Two-dimensional Totalistic Code 52*

Todd Rowland†

Senior Research Associate,
Wolfram Research, Inc.
100 Trade Center Drive,
Champaign, IL

The totalistic two-dimensional cellular automaton code 52 is capable of a wide variety of behavior. In this paper we look at its generic behavior, its complex behavior on simple backgrounds, its versatility in performing finite computations, and its block emulations of elementary cellular automata (ECA). In all, we found 65 ECA which are emulated by code 52. Taken together, the evidence is strong that code 52 may be a universal rule.

Introduction

Part of the Principle of Computational Equivalence [1, p. 715] tells us that complex behavior in a simple rule implies the rule is likely to be universal. Code 52 has this sort of unpredictability. But even if we did not have this principle, we might think that code 52 is universal since it shows great flexibility with finite computations, as well. Recall that to be universal means that it can emulate any other rule.

Past work has suggested two different fates for code 52. Stephen Wolfram has pointed out that the generic behavior of code 52 involves some localized structures, like the class 4 elementary cellular automata (ECA) [1, p. 692]. That would suggest its universality. On the other hand, there is the idea that the totalistic five-neighbor two-dimensional (2D) cellular automata cannot be universal, a misconception arising from [2].

As we show in this paper, the difference between the points of view can be understood in terms of the backgrounds allowed. On a pure white background, code 52 cannot grow, in which case no universal computations can be performed. If one allows regular backgrounds, it is a completely different story. There is complex behavior, as well as emulations of at least 65 ECA.

We begin by investigating the generic behavior, which is a sort of clumping. The limit cycles on finite grids appear to be nearly regular bands, where the boundary is an emulation of a one-dimensional (1D) cellular automaton. Almost all of the time, it is rule 150, first pointed out

*Matthew Szudzik, Guest Editor.

†Electronic mail address: rowland@wolfram.com.

in [3]. We also look at the density behavior. Comparing the theoretical predictions to varying degrees of accuracy suggests that there is a fair amount of instability.

Besides complex behavior, the intuition coming from the definition of a universal rule is that it should be programmable. So the ease in which one can program finite computations is one form of evidence for universality. Code 52 is able to perform the 16 boolean functions of two variables, with nearly a minimal amount of space and time. In addition, configurations can be found which are somewhat uniform.

Block emulations of a cellular automaton, where each color is represented by a fixed block of cells, are somewhat rare. Even more rarely are they interesting or useful. But they are the most straightforward way for one cellular automaton to emulate another. In the case of ECA emulating other ECA, the one with the most known block emulations is rule 41 with 12 [1, p. 702]. The 2D code 52 has at least 65 (out of 256), using block emulations on the cylinder.

To be proven universal requires more than this. Unfortunately, a block emulation of rule 110 was not found. So there is not yet a proof, but it would be quite surprising if code 52 were not universal.

Rule description

The rule depends on the total of the five neighbors, including the center cell. It is almost a majority rule, except it differs when the average is closest to 2.5. Figure 1 shows the color of the updated cell as a result of the average in its five-cell neighborhood, where the corresponding neighborhoods are listed underneath.

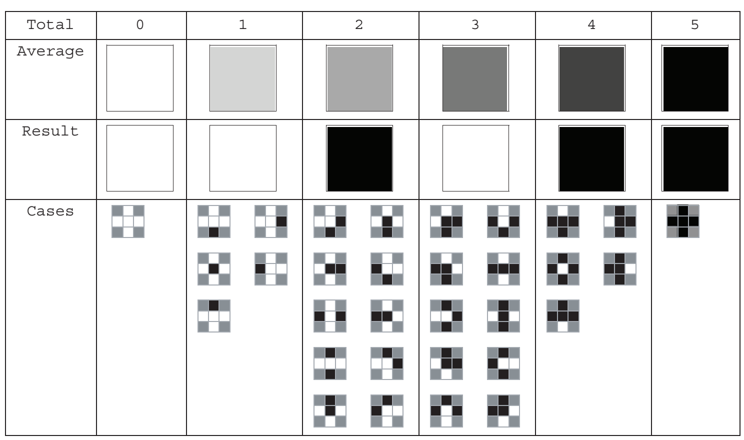


Figure 1.

Past work

Code 52 has been mentioned as an interesting rule in some of Wolfram's early papers on cellular automata, and also in [1].

For instance, he showed that a typical long term behavior involves a sort of clumping [1, p. 233] and that on a finite grid it can settle down into linear light and dark regions, with a boundary performing ECA rule 150 (Figure 2).

Wolfram has also pointed out that the slices of code 52 obey a class 4 type of behavior [1, p. 692].

Since this work was presented at the NKS 2006 conference, other past work has been pointed out to me. In [2], it is claimed that none of the five-neighbor totalistic rules, including code 52, can be universal. The definition of universal there is more restrictive than that used here, as it requires that the background be all white. As mentioned in the following, if one has a white background then any finite configuration of code 52 cannot grow, and thus any system restricted to all white backgrounds cannot be universal. We will also demonstrate that on other backgrounds a wide variety of behavior is possible.

Trivial observations

There are a few trivial observations of code 52 which help one understand its generic behavior. The first trivial observation is that the rule is symmetric with white and black.

Since it is like a modified majority rule, it is easy to see that, when there are walls between regions, thickness at least two, they are stable (Figure 3). Because at the boundary, within the black region there are always at least four black cells in each neighborhood.

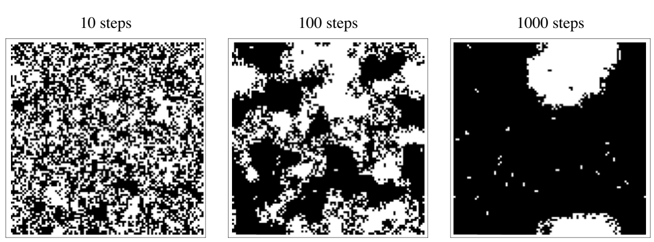


Figure 2. Clumping behavior from random initials on size 100.

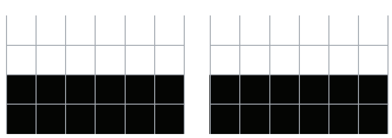


Figure 3.

Hence, on a plain background there is no growth. Any bounded region stays bounded. In Figure 4 red marks the original boundary. Another interesting case of bounded behavior is shown in Figure 5.

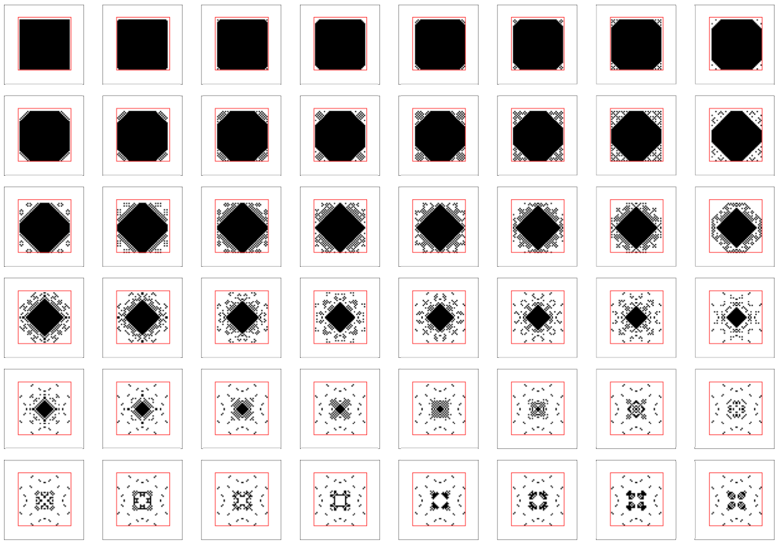


Figure 4.

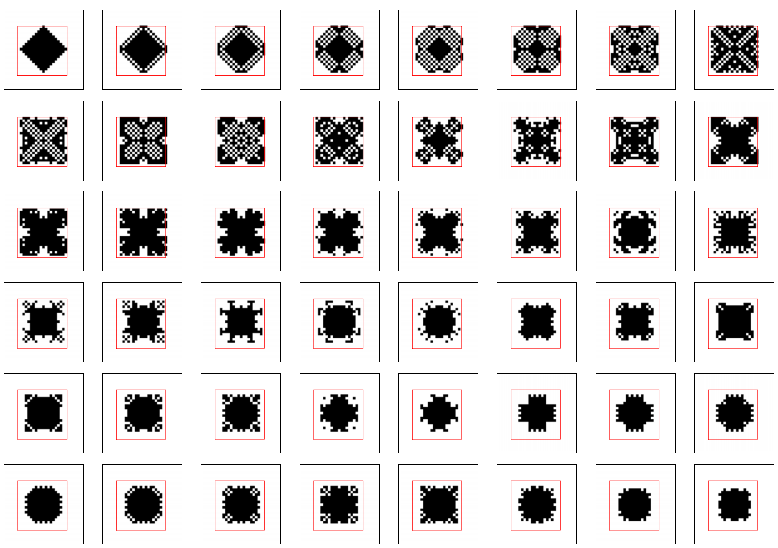


Figure 5.

There are many stationary states, the simplest includes two black cells (Figure 6).

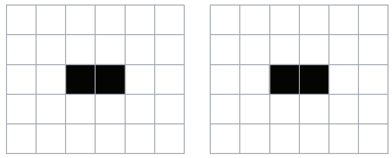


Figure 6.

Generic behavior

By *generic* behavior, I mean behavior from random initial conditions on a large grid. One global parameter is the density of white and black cells.

With initial conditions 50% black or white, it shows a sort of clumping behavior (Figure 7).

The testing was done on grids of size 40 by 40, with comparisons made to smaller grid sizes as well.

As time progress on a finite grid, it resolves, usually as either mostly white or black with a few stationary structures (Figure 9).

About 25% of the time it resolves into a single band (at size 20). This appears to be because it is on a finite grid.

The band persists, and most of the time has a boundary which follows rule 150, as noted by Wolfram and Norman Packard in [3].

Somewhat more rare is the structure shown in Figure 10 which follows rule 90, about 1/2% of the time on grids of size 10, down to less than .01% on grids of size 20.

Theoretically, it is possible to have other boundaries, some following other ECA, but I was not able to observe any coming from random initial conditions. We will see those examples when we get to the ECA emulations.

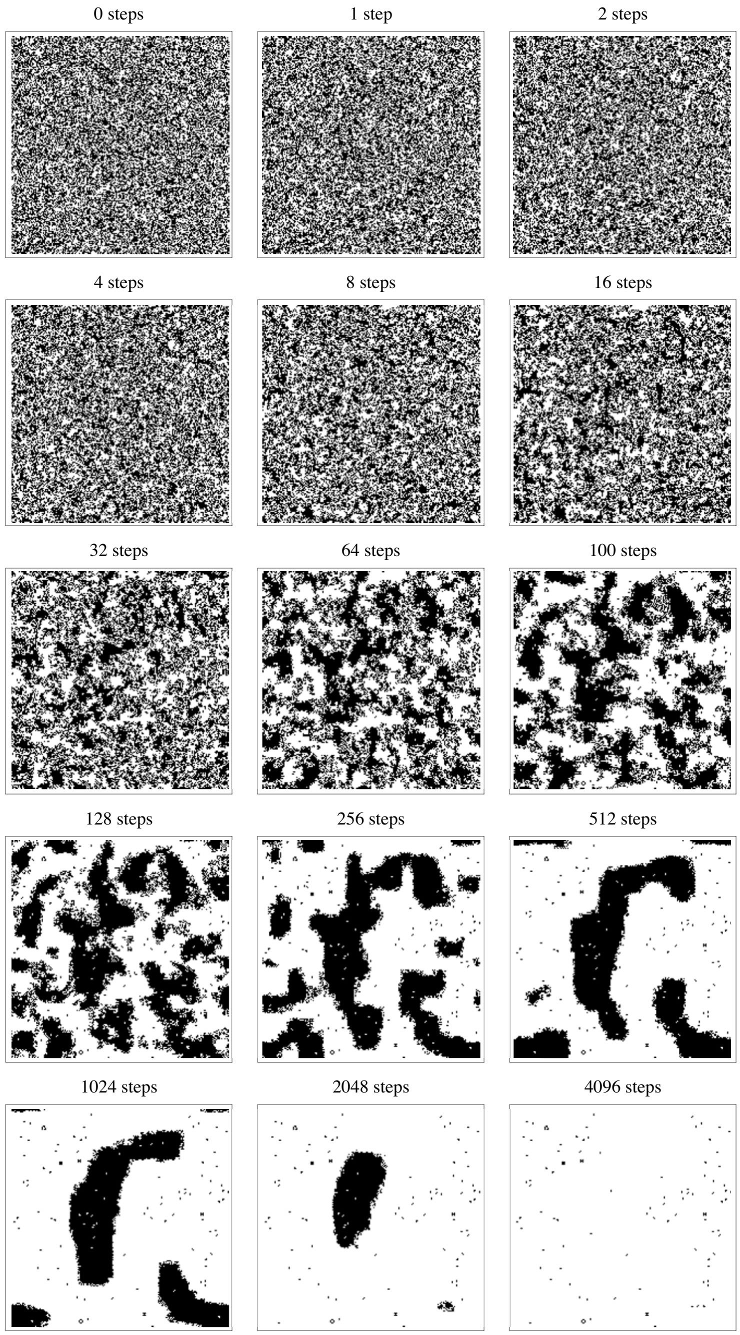

0 steps
1 step
2 steps
4 steps
8 steps
16 steps
32 steps
64 steps
100 steps
128 steps
256 steps
512 steps
1024 steps
2048 steps
4096 steps

Figure 7.

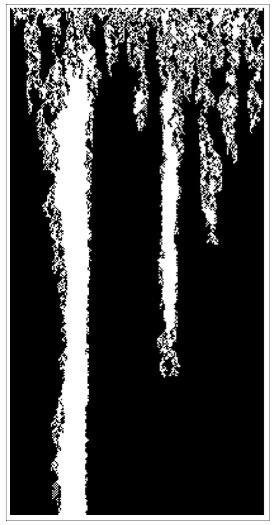


Figure 8. First row on a size 200 grid for 400 steps.

Figure 9.

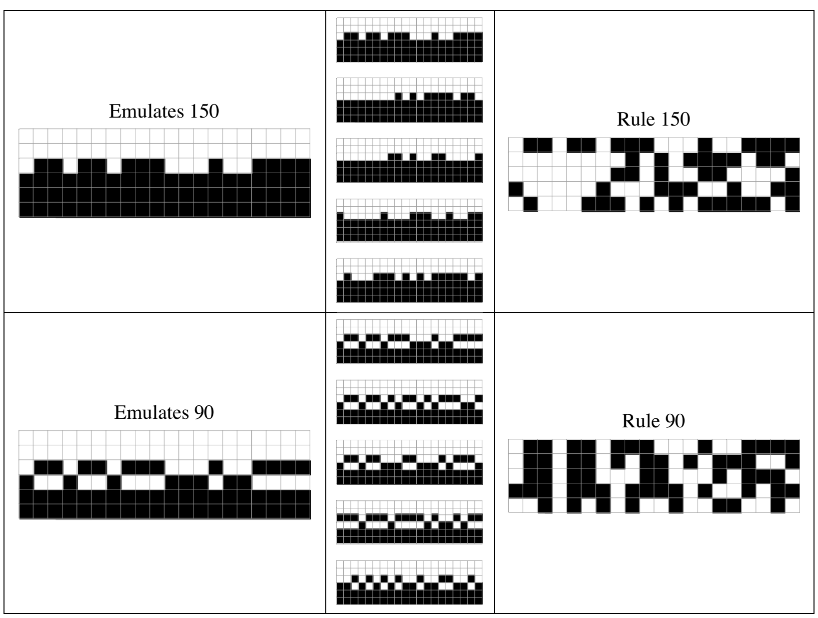


Figure 10.

Density behavior

In a mean-field approximation, one assumes that the cell colors are not correlated and are given by a uniform probability. The implications are that the density p updates by the following (see Figures 11 through 13):

$$p^5 + 5(1-p)p^4 + 10(1-p)^3 p^2.$$

Recall that code 52 is symmetric with black and white, which explains the symmetry of the curve.

Note also that it is close to the straight line, and we can easily see its tendencies when repeated by subtracting that off. When it is less than about 0.17 it is decreasing, tending to zero, between 0.17 and 0.83 it tends to 1/2, and above that it tends to 1.

Iterating the map six times shows this prediction (Figure 14). As we can see it is pretty far from the experimental results.

Looking more steps ahead, in an attempt to make corrections as in [1, p. 953], shows that the experimental results are not a fluke.

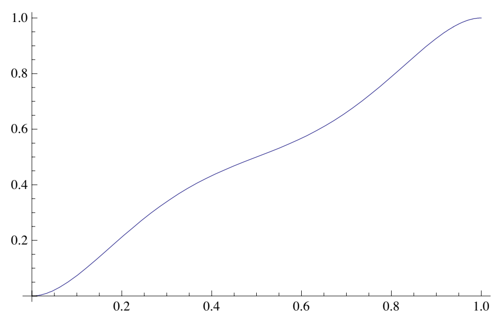


Figure 11. Uncorrelated density update after one step with the likelihood of 5, 4, or 2 neighbors being black.

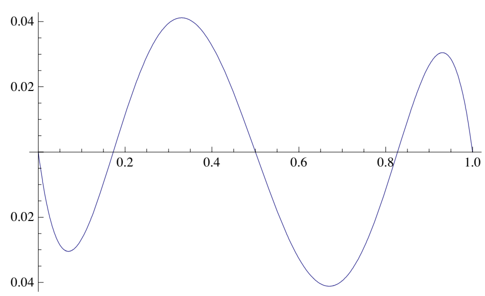


Figure 12. Difference from identity map.

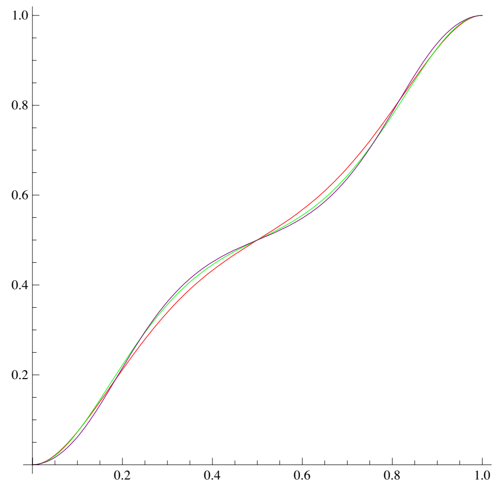


Figure 13. Uncorrelated density update after one step (red), after two steps (green), and after three steps (purple).

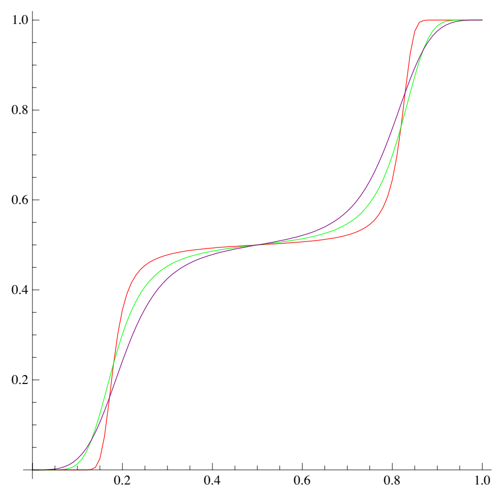


Figure 14. Three predictions for six steps ahead: one step map, 6× (red); two steps, 3× (green); and three steps, 2× (purple).

We see that the mean field approximation is not right, which means that the rule is not as random as it appears (Figure 15). The arrangements are correlated.

If one takes the main clumping behavior as an assumption, then it makes sense instead to assume only that the overall density is a value p, rather than the assumption that each cell is individually white or black

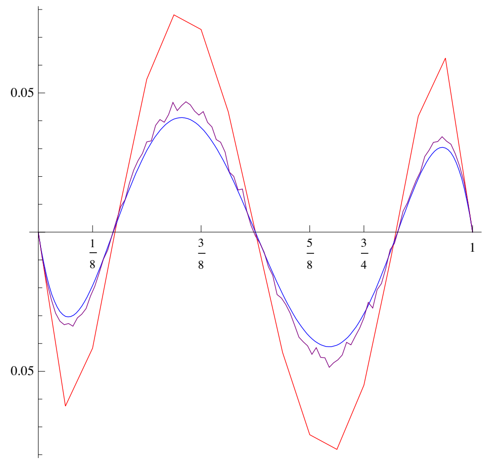


Figure 15. Theory *versus* experiment: one step difference from identity. Based on all 4×4 grids (red), random sample on 10×10 grid (purple), and theoretical prediction (blue).

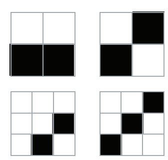


Figure 16.

with probability p. This gives a better analytical approximation to the actual behavior of the density function.

Other backgrounds

In examining this rule, we have just looked at the case of generic situations, but now let us look at specific configurations where code 52 does something special.

As is common in the study of the ECA, we can look for interesting behavior on cyclic backgrounds. The simplest cyclic backgrounds of code 52 in two dimensions are shown in Figure 16. The more interesting cyclic backgrounds begin at size 5, but it makes sense to start with the simplest.

Recall that on the white background, bounded regions stay bounded, but there are plenty of stationary structures. Looking, for instance, for moving localized structures, we have to go to other backgrounds.

On many of the other backgrounds it is possible to get growing structures. The red square is to help see how fast it grows from a

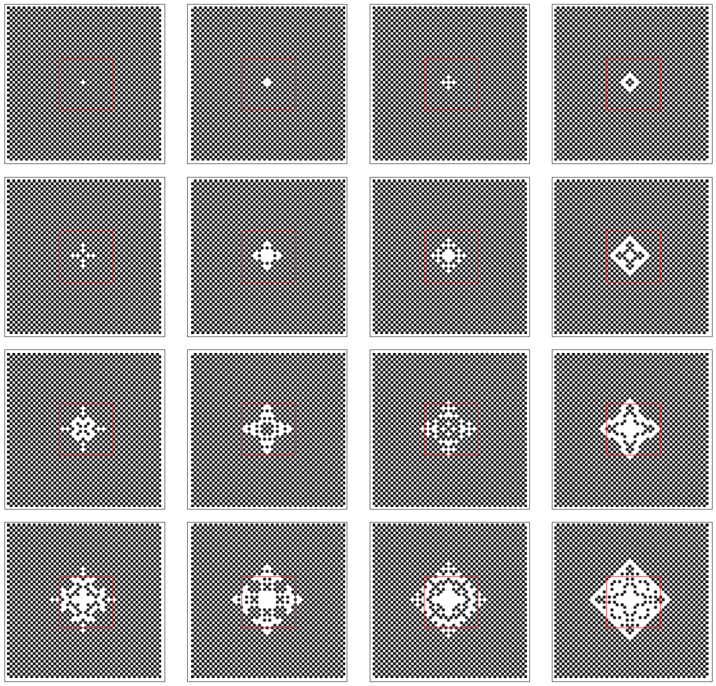


Figure 17.

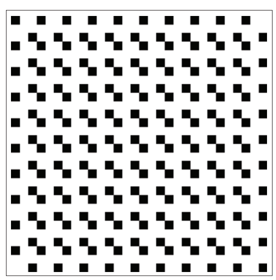


Figure 18.

singular cell on a checkerboard grid. The plots shown in Figure 17 are typical of 2D cellular automata, as in some of the pictures in [1].

On the background shown in Figure 18 there is even a moving localized structure. It leaves behind a trail. It begins in Figure 19 as a random tangle.

The structure's leftover tail has the property of forming walls which stop other structures. Gliders like these have been a main ingredient in traditional universality arguments [4, 5].

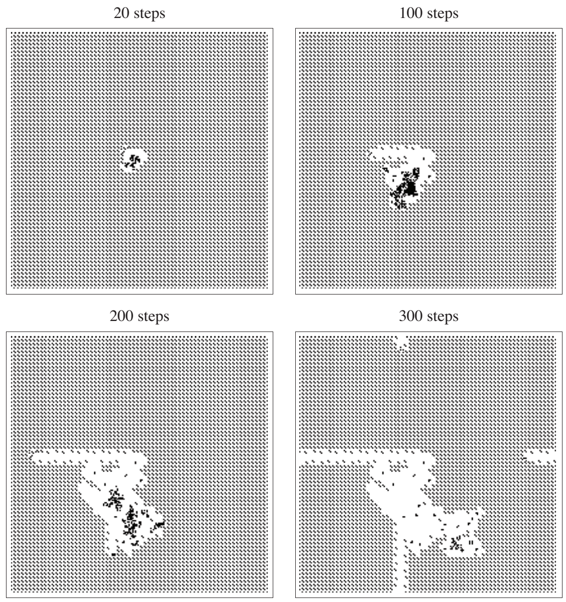


Figure 19.

Logic

Moving onto the cases of special initial conditions, and starting with finite computations, there is the problem of emulating boolean expressions.

What kind of finite computations are supported? It seems like it is fairly easy to find initial configurations which perform any specific finite computation.

For instance, take the boolean operators shown in Figure 20. One asks whether the center cell is white or black after two steps. Since the rule depends on immediate neighbors, the outcome of the center cell after two steps depends on the cells at distance two. One can pick cells whose color represents the inputs to the boolean function, as well as picking a fixed arrangement to impose the restriction that the input values make the center cell have the value determined by the boolean function.

In Figure 20, the blue cells are fixed as black, the green and red cells are the two inputs. The output is the center cell after two steps. The top row shows the boolean operator, the grid with the colors shows the initial configuration, and the bottom row shows the two-step evolutions in the four cases. In the evolutions, the cells which do not affect the center cell are grayed out.

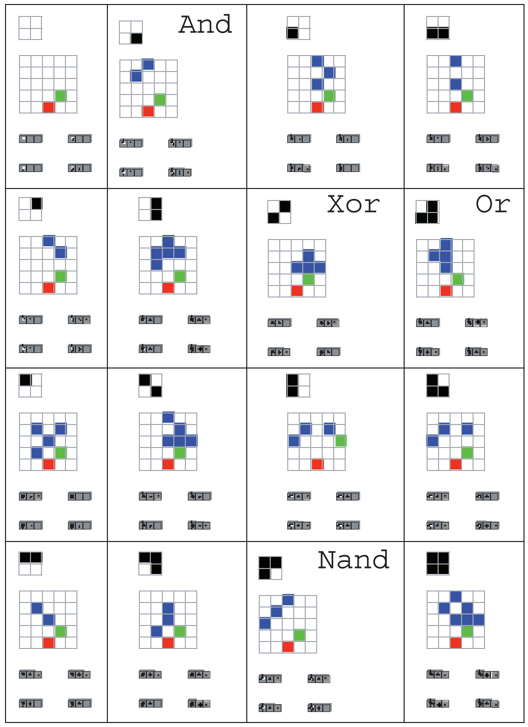


Figure 20.

Emulations of one-dimensional elementary cellular automata

There are a few obvious ways to emulate a 1D CA with a 2D rule. One can consider block emulations on a 2D cylinder. Each band in Figure 21 corresponds to white or black.

Because code 52 stays within its boundaries, it is also possible to have block emulations on a strip bounded by white cells on the infinite grid.

Figure 22 shows ECA rule 72 emulated by the 2D code 52. Four steps of code 52 perform one step of rule 72. The blocks have height 7 and width 4, and I call this a "w-w" strip since it has two rows of white on the top and bottom.

There is a further possibility of having black cells in the lower half. Having black cells in both halves is taken care of by symmetry since code 52 is the same when white is switched with black.

In addition to block emulations, where one block is white and another block is black, there are the coarse-grain emulations, and also the coarse graining of a subalgebra.

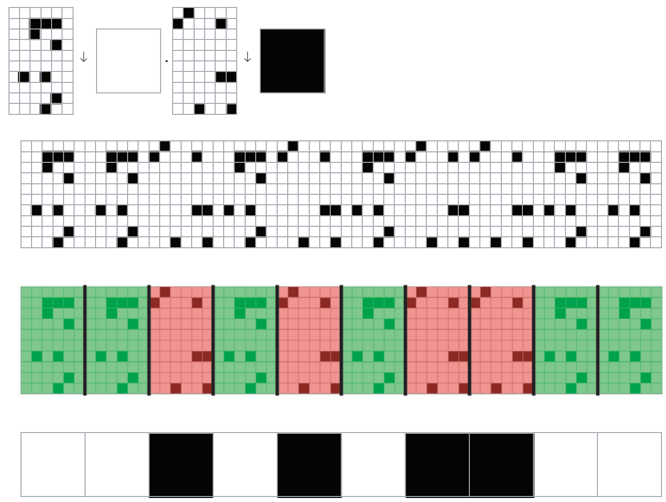


Figure 21.

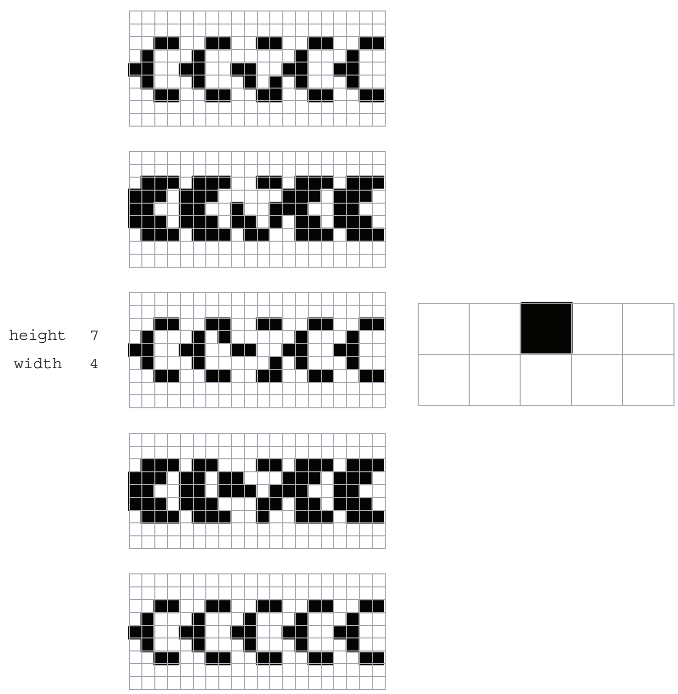
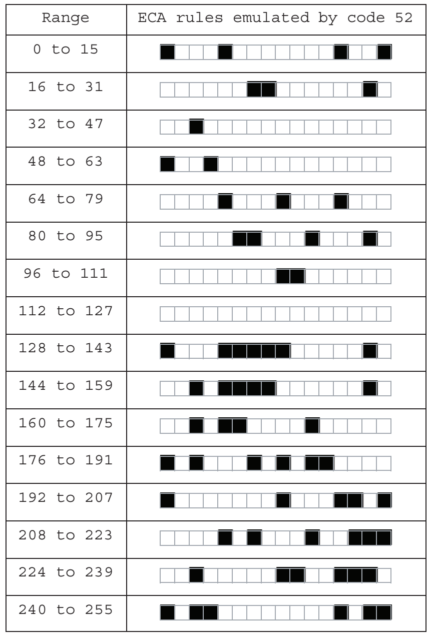


Figure 22.



0	4	12	15	22	23	30	34	48	51	68	72	76
85	86	90	94	104	105	128	132	133	134	135	136	142
146	148	149	150	151	158	162	164	165	170	176	178	182
184	186	187	192	200	204	205	207	212	214	218	221	222
223	226	232	233	236	237	238	240	242	243	252	254	255

Figure 23.

In all, I was able to find emulations for 65 out of the 256 ECA. They were all found by brute search. Figure 23 shows which ECA rule numbers were found to have emulations.

By comparison, the ECA rule 41 can do the 12 rules shown in Figure 24 [1, p. 692].

Figure 25 summarizes what those rules emulated by code 52 do from random initial conditions.

To me, the most interesting rules it can emulate are 22, 30, and 94, which can perform nested and complex behavior (Figure 26). Other notables include 90 and 184.

Figure 26 shows those rules in typical initial conditions, to get a sense of what this means for the range of behaviors code 52 is capable of in special initial conditions.

The list of equivalence classes of ECA found to be emulated by code 52 is shown in Figure 27. Each rule is block emulated on a cylinder running code 52, using a block of dimensions height and width, with the given code number in binary for the white and black cells. The ECA

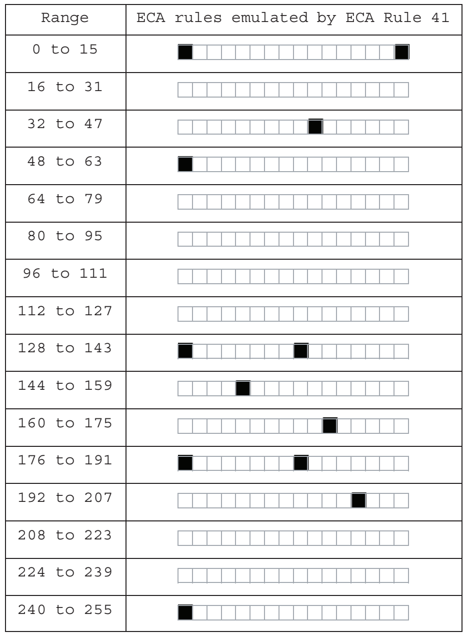


Figure 24.

rules equivalent to the given rule are in the last column. Some of the code numbers are large because they are more naturally given as a strip on the plane with either an all white background, or all white on top and black on the bottom.

One reason this is interesting is that there is often in nature the situation where something happens extremely rarely. For instance, errors in the process of DNA replication, or certain radioactive decays. Here we have something which is definitely, easily observable, happening completely deterministically, even in a relatively small system.

Conclusion

Code 52 has a variety of behaviors, not just in the generic situation, but also in special initial conditions. Its flexibility to perform very different sorts of calculations suggests that more examples of simple universal behavior will be found. The other 63 rules deserve further study.

As the number of emulation examples increases, it becomes easier to demonstrate more emulations. One eventual goal will be a more detailed statement of the Principle of Computational Equivalence (PCE). We presented a few bits of evidence here, both of the PCE and of the universality of code 52 in particular, and hopefully other forms of evidence will be discovered.

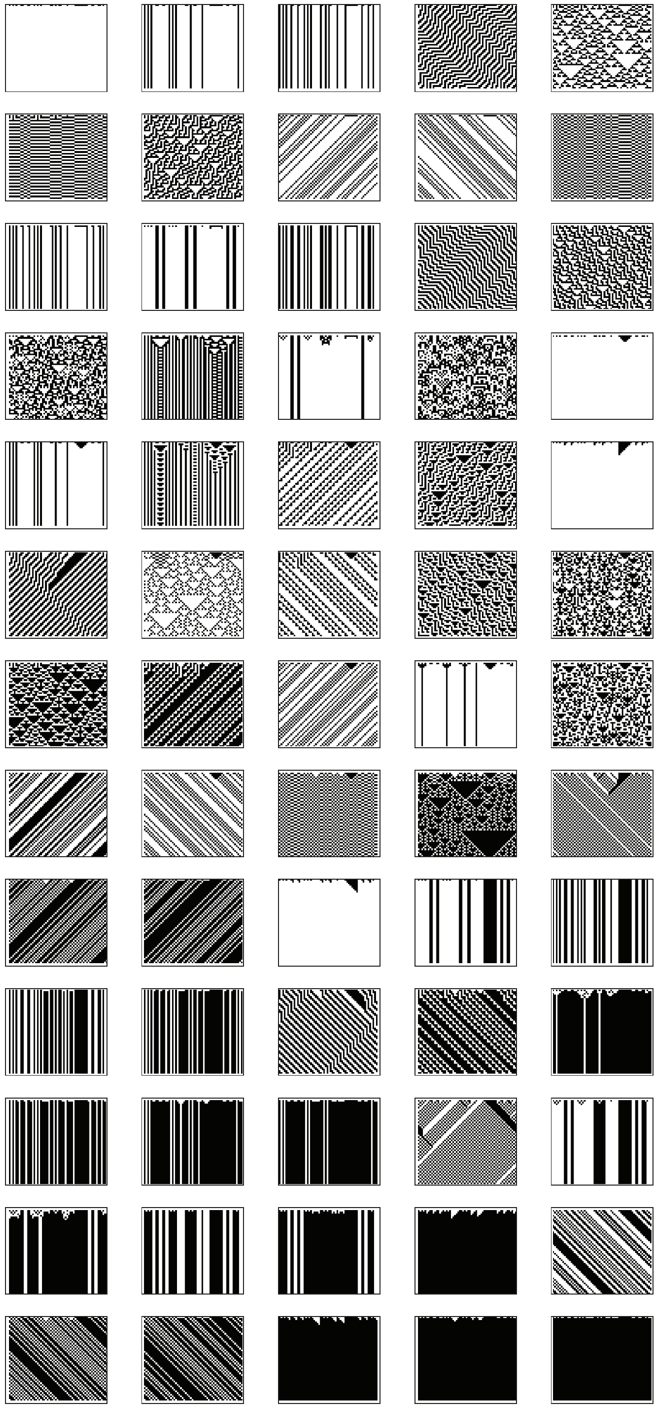


Figure 25.

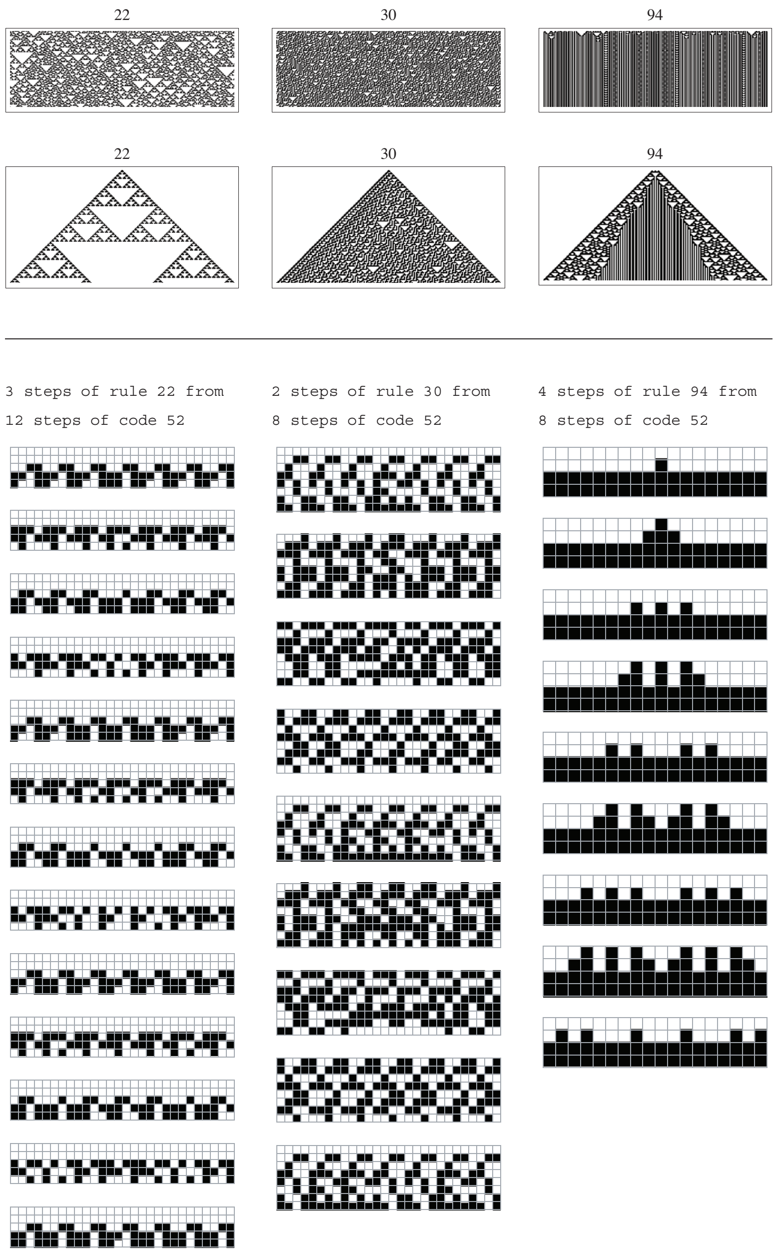

22
30
94
22
30
94
3 steps of rule 22 from
12 steps of code 52
2 steps of rule 30 from
8 steps of code 52
4 steps of rule 94 from
8 steps of code 52

Figure 26.

rule	height	width	white code	black code	rule equivalences
0	1	3	0	1	{255}
4	3	2	0	3	{223}
12	4	6	40 329	38 089	{68, 207, 221}
15	7	10	881 712 501 938 480 742 399	881 703 490 341 179 490 303	{85}
22	5	4	985	989	{151}
23	8	4	128 891 465	55 230 125	{}
30	8	4	55 354 013	55 483 567	{86, 135, 149}
34	7	8	63 111 164 275 523 583	63 111 228 700 033 023	{48, 187, 243}
51	2	3	1	8	{}
72	7	4	216 131	199 763	{237}
76	5	8	12 850 321	15 090 099	{205}
90	2	2	6	7	{165}
94	4	2	15	31	{133}
104	3	1	0	1	{233}
105	2	1	1	2	{}
128	1	2	0	1	{254}
132	3	2	0	6	{222}
134	7	4	568 063	582 911	{148, 158, 214}
136	2	4	51	57	{192, 238, 252}
142	7	4	795 391	811 775	{212}
146	7	4	795 391	860 927	{182}
150	2	2	3	5	{}
162	7	4	540 159	669 183	{176, 186, 242}
164	4	1	0	3	{218}
170	3	2	1	47	{240}
178	7	4	567 039	583 423	{}
184	7	4	568 063	453 375	{226}
200	7	4	689 546	689 578	{236}
204	2	2	0	15	{}
232	1	1	0	1	{}

Figure 27.

References

[1] S. Wolfram, *A New Kind of Science* (Wolfram Media, Inc., Champaign, IL, 2002).

[2] E. Codd, *Cellular Automata* (Academic Press, New York, 1968).

[3] N. Packard and S. Wolfram, "Two-dimensional Cellular Automata," *Journal of Statistical Physics*, **38** (1985) 901.

[4] E. Berlekamp, J. Conway, and R. Guy, "What Is Life?" Chapter 25 in *Winning Ways for Your Mathematical Plays, Volume 2: Games in Particular* (Academic Press, London, 1982).

[5] M. Gardner, "The Game of Life, Parts I–III." Chapters 20–22 in *Wheels, Life, and other Mathematical Amusements* (W. H. Freeman, New York, 1983).

[6] S. Wolfram, "Universality and Complexity in Cellular Automata," *Physica D*, **10** (1984) 1.

[7] S. Wolfram, *Cellular Automata and Complexity: Collected Papers* (Addison-Wesley, Reading, MA, 1994).