

Self-reproduction by glider collisions: the beehive rule

Andrew Wuensche

Discrete Dynamics Lab, Santa Fe, NM 87506

andyw@cybermesa.com, www.ddlab.com

Abstract

We present a 3-value cellular automaton which supports self-reproduction by glider collisions. The complex dynamics emerge spontaneously in both 2d and 3d according to the 6-neighbor, k -totalistic, “beehive” rule; the 2d dynamics on a hexagonal lattice is examined in detail. We show how analogous complex rules can be found, firstly by mutating a complex rule to produce a family of related complex rules, and secondly by classifying rule-space by input-entropy variance. A variety of complex rules opens up the possibility of seeking a common thread to distinguish those few rules from the rest: an underlying principle of self-organization?

Introduction

Structure emerging by local interactions, self-reproduction and evolution; these themes are central to understanding natural processes. A system’s complexity, according to this approach, relates to the number of levels on which it can be usefully described (Wuensche, 1994; Wuensche, 1999).

The simplest artificial systems able to capture the essence of these dynamical processes are cellular automata (CA), where “cells” connected on a regular lattice synchronously update their color by a logical function of their neighbor’s colors. Just a tiny proportion of possible logics (complex rules) allow higher levels of description, greater complexity, to emerge from randomness.

In a movie of successive patterns on the lattice, recognizable sub-patterns emerge; mobile structures (gliders¹) interact, aggregate, make glider-guns, and gliders self-reproduce or self-destruct by colliding.

In discrete CA everything can be precisely specified: rules, connections, dynamics. So for a given complex rule it should be possible to find causal links between the underlying “physics” and the ascending levels of emergent structure. We can also ask if there is a common thread that distinguishes those few rules that support complex dynamics from the vast majority that do not: an underlying princi-

¹“Gliders” and other terminology is taken from John Conway’s famous Game-of-Life (Conway, 1982). Gliders can also be regarded as particles or waves

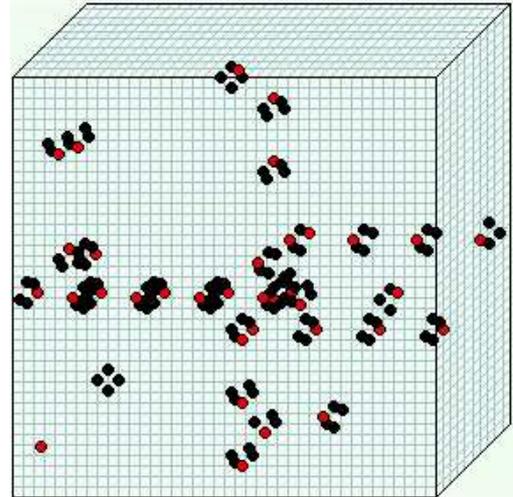


Figure 1: A snapshot of the beehive rule running on a 3d (40x40x10) lattice. The $k = 6$ neighborhood is shown in figure 3. The complex dynamics includes the spontaneous emergence of gliders, self-reproduction by glider collisions and glider guns, analogous to the 2d case. Gliders move in the direction of their red heads. Read this figure as if looking down into a shallow box.

ple of self-organization? That investigation would require a good sample of complex rules, which is now accessible.

Of the variety of complex behaviors in CA, self-reproduction (or self-replication) is perhaps the most prized (von Neumann, 1966; Conway, 1982; Langton, 1984) and provided the early motivation for ALife - but there have been few further examples of non-trivial self-reproduction until recent work by Antonio Lafusa (Bilott et al., 2003). He has been searching among the multi-value k -totalistic rules by genetic algorithm, using a fitness function of high input-entropy variance (Wuensche, 1999) and related measures.

We have also looked at multi-value k -totalistic rule-space, but for much smaller lookup tables than Lafusa’s. We have limited both the value-range v (range of colors) and the neighborhood k to keep our look-up tables short, and make it easier to understand how specific entries relate to gliders. We have results for $v = 3$ and $k = 4$ to $k = 9$, but we will mainly describe results for $k = 6$ on a hexagonal 2d lattice.

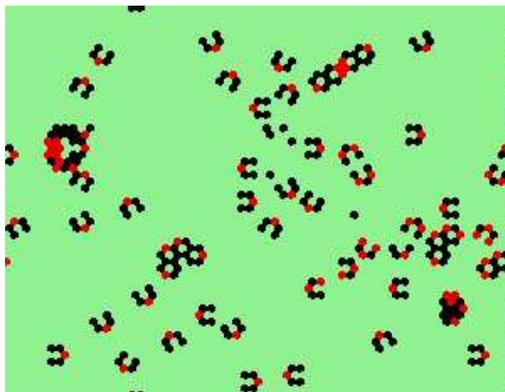


Figure 2: A snapshot of the beehive rule running on a 2d hexagonal lattice. Gliders move in the direction of their red heads.

Complex rules are easily found in these smaller rule-spaces by the classification methods in (Wuensche, 1999). Small lookup tables also make it easier to study mutations. It turns out that a large proportion of 1-value mutations are quasi-neutral; they make little difference to the complex dynamics. Some mutations result in modified but equally interesting complex dynamics. So mutations create families of related complex rules. Of course, there are also sensitive positions in the lookup table where a mutation completely disrupts the complex dynamics.

This paper outlines the ideas and methods. The “beehive” rule, which supports spontaneous self-reproduction by glider collisions in both 2d and 3d, is examined in detail for 2d, and other examples are presented. Further details and results can be found at www.ddlab.com (Wuensche, 2004).

k-totalistic rules

We will consider a subclass of CA rules, the k -totalistic rules (Adamatzky, 1994; Bilott et al., 2003)², where a cell’s update depends only on the frequency of values (colors) in its neighborhood, not their position (figure 4). Because of this, the dynamics conserve symmetry; whatever happens in one direction or reflection can also happen in all others. k -totalistic lookup tables (kcode) are much smaller than the general case, $G = v^k$. The size L of the kcode is given by $L = (v + k - 1)! / (k!(v - 1)!)$. For $[v, k] = [3, 6]$, $L = 28$, as opposed to $G = 729$. For greater $[v, k]$, L increases rapidly. If complex behavior can indeed be found for small $[v, k]$, it is of course worthwhile to think small and deal with short kcode.

The beehive rule

The beehive rule is a multi-value k -totalistic rule with $[v, k] = [3, 6]$. The rule created the snapshots in figures 1 and 2, and spontaneously self-organizes a basic glider which becomes

²Thanks to Antonio Lafusa for introducing this class of rules to us. There is a prior attribution to (Adamatzky, 1994), and his identical class “ATOT”.

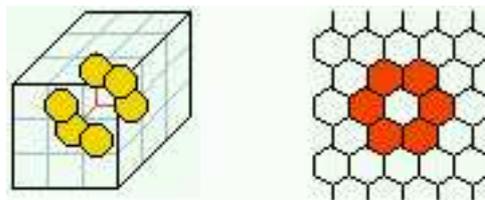


Figure 3: The $k = 6$ neighborhoods of 3d, and 2d hex, CA.

```

kcode = 0022000220022001122200021210
          kcode index
          / totals: 2s+1s+0s=k=6
          / kcode
basic
glider  / / / / / mutations
----- / 2_1_0 / 2_1_0
background-> 0: 0 0 6 -> 0 o c -
head+--> 1: 0 1 5 -> 1 0 - 0
          2: 0 2 4 -> 2 - Sg cg
          3: 0 3 3 -> 1 +- G - G
out4      4: 0 4 2 -> 2 +- - G G
out3      5: 0 5 1 -> 0 +- G G -
out1      6: 0 6 0 -> 0 +- G G -
side2-> 7: 1 0 5 -> 0 c c -
side1-> 8: 1 1 4 -> 2 - c c
side1+   9: 1 2 3 -> 2 - cg G
          10: 1 3 2 -> 2 +- - G G
out2     11: 1 4 1 -> 1 +- G - G
tail     12: 1 5 0 -> 1 +- G - G
head-> 13: 2 0 4 -> 0 c c -
          14: 2 1 3 -> 0 Gs c -
          15: 2 2 2 -> 2 - gc gc
          16: 2 3 1 -> 2 +- - G G
          17: 2 4 0 -> 0 +- G G -
          18: 3 0 3 -> 0 g c -
          19: 3 1 2 -> 2 - c cg
          20: 3 2 1 -> 2 - cg Gd
          21: 3 3 0 -> 0 +- G G -
          22: 4 0 2 -> 0 G c -
center-> 23: 4 1 1 -> 0 g cg -
          24: 4 2 0 -> 2 - cg G
          25: 5 0 1 -> 2 - cg G
          26: 5 1 0 -> 0 g gc -
          27: 6 0 0 -> 0 G Gd -

```

key to mutations:
quasi-neutral G=25/56, wildcards +- 10/28
G/g=gliders, G=same/similar dynamics,
g=weak/different, S=spirals, d=dense,
s=sparse, c=chaos, o=order, 0=all 0s

Figure 4: The lookup table (kcode) of the k -totalistic beehive rule, showing its construction. This also shows the entries that make the basic glider in figure 5, and the consequences of all 56 possible 1-value mutations, 25 of which are quasi-neutral.

the predominant structure in both a cubic 3d and hexagonal 2d lattice, with neighbors as in figure 3; the cell itself is not included in its neighborhood.

The complex dynamics includes self-reproduction by glider collisions (figure 6), and polymer-like gliders and glider-guns (figures 6 and 7), but no permanently static patterns. We chose the beehive rule for closer scrutiny because its self-reproduction is especially clear in a live simulation.

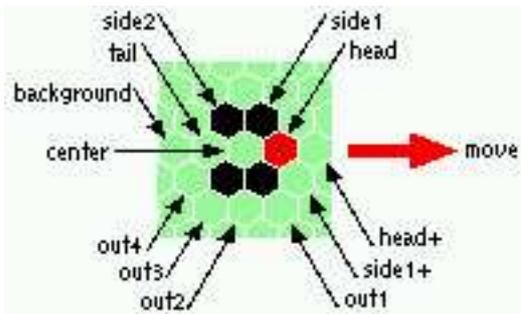


Figure 5: The basic $k = 6$ 2d glider moves in the direction of its red head. Each cell that forms the glider and its surroundings must blink to the correct color at the next time-step according to the kcode. 12 cells are indicated which cover all cases because of symmetries. The cells are controlled by 6 kcode entries in figure 4; mutation of these disrupts the dynamics (except “center”).

Self-reproduction by glider collisions

We will look in some detail at the 2d dynamics, firstly the outcomes of all possible, non-equivalent, types of collisions between pairs of basic gliders, bearing in mind that different direction on the hex lattice, and reflections, are equivalent. Self-destruction, survival, conservation and self-reproduction all occur, depending on the exact point and direction of impact, summarized in the table below. Of the 21 collision types (8 head-on and 13 angular), 4 lead to self-reproduction, where 2 gliders release either 4, 5, or 6 after an interaction phase of several time-steps. Figure 6 shows some examples.

		gliders		
	type	no before	after	
self-destruction:	2->0	10	20	0
one-survivor:.....	2->1	4	8	4
conservation:.....	2->2	3	6	6
self-reproduction:	2->4	1	2	4
	2->5	1	2	5
	2->6	2	4	12

	totals	21	42	31

The glider before/after ratio is 31/42, so if collision types were equiprobable, and ignoring other interactions, we would expect a high population density of gliders to decrease over time; though this is observed in the long run, other structures and interactions make the dynamics more complex. Gliders can crash into the transient patterns following collisions. An isolated red cell, from collision debris, explodes to make 6 new gliders, so outside perturbations, noise, would tend to repopulate the space with gliders; the dynamics in general is robust to noise. Polymer-like gliders made up of sub-units, also emerge.

Most notably, there are a variety of glider-guns³ that eject from 1 to 4 glider streams in different directions.

Some examples of all of the above are given in figure 7. These processes combine with self-reproduction to produce an extremely complex hive of activity.

³Strictly speaking these are a cross between “glider-guns” and “puffer trains” (Conway, 1982)

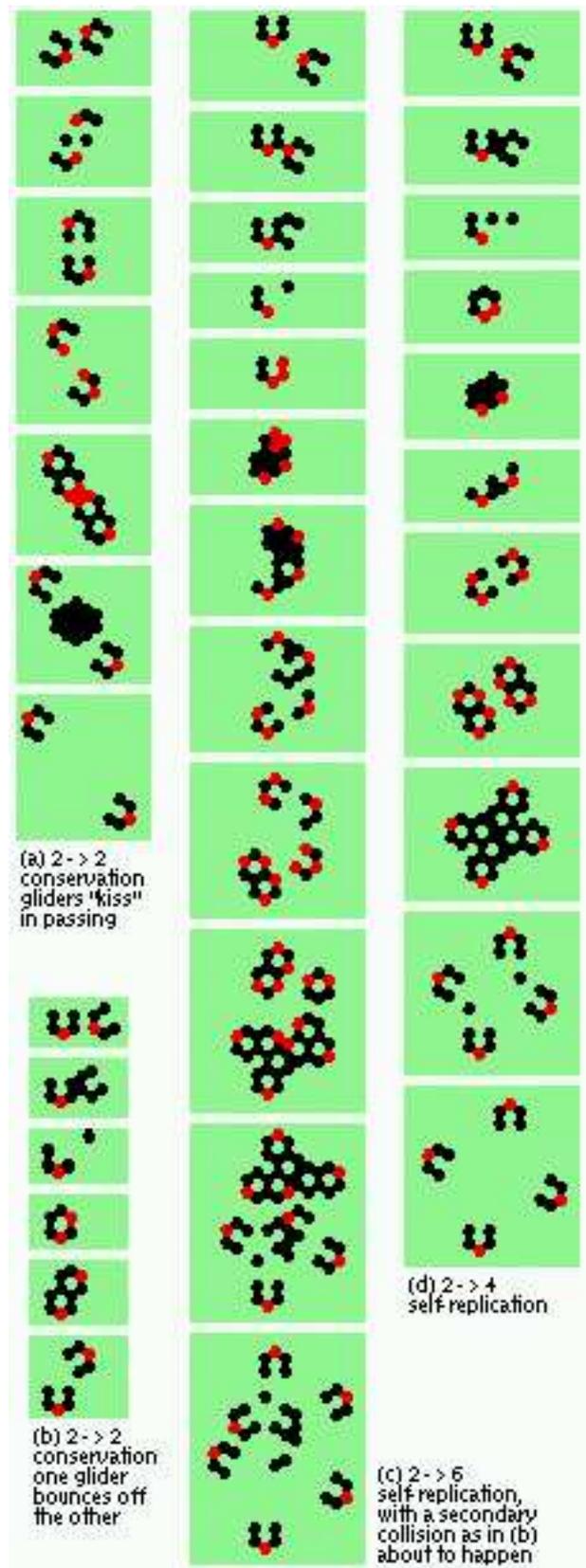


Figure 6: 4 of the 21 types of collisions between 2 gliders (time-steps from the top). Conservation: (a) and (b). Self-reproduction: (c) and (d). For all collisions types see (Wuensche, 2004).

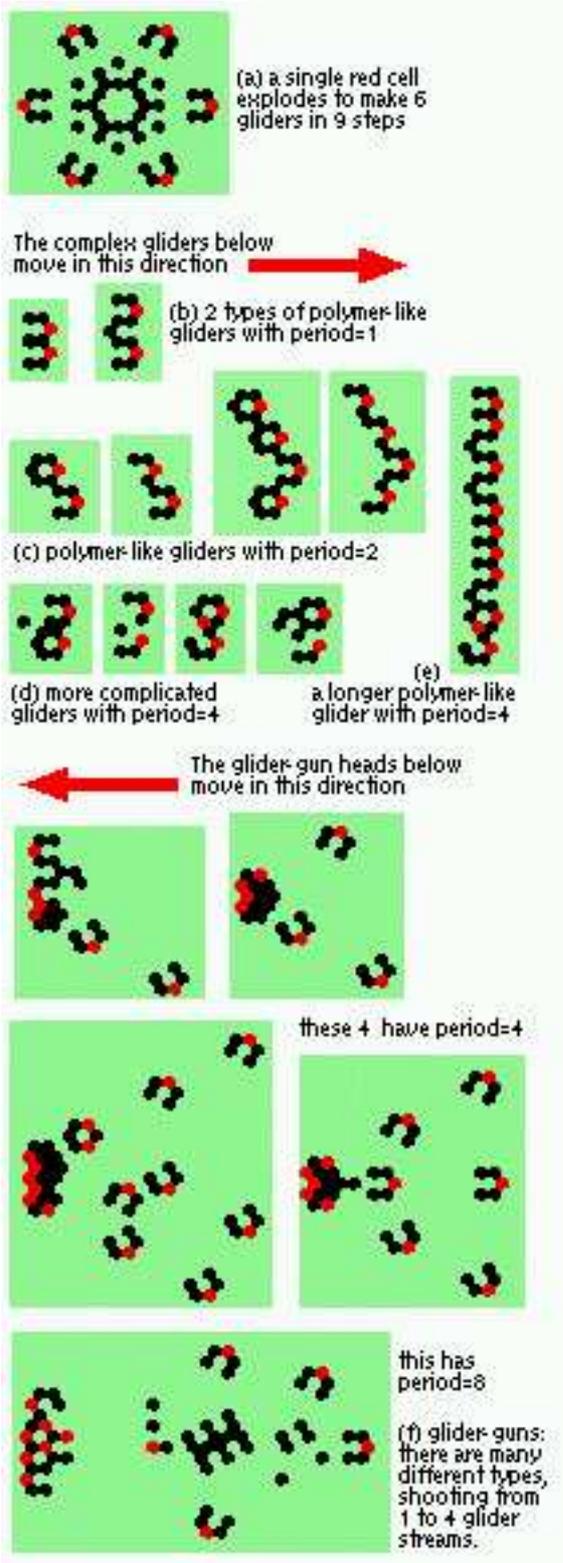


Figure 7: (a) an exploding red cell makes 6 new gliders. (b,c,e) polymer-like gliders made from subunits. (d) a glider that is also present in (e) as a subunit, (e) a longer polymer-like glider made of subunits from (b) and (d). (f) 5 examples of the various types of glider-gun, which shoot from 1 to 4 glider streams. For more examples see (Wuensche, 2004)

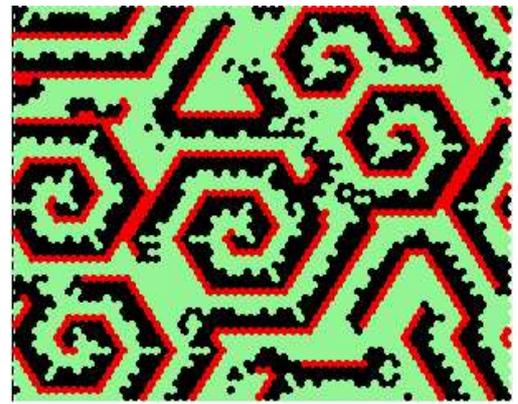


Figure 8: The result of a 1-value mutation to the beehive rule, at index 2 (the output 2 is changed to 1). Glider activity is gradually overwhelmed by spirals. A snapshot on a 2d (60 × 60) hex lattice

Mutations

The consequences of all possible 1-value mutations to the beehive rule are tabulated in figure 4, and snapshots of all can be found at (Wuensche, 2004).

The lookup table has 28 entries, and each can be changed from its present value to two alternatives, giving 56 possible minimal (1-value) mutations. The results of this experiment (Wuensche, 2004) show that for 10 of the entries, changing to either alternative (20 mutations) is quasi-neutral; it appears not to make much difference to the dynamics; experiment confirms that these 10 entries can actually be wild-cards. A further 5 mutations elsewhere, to just one value, are also quasi-neutral, making 25/56. Multiple mutations in these neutral regions needs examining.

On the other hand, mutations to any of the 6 sensitive entries that maintain the basic glider destroy the dynamics - with one exception - a mutation at index 23 (the glider's center) which sets the glider's tail at the next time-step. This mutation closes the glider's tail (a black cell, value 2), but otherwise conserves complex dynamics.

Another interesting mutation is at index 2, which causes glider activity to be gradually overwhelmed by spirals, as shown in figure 8.

The beehive kcode is set out below, indicating these mutations, the 10 wildcards (+), and the 6 glider entries (^),

```

index  23                2
      |                   |
002200+220++200+++220+++210
      ^                   ^

```

It would be possible then, to explore the family of related rules by gradually mutating away from the beehive rule, and entering into the network of related complex rules in rules-space.

Finding complex rules

To find new complex rules from scratch (without mutating old ones), and in particular rules that support gliders, we use

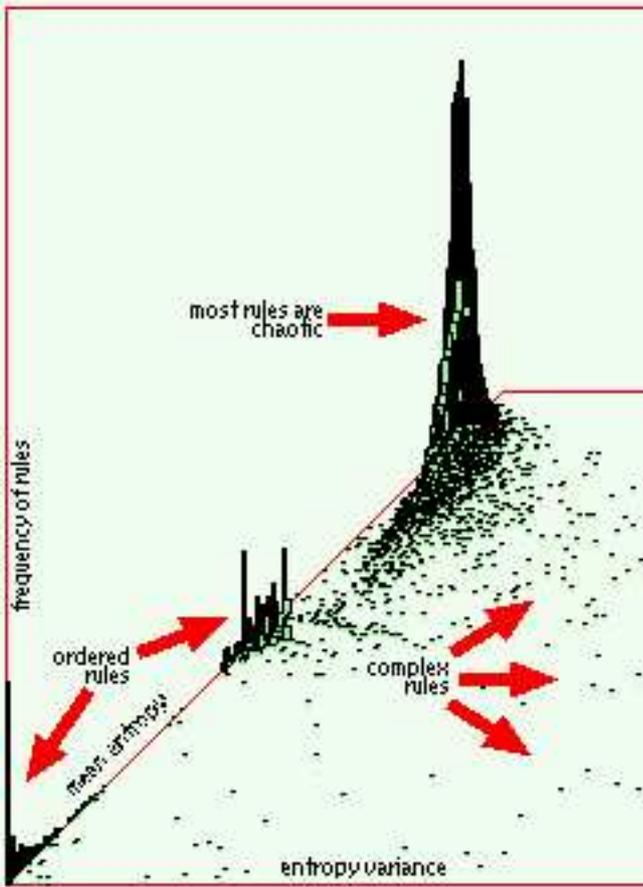
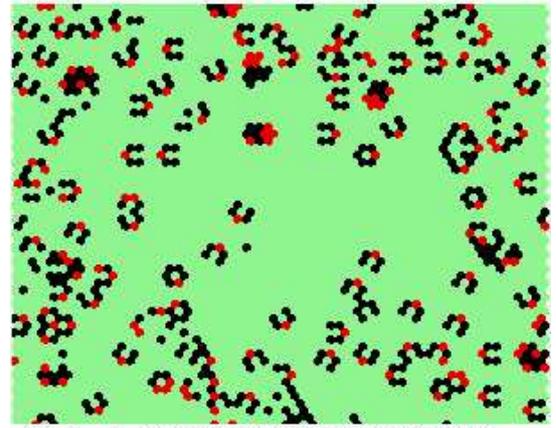


Figure 9: About 15800 $[v,k] = [3,6]$ k -totalistic rules classified by input-entropy variance.

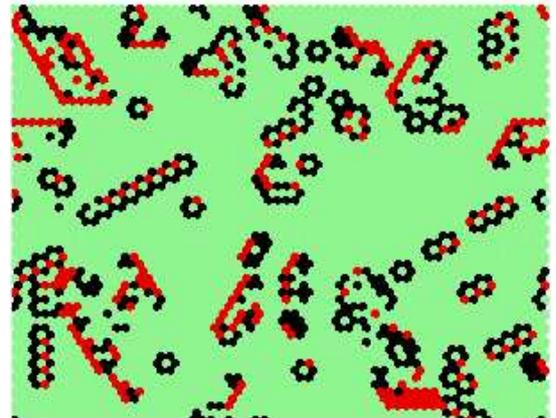
the method for automatically classifying 1d rule-space by input-entropy variance (Wuensche, 1999), but which applies equally well to k -totalistic rules, and to 2d and 3d.

We track how frequently the different entries in the kcode (as in figure 4) are actually looked up, once the CA has settled into its typical behavior. The Shannon entropy of this frequency distribution, the input-entropy S , at time-step t , for one time-step ($w=1$), is given by $S^t = -\sum_{i=0}^{L-1} \left(\frac{Q_i^t}{n} \times \log \left(\frac{Q_i^t}{n} \right) \right)$, where Q_i^t is the lookup frequency of neighborhood i at time t , L is the kcode size, and n is the size of the CA. In practice the measures are smoothed by being averaged over a moving window of $w = 10$ time-steps. The measures are started only after 200 time-steps, and are then taken for a further 300 time-steps. The 2d CA 100×100 is run from a sample of 5 random initial states. The sizes of these parameters can be varied, of course.

Average measures are recorded for (a) entropy variance (or standard deviation), and (b) the mean entropy. This is repeated for a sample of randomly chosen rules. The sample is then sorted by both (a) and (b), and data plotted as in figure 9. The plot classifies rule-space between chaos, order and complexity. Individual rules can be selected by various



(a) kcode=22000021000222201110201212210



(b) kcode=0222200220000200100201102110

Figure 10: Complex rules on a 60×60 hexagonal lattice.

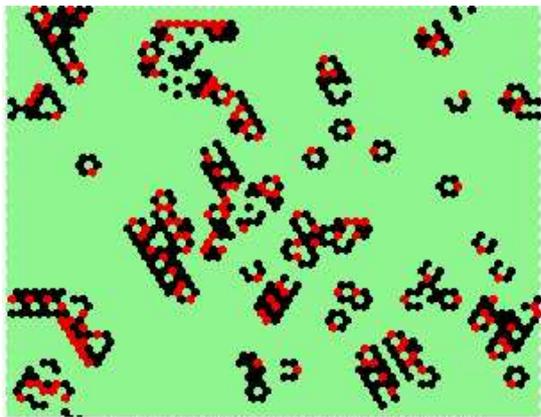
methods, including directly from the plot, to check their behaviors.

The basic argument is that if the entropy continues to vary in settled dynamics, moving both up and down, then some kind of self-organizing collective behavior must be unfolding. This might include competing zones of order and chaos, or two different types of chaos, as well as glider dynamics.

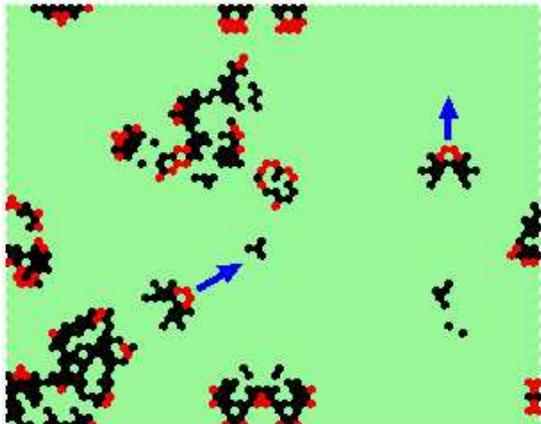
In the case of the beehive rule and other glider rules, at any given moment there may be a bias in the dynamics towards a preponderance of gliders (low entropy) or post-collision transient patterns (high entropy). The lattice (or a patch undergoing the analysis) must not be too large in relation to the scale of possible emergent structures, otherwise the effects would cancel out. By contrast, stable/high entropy indicates chaos (most rules); stable/low entropy indicates order - in both cases the entropy variance is low.

Other complex rules

In figures 10 and 11, we show 4 examples of $[v,k] = [3,6]$ complex rules, found independently by the input-entropy variance method (more can be seen at (Wuensche, 2004), also for $k = 7, 8, 9$). The basic beehive glider is sometimes present, but we also see different gliders and complex struc-



(a) kcode=0200001120100200002200120110



(b) kcode=0200202022222200012100002100

Figure 11: Complex rules on a 60×60 hexagonal lattice. (b) note 2 large slow moving gliders (period=3), their motion is indicated by arrows

tures, which we have not yet examined in detail. The example in figure 11(b) has a remarkably complex glider and glider-gun.

In these examples, the kcode has been transformed with a value-swapping algorithm to an equivalent kcode, but with the colors (values) made to correspond with the beehive rule, where the background value is 0 (green), the leading head of gliders is 1 (red). This allows the different kcode tables to be compared to look for common biases. Below we compare the kcodes of our 4 examples with the beehive rule. The wildcards(+) and glider entries (^) are indicated.

		frequency of values								
26	23			+	++	+++	++++	2	1	0
0022000220022001122200021210								11	4	13
		^				^	^			
2200021000222201110201212210	- 10a							11	7	10
0222200220000200100201102110	- 10b							9	5	14
0200001120100200002200120110	- 11a							6	6	16
0200202022222200012100002100	- 11b							11	3	14
		2	4			34	34			
										- matches

We can see that there is a high correlation with glider entries, except for index 23 which we have already noted is

exceptional; a 2 at index 23 closes the basic glider's tail, and a 2 at index 26 keeps it closed (a black cell, value 2). There is also a correlation with the frequencies of values.

If a common thread or bias in kcodes can be identified among these and other complex rules, which distinguishes them from the vast majority of rules-space, then this could become the basis for an underlying principle of self-organization in k-totalistic cellular automata.

Discussion

There is a network of complex rules in k-totalistic rule-space, connected by mutations, where large scale collective behaviors emerge spontaneously. The complex dynamics includes self-reproduction by glider collisions, polymer-like gliders, glider guns, and possibly other structures and interactions. This implies higher levels of description beyond the underlying "physics", the kcode. The levels could conceivably unfold without limit given sufficient time and space; the number of these emergent levels is our qualitative measure of complexity.

Some questions arise; what is the mechanism of self-reproduction? how do glider-guns self-assemble? are these systems computation universal? how does complexity scale with greater v or k ? how do the various complex rules relate? is there an underlying principle of self-organization? and what is it?

Discrete Dynamics Lab

The software used to research and produce this paper was multi-value *DDL*ab, in which the dynamics can be seen live, and the rules are provided. It is available at www.ddlab.com.

References

- Adamatzky, A. (1994). *Identification of Cellular Automata*. Taylor and Francis.
- Bilott, E., Lafusa, A., and Pantano, P. (2003). Is self-replication an embedded characteristic of the artificial/living matter? In Standish and Bedau, editors, *Artificial Life VIII*, pages 38–48. MIT Press.
- Conway, J. (1982). *What is Life?*, chapter 25 in "Winning ways for your mathematical plays", Vol.2, by Berlekamp, E, J.H.Conway and R.Guy. Academic Press, New York.
- Langton, C. (1984). Self-reproduction in cellular automata. *Physica D* 10, 10:135–144.
- von Neumann, J. (1966). *Theory of Self-Reproducing Automata*. Univ. of Illinois Press. edited and completed by A.W.Burks from 1949 lectures.
- Wuensche, A. (1994). Complexity in one-d cellular automata. Santa Fe Institute working paper 94-04-025.
- Wuensche, A. (1999). Classifying cellular automata automatically. *COMPLEXITY*, 4/no.3:47–66.
- Wuensche, A. (2004). www.ddlab.com (follow the links to "self-reproduction" and "dd-life").