

Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata.

Cosma Rohilla Shalizi

Abstract: All self-respecting nonlinear scientists know self-organization when they see it: except when we disagree. For this reason, if no other, it is important to put some mathematical spine into our sloppy intuitive notion of self-organization. Only a few measures of self-organization have been proposed; none can be adopted in good intellectual conscience.

To find a decent formalization of self-organization, we need to pin down what we mean by organization. The best answer is that the organization of a process is its *causal architecture* - its internal, possibly hidden, causal states and their interconnections. *Computational mechanics* is a method for inferring causal architecture represented by a mathematical object called the *ε-machine* - from observed behavior. The *ε-machine* captures all patterns in the process which have any predictive power, so computational mechanics is also a method for *pattern discovery*. In this work, I develop computational mechanics for four increasingly sophisticated types of process - memoryless transducers, time series, transducers with memory, and cellular automata. In each case I prove the optimality and uniqueness of the *ε-machine's* representation of the causal architecture, and give reliable algorithms for pattern discovery.

The *ε-machine* is the organization of the process, or at least of the part of it which is relevant to our measurements. It leads to a natural measure of the *statistical complexity* of processes, namely the amount of information needed to specify the state of the *ε-machine*. Self-organization is a self-generated increase in statistical complexity. This fulfills various hunches which have been advanced in the literature, seems to accord with people's intuitions, and is both mathematically precise and operational.

1. Introduction

1.1 Self-Organization. There is no current scientific measure of "self-organizing" other than "I know it when I see it."

1.2 Formalizing an Intuitive Notion. We need a formal definition than matches the intuitive one in all the easy cases; resolve the hard ones in ways that don't make us boggle; and let us frame simple and fruitful generalizations.

1.3 The Strategy. Pattern discovery problem is soluble.

1.4 A Summary. Of the book structure.....

1.5 Historical Sketch. Origins of the Concept. Ross Ashby (1974) A system cannot change its own organization, organization is invariant, there may be an apparent change in organization but the underlying dynamics stay the same. **Uses of the Concept.** Many disciplines have use "self-organizing" but few with rigor **History of Pattern Discovery and Computational Mechanics.** **Algorithmic** pattern discovery was the goal of much work in computer science and unsupervised learning. 70's saw model-identification and selection techniques which sought to balance accuracy against complexity. 80's The "equations of motion from a data series" method fitted a vector-field of specified functional form to each small region of state-space. The crucial step was to realize that a "pattern basis" could be constructed directly from the data, and that it could give the optimal predictor, as well as the equations of motion.

2 Measuring Pattern, Complexity & Organization.

2.1 Organization thermodynamic. Entropy fails as a measure of organization in many ways. The most basic problem is that it doesn't distinguish between the many different kinds of organization matter can exhibit

2.2 Complexity Measures, or, the History of One-Humped Curves It is altogether too easy to come up with complexity measures. The first and still classic measure of complexity is Kolmogorov's, which is (roughly) the shortest computer program capable of generating a given string (this quantity is in general uncomputable). So we don't want to cognate complexity with randomness, while at the same time we don't want to say that things which are completely uniform and orderly are complex either. Complex and interesting stuff should be some place "in the middle".

2.3 Patterns I'm asking what patterns are and how patterns should be represented. I want pattern discovery, not pattern recognition. **Algebraic Patterns** the first attempt to make the notion of "pattern" mathematically rigorous was that of Whitehead and Russell in Principia Mathematica. They viewed patterns as properties, not of sets, but of relations within or between sets. A more recent attempt at developing an algebraic approach to patterns builds on semi-group theory and its Krohn-Rhodes decomposition theorem. Yet another algebraic approach has been developed by Grenander and co-workers, primarily for pattern recognition. **Turing Mechanics: Patterns and Effective Procedures** the Kolmogorov-Chaitin framework establishes, formally at least, the randomness of an individual object without appeals to probabilistic descriptions or to ensembles of reproducible events. And it does so by referring to a deterministic, algorithmic representation - the UTM. There are many well-known difficulties with applying Kolmogorov complexity to natural processes. First, it is uncomputable in general, owing to the halting problem. Second, it is maximal for random sequences; this is either desirable, as just noted, or a failure to capture structure. Third, it only applies to a single sequence; again this can be either good or bad. Fourth, it makes no allowance for noise or error, demanding exact reproduction. Finally, it may not converge. **Patterns with Error** Dennett notes that there is generally a trade-off between the simplicity of a predictor and its accuracy, and he plausibly describes emergent phenomena as patterns that allow for a large reduction in complexity for only a small reduction in accuracy. **Causation** We want our representations of patterns in dynamical processes to be causal. **Synopsis of Pattern** We want an approach to patterns which is at once

- Algebraic, giving us an explicit breakdown or decomposition of the pattern into its parts;
- Computational, showing how the process stores and uses information;
- Calculable, analytically or by systematic approximation;
- Causal, telling us how instances of the pattern are actually produced; and
- Naturally stochastic, not merely tolerant of noise but explicitly formulated in terms of ensembles.
- Computational mechanics satisfies all these desiderata.

3 Memoryless Transducers

3.1 The Setup

3.2 Effective States Any prediction scheme treats some inputs the same when it calculates its

predictions. **Minimality and Prediction** we want to minimize statistical complexity, subject to the constraint of maximally accurate prediction.

3.3 Causal States Each causal state σ has a unique associated distribution of outputs $P(Y = y|S = \sigma)$, called its morph. In general every effective state has a morph, but two effective states in the same state class may very well have the same morph. Moreover, the causal states have the important property that all of their parts have the same morph. **Homogeneity, Optimalities and Uniqueness** the entropy of a mixture of distributions is at least the mean of the entropies of those distributions.

3.4 Other Approaches to Memoryless Transduction. Graphical Models, The Information-Bottleneck Method and The Statistical Relevance Basis.

3.5 Summary We start with one variable (or set of variables) which causes, in some statistical fashion, another variable. We want to predict the output, given the input, as accurately and as simply as possible. We summarize the input in an effective state, and measure predictive power by the entropy of the output conditional on the effective state, and the complexity of the predictor by the entropy of the effective state, i.e., the amount of information the state retains from the input. The predictive power of effective states is limited by that of the original input; states which attain this limit are prescient. Our goal is to minimize complexity, subject to the constraint of prescience.

We introduce a particular partition of the inputs, the causal states, which treats inputs as equivalent if they lead to the same conditional distribution of outputs. This is prescient, since the distribution of outputs conditional on the causal state is, by construction, the same as that conditional on the input. We then use homogeneity to prove a refinement lemma, telling us that any prescient rival to the causal states must be a refinement of them almost everywhere. The refinement lemma, in turn, leads directly to the result that the causal states are the minimal prescient states, and to the uniqueness of the causal states.

4 Time Series

4.1 Paddling Around in Occam's Pool. Processes. Let's restrict ourselves to discrete-valued, discrete-time stationary stochastic processes. Intuitively, such processes are sequences of random variables, the values of which are drawn from a countable set. We can define a process in terms of the distribution of such sequences. Intuitively, we can imagine starting with distributions for finite-length sequences and

extending them gradually in both directions, until the infinite sequence is reached as a limit. A stationary process is one that is time-translation invariant. Time-invariant transition probabilities are this *conditionally stationary*. **The Pool** Our goal is to predict all or part of the future using some function of some part of the past. All the histories belonging to a given effective state are treated as equivalent for purposes of predicting the future. **Patterns in Ensembles** We say R captures a pattern when it tells us something about how the distinguishable parts of a process affect each other: R exhibits their dependence. Supposing that these parts do not affect each other, then we have independent, identically distributed (IID) random variables, which is as close to the intuitive notion of "patternless" as one is likely to state mathematically. **The Lessons of History** we want to forget as much of the past as possible and so reduce its burden.

4.2 The Causal States The causal states of a process are the members of the range of the function ϵ that maps from histories to sets of histories. Alternately and equivalently, I could define an equivalence relation \sim_ϵ such that two histories are equivalent if and only if they have the same conditional distribution of futures, and then define causal states as the equivalence classes generated by \sim_ϵ . We can make the original definition more intuitive by picturing a sequence of partitions of all histories in which each new partition, induced using futures of length $L + 1$, is a refinement of the previous one induced using L . At the coarsest level, the first partition ($L = 1$) groups together those histories that have the same distribution for the very next observable. These classes are then subdivided using the distribution of the next two observables, then the next three, four, and so on. The limit of this sequence of partitions - the point at which every member of each class has the same distribution of futures, of whatever length, as every other member of that class is the partition of histories induced by \sim_ϵ . **Morphs** Each causal state has a unique morph, i.e., no two causal states have the same conditional distribution of futures. **Causal State-to-State Transitions** The causal state at any given time and the next value of the observed process together determine a new causal state. **ϵ -Machines** The combination of the function ϵ from histories to causal states with the labeled transition probabilities is called the ϵ -machine of the process. Starting from a fixed state, a given symbol always leads to at most one single state. But there can be several transitions from one state to another, each labeled with a different symbol. ϵ -Machines Are Markovian - given the causal state at time $t-1$, the causal state at time t is independent of the causal state at earlier times.

4.3 Optimalities and Uniqueness. Causal states are maximally accurate predictors of minimal statistical complexity; they are unique in sharing both properties; and their state-to-state transitions are minimally stochastic. In other words, they satisfy both of the constraints borrowed from Occam, and they are the only representations that do so. The overarching moral here is that causal states and ϵ -machines are the goals in any learning or modeling scheme. Causal states are as good at predicting the future - are as prescient - as complete histories. the causal states are the minimal prescient states, they are also the minimal deterministic states which get the distribution of the next symbol right. No rival pattern, which is as good at predicting the observations as the causal states, is any simpler than the causal states. Causal states are minimal sufficient statistics for predicting futures of all lengths. Due to the minimality of causal states, the statistical complexity measures the average amount of historical memory stored in the process.

4.4 Bounds The excess entropy of a process is the mutual information between its semi-infinite past and its semi-infinite future. The excess entropy is a frequently-used measure of the complexity of stochastic processes and appears under a variety of names; e.g., "predictive information", "stored information", "effective measure complexity", and so on. It measures the amount of apparent information stored in the observed behavior about the past; rather it is only a lower bound.

4.5 The Physical Meaning of Causal States Put slightly differently, what we have done is construct a partition of the phase space \square which is Markovian, starting from an arbitrary observational partition. Each causal state thus corresponds not only to a history of observations, but also to a region in phase space. Even better, since the causal states form a Markov chain, the distribution of sequences of causal states is a Gibbs distribution.⁸ Yet we haven't had to assume that our system is in equilibrium, or in a steady state, or has any particular kind of ensemble (such as a maximum entropy ensemble).

5 A Reconstruction Algorithm

5.1 Reconstructing States by Merging. Previous procedures for reconstructing the states operate by using what one might call compression or merging. The default is that each distinct history encountered in the data is a distinct causal state. Histories are then merged into states on the basis of equality of conditional probabilities of futures, or at least of closeness of those probabilities. The standard Crutchfield-Young merging algorithm is a tree method.

5.1.1 What's Wrong with Merging Methods? The basic problem with all merging methods is that their default is to treat each history as belonging to its own causal state, creating larger causal states only when they must. The implicit null model of the process is thus the most complicated one that can be devised, given the length of histories available to the algorithm. This seems perverse, especially given computational mechanics's strong commitment to Occam's Razor and the like. Worse, it makes it very hard, if not impossible, to apply standard tools of statistical inference to the estimation procedure.

5.2 Reconstructing States by Splitting.

Description of the Method If we have a sequence of characters from a finite alphabet A . Then a *history* is a sub-sequence of the character sequence. A *state* σ is a set of histories, or suffixes to histories. Each state has a *morph*, which is a probability distribution for the subsequent character in the sequence (ie for each a in A). A function ϵ , maps a finite history s to the state σ containing the longest suffix of s . **Initialization:** We begin by mapping all histories to the null sequence $\sigma_0 = \{\emptyset\}$. So the initial model is that the process is a sequence of independent, identically distributed random variables. **Homogeneity:** First, we generate states whose members have no significant differences in their morphs. Next, we calculate the morph of each state - the weighted average of the morphs of its member histories. This is done by examining every history (sub-sequence), of length $1,2,3\dots L_{\max}$. If a history has a morph which is significantly different to the morph of all other states then it is assigned to a new state and all ancestor parent sequences are removed from the parent state. .

At this point I despaired in trying to understand the algorithm provided and located a copy of the more recent paper "Blind Construction of Optimal Nonlinear Recursive Predictors for Discrete Sequences" which includes an improved algorithm and pseudo-code!

At the end of this procedure, no history is in a state whose morph is significantly different from its own. Moreover, every state's morph is significantly different from every other state's morph. **Reliability of Reconstruction** It is reliable! (particularly as $N \rightarrow \infty$) **Advantages of the Method.** Everything that was noted in merging methods. **Problems with the Method.** The states produced may not be minimal. L_{\max} needs to be set. Only a single state class is returned rather than all adequate state classes. It will not detect a strictly sofic pattern (eg the total of all 0 & 1's is even). **Some Notes on an**

Implementation. Preliminary investigations indicate good performance. **Statistical Analysis and Rates of Convergence** The value chosen for significance will determine the power of the algorithm to discriminate. Convergence is achieved at ∞ but it would be good to have a measure of convergence – but this will likely be tricky.

6 Connections. Time Series Modeling, Decision-Theoretic Problems, Stochastic Processes, Formal Language Theory and Grammatical Inference, Computational and Statistical Learning Theory, Description-Length Principles and Universal Coding Theory, Measure Complexity, Hierarchical Scaling Complexity, Continuous Dynamical Computing.

7 Transducers

"We watch an ant make his laborious way across and wind and wave-molded beach. He moves ahead, angles to the right to ease his climb up a steep dunelet, detours around a pebble, stops for a moment to exchange information with a compatriot. So as not to anthropomorphize about his purposes, I sketch the path on a piece of paper. It is a sequence of irregular, angular segments | not quite a random walk, for it has an underlying sense of direction, of aiming towards a goal.

Viewed as a geometric figure, the ant's path is irregular, complex, hard to describe. But its complexity is really a complexity in the surface of the beach, not a complexity in the ant. On that same beach another small creature with a home at the same place as the ant might well follow a very similar path.

The ant, viewed as a behaving system, is quite simple. The apparent complexity of its behavior over time is largely a reaction of the complexity of the environment in which it finds itself. Herbert Simon (1996, pp. 51-52)

7.1 Introduction The picture is that one series, called the input, is fed into a transducer, box (or other physical process), resulting in an output series. This differs from the case of memoryless transduction because the transducer has internal states, and so a kind of memory for both the past of the input process and its own internal dynamics (which may well be stochastic). The goal is to be able to identify the internal states of the transducer and their structure of connection - to find the e-transducer.

Put another way: we have two time series, and the future values of the output are a stochastic functional of the history of the input. We want to put this relationship in "transducer form", replacing the stochastic functional of the series with a stochastic function of an internal or hidden state of a transducer, which in turn is a functional of the history.

7.2 Simplifying the Transducer Problem It is commonly assumed that you completely specify a transducer by giving the conditional probabilities of all finite-length input-output pairs.I assumed that the future of the input is independent of the past of the output, given the past of the input. This is true just when there is no feedback from output to input. I'll deal with the feedback case below.... Finding transducer states reduces to finding states which "get right" the next output, given the complete input and output histories.

7.3 Effective Transducer States I want to attend only to how well the effective states capture the internal structure of the transducer, and the relation it imposes between inputs and outputs, not how well they do that and predict the future of the input series.

7.3.1 Determinism This definition of determinism implies that transitions from one state to another happen after seeing both a new input and a new output. In the theory of finite state transducers (Booth 1967), this is a "Mealy machine", as opposed to a "Moore machine," which has a single output for each state, and makes transitions only on inputs.

7.4 Causal States The second clause of the definition of e ensures that the causal states are deterministic, which (as will be seen) is important for much of what follows.

7.5 Transduction with Feedback I assumed above that the output has no influence on the input. This is often true, and it's the classic transducer problem, but there is no logical necessity for this to be so. But we can go through an entirely parallel construction for predicting the input on the basis of the joint history. **An Aside: The Elimination of Dialectics in Favor of Mechanics** ... even when it's most reasonable to talk about dialectical relationships, we can always replace the dialectical representation with a purely (statistical) mechanical one, without any loss of information.

7.6 Kindred Approaches e -transducers completely cover other approaches such as computer science "finite state transducers" and stochastic model of discrete transduction.

7.7 Reconstruction The state-splitting algorithm of Chapter 5 can easily be adapted to deal with transducers without feedback, simply by considering the joint history, and splitting joint histories when they produce significantly different distributions for the next output. The reconstruction of the feedback state would go in the same way. The reliability

analysis proceeds on exactly the same lines as for time series.

8 A Very Brief Introduction to Cellular Automata

The best non-technical introduction to cellular automata is the book by Poundstone (1984). The standard modern reference is Gutowitz (1991), but it will probably be superseded by Griffeath and Moore (forthcoming)

9 Domains and Particles: Spatial Computational Mechanics

People notice patterns when they look at CAs, though whether this says more about what CA are apt to do, or what people like to look at, is a nice question. Two very common kinds of patterns noted in CAs are domains | patches of space-time where everything looks the same, where some "texture" is repeated over and over - and particles, localized blobules which propagate across the lattice. A review of the literature indicates that particles are generally felt to be about the most interesting things in CAs. Part of the reason for this is that propagating blobules are observed in real physical systems..... Many people have long suspected that particles and domains are emergent structures. A general theoretical analysis shows that this is true.

The burden of this chapter is to expound the "pure-space" computational mechanics of cellular automata of Hanson and Crutchfield. This is a method for analyzing particles and domains in one-dimensional CAs in terms of regular languages and the states of machines associated with them. The next chapter constructs a fully spatio-temporal, multi-dimensional computational mechanics

9.1 Domains A regular domain of a CA is a process language, representing a set of spatial lattice configurations, with the following properties: 1. Temporal invariance (or periodicity) and 2. Spatial homogeneity: The process graph of each temporal iterate is strongly connected. That is, there is a path between every pair of states. According to the first property - temporal invariance or periodicity - a particular domain consists of temporal phases for some temporal periodicity of the domain. Each of the temporal phases of a domain is represented by a process graph which, according to the second property (spatial homogeneity), is strongly connected. Each of these process graphs consists of a finite number of states. The process graphs of all temporal phases of all domains can be connected together and transformed into a finite-state transducer, called the domain transducer, that reads in a spatial configuration and outputs various kinds of information about the sites. Variations on this transducer can do useful recognition tasks. For example, a domain transducer can be used to filter

CA lattice configuration, mapping all domain regularities to D and mapping all domain violations to output symbols w that indicate domain walls of various kinds.

9.2 Particles When domain violations form a spatially localized (finite width), temporally periodic boundary between two adjacent domains, they are called particles. Since a particle is a bounded structure, it does not have a spatial periodicity. "Periodicity of a particle" therefore always means temporal periodicity. **Structural Complexity of a Particle** The preceding definitions and discussion suggest that one can think of particles as having an internal clock or, in the more general case that includes aperiodic particles, an internal state, much as the solitary-wave solutions of continuum envelope equations have internal states (Infeld and Rowlands 1990). One can ask about how much information a particle stores in its states. This is the amount of information that a particle transports across space and time and brings to interactions. These considerations lead one to a natural measure of the amount of structural complexity associated with individual particles. **Domain Transducer View of Particle Phases** A particle is bounded on either side by two patches of domain.

9.3 Interactions In many CAs, when two or more particles collide they create another set of particles or mutually annihilate. There are also unstable walls that can spontaneously decay into particles. Often, the actual product of a particle interaction depends on the phases in which the interacting particles are at the time of collision.

9.4 Bounding the Number of Interaction Products Restricting ourselves to particle interactions with just two colliding particles, we'll now derive an upper bound on the number of possible interaction products from a collision between them.

9.5 Examples. ECA 54 and Intrinsic Periodicity see Hanson and Crutchfield (1997) **An Evolved CA** The second example for which we test the upper bound is a CA that was evolved by a genetic algorithm to perform a class of spatial computations: see Hordijk, Mitchell and Crutchfield (1998). **Another Evolved CA** see Crutchfield, Hordijk and Mitchell (2000b). **ECA 110** see Crutchfield and Shalizi (2001).

9.6 Conclusion. Summary. We've established a number of properties of domains and particles - structures defined by CA computational mechanics. The examples showed that the upper bound is tight and that, in complex CAs, particle interactions are substantially less complicated than they look at first blush. Moreover, in developing the bound for complex domains, the analysis elucidated the

somewhat subtle notion of a particle's intrinsic periodicity a property not apparent from the CA's raw space-time behavior: it requires rather an explicit representation of the bordering domains' structure. Understanding the detailed structure of particles and their interactions moves us closer to an engineering discipline that would tell one how to design CA to perform a wide range of spatial computations using various particle types, interactions, and geometries. In a complementary way, it also brings us closer to scientific methods for analyzing the intrinsic computation of spatially extended systems. **Open Problems** It would be preferable to directly calculate the number of products coming out of the interaction region, rather than (as here) the number of distinct particle-domain-particle configurations coming into the interaction region. Two very desirable extensions of these results suggest themselves. The first is to go from strictly periodic domains to cyclic (periodic and "chaotic") domains and then to general domains. The second extension would be to incorporate aperiodic particles. A third extension, perhaps more tractable than the last, is to interactions of more than two particles. Does there exist an analogous lower bound on the number of interactions? If so, when do the upper and lower bounds coincide?

10 Spatio-temporal Computational Mechanics 91

10.1 The Difficulties of Higher Dimensions 91 we could apply computational mechanics to CA, at the global level, in a very straightforward manner and this would capture the causal states including those related to spatial structure - however it would be very hairy. We'd really like something where the spatial structure was transparent, just as the e-machine makes the causal structure transparent. No-one has a solution so we are in the uncomfortable position of having to strike out more or less on our own.

10.2 Global Causal States for Spatial Processes We'll consider, not spatial processes in general, but those whose space, time and state are all discrete, and space is a regular lattice. **Why Global States Are not Enough.** First, there is no explicit representation of the spatial structure. Second, the number of global causal states is apt to be very large. Third, getting adequate statistics to empirically estimate the global causal states is simply not practical. For all these reasons, the global causal states approach to spatial processes, while valid, is useless. What we would like, instead, is some way of factoring or distributing the information contained in the global causal state across the lattice - of finding local causal states. It is to this question that we now turn.

10.3 Local Causal States. Light Cones and Their Equivalence Classes. Let x be a single cell at a single time, or a point-instant. We define the past light-cone of x , denoted $L(x)$, as all other point-instants such which could disturb x given the maximum speed that a disturbance can propagate C - the "speed of light," as it were. **The Local Causal States. Composition of the Global State from Local States.** Rather than considering the past and future light-cones of a single point-instant, we can consider those of a patch of points at the same time. It will be convenient to only consider connected patches. The past and future cones of the patch are simply the unions of the cones of the patch's constituent cells. Transparently, the patch causal states are prescient (for the patch future light-cone), minimal among the prescient patch states, and render the patch's future light cone conditionally independent of its past light cone. We have thus shown that the global causal state can be decomposed into local causal states, as we have defined them, without losing its global properties or indeed any information.

10.4 Connections Among Local States; the e-Machine 100

- 10.4.1 Temporal Transitions 100
- 10.4.2 Spatial Transitions 102
- 10.4.3 Arbitrary Transitions 103
- 10.4.4 The Labeled Transition Probabilities 104
- 10.4.5 e-Machine Reconstruction 105
 - 10.4.5.1 Reliability of Reconstruction 106
- 10.5 Emergent Structures 106
- 10.6 Examples 107
 - 10.6.1 ECA Rule 54: Domain and Particles 107
 - 10.6.2 ECA Rule 110: Domains and Particles 107

10.7 Summary The main point of this chapter has been to show how to define local causal states for well-behaved spatial processes. By using light cones for our histories and futures, we can assign a causal state to each point-instant, and these are the unique minimal optimal predictors, as we'd hope; indeed, almost all of the familiar, comforting properties of causal states in purely temporal processes carry over. We can also compose these local causal states into the causal states for extended regions, even the entire lattice, thereby recovering the global causal state. We can define the most common sorts of emergent structure (domain, particle, etc.) in terms of the e-machine connecting the local causal states, and so put all the results of Chapter 9 on a much firmer footing.

I have assumed throughout that space is a regular lattice, that every cell's connections look like every other cells. But a lot of the math developed here doesn't depend on that. Space could be an arbitrary

graph, for instance, and we could still define past and future light cones, and so local causal states

11 Conclusion

11.1 What Has Been Accomplished The main line of this book has been the exposition of computational mechanics for increasingly sophisticated processes. We started, in Chapter 3 with memoryless transducers, where we constructed causal states as equivalence classes of inputs | two inputs are causally equivalent when they have the same conditional distribution of outputs. The causal states, we saw, were optimal predictors, and the unique minimal optimal predictors. Since they are both unique and minimal, we could identify the complexity of the process with the complexity of the causal states, defined as the amount of information needed to specify the current causal state. The rest of the book showed how the same basic idea of causal state works with different sorts of process: time series, transducers and CAs. Chapter 7 introduced the computational mechanics of interacting time series. Chapter 9, following a long tradition of spatial computational mechanics, assigns a causal state to each point in one-dimensional space, effectively treating the spatial coordinate as Chapter 4 treated time. Finally, Chapter 10 went beyond the older temporal and spatial computational mechanics, to a fully spatio-temporal version of the theory, with the advantage of working in any number of spatial dimensions. Along the way, we saw how to estimate the causal states and the e-machine from data, and how spatial computational mechanics lets us begin to get a handle on the computational powers of cellular automata. Now we'll see how to define emergence and self-organization

11.2 Emergence "Emergence" is an extremely slippery concept, used in an immense number of ways, generally with no attempt at precision whatsoever. However, one useful view of emergent properties are ones which arise from the interactions of the lower-level entities, but which the latter themselves do not display. Two points: First, the variables describing emergent properties must be fully determined by lower-level variables | must supervene on them, as the philosophers say (Kim 1998). Second, higher-level properties are worthy of being called emergent only if they are "easier to follow," or "simplify the description," or otherwise make our life, as creatures attempting to understand the world around us, at least a little easier.

Emergent Processes The efficiency of prediction of a process is the ratio between its excess entropy and its statistical complexity. One process derives from another if for some measurable function it is called the derived or filtered process of the original,

underlying or raw process. A derived process is emergent if it has a greater predictive efficiency than the process it derives from. We then say the derived process emerges from the underlying process. A process is intrinsically emergent if there exists another process which emerges from it. It may help to contrast this notion of emergence with what people attempt to accomplish with statistical regression. There the goal is to "explain" all of the variance in the output by accounting for the effects of all possible input variables. What we are attempting to do in looking for an emergent process, on the other hand, is to filter out everything we can - get rid of all the small-but-significant inputs - so as to simplify the relationship. We are not trying to explain everything we can measure; we are trying to find what's intrinsically important in our measurements. Emergence is anti-regression. **Emergent Structures Are Sub-Machines of the e-Machine** Divide the e-machine into sub-machines, i.e., strongly connected components, and label them all. Find all the transitions between sub-machines, and give those labels too. Then apply the following filter: at each time-step, check whether the current causal state and the previous state were in the same sub-machine. If they were, output the label of that sub-machine. If they weren't, then the process has moved from one sub-machine to another; output the label of that transition. **Example: Thermodynamics and Statistical Mechanics** Hence thermodynamics emerges from the statistical mechanics, and does so very strikingly, since almost all of the information needed at the statistical-mechanical level is simply irrelevant thermodynamically.

11.3 Self-Organization Defined an increase in statistical complexity is a necessary condition for self-organization. While the fundamental causal architecture remains unchanged, the degree of organization - measured by the amount of information needed to place the process in a state within the architecture - is variable. If we compare this criterion for self-organization with the definition of emergence in chapter 11.2, we see that self-organization increases complexity, while emergence, generally speaking, reduces it, or requires us to use it more effectively for prediction. However, self-organization is something a process does over time, like being stationary, or having a growing variance. Emergence is, primarily, a relation between two processes, one of which is derived from the other. There is nothing contradictory in saying that a process is becoming more structurally complex, while at the same time saying that there is another description of the process which is always simpler than the raw data. "it is conceivable that there are processes which organize themselves into conditions so complex that no human being can

grasp them. They would be so organized, in other words, that they would look very like noise." Pg 119.

11.4 Things That Are Not Yet Theorems. Non-Stationarity. The Permutation City Problem. Continuity.

11.5 What Is to Be Done, or, Neat Things to Hack. **The Real World** almost any application domain where nonparametric or data-mining methods are used, computational mechanics is at least a contender. **Turbulence** We might settle the question of whether the transition to turbulence is self-organizing, with which we began. **Physical Pattern Formation. Biosequences.** DNA analysis. **Neural Coding.** Spike train decoding **Signal Transduction, Gene Regulation, and Metabolic Networks. Agents** Model ants for a start. **The Dynamics of Learning** Computational mechanics sets limits on how well processes can be predicted, and shows how, at least in principle, those limits can be attained. e-machines are what any prediction method would build, if only they could. But any learning problem which is formal and definite enough that we can say whether or not it's been successfully solved is also a prediction problem, or at least equivalent to one. So, in a sense, e-machines are also what every learning method wants to build. Computational mechanics thus has some important things to say about how well learning can succeed in different environments, and what optimal learning looks like. **Phenomenological Engines** 125

A Mathematical Review

A.1 Equivalence Relations and Partitions

A.2 Information Theory Information theory appeared in essentially its modern form with Shannon (1948), though there had been predecessors in both communications (Hartley 1928) and statistics, notably Fisher (see (Kullback 1968) for an exposition of these notions), and similar ideas were developed by Wiener and von Neumann, more or less independently of Shannon (Wiener 1961). Shannon and Weaver (1963) contains the classic papers; Pierce (1961) is a decent popular treatment. Appendix A.2.4 lists a number of useful information-theoretic formula, which get called upon in our proofs. Throughout, our notation and style of proof follow those in (Cover and Thomas 1991), the definitive modern reference.

A.3 Statistical Independence and Conditional Independence

A.4 Automata Theory

A.5 Sufficient Statistics

B Mathematical Annex to Chapter 4

Further Reading: pg 22, Ross Ashby (1974). pg 69, Sinha and Ditto (1998). See page 69 section 6.9 regarding "Continuous Dynamical Computing". Hanson and Crutchfield (1997). Griffeath and Moore (forthcoming)

Cover, Thomas M. and Joy A. Thomas (1991). Elements of Information Theory. New York: Wiley. The definitive information theory reference.

Interesting Comments.

- Pg 15 "Computational, showing how the process stores and uses information" I don't know how CSSR shows how the process uses information.
- Pg 21 "Theorem 4 (Control Theorem for Memoryless Transduction)" there might be room here to attach a new controller to a known effective state.
- Pg 25 "This property, of time-invariant transition probabilities, should I guess be named some form of 'homogeneity,'" by analogy with the corresponding property for Markov processes, but that name is preempted. So let's call this conditional stationarity instead.
- Pg 28 "Using this we can make the original definition, Eq. 4.10, more intuitive by picturing a sequence of partitions of the space S of all histories in which each new partition, induced using futures of length $L + 1$, is a refinement of the previous one induced using L . At the coarsest level, the first partition ($L = 1$) groups together those histories that have the same distribution for the very next observable. These classes are then subdivided using the distribution of the next two observables, then the next three, four, and so on. The limit of this sequence of partitions is the point at which every member of each class has the same distribution of futures, of whatever length, as every other member of that class. This is the partition of S induced by
- pg 35 "Here it becomes important that we are trying to predict the whole of $!S$ and not just some piece, $!S L$. Suppose two histories s and s' have the same conditional distribution for futures of lengths up to L , but differing ones after that. They would then belong to different causal states. An η -state that merged those two causal states, however, would have just as much ability to predict $!S L$ as the causal states. More, these R -states would be simpler, in the sense that the uncertainty in the current state would be lower. Causal states are optimal, but for the hardest job - that of predicting futures of all lengths."
- pg 37. "The excess entropy E of a process is the mutual information between its semi-infinite past

and its semi-infinite future: $E = I[!S; S] : E$ measures the amount of apparent information stored in the observed behavior about the past. But E is not, in general, the amount of memory that the process stores internally about its past; that's C_μ .

- Pg 37 "At first glance, it is tempting to see E as the amount of information stored in a process. As Theorem 10 shows, this temptation should be resisted. E is only a lower bound on the true amount of information the process stores about its history, namely C . You can, however, say that E measures the apparent information in the process, since it is defined directly in terms of observed sequences and not in terms of hidden, intrinsic states, as C_μ is.
- Pg 38 Lemma 13 and Theorem 11 have additional discussion regarding Control.
- Pg 40 "If you measure several macro-variables $S; R : : \dots$ simultaneously (which is always possible, classically), the induced partition of \square is simply the product of the partitions of the individual variables, $A \times B \times \dots$. We may regard this joint variable as simply yet another macroscopic variable, which could be measured directly with the appropriate instrument. So, without loss of generality, let's just think about a single macro-variable"
- pg 48 "Problems with the model We have no assurance that the set of states produced by this algorithm will be minimal.....It is possible, however, to independently test for the Markov order of the data stream (Billingsley 1961; van der Heyden, Diks, Hoekstra and DeGoede 1998), and so place bounds on L_{max} , if we want to. The algorithm returns a single state class. any pattern which is strictly sofic | where there are sub-words of allowed words which are forbidden | the algorithm will fail to pick up the pattern.
- **Connections to Other Approaches.**
- Pg 54 "A productive line of future work would be to investigate the relationship between hierarchical scaling complexity and computational mechanics, and to see whether they can be synthesized."
- **Transducers with Memory.**
- pg 56 "Both input and output values are in general multidimensional variables, but we don't care about that."
- pg 57 "I include X^{L-1} in the conditioning variables because I want to attend only to how well the effective states capture the internal structure of the transducer, and the relation it imposes between inputs and outputs, not how well they do that and predict the future of the input series."
- pg 57. "Definition 24 (Determinism for State Classes) A class of effective states R is deterministic if the current state and the next

input and next output fix the next state. This definition of determinism implies that transitions from one state to another happen after seeing both a new input and a new output. In the theory of finite state transducers (Booth 1967), this is a “Mealy machine”, as opposed to a “Moore machine”, which has a single output for each state, and makes transitions only on inputs. Translation between the two representations is always possible for non-stochastic transducers, but is sometimes very awkward. Formulating a “Moore” version of the computational mechanics of transducers is an interesting exercise, but outside the scope of this book.

- Pg 77 “If the number of outcomes is less than n , the interaction effectively performs an irreversible logical operation on the information contained in the input particle phases.”
- pg 100 “The new data obtained from a transition in a spatial process consists of the values of point-instants which are in the past light cone of the new point-instant, but were inaccessible from the old one.”
- pg 106 “Definition 43 (Domain) A domain phase is a sub-machine of the α -machine which is strongly connected for transitions in all spatial directions. A domain is a strongly-connected set of domain phases.” does this mean then that any strongly connected subgraph can be used as a filter?
- Pg 107 “Observe that the probability of staying within a domain phase, once entered, is much higher than that of leaving it, so that grouping the domain states together (by filtering on the domain) will improve the efficiency of prediction. That is, the domain-filtered process is emergent.” So at the very least if a system is operating within a domain we can pretty much ignore its inner workings and place sensors on the exit points to signal something interesting happening.
- Pg 113 “If the ideas in this chapter are the right way of thinking about patterns and complexity in spatial processes, then it really doesn't make much sense to try to work out the complexity of (say) static images, or of individual configurations. Complexity, on this view, must be a function of the process which generates configurations”
- pg 113 “two area for future work. One has to do with irregular lattices. I have assumed throughout that space is a regular lattice, that every cell's connections look like every other cells. But a lot of the math developed here doesn't depend on that. In particular, many technical, biological and social networks seem to be “small world” networks, and it would be nice to understand how they work, and particularly nice to understand their emergent structures (if any) (Watts 1999; Shalizi 2000). We might also look at these networks as so many interconnected transducers, along the lines of Chapter 7 | which may be formally equivalent! But the transducer view may be more valuable when we do not know what the network is to start with | and, after all, a network, in these sense, is a pattern of causal interaction, so it ought to be something we infer.”
- pg 116 “There is a sense in which the dynamics of a process are completely summarized by its α -machine | so why can't we use it to build a filter? The following procedure, in fact, suggests itself. Divide the α -machine into sub-machines, i.e., strongly connected components, and label them all. Find all the transitions between sub-machines, and give those labels too. Then apply the following filter: at each time-step, check whether the current causal state and the previous state were in the same sub-machine. If they were, output the label of that sub-machine. If they weren't, then the process has moved from one sub-machine to another; output the label of that transition.
- Pg 118 “When something self-organizes, it becomes more statistically complex, i.e., optimal prediction requires more information. A cognitively-limited observer (such as a human scientist) is therefore motivated to look for a new way of describing the process which has a higher predictive efficiency That is, the desire to describe things simply makes us look for emergent behavior in self-organizing systems. Emergence may be a precondition of detectable self-organization.”
- pg 119 “Non-Stationarity” It seems to me that if we are trying to learn about the world then it is a very big e-machine to learn. It is not so much that we need to cope with the world being non-stationary but rather that we are now having to learn about a different part of the world that is represented by a part of the e-machine that might have some similarities and some differences. If the differences are overwhelming then we need to start again but if the differences are relatively minor then we should be able to perhaps use some of the structure that has been learned elsewhere.
- Pg 120 “The Permutation City Problem” It seems to me that the explanation may be valid but has the wrong end of the stick. The information in a time series is in the order of the data points, not in the data points per se. Re-ordering a time series is by (this) definition destroying information.
-