What Makes the Game of Life Special?

Peter D. Turney*

Submitted to: Artificial Life. **Category:** Fast Track Article. **Date:** October 15, 2020.

Abstract

Conway's Game of Life (GoL) is the best-known cellular automaton. It is a classic model of emergence and self-organization, it is Turing-complete, and it can simulate a universal constructor. GoL belongs to the set of semi-totalistic cellular automata, a family with 262,144 members. In such a large family, what makes GoL stand out? Packard and Wolfram (1985) proposed a set of four qualitative classes for cellular automata. Eppstein (2010) proposed four objective classes. Both classification systems are interesting, but neither can distinguish GoL from its many near relatives. Instead of a classification system, we present a simple and objective metric that assigns a numerical score to all semi-totalistic cellular automata. The metric uses four properties to distinguish GoL from its relatives: GoL is evenly balanced between growth and decay, measured by (1) the number of living cells and (2) the area covered by the living cells; (3) the rule for GoL is simple; and (4) GoL has low density, measured by living cells per area. We combine these four properties in a simple mathematical expression. When we score the 262,144 semi-totalistic cellular automata with this formula, GoL achieves the maximum score. We show that the metric favours Turing-complete automata, although it was not designed to do so, which suggests it captures the spirit of the Game of Life.

Keywords: Game of Life, cellular automata, metrics for automata, Turing-completeness, semi-totalistic, outer-totalistic.

In celebration of the 50th Anniversary of the Game of Life. In memory of John Horton Conway.

Fast Track Article

(about 2,000 words)

* Ronin Institute, 127 Haddon Place, Montclair, NJ 07043-2314, USA,
peter.turney@ronininstitute.com, 819-661-4625.

1 Introduction

The *Game of Life* (GoL) is a solitaire game invented by John Conway and introduced to the world by Martin Gardner in *Scientific American* [3]. It is played on a potentially infinite, two-dimensional grid of square cells. Each cell is either *dead* (state 0) or *alive* (state 1). The state of a cell changes with time, based on the states of its eight nearest neighbours (called the *Moore neighbourhood*). Time passes in a series of discrete intervals. At time $t = 0$, the player of the game chooses the initial states of the grid. The initial states form a *seed pattern* that determines the course of the game. The states at time t uniquely determine the states at time $t + 1$. With each increment of t, all of the cells are updated. As the game runs, patterns grow and decay, resembling living organisms.

The rule for changing states in GoL can be compactly represented as B3/S23, where B means *born* and S means *survives*. A cell is *born* (it switches from state 0 to state 1) when exactly three of its eight nearest neighbours are alive (in state 1). A cell *survives* (remains in state 1) when it has either two or three living neighbours. Otherwise, a cell *dies* (it switches to state 0 or remains in state 0).

GoL is a *cellular automaton*, a discrete, abstract computational system. It is the best-known member of the family of cellular automata. It is popular as a model of emergence and self-organization [1], it is Turing-complete [10], and it is a universal constructor [5].

GoL is a member of the family of *semi-totalistic cellular automata* (also called *outer-totalistic*). The rules for this family have the general form Bx/Sy, where x and y are generated by deleting digits from the string 012345678, including deleting no digits or deleting all digits [2]. Since there are nine digits available for B and nine digits available for S, there are 2 to the power of 18 (262,144) possible semi-totalistic rules.

The question we explore here is, what makes the Game of Life special? Is it merely a historical accident that GoL was chosen out of 262,144 possible semi-totalistic rules, or does GoL have a property that distinguishes it from the other rules?

Packard and Wolfram [9] introduced a four-class system for characterizing two-dimensional cellular automata. Given a random initial seed pattern, class-1 automata eventually settle into one state (all zero or all one). Class-2 automata settle into separated stable or oscillating patterns. Class-3 automata are chaotic, yielding patterns without regular cycles. Class-4 automata, such as GoL, are capable of universal computation (they are Turing-complete).

Eppstein [2] pointed out that the four classes of Packard and Wolfram [9] are subjective and difficult to apply in practice. He offered an alternative four-class system, based on analysis of the Bx/Sy rule forms. This is an interesting and useful classification system, but no classification system can tell us what makes

GoL special, unless GoL belongs to a one-member class, *sui generis*. Instead of a classification system, we need a numerical scoring system, such that GoL stands out as the rule with the highest score. This scoring system could (potentially) tell us what is special about GoL (if the system is not *ad hoc* and arbitrary).

Gardner reported that Conway selected the rule of GoL to meet three criteria [3, p. 120]:

1. There should be no initial pattern for which there is a simple proof that the population can grow without limit.
2. There should be initial patterns that *apparently* do grow without limit.
3. There should be simple initial patterns that grow and change for a considerable period of time before coming to end in three possible ways: fading away completely (from overcrowding or becoming too sparse), settling into a stable configuration that remains unchanged thereafter, or entering an oscillating phase in which they repeat an endless cycle of two or more periods.

Inspired by Gardner's report of Conway's criteria [3], Packard and Wolfram's classification system [9], Eppstein's system [2], and our own experience with GoL, we created a simple metric that assigns a score to all 262,144 possible semi-totalistic rules. The metric is based on a combination of four properties: GoL is evenly balanced between growth and decay, measured by (1) the number of living cells and (2) the area covered by the living cells; (3) the rule for GoL is simple; and (4) GoL has low density, measured by living cells per area. The source code for calculating the metric is available, along with the scores for all 262,144 possible semi-totalistic rules [12]. (The source code uses Python and Golly [11].)

In Section 2, we precisely define the metric and explain how it is computed. In Section 3, we apply the metric to a list of rules that are known to support Turing-complete automata. The results support the claim that the metric is capturing an interesting quality; it is not merely *ad hoc*. We conclude in Section 4.

2 Scoring the Semi-totalistic Cellular Automata

To score each rule, we create a series of random initial seed patterns (random *soups*) and run the patterns to observe how they develop. By comparing the initial state with the final state, we can calculate whether patterns tends to grow or shrink.

The initial random soup is contained in a 16×16 square of cells. The soup is generated in two steps. First, we randomly select a number d using a continuous uniform distribution between 0 and 1. This d gives us the desired density of live cells for the 16×16 square. Second, we iterate through the cells in the square, randomly assigning state 1 with probability d and state 0 with probability $1 - d$. We then run the soup for

50 steps, to see how it develops. This process is repeated 1000 times, each time with a different density d. From the 1000 soups, we calculate the four statistics, as follows:

Probability of population increase: For each soup, we calculate the number of live cells at the start of a run and the end of a run, to determine whether the population increases. We estimate the probability of population increase by counting the number of soups that experience an increase and dividing the count by 1000.

Probability of area increase: For each soup, we assume the initial area is 16×16 = 256. (We ignore that the area could be less than 16×16 if d is quite small.) We calculate the final area by multiplying the height and width of the smallest rectangle that covers all living cells. We estimate the probability of area increase by counting the number of soups that experience an increase and dividing the count by 1000.

Average final density: For each soup, we calculate the density at the end of a run by the number of living cells divided by the area of the smallest rectangle that covers all living cells. We then calculate the average density over the 1000 runs.

Rule complexity: The complexity of a rule is determined by its length. For example, the rule for GoL is B3/S23, which has a length of 6 characters. We subtract 2 from this, because B and S are not strictly necessary; they are only present as a helpful hint for humans. (The / is necessary.) Thus the complexity of the rule for GoL is 4. Rules that begin with B0 are a special case, because B0 rules cause an infinite population. Therefore it is standard practice to modify B0 rules to remove infinities. For example, B03/S23 becomes B1245678/S0145678 for even values of t and B56/S58 for odd values of t [4]. To compensate for this extra level of complexity, we do not subtract 2 from the length of B0 rules. For example, B03/S23 has a complexity of 7.

The motivation for including rule complexity in the score is the intuition that Conway, as a mathematician, likely had a preference for simplicity and elegance. Furthermore, his initial investigation of cellular automata did not involve a computer, so simple rules would have made his work easier.

Let p_{p} be the probability of population increase and let p_{a} be the probability of area increase. Let d_{f} be the average final density and let c_{r} be the rule complexity. As we mentioned in Section 1, we believe that GoL is evenly balanced between growth and decay. To express this in a formula, we apply the geometric mean to p_{p} and p_{a}. Let $g(x)$ be the geometric mean of x (the probability of growth) and $1 - x$ (the probability of decay), $g(x) = \sqrt{x(1-x)}$, where $0 \leq x \leq 1$. The function $g(x)$ is at its maximum value of 0.5 when $x = 0.5$ (growth and decay are perfectly balanced). We apply $g(x)$ to p_{p} and p_{a} in order to reward rules that have a balance between growth and decay. Our scoring formula is as follows:

$$\frac{g(p_\mathrm{p}) \cdot g(p_\mathrm{a})}{d_\mathrm{f} \cdot c_\mathrm{r}}$$

This formula is near its maximum value when p_p and p_a are near 0.5 (an even balance of growth and decay) and d_f and c_r are small (low density and low complexity). To handle the problem of division by zero, if the denominator in this formula is zero, then we take the value of the whole formula to be zero.

3 Observations Concerning the Scores

When we score the 262,144 semi-totalistic rules with the above formula, the Game of Life achieves the highest score. Of course, the formula was designed to give GoL a high score, so this is not surprising, although it may be surprising that the formula is relatively simple. At this point, some further evidence is required to show that the formula is not merely a random fluke.

A compelling property of GoL is its Turing-completeness. If a scoring formula is truly capturing the character of GoL, then we should expect that other rules with high scores should also be Turing-complete. Naszvadi has collected a list of 35 semi-totalistic cellular automata that are known to be Turing-complete [8]. We use Nasvadi's list to validate our formula.

In Table 1, from the set of 262,144 semi-totalistic rules, we take the top 35 rules that have the highest scores, according to our formula, and then we look at Nasvadi's list to see how many of his 35 rules also appear in our top 35 list. Fifteen of his rules appear on our list. This is extremely unlikely to happen by chance. The expected number of matches between two lists of 35 items, both randomly sampled from a list of 262,144 items, is $\left(\frac{35}{262{,}144}\right)^2 = 1.783 \times 10^{-8}$.

Insert Table 1 here.

In Table 2, we show the rank of each of the 35 rules in Nasvadi's list. In an earlier version of the list, from 2017, he reported 30 rules. In 2020, he added five more rules. The lowest rank among the original 30 rules from 2017 is 1,282, which puts this rule in the top 1% of the semi-totalistic cellular automata (1,282 ÷ 262,144 = 0.00489). However, the more recent rules in 2020 show a different behaviour. Interestingly, two of these rules were shown to be Turing-complete by a novel approach, which does not use gliders. It seems that some of the 2020 rules are quite different from GoL, and hence it is appropriate that they should have low scores. GoL researchers are now entering a new realm in their search for Turing-complete rules.

Insert Table 2 here.

4 Conclusion

What makes the Game of Life special? It balances growth and decay, it tends towards low density, and it is simple. Other rules with these characteristics tend to be similar to the Game of Life, in that they are likely to support Turing-completeness. We may also expect many of them will support gliders and perhaps they will be universal constructors.

Given that there are 262,144 semi-totalistic rules, researchers in cellular automata have a large space to explore. Beyond semi-totalistic cellular automata, there are infinitely many other kinds of cellular automata. Defining various classes of automata [2, 9] is one way to manage this abundance. Our results suggest that defining various metrics on the spaces is another way.

Acknowledgments

Thanks to John Conway for discovering the Game of Life.

References

1. Bak, P., Chen, K., & Creutz, M. (1989). Self-organized criticality in the ‘Game of Life’. *Nature*, 342(6251), 780-782.
2. Eppstein, D. (2010). Growth and decay in life-like cellular automata. In Adamatzky, A. (Ed.), *Game of Life Cellular Automata* (pp. 71-97). Springer, London.
3. Gardner, M. (1970). Mathematical Games: The fantastic combinations of John Conway’s new solitaire game ‘life’. *Scientific American*, 223(4): 120-123.
4. Golly (2020). *QuickLife: B0 emulation*. http://golly.sourceforge.net/Help/Algorithms/QuickLife.html#b0emulation.
5. LifeWiki (2020). *Universal constructor*. https://www.conwaylife.com/wiki/Universal_constructor.
6. LifeWiki (2020). *Other cellular automata: Day & night.* https://www.conwaylife.com/wiki/OCA:Day_%26_Night.
7. LifeWiki (2020). *Other cellular automata: Move.* https://www.conwaylife.com/wiki/OCA:Move.
8. Naszvadi, P. (2020). *List of the Turing-complete totalistic life-like CA.* https://conwaylife.com/forums/viewtopic.php?f=11&t=2597.
9. Packard, N. H., & Wolfram, S. (1985). Two-dimensional cellular automata. *Journal of Statistical Physics*, 38(5/6): 901-946.
10. Rendell, P. (2016). *Turing machine universality of the Game of Life*. Springer, New York, NY.

11. Trevorrow, A., Rokicki, T., Hutton, T., Greene, D., Summers, J., Verver, M., Munafo, R., Bostick, B., & Rowett, C. (2020). *Golly 3.4 for Windows*. http://golly.sourceforge.net/.
12. Turney, P. D. (2020). *Life is special: Source code*. https://github.com/pdturney/life-is-special.

Table 1. This table shows the top-thirty-five highest-ranked rules among the 262,144 semi-totalistic rules, where high rank corresponds to high score. The Game of Life rule, B3/S23, is first. Fifteen of Nasvadi's thirty-five rules [8] are in the top-thirty-five highest-ranked semi-totalistic rules. These fifteen rules are marked with *N* in this table. The high degree of overlap between the top-thirty-five highest-ranked rules according to our scoring formula and Nasvadi's list of thirty-five Turing-complete rules supports the hypothesis that our formula is capturing the spirit of what it means for a rule to be life-like.

Rank	Rule	Nasvadi's list	Probability of population increase	Probability of area increase	Average final density	Rule complexity	Score
1	B3/S23	N	0.091	0.730	0.132	4	0.242
2	B34/S12		0.316	0.733	0.186	5	0.221
3	B2/S		0.941	0.976	0.083	2	0.216
4	B3/S237	N	0.156	0.818	0.133	5	0.211
5	B3/S236	N	0.542	0.840	0.175	5	0.208
6	B3/S023		0.293	0.896	0.135	5	0.207
7	B38/S23	N	0.109	0.710	0.139	5	0.204
8	B34/S13		0.234	0.750	0.188	5	0.195
9	B37/S23	N	0.119	0.763	0.142	5	0.194
10	B3/S238	N	0.094	0.737	0.133	5	0.194
11	B345/S02		0.205	0.776	0.146	6	0.192
12	B37/S023		0.384	0.868	0.144	6	0.191
13	B3/S134		0.243	0.761	0.192	5	0.191
14	B3/S0238		0.300	0.870	0.137	6	0.188
15	B34/S24		0.502	0.717	0.240	5	0.188
16	B348/S12		0.312	0.695	0.190	6	0.187
17	B38/S023		0.316	0.878	0.138	6	0.184
18	B3/S2378	N	0.169	0.801	0.140	6	0.178
19	B37/S238	N	0.139	0.766	0.137	6	0.178
20	B36/S23	N	0.086	0.689	0.154	5	0.168
21	B37/S237	N	0.204	0.829	0.151	6	0.168
22	B347/S12		0.370	0.777	0.200	6	0.168
23	B38/S236	N	0.543	0.850	0.177	6	0.167
24	B36/S023		0.385	0.882	0.157	6	0.166
25	B367/S23		0.186	0.769	0.165	6	0.166
26	B3/S2368	N	0.534	0.849	0.180	6	0.165
27	B348/S13		0.277	0.774	0.191	6	0.163
28	B368/S23	N	0.120	0.707	0.152	6	0.162
29	B378/S23	N	0.134	0.742	0.156	6	0.159
30	B38/S0238		0.321	0.877	0.139	7	0.158
31	B34/S128		0.432	0.822	0.203	6	0.155
32	B36/S237	N	0.191	0.797	0.170	6	0.155
33	B356/S23		0.220	0.747	0.194	6	0.155
34	B35/S237		0.270	0.791	0.194	6	0.155
35	B35/S023		0.380	0.890	0.164	6	0.154

Table 2. This table lists all thirty-five of Nasvadi's Turing-complete rules, in order of their rank in the 262,144 semi-totalistic rules. Thirty of the rules were listed by Nasvadi in 2017. Five more rules were added in 2020. To facilitate reading the table, we only show the year for the five new rules added in 2020. Four of the five new rules have a low rank, suggesting that the search for Turing-complete rules is now heading into new territory, away from the Game of Life. Near the bottom of the table, rules B3678/S34678 (Day & Night [6]) and B368/S245 (Move [7]) were shown to be Turing-complete by constructing a Rule 110 unit cell, which is quite different from the usual constructions based on gliders.

Rank	Rule	Year 2020	Probability of population increase	Probability of area increase	Average final density	Rule complexity	Score
1	B3/S23		0.091	0.730	0.132	4	0.242
4	B3/S237		0.156	0.818	0.133	5	0.211
5	B3/S236		0.542	0.840	0.175	5	0.208
7	B38/S23		0.109	0.710	0.139	5	0.204
9	B37/S23		0.119	0.763	0.142	5	0.194
10	B3/S238		0.094	0.737	0.133	5	0.194
18	B3/S2378		0.169	0.801	0.140	6	0.178
19	B37/S238		0.139	0.766	0.137	6	0.178
20	B36/S23		0.086	0.689	0.154	5	0.168
21	B37/S237		0.204	0.829	0.151	6	0.168
23	B38/S236		0.543	0.850	0.177	6	0.167
26	B3/S2368		0.534	0.849	0.180	6	0.165
28	B368/S23		0.120	0.707	0.152	6	0.162
29	B378/S23	2020	0.134	0.742	0.156	6	0.159
32	B36/S237		0.191	0.797	0.170	6	0.155
44	B38/S238		0.102	0.745	0.147	6	0.150
47	B38/S2378		0.157	0.775	0.146	7	0.149
52	B37/S236		0.623	0.868	0.187	6	0.146
55	B378/S237		0.184	0.810	0.149	7	0.146
57	B38/S237		0.134	0.791	0.159	6	0.145
58	B38/S2368		0.572	0.844	0.177	7	0.145
59	B36/S2378		0.219	0.774	0.171	7	0.145
61	B378/S238		0.147	0.780	0.145	7	0.144
64	B36/S238		0.101	0.710	0.159	6	0.143
71	B3/S2367		0.656	0.868	0.190	6	0.141
117	B3/S23678		0.595	0.850	0.193	7	0.130
185	B378/S2378		0.181	0.819	0.152	8	0.122
192	B368/S238		0.108	0.725	0.164	7	0.121
261	B38/S2367		0.668	0.879	0.194	7	0.113
266	B38/S23678		0.586	0.860	0.189	8	0.113
745	B368/S12578	2020	0.188	0.814	0.186	9	0.091
1,282	B35/S236		0.862	0.871	0.246	6	0.078
5,878	B3678/S34678	2020	0.095	0.275	0.315	10	0.042
241,984	B368/S245	2020	0.000	0.057	0.276	7	0.000
247,567	B36/S245	2020	0.000	0.046	0.265	6	0.000