Getting to terms with complexity in Cellular Automata

by Markus Echterhoff

December 8, 2010

Contents

Abstract

Because of their simplicity and capacity to display interesting global behavior emerging from local interaction, cellular automata (CA) have been extensively studied as a model for complexity. In this assignment, I explore attempts at capturing that what makes a CA complex, or "interesting".

The purpose of this exploration is for me to gain more insight into the nature of CA, how they can be defined and what makes them interesting, with special interest in finding objective measures of complexity that match our subjective notion of complexity being not too ordered and not too chaotic. This is done by stepping chronologically through select publications in the field of CA research.

Input-Entropy variance is explored experimentally as an objective measure of subjective experience of the order-chaos relation.

1 Introduction

Science is done with the tools and the language available. In the early history, CA were looked at using analytical mathematics or engineering of specific CA. The first attempt at *general* classification of CA that I am aware of, came about when the tools for massive simulation were available to a wide enough academic public, with Stephen Wolfram in the mid '80s. While still referring to CA as mathematical constructs in a very analytical way, he deducts the classification scheme from gathering extensive statistics from simulations, enabling statistical, qualitative classification for the first time. Although seemingly unpublished and by far less general, the schema of David Eppstein offers a refreshing approach to classifying CA, that captures some aspects of CA which are similar to Conway's Game of Life, that Wolfram's classes do not. Chris Langton, in the beginning of the '90s, using the language of modern computer science, provided a measure of complexity of CA that includes the classes discovered by Wolfram in a spectrum of complexity. In the late '90s then, Andrew Wuensche did extensive research using the knowledge of Wolfram and Langton, uncovering more measures of complexity and means of automatically classifying CA by these measures. Wuensche complements the mathematical and computational view on CA with insights from dynamical systems analysis.

Due to time constraints, some work on CA is not explicitly covered even though influential at least for later research. This includes but is not limited to J.P. Crutchfield and H.A. Gutowitz as well as related work in the fields of dynamical systems, especially discrete dynamical systems such as S.A.

Kauffmann's Boolean Networks.

2 Defining Cellular Automata

Cellular automata have been formally described in mathematical [3] and computational [2] terms. Both vocabularies enabled the authors to describe their observations. Since I am writing about both classification schemata (and then some), it does not seem sensible to use one or the other definition here. I could introduce both and switch between them, but that seems confusing rather than helpful. Therefore, I shall introduce CA in natural language and introduce formal language when it is needed to understand specifics I wish to elaborate on.

A cellular automaton is a lattice of connected things that have state. These things are usually referred to as *cell* and their state is usually discrete. The states of all cells change dynamically and synchronously in discrete time, according to an external specification of local interaction. This means that the state of any cell in the next time step holds a value calculated based on the states of other nearby cells in a certain radius, at the current time step. This specification has described as a function (mathematical vocabulary), a look-up table (computational vocabulary) or a set of rules (causality in natural language). From this local interaction, patterns emerge that we seem to be somehow attracted to and that are suspected to be a model of diverse processes in all kinds of scientific fields.

CA can also be seen as the special case of boolean networks with a special scale free topology as illustrated in figure 1.

3 Wolfram classes

A classification scheme for cellular automata was introduced by Wolfram[3]. According to Wolfram, cellular automata fall into four classes now also called "Wolfram classes". The first class consists of automata which evolve into states where all cells have the same value for all but exceptional initial configurations. These are said to be a discrete equivalent of nonlinear dynamical systems which have a fixed point attractor. Class 2 CA evolve into "a set of separated simple stable or periodic structures"[3] for all but exceptional initial configurations. That is, they transform initial configurations and "act as a filter"[3]. The dynamical systems analogue here are systems attracted to limit cycles. Class 3 is where things become chaotic. Wolfram describes this class to refer to CA where the value of a single site depends on a number of

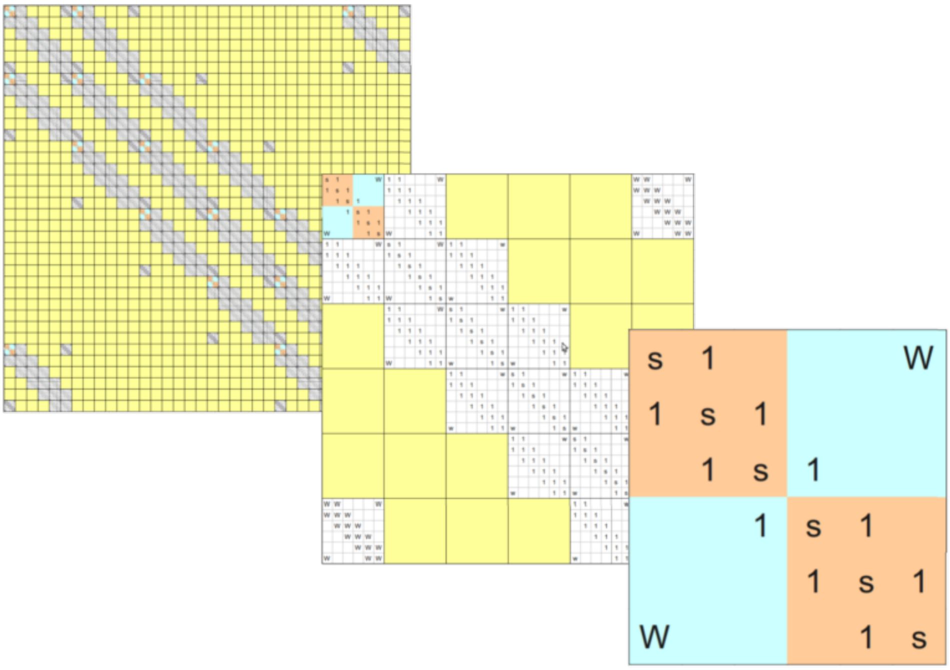


Figure 1: Illustration of CA as boolean networks with scale-free topology. The structure of the network connectivity is repeated as spacial dimension is increased. The rightmost graphic illustrates the connectivity matrix of a 1D CA with 6 cells and neighborhood size 3^1. The structure is repated as dimensions increase from 1 to 2 to 3 and neighborhood size from 3^1 to 3^2 to 3^3. ('1' marks a regular connection, 's' a connection to the same node and 'w' stands for a wrapped connection due to periodic boundary conditions.

initial values for itself and other cells. As the system develops over time, the value of a given site depends on more and more initial values of other sites and thus is considered chaotic. However, defining the upper bounds of this class, Wolfram states that given the initial configuration, the state of a given site "may be determined by a simple algorithm". Further, "after sufficiently many time steps, the statistical properties of these patterns are typically the same for almost all initial states". The dynamic system equivalent here is a strange attractor. Class 4 then holds CA which cannot be reduced and are therefore universal in that the only way to find out whether a certain configuration in a class 4 cellular automaton is reachable, is to actually run it. No other algorithm can be used to predict whether a CA in this class reaches a certain state (reminiscent of the halting problem for Turing machines). Statistical properties do not stay the same as complex, sometimes long lived structures can emerge and govern the evolution of the CA. No attractor can be specified for these systems.

A degree of uncertainty in the classification becomes apparent when reading [3]. Carefully inclusive formulations are used as to what extend a given class exhibits certain mathematical characteristics. Even for the seemingly uninteresting class 1, Wolfram allows for variance which goes so far as to allow to "enter non-trivial cycles" instead of the proclaimed homogeneous equilibrium. These configurations are said to decrease as the number of sites increases so that the strict classification for class 1 CA holds for infinite size automata, which is why these CA still belong to class 1 despite their class 2 behavior for "exceptional" initial configurations. Since infinite CA cannot exist in finite space, no implementation of a class 1 CA should be expected to exhibit class 1 behavior for all initial configurations. It seems to me that Wolfram forced certain CA into his classes rather than have them "fall" into them, as he claimed.

4 Thinking differently - a schema by Eppstein

Eppstein formulated a classification scheme that operates not on the complexity of CA but on two simple characteristics [1].

Its purpose is to give a more precise estimate for whether or not a given CA is capable of *supporting* gliders rather than insisting on having them emerge from unordered initial configurations. The support of gliders is insofar important as these are seen as an indicator for Turing-universality. Wolfram, according to Eppstein, did not consider ordered initial states. The argument is that for a sufficiently large number of cells these ordered states might occur in unordered initial conditions. In [3] this seems covered by the vague formulation that the classification holds in "almost all" cases, conveniently declaring cases such as those discovered by Eppstein as "exceptional".

In Eppstein's schema, CA are differentiated by their possibility of both expansion and contraction to be present in the same automaton. Expansion means an off cell in a Life-like CA to switch on if a number of its neighboring cells are on. Contraction allows it to switch to off again. The interaction of expansion and contraction allows for gliders to exist.

An argument has been made by Tyler and acknowledged by Eppstein [1] that non-contracting CA can exhibit universality. However, the impact of this observation seems minor as it involves viewing a 2D evolution of a universal 1D CA as a CA on its own right. In doing so, the CA becomes spatially infinite and can therefore be seen as much a mathematical shenanigan as Wolfram's elaboration on CA with infinite number of cells (as in [3]).

However Eppstein's system is very limited as it only accounts for Life-like, outer- (or as he termed it semi-) totalistic CA with a neighborhood of radius 1. It can be conceived of a 2D CA that can contract in only one dimension and expand only in the orthogonal direction. The CA rule allows for both, but no gliders could be sustained, let alone emerge. Eppstein's schema does not consider these CA while they are included in the Wolfram classes. As totalistic rules might be an interesting subset of CA rules, they form a minority of the rule space. Therefore, Eppstein's schema while being an interesting alternative (and therefore listed here) is not universal enough to be useful for the general study of CA.

5 Langton's λ parameter

Following Wolfram, Langton studied the behavior of CA statistically. However, he describes CA using a computational vocabulary rather than a purely mathematical one. This seems sensible as the goal is to search for the foundations of Turing-universality, or simpler: computation. In his own words: "Under what conditions will physical systems support the basic operations of information transmission, storage, and modification constituting the capacity to support computation?"[2]. To achieve this, he introduced the λ *parameter*. For his analysis, he considered CA with periodic boundary conditions only, so that any cell properties are equal among all cells.

For a given CA rule Δ, let K be the number of possible states of a cell and N be the number of neighbors considered in determining the next state, then the total number of possible states for a cell after one transition is K^N. An arbitrary state is then picked and named the *quiescent* state s_q. Let n be the number of possible transitions that Δ makes possible from any of the

K states to s_q in a single time step, then λ is defined as

$$\lambda = \frac{K^N - n}{K^N}.$$

Put simply, if we look at a specific CA rule, then λ is exactly the probability of this rule to reach any state other than s_q in one time step. This means that λ roughly matches our subjective impression of order being "things staying the same". The smaller λ is, the faster things will "be the same" in the CA development.

An important additional part of the definition of λ is the assumption that the $K - 1$ non-special states are picked randomly with equal chances. Imagine a CA with $K = 10$ states referred to by integers ranging from 0 to 9 and $s_q = 0$. If the transition rule was biased to one of the other states, we would quickly label the system ordered as most cells assume this state while λ would be very large, which, according to Langton, correlates to the system being chaotic. Therefore, for all measurements of λ, a sufficiently large set of rules must be sampled for biased rules to cancel each other out. Langton uses lower bounds of $K \geq 4$ and $N \geq 5$ for λ to become a descriptive measure of complexity. While this does not lessen the significance of the λ parameter for the study of emergent computation in general, it raises the question as to whether measures could be found that remain viable for small transition tables.

Besides this limitation, using the λ parameter enables us to gain insights into CA that have been clouded by the shortcomings of Wolfram's classification system. λ is used to impose an order on CA rules that correlates to subjective differentiation between order and chaos. On the left hand side of this spectrum, the CA evolution quickly settles for fixed points or limit cycles in the state space, with little to no influence of the initial configuration. Viewing the initial configuration as information that is to be processed by the CA, this information is lost. On the right hand site, the system behaves chaotically. Information, again, is lost. In agreement with Wolfram, Langton placed most CA rules on both ends of the spectrum, with only a few remaining near a point that corresponds to Wolfram class 4, a class that is truly a point, rather than a spectrum. CA rules with λ near this critical value, become very sensitive to the initial configuration. Information is processed and gives rise to complex behavior. Figure 2 illustrates the λ spectrum and the mapping of the Wolfram classes.

One important thing the λ parameter captures is the length of transients in state space that a random CA displays. CA with $\lambda = 0.5$ are said to have transients of arbitrary length[2]. Arbitrary here means that there exists a strong (quadratic) correlation between the number of cells in the CA and

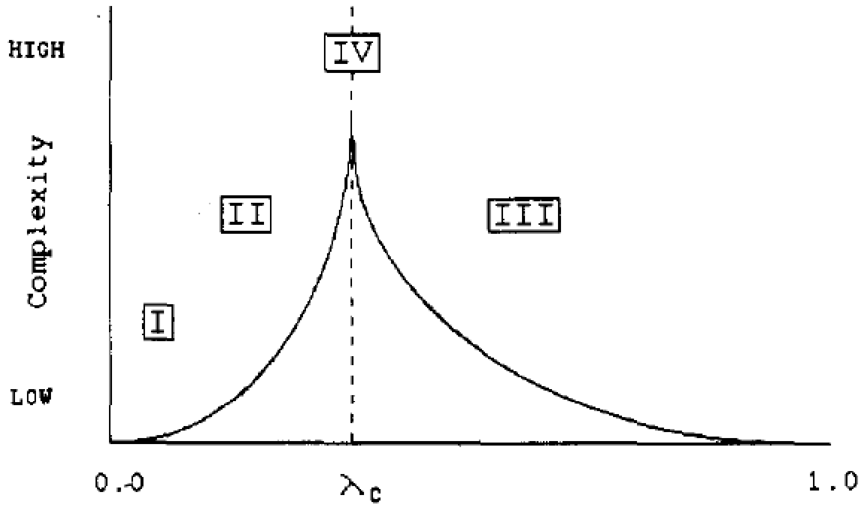


Figure 2: Locating the Wolfram classes in Langton's λ spectrum. Source: [2]

the length of transients at $\lambda = 0.5$. Figure 3 illustrates both the correlation of λ and transient length and that of CA size for CA for which $\lambda = 0.5$.

One other particularly interesting observation by Langton is a relationship between the Shannon-Entropy and λ. Investigation of this relation lead him to the conclusion that there exists a phase transition between order and chaos. This is best illustrated with a plot of Shannon-Entropy against λ as in figure 4. The observation that there exists a phase transition is deducted by a lack of CA rules with entropy around $H = 0.5$. This gap is the biggest at $\lambda = 0.5$, where low values for H cease to exist (in λ dimension). For lower values of λ, the entropy is generally on the lower end of the scale and for values higher than that, usually on the upper end.

I wish I had more time to investigate Langton's use of Shannon-Entropy because I have my doubts about it; e.g. did Langton not investigate the whole spectrum of λ, the missing spectrum of entropy might be hiding there? I am overall skeptical as to what extend Shannon-Entropy as used in [2] correlates to our subjective notion of order and chaos, because high Shannon-Entropy equals efficient information encoding. This would imply that chaos is the most efficient way to encode information, only that we as receivers are too stupid to decode it. It might not have been Langton's intention at all to correlate his findings to human experience, but because CA generate lots of patterns and we are good enough at pattern matching to crack captchas on the Internet, it might be sensible to do so. Subjective terming might open

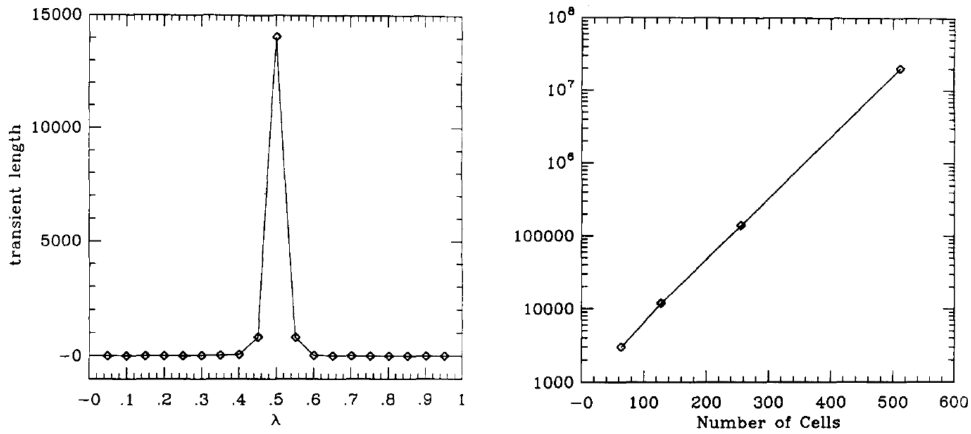


Figure 3: The relation between λ and transient length (left). CA with $\lambda = 0.5$ have a quadratic correlation of the size of the CA and the average length of transients (right). Source: [2]

some doors, as in "interesting" and "uninteresting", because order and chaos are equally uninteresting. Understanding what is interesting and why it is interesting might enable us to view what has been said so far differently. Unfortunately, I cannot think of a way to put enough numbers and Greek letters into this observation to justify elaborating on it in a mathematics homework any more than I already did.

6 Automatic classification with Wuensche

Andrew Wuensche has done research on classifying CA and found a measure of complexity that he claims also correlates to our subjective experience of complexity[5]. He adopted Langton's classification as a "readjustment" of the Wolfram classes into the three classes *ordered* (W. class 1, 2; low λ), *complex* (W. class 4; $\lambda \sim 0.5$) and *chaotic* (W. class 3; high λ)[4].

Wuensche[4] offers several measures of complexity in CA. Using a reverse algorithm, he constructed the state space of CA as what he called a *field of basins of attraction* which can be seen as a discrete equivalent of phase space diagrams. In fact, they seem more convenient as their real valued counterparts as it is possible to fully map out the dynamics of a given CA of arbitrary size and neighborhood in two dimensions, resulting in useful (and pretty) graphs such as figure 5. These plots allow direct measures such as the *G-density*, meaning the density of *Garden of Eden states*, i.e. states without predecessor that are found by the reverse algorithm. Another

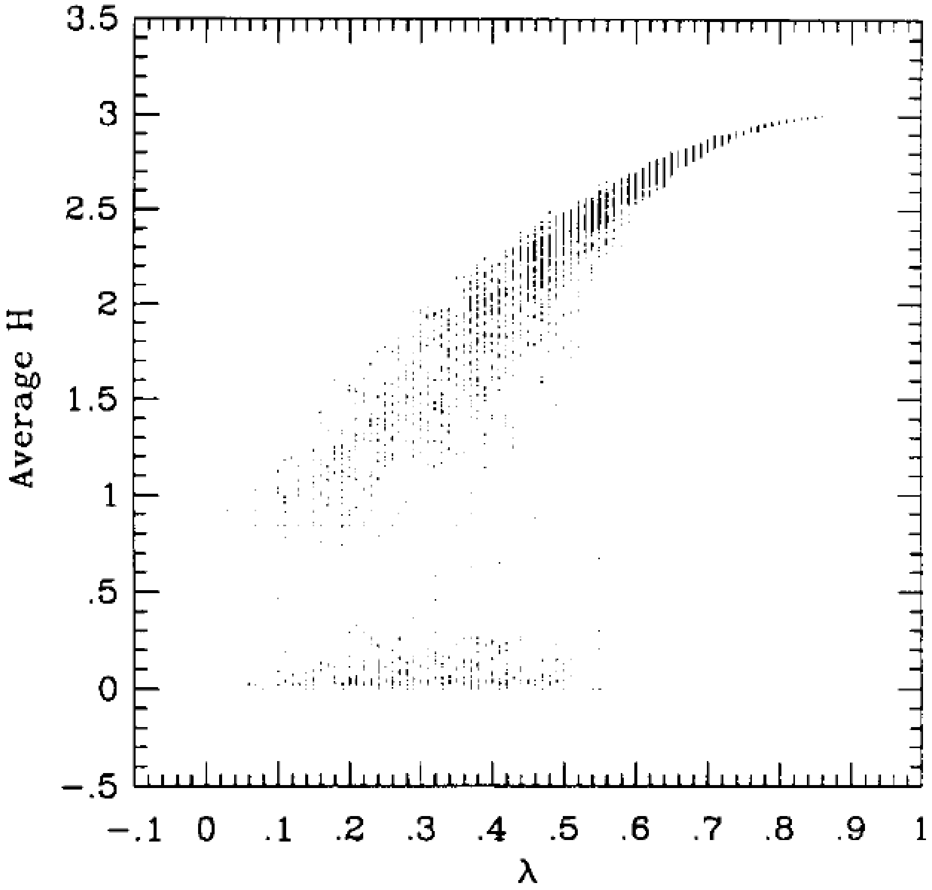


Figure 4: Plotting average Shannon-Entropy per cell against the spectrum of λ for $\sim 10,000$ CA runs. Each point corresponds to a different CA rule. Source: [2]

measure is called the *in-degree* which is the transient length of states in state space before they reach a limit point or enter a limit cycle. Yet another is a by product of Wuensche's reverse algorithm, called the *Zparameter*. For this assignment I decided to focus on the one measure for which Wuensche claims correlation to human experience of the order-chaos spectrum, being the *Input-Entropy variance*.

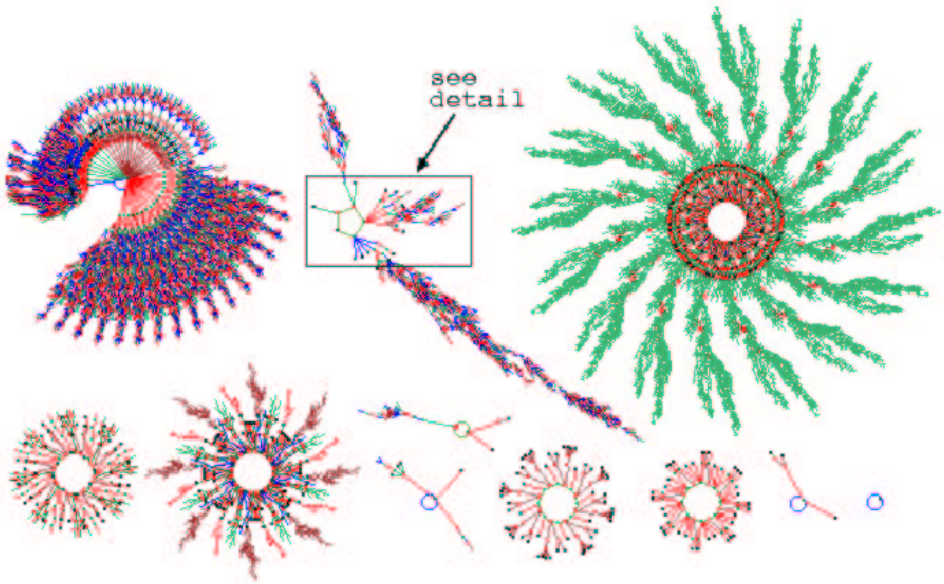


Figure 5: Basin of attraction field of a 16 cell 1D CA with neighborhood size 7, fully describing the CA dynamics (equivalent basins of attraction not shown). Source: [5] (detailed picture not shown here)

In the papers at hand, only one-dimensional, binary CA with periodic boundary conditions were considered. I shall broaden this definition to include multi-dimensional n-nary state CA (with periodic boundary conditions). Let c denote the number of cells, and k the size of the neighborhood, then there exists a lookup table mapping n^k possible neighborhoods to one of the n states. The *lookup frequency* Q_i^t is the number of lookups of a particular neighborhood-constellation i per time step t. The Input-Entropy for time step t is then given by the Shannon-Entropy using the set of possible neighborhoods as alphabet and the average input frequency per cell.

$$S^t = -\sum_{i=1}^{n^k} \left(\frac{Q_i^t}{c} \log \left(\frac{Q_i^t}{c} \right) \right)$$

If the CA evolution, after an initial phase of what Wuensche called "settling down" displays low mean Input-Entropy with low variance, the system has settled for an ordered state. If the mean Input-Entropy is high, and the variance still low, the system has settled for a chaotic state. If the variance of the Input-Entropy is high, however, the system is considered complex. See figure 6 for illustration.

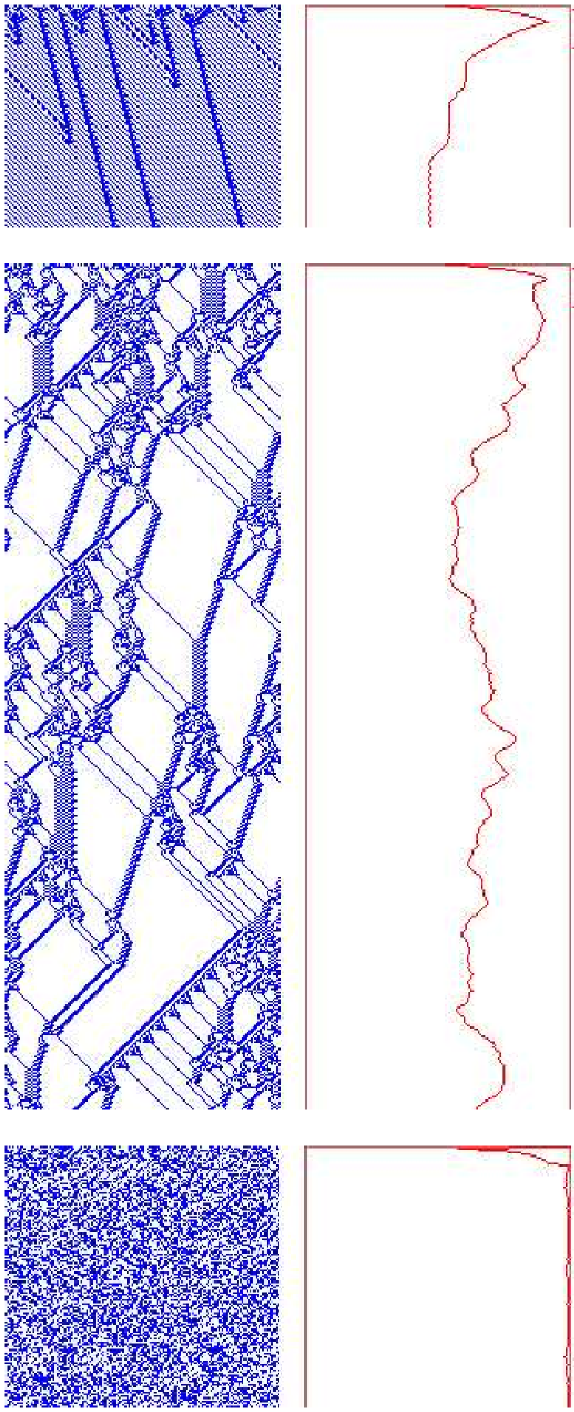
One thing, [4, 5] do not mention explicitly, is that Input-Entropy is by no means a general measure of the complexity of a CA rule. It should be stressed that complex CA are highly dependent on their initial configuration. That means that depending on the initial configuration, they might be falsely classified as either ordered or chaotic. A concrete and prominent example of a complex CA rule is Conway's game of life. In Life most initial configurations lead to highly ordered states consisting of very short limit cycles, if any. If you were to measure Input-Entropy for a random initial configuration, you would be likely to classify Life as being an ordered CA. The more complex a CA, the more samples are needed to classify a given CA. This would not be too bad if the spectrum of complex CA was not of measure0. The consequence is that to confidently determine whether a given CA is complex, one would have to start the CA from all possible initial configurations.

However, using the Input-Entropy variance one can randomly sample the CA rule / initial configuration space for combinations that yield complex behavior and Wuensche did so with some success.

Furthermore, the classification via Input-Entropy only really works for smallish networks. Wuensche acknowledges the problem: "...the entropy variance will be reduced, to zero in the limit of infinite size"[4]. I believe this limitation can be overcome though, by tiling the lattice into smaller areas and applying the same algorithm. If the highest overall Input-Entropy variance is used, very large networks can be screened for emergent gliders.

7 Getting down and dirty with Input-Entropy variance

As suggested, I picked a measure and implemented it (see Matlab files, submitted separately). Because Matlab is fairly fast to program, I managed to implement a generator for 1D CA with periodic boundary conditions, arbitrary numbers of states per cell and arbitrary lattice- and neighborhood sizes without sacrificing too much sleep. The program endlessly samples randomly generated CA and keeps a log of explored rules. It logs only rules that have higher variance in Input-Entropy than those already explored. This way, the log is automatically ranked. A plot of CA evolution and corresponding



a) Ordered dynamics (class 2), about 100-time steps.
Rule af d1 db 47

b) Complex dynamics (class 4), about 500 time-steps.
Rule 6c 1e 53 a8

c) Chaotic dynamics (class 3), about 140 time-steps.

Figure 6: Ordered, complex and chaotic system behavior (left) and corresponding Input-Entropy variance (right). Source: [4]

Input-Entropy is generated for the latest CA (see figure 7 for an example).

Besides proving Wuensche right in that CA evolution becomes increasingly complex as variance in Input-Entropy increases and that this measure seems to accurately capture the human notion of "interesting behavior" being somewhere in between our experiences of order and chaos, my implementing this algorithm has lead to a deeper understanding of the halting problem in the context of CA (also referred to as the *freezing problem* [2]).

For measuring complexity using Input-Entropy variance, any values before the end of an initial "settling-down phase" have to be ignored. This is because ordered and chaotic systems agree on a level of entropy during this phase and display variance before they settle. In order to distinguish them from complex CA, this variance has to be ignored. However, since we cannot run the CA forever to find out whether or not it might eventually settle, any CA that displays considerable variance immediately after the settling-down phase is considered complex by the search algorithm. The problem is that often systems display complex behavior for a considerable amount of time until finally settling anyways. This can be observed in figure 7 around time step 110. The system settles for order, yet due to the short length of the settling-in phase, the system is considered complex.

I propose to use the variance over a certain time frame whose length has to be determined. The time frame for settling-down phase and measuring variance should be as small as possible to save time during sampling, however big enough to sort out quickly ordered/chaotic evolutions and capture slower complex dynamics respectively (as they cause slow changes in variance). Those CA displaying high variance (and the initial configuration used and/or last known state) can then selectively be run either for a longer time to study their evolution, or reverse engineered to analyze their basin of attraction. However, I do not expect the results to turn up anything particularly interesting that has not been already described elsewhere. In fact, I would expect to at best find some sort of quadratic relationship of some version of Input-Entropy to transient length just as uncovered by Langton for the Δ parameter (see 3).

Again, what this method definitely allows for is automatically finding interesting CA/configuration pairs which then can be used to study their dynamics at a higher descriptive level. Any feasible implementation should be run with optimized code, ideally compiled natively, as the computations take a very long time (outright painfully long in Matlab).

The biggest improvement would be to limit the search space by keeping rules known to be ordered or chaotic from being considered for sampling. This might be done by generating CA with $\Delta \sim 0.5$ and sampling their initial configuration space.

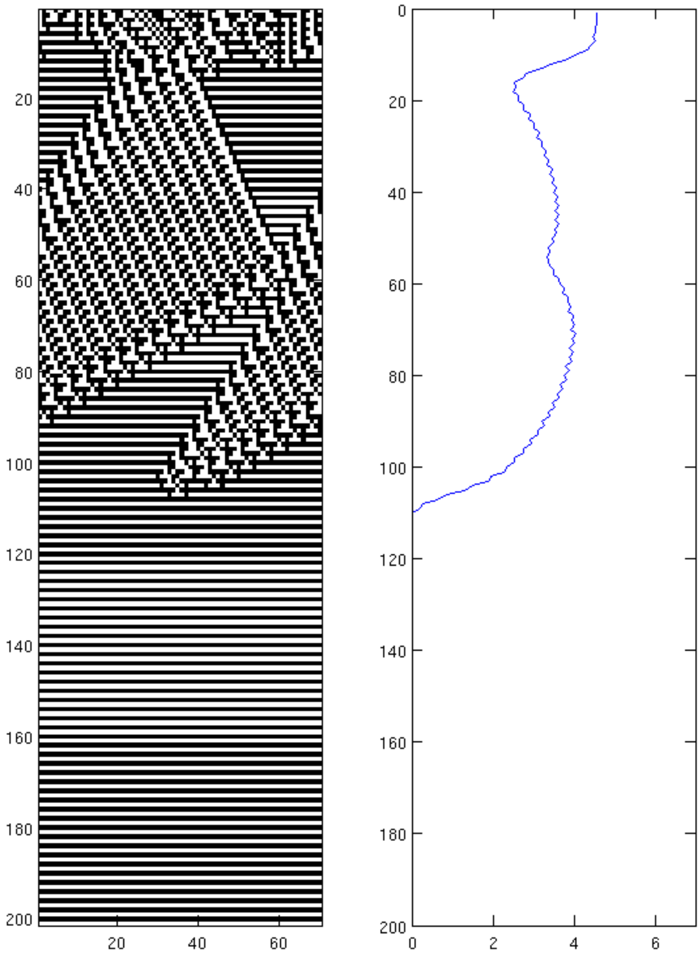


Figure 7: A CA found by my implementation of Input-Entropy variance screening. This was taken for a 1d 70 cell CA with periodic boundary conditions. Each cell can assume one of 2 states, neighborhood size is 5 (2 left, 2 right), the simulation was run over 200 time steps with a settling time of 20 steps. The Input-Entropy variance is 2.96. The plot on the right has been smoothened over a window of 5 steps. The CA rule is 353121653 and the initial configuration used (1 indexed) is: [2 1 2 2 1 2 1 1 2 1 2 2 1 2 2 1 2 2 1 1 1 2 2 1 2 1 2 1 1 1 1 2 2 1 2 1 2 1 2 2 2 2 1 1 2 2 1 2 2 2 2 1 1 2 2 2 2 1 1 1 2 2 2 2 2 1 1 2 1 2].

8 Remarks

Besides taking forever and being most likely flawed in more than one way, my work has allowed me to gain an interesting introduction to basic measures of complexity in the special case of cellular automata. “Incidentally”, I will be looking at higher level organization in Conway’s Game of Life or maybe even CA in general, for my ALife project and I think I will try to find the size limits of the Input-Entropy variance measure, implement my tiling idea as well as the moving variance window to check for longevity of complexity and hopefully be able to sample the space of initial configurations sufficiently to provide me with a bunch of complex evolutions. Markus out.

References

[1] David Eppstein. Wolfram's classification of cellular automata. `http://www.ics.uci.edu/~eppstein/ca/wolfram.html` (accessed 2010/12/03).

[2] Chris G. Langton. Computation at the edge of chaos: phase transitions and emergent computation. *Phys. D*, 42:12–37, June 1990.

[3] S. Wolfram. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):1–35, January 1984.

[4] Andrew Wuensche. Classifying cellular automata automatically. 1998.

[5] Andrew Wuensche. Classifying cellular automata automatically; finding gliders, filtering, and relating space-time patterns, attractor basins, and the z parameter. *Complexity*, 4:47–66, 1998.