8.4 The Edge of Chaos

The idea of "the edge of chaos" originated with a few simple computational experiments by Packard (1988b) and Langton (1990), and quickly became part of the lexicon of complex systems. These experiments suggested that systems poised at the edge of chaos had the capacity for emergent computation. The intuition behind this claim has tremendous appeal: systems that are too simple are static and those that are too active are chaotic, and thus it is only on the edge between these two behaviors where a system can undertake productive activity. In its most grand incarnation, the edge of chaos captures the essence of all interesting adaptive systems as they evolve to this boundary between stable order and unstable chaos.

The notion of the edge of chaos is both intuitively appealing and metaphorically rich. While the examples of Packard and Langton are intriguing—and scientific progress often proceeds from examples, to ideas, to understanding—they may also be misleading, lacking adequate logical underpinnings. That said, there remains a grain of truth in the basic intuition that, loosely speaking, complexity lies somewhere between order and chaos.

Below we explore the edge of chaos in a simplified framework. Our goal is to begin to solidify the boundaries of our understanding of this concept through a more formal analysis. We will investigate two questions: is there an edge and, if so, is it important? If the idea does have merit, then its impact on our understanding of complex adaptive social systems could be substantial.

8.4.1 Is There an Edge?

A key first step in understanding whether or not there is an edge of chaos is being careful about defining the space that we are exploring. In models of the edge of chaos there is an attempt to detect whether *a given rule* will tend to imply a system that is either chaotic, stable, or poised somewhere in between. A casual reading of this statement (without the added emphasis) often causes the misleading perception that the focus here is on the implied phase space of the rule rather than the rule itself.

The "edge" in the edge of chaos is not in phase space but in the space of rules. The idea is that if we slightly perturb a rule that generates complexity we will get a rule that either generates chaos or stasis. Therefore, the search for the edge of chaos focuses on how small changes in a rule impacts its

Situation	**Left Neighbor**	**Self**	**Right Neighbor**	**Rule 110**
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Table 8.11: Rule 110.

behavior.

Below we will discuss the edge of chaos in terms of a one-dimensional, two-state, nearest-neighbor cellular automata. In such a system, a rule lies at the edge of chaos if small changes in its rule table move it back and forth between chaotic and non-chaotic behavior.

Our investigation requires us to classify each rule according to the behavior it generates. As previously discussed, Wolfram (2001) classifies automata as either being fixed (Class I), periodic (Class II), chaotic (Class III), or complex (Class IV). Li and Packard (1990) use a slightly different and finer classification. For our purposes, the important difference between these two classifications is that Li and Packard pay special attention to rules that generate two period cycles, what we will call *blinkers*. Li and Packard's definition of complex is also worth noting. Like Wolfram, they define a rule as complex if the time it takes to get to the limiting distribution is long, but in addition they require that this time increase linearly with the number of cells.

Consider Rule 110 shown in Table 8.11. This rule is classified as complex by both Wolfram and Li and Packard. To see whether or not it is poised at the edge of chaos, we need to define the set of neighboring rules. The most obvious notion of neighborhood here is to consider all the rules that have a rule table that differs by only one situation. Using this convention Rule 110 has the eight neighboring rules given in Table 8.12.

Is Rule 110 at the edge of chaos? The last two rows of Table 8.12 reveal

Situation	**Rule 111**	**Rule 108**	**Rule 106**	**Rule 102**	**Rule 126**	**Rule 78**	**Rule 46**	**Rule 228**
0 (000)	1	0	0	0	0	0	0	0
1 (001)	1	0	1	1	1	1	1	1
2 (010)	1	1	0	1	1	1	1	1
3 (011)	1	1	1	0	1	1	1	1
4 (100)	0	0	0	0	1	0	0	0
5 (101)	1	1	1	1	1	0	1	1
6 (110)	1	1	1	1	1	1	0	1
7 (111)	0	0	0	0	0	0	0	1
λ	3/4	1/2	1/2	1/2	3/4	1/2	1/2	3/4
Wolfram	II	II	III	III	III	I	II	I
Li-Packard	2-C	2-C	Ch	Ch	Ch	F	F	F

Table 8.12: Neighbors of Rule 110 and their respective classification measures. With Wolfram class and Li-Packard classification (where 2-C is 2-cycle, Ch is chaotic, and F is fixed).

that three of its eight neighbors are chaotic[9] and three lead to fixed points. Thus, this rule appears to be poised between chaos and stasis. Superficially, at least, this seems to indicated that Rule 110 is on the edge of chaos.

Langton's (1990) model of the edge of chaos classifies one-dimensional cellular automata according to a single parameter, λ. In his model each site has s possible states and is connected to k neighbors in both directions. The λ value for a given rule equals the percentage of all rule table entries that map into some predefined *quiescent* state. Thus, if all rule table entries map into the quiescent state, then λ will equal one. In this case, the system will immediately freeze in the quiescent state. In his experiments, Langton explored randomly generated rule tables and tried to connect the λ value for a given rule table to a measure of subsequent system activity. He found that λ had some explanatory value—as it decreased from one, the average behavior of the implied systems went from rapidly freezing, to long transients, to chaos. For Langton, the edge of chaos was the value of λ at which the average behavior first showed evidence of chaos.

[9]The usual caveats of rule classification apply, for example, all three chaotic neighbors have fixed points at all zeros.

It is not too surprising that chaotic system behavior can be roughly correlated with a parameter like λ. As a crude approximation, suppose that there are only two possible states and that at any time step the state of each site is randomly chosen.[10] With random sites the λ parameter gives the probability that any given site will be mapped to the quiescent state. Thus, when λ approaches either of its extreme values (zero or one) the sites are likely to lock into the quiescent state. As λ gets closer to $1/2$, chaos will reign as each site will have an equal chance of being either value. Thus, an "edge" associated with monotonic changes in λ will appear between these two types of behaviors.

This crude approximation of what occurs as λ varies can be refined by looking at particular rules in more detail. Our approach here is to explore the entire space of two-state, nearest-neighbor cellular automata rules. Of the two hundred and fifty six rules of this type, thirty two are classified as chaotic. Since there are eight rule table entries for this type of cellular automaton, λ values are of the form $j/8$, where j belongs to $\{0, 1, \ldots, 8\}$. The number of rules associated with each λ value varies widely, from one rule for λ equal to 0 or 1, to seventy rules when λ equals $1/2$. In Table 8.13 we show the number of possible rules for each λ and the number of such rules that are classified as chaotic and complex.[11] The table shows that the chaotic rules are strongly biased toward the middle of the distribution of λ (with most having a λ value of one half). The table also shows the distribution of the six complex rules in this space. As expected, these rules too are biased toward the middle of the λ distribution.

Given symmetry, both zero and one can be thought of as quiescent states. Therefore, a λ of $5/8$ and one of $3/8$ are equivalent, and so on. If we define $\hat{\lambda}$ to be the minimum of λ and $(1 - \lambda)$, we can condense Table 8.13 into Table 8.14. Using this data, we can compute the average $\hat{\lambda}$ for the different types of rules. Chaotic rules have an average $\hat{\lambda}$ of 0.44 and complex rules average 0.46. Thus, Langton's inference, namely that complexity occurs at the edge of chaos, seems to hold in this example.

However, the edge that appears in the aggregate data is not apparent in the individual cases. That is, there are multiple edges, not just a single

[10] We know, of course, that as time passes there is often much more structure to the sites (for example, triangles and such), but for the moment we will ignore this issue.

[11] The chaotic rules are: 18, 22, 30, 45, 60, 73, 75, 86, 89, 90, 101, 102, 105, 106, 109, 120, 122, 126, 129, 135, 146, 149, 150, 151, 153, 161, 165, 169, 182, 183, 195, and 225. The complex rules are: 54, 110, 124, 137, 147, and 193.

λ	**All Rules**	**Chaotic Rules**	**Complex Rules**
0	1	0	0
1/8	8	0	0
1/4	28	2	0
3/8	56	4	1
1/2	70	20	4
5/8	56	4	1
3/4	28	2	0
7/8	8	0	0
1	1	0	0

Table 8.13: λ-Distribution over chaotic and complex rules in the space of two-state, nearest-neighbor, one-dimensional cellular automata.

$\hat{\lambda}$	**All Rules**	**Chaotic Rules**	**Complex Rules**
0	2	0	0
1/8	16	0	0
1/4	56	4	0
3/8	112	8	2
1/2	80	20	4

Table 8.14: $\hat{\lambda}$-Distribution over chaotic and complex rules.

one. For example, consider the neighbors of Rule 110 (see Table 8.12). If the picture of an easily tuned world existing between complexity on the one hand and chaos or stasis on the other was accurate, then we would expect that the neighbors with λ equal to 1/2 would generate chaos and those with λ equal to 3/4 would be cyclic or stable. Yet, we see from Table 8.12 that this is not the case. When λ equals 1/2, two rules are chaotic, two are fixed, and one is cyclic. When λ equals 3/4 one rule with each type of behavior exists.

To investigate this issue a bit more, consider Rule 126 in Table 8.12. This rule has a λ value equal to 3/4 yet it is chaotic. A look at its rule table indicates that it produces a one unless the site and its neighbors all agree, in which case it goes to zero. While this rule could potentially lock into all zeros, its general tendency is to propagate ones in the system while eating away at the edges of any long strings of zeros. Moreover, as soon as three consecutive ones appear they are destroyed by having a zero inserted at their center. As long as the system begins at any configuration other than all ones or all zeroes, Rule 126 tends to create ones and then destroy them, resulting in long, convoluted cycles of activity.

Now consider Rule 46, which has periodic behavior with a λ of 1/2. Some initial intuition can be gained by considering the behavior of the rule after one time step. To do so, we can look at a neighborhood of size five (which has thirty two possible configurations) and track what happens to the middle three sites. We find that the sequence 010 is not a possible outcome—that is, after one iteration of the rule we can never have a single one bordered by zeros. This means that any future configurations that require isolated ones will also be ruled out after the second iteration, and in the case of Rule 46 this rules out the sequence 111. Therefore, after two periods, we will never see the sequences 010 or 111, and we can reduce the relevant parts of the rule table to that given in Table 8.15. In this restricted domain, Rule 46 is much simpler: copy the site to the right. Such a rule is Class II according to Wolfram and fixed based on Li and Packard—in either case, it is not chaotic.

The failure of a nicely behaved λ-edge in the neighborhood of Rule 110 is not particular to that rule. Every one of the rules classified as complex in this space has at least one chaotic neighbor with a lower λ value and one with a higher value. Therefore, while it is true that complex rules have chaotic edges, they do not lie poised at the edge of chaos in the traditional sense implied by λ. That being said, the edge of chaos idea is not wrong *per se*, as complex rules do appear to lie next to chaotic rules. However, collapsing a

Situation	**Rule 46**
0 (000)	0
1 (001)	1
2 (010)	–
3 (011)	1
4 (100)	0
5 (101)	1
6 (110)	0
7 (111)	–

Table 8.15: Relevant rule table for Rule 46 after two iterations.

multi-dimensional phenomenon onto one dimension obscures the details. In the case of these simple automata, what drives complexity, chaos, and order, is the microstructure within a rule such as we saw in Rule 46.

To show how microstructure undermines attempts to create a a "complexity dial," we can partition the set of rules into four equal-sized groups (with sixty four rules in each) based on how the rule maps the sequences 000 and 111. We will name each sequence based on the implied behavior when we have a string of all zeros or all ones. Let the *identity rules* be the ones that map 000 to 0 and 111 to 1. We call rules that map 000 to 1 and 111 to 0, *blinker rules*, as a string of all ones goes to all zeros then back to all ones and so on. The third and fourth sets are those rules for which 000 and 111 both get mapped to either 0 (*0-attractor rules*) or 1 (*1-attractor rules*). In these latter rules, a string of identical elements falls into, and remains in, the attractor state after one time period. Both identity and blinker rules have an identical distribution of the number of ones in the rule tables (with one rule each having one or seven ones, six having two or six ones, fifteen having three or five ones, and twenty having four ones). The two attractor rules have the identical distribution except for the number of ones is offset either down by one (0-attractor rules) or up by one (1-attractor rules). The λ distribution for each class is directly tied to the distribution of ones.

Table 8.16 classifies the rules within each of the above four sets using Wolfram's classification. Note that the blinker rules are far more likely than the other types of rules to be periodic. This is not surprising, as once long sequences of zeros or ones emerge, they inherently embody two-period cycles. Indeed, of the fifty blinker rules that lead to periodic behavior, forty six of

Rule Set	**Class I: Fixed**	**Class II: Periodic**	**Class III: Chaotic**	**Class IV: Complex**
Identity	48	13	3	0
Blinker	7	50	7	0
0-Attractor	29	17	11	3
1-Attractor	29	17	11	3

Table 8.16: Classification of our rule partition into behaviors.

them generate cycles of period two.[12] Also note that the identity rules and the attractor rules tend to be far more stable then the blinker rules. Again, given the tendency of these rules to lock-in long sequences of zeros or ones, this is not surprising.

The one potentially puzzling result revealed in Table 8.16 is that all of the complex rules belong to the set of attractor rules. Recall that attractor rules map 000 and 111 into the identical value, and thus it would seem that these rules are the most likely candidates for boring behavior among the four sets. When we look at the complex rules within these sets, we find a common characteristic, namely that their rule tables are heavily biased toward the opposite value that is being mapped to under 000 and 111. This leads to an inherent amount of, in Schumpeter's words, creative destruction. Long sequences of identical values that are induced and stabilized by either the 000 or 111 mappings are destroyed at the edges by the other elements of the rule table. During this destruction, long sequences of the opposite value begin to accumulate, and these form the basis for the creation of the original values. This process of creative destruction results in long transients, where one value is churned into another. This also explains why the identity rules do not generate complex behavior, as long sequences are stabilized rather than destroyed under such rules.

Our decomposition into the four sets above also proves useful when thinking about neighbors and the edge of chaos. Six of the eight neighbors of any rule belong to the same set within the decomposition. Thus, six of the eight neighbors of an identity rule, will themselves be identity rules and thereby likely to generate fixed points. None of an identity rule's immediate neigh-

[12]In contrast, of the thirteen identity rules that are periodic, only nine generate period-two cycles.

bors can be blinker rules (as this requires flips in the rule table for both 000 and 111), and therefore stability does not appear likely to lie at the edge of periodicity. Attractor rules, which have exactly one identity- and one blinker-rule neighbor, tend to border attractor rules. More than two thirds of the chaotic rules are attractor rules and all of the complex rules fall into this set. This partly explains why complex rules have chaotic edges, as the complex and chaotic rules belong to the same set. Therefore, our decomposition (by construction) lends support to the notion of an *edge of chaos*, but not a single edge as usually supposed, but rather a multitude of edges contained within the set of attractor rules.

A crude measure like λ has its uses and limitations. As the above analysis indicates, knowing the λ value of a particular rule does not necessarily tell us much about that rule's behavior. The λ value is, however, probably a good way to identify broad areas of the rule space that might harbor the potential for interesting behavior. In this view, λ is a necessary but not sufficient condition for interesting behavior. Note that the analysis above was confined to very simple automata (one-dimensional, two-state, with one-nearest-neighbor), and we know that such systems may have limitations (though sometimes even advantages) over more complicated systems. That being said, we suspect the tenor of the insights above are relatively general.

8.4.2 Computation at the Edge of Chaos

For most physical, biological, and social processes, an important aspect of rule behavior is whether the system can undertake productive behavior. The productive behavior we are concerned with here is the ability of a system to solve a computational problem. The measures of system behavior we used above were ways to indirectly capture an automaton's ability to compute answers by tracking its ability to transmit and process information. In this section we look at the process of computation more directly.

Cellular automata become computational systems when they produce "answers" to "questions." In these systems the questions are posed by setting up initial conditions and then activating the automata. Answers come from some interpretation of the system after it has undergone sufficient iterations. For example, if the system must determine a binary answer to some question, we might allow the automaton to run and then take the state of a randomly chosen site as the answer. A well known example of such a compu-