, A., Santitissadeekorn, N., & Hughes, S. (2021). Calibration of cellular automata urban growth models from urban genesis onwards: A novel application of Markov chain Monte Carlo approximate Bayesian computation. Computers, Environment and Urban Systems, 90, 101689–101702.
34. Constantin, D., & Bălcău, C. (2025). A cellular automata-based crossover operator for binary chromosome population genetic algorithms. Applied Sciences, 15, 8750–8781.
35. Enescu, A., Dumitru, D., Andreica, A., & Dioşan, L. (2020). Unsupervised edge detector based on evolved cellular automata. Procedia Computer Science, 176, 470–479.
36. Cirillo, E. N. M., Nardi, F. R., & Spitoni, C. (2021). Phase transitions in random mixtures of elementary cellular automata. Physica A: Statistical Mechanics and Its Applications, 573, 125942–125957.
37. Wetterich, C. (2021). Probabilistic cellular automata for interacting fermionic quantum field theories. Nuclear Physics B, 963, 115296–115320.
38. García Otero, J., Belmonte-Beitia, J., & Jiménez-Sánchez, J. (2025). Exploring neuroblastoma's cellular microenvironment: A novel approach using cellular automata to model Celyvir treatment. Computers in Biology and Medicine, 188, 109782–109812.
39. Krawczyk, M. J., & Kułakowski, K. (2025). A cellular automaton towards structural balance: Long cycles of link dynamics. Journal of Computational Science, 92, 102712–102719.
40. Podolski, P., & Nguyen, H. S. (2021). Cellular automata in COVID-19 prediction. Procedia Computer Science, 192, 3370–3379.
41. Matsumoto, S., Morikawa, K., Goda, K., Goto, K., & Fukuyama, H. (2025). Cellular automaton modeling of the local corrosion of $ZrO_2$–C refractories at slag/metal interfaces. Open Ceramics, 23, 100831–100844.
42. Ding, Y., Cui, Y., Rao, W., & Liu, J. (2025). Cellular automata modeling on uniform corrosion behavior of solid copper in gallium-based liquid metals. Corrosion Communications, 17, 66–76.
43. Zhang, Y., Zhou, J., Yin, Y., Shen, X., & Ji, X. (2021). Multi-GPU implementation of a cellular automaton model for dendritic growth of binary alloy. Journal of Materials Research and Technology, 14, 1862–1872.
44. Su, F., Liu, W., & Wen, Z. (2020). Three-dimensional cellular automaton simulation of austenite grain growth of Fe–1C–1.5Cr alloy steel. Journal of Materials Research and Technology, 9(1), 180–187.
45. Dissanayake, S., Salehi, H., Zigan, S., Deng, T., & Bradley, M. (2025). Modelling segregation phenomena in large industrial silos: A cellular automaton approach. Powder Technology, 459, 120998–121012.
46. Hua, L., Hu, Z., Wu, W., Mei, Z., Cao, M., Peng, L., & Liu, C. (2025). Investigation of marine corrosion characteristics of 10CrNiCu steel subjected to stress fields using an improved 3D cellular automata modeling. Materials & Design, 250, 113606–113624.
47. Copertari, L. F. (2016). Free will for natural machines with artificial intelligence. Open Access Library Journal, 3, e2355. https://doi.org/10.4236/oalib.1102355
48. Kurzweil, R. (2005). The singularity is near: When humans transcend biology. Penguin Books.
49. Qu, Y., Du, P., Che, W., Wei, C., Zhang, C., Ouyang, W., Bian, Y., Xu, F., Hu, B., Du, K., Wu, H., Liu, J., & Liu, Q. (2024). Promoting interactions between cognitive science and large language models. The Innovation, 5(2), 100579–100580.
50. Han, S. J., Ransom, K. J., Perfors, A., & Kemp, C. (2024). Inductive reasoning in humans and large language models. Cognitive Systems Research, 83, 101155–101182.
51. Ilić, D., & Gignac, G. E. (2024). Evidence of interrelated cognitive-like capabilities in large language models: Indications of artificial general intelligence or achievement? Intelligence, 106, 101858–101868.
52. Karlsen, S. S., Yamin, M. M., Hashmi, E., Katt, B., & Ullah, M. (2026). Securing large language models: A quantitative assurance framework approach. Journal of Information Security and Applications, 97, 104351–104369.
53. Zheng, Z., Wang, Y., Huang, Y., Song, S., Yang, M., Tang, B., Xiong, F., & Li, Z. (2025). Attention heads of large language models. Patterns, 6(2), 1–20.
54. Xu, F., Hao, Q., Shao, C., Zong, Z., Li, Y., Wang, J., Zhang, Y., Wang, J., Lan, X., Gong, J., Ouyang, T., Meng, F., Yan, Y., Yang, Q., Song, Y., Ren, S., Hu, X., Feng, J., Gao, C., & Li, Y. (2025). Toward large reasoning models: A survey of reinforced reasoning with large language models. Patterns, 6(10), 1–30.
55. Hu, J., Bi, W., Zhou, Y., Zheng, J., Zhang, S., Chu, H., Wang, L., & Li, Q. (2026). Plang: Efficient prompt engineering language for blending natural language and control flow in large language models. Expert Systems with Applications, 300, 130118–130131.
56. Hui, B. P. H., Lau, C. L., Sun, R., LC, R., Hendra, L. B., & Kogan, A. (2025). Decoding moral responses in AI: A

quantitative analysis of large language models. Computers in Human Behavior Reports, 20, 100854–100867.