Fall 1992

Order, chaos and complexity in discrete dynamical systems

John E. Bates
University of New Hampshire, Durham

Order Number 9236297

Order, chaos and complexity in discrete dynamical systems

Bates, John E., Ph.D.

University of New Hampshire, 1992

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

ORDER, CHAOS AND COMPLEXITY IN DISCRETE DYNAMICAL SYSTEMS

BY

JOHN E. BATES
B. S. University of New Hampshire, 1977
M. S. University of New Hampshire, 1979

DISSERTATION

Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy

in

Physics

September, 1992

This dissertation has been examined and approved.

Harvey K. Shepard
Dissertation director, Harvey K. Shepard
Professor of Physics

L. Christian Balling
Professor of Physics

Michael J. Carter
Assistant Professor of Electrical Engineering

John F. Dawson
Professor of Physics

Dawn C. Meredith
Associate Professor of Physics

July 23, 1992
Date

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

ORDER, CHAOS AND COMPLEXITY IN DISCRETE DYNAMICAL SYSTEMS

by

John E. Bates
University of New Hampshire, September, 1992

A method for analyzing discrete dynamical systems is presented that provides a unified quantitative description of order, chaos and complexity in terms of information flow across system boundaries. Complexity is identified with variability in the relative dominance of order and chaos as systems evolve in time; therefore, purely ordered or purely chaotic behavior is considered simple. This notion of complexity is quantitatively expressed as fluctuation in net information gain.

The method is applied to one-dimensional cellular automata, which are spatially and temporally discrete systems. Evidence is presented for a correlation between information fluctuation and the existence of internally complex propagating structures known as gliders. Gliders have been used in the construction of computing machines within cellular automata. This indicates that information variables may provide a connection between dynamical and computational notions of complexity.

The method is also applied to one-dimensional maps, which are temporally discrete but spatially continuous, by partitioning the spatial dimension. For the logistic map, information fluctuation is maximum at the threshold between ordered and chaotic behavior, in agreement with the results of other researchers.

1. Introduction

The notion of opposites combining to produce harmony, creativity, or dynamic change is not new. The theme is familiar in many non-scientific disciplines. "Great music is a pleasing mixture of the expected and unexpected." "There is a fine line between genius and insanity." What are the opposing forces? Yin and yang? However general the principle may be, let us restrict our attention to the opposite pair, order and disorder, and ask what role they have played in science.

For centuries the Newtonian world view presumed to explain everything in terms of immutable, orderly laws. If anything unexpected happened, it was only because of inadequate attention to detail. Laplace believed that if the positions and velocities of all masses were known at any instant, the future fate of the Universe could be known with certainty. Apparent disorder merely reflected ignorance of initial conditions. Poincaré's early insights about deterministic chaos and then the quantum revolution shattered this world view. The pieces are still being picked up and reassembled. Only recently has a full appreciation arisen of the importance of the interplay between order and disorder in physical and biological systems. Disorder is no longer seen as the destroyer of order but as an essential partner in the creation of complexity.

The delicate balance between order and disorder is evident in biological evolution. If organisms were totally ordered they would be rigid, inflexible, and unable to adapt and evolve. The element of disorder provided by mutation makes change possible. At another level, unexpected events in the environment give survival advantage to organisms that are resourceful,

creative and hence complex. Naturally, order is also indispensable. Without it, there would be no stability. Organisms could not maintain their form long enough to reproduce, and survival advantages could not be preserved.

That biological organisms are complex compared to, say, rocks is intuitively obvious. Furthermore, it is generally accepted that the biosphere in which we live has increased in complexity over the eons, in seeming defiance of the spirit of the second law of thermodynamics. Complexity researchers have sought to define complexity precisely and to explain how complexity can be created. A theory of complexity should be consistent with our intuition and qualitative understanding; thus it should incorporate the interplay between order and disorder mentioned above. It should also be abstract and general, although it is unclear whether there can ever be a single theory that encompasses the great variety of systems and objects found in nature or created by human beings.

In the past decade there have been numerous proposals for a quantitative measure of complexity, some of which will be discussed in chapter 2. None of them is preeminent; they all have limitations. I have developed a method for analyzing discrete dynamical systems based on information flow in state space. It contains elements of other methods, but has several unique aspects. It provides an integrated, coherent, compact, quantitative description of order, chaos and complexity. My *information variable method* may be quite general in its applicability, although this remains to be demonstrated.

A few comments are in order regarding the relevance of complexity research to physics. Complexity research goes against the grain of traditional physics research, which has had a strong reductionist tendency. Reductionists believe that the way to understand complex objects is to decompose them into simpler constituent parts. The logical conclusion of this line of reasoning is

that the (hypothetical) Super-Grand Unified Theory, which reduces the Universe to just one fundamental force, will implicitly explain everything in the Universe [Davies].

There is a growing awareness that "More is Different." In a 1972 article with that title, Anderson "emphasized the concept of *broken symmetry*, the ability of a large collection of simple objects to abandon its own symmetry as well as the symmetries of the forces governing it and to exhibit the *emergent property* of a new symmetry" [Anderson]. He cites the Bardeen-Cooper-Schrieffer theory of superconductivity as an example of a broken gauge symmetry. He also notes the role of disorder in emergent properties as exemplified by spin glasses [Stein-1], which require a new version of statistical mechanics. Examples of emergent behavior are spontaneous pattern formation (often of fractal dimension) [Campbell, Newell] and self-organized criticality, when a system organizes itself into a critical state with scaling laws like those at critical points of phase transitions [Bak]. An essential ingredient in all these cases is nonlinearity. Indeed, nonlinear systems with only a few degrees of freedom can exhibit complex as well as chaotic behavior.

Emergent properties disappear when an attempt is made to reduce complex systems to simpler ones. Davies suggests that there are undiscovered laws at work that he calls *software laws*. [Davies]. They are laws of organization, structure, and behavior that are independent of the physical medium (the hardware). Thus, they apply equally well to physical, chemical, biological, social and economic systems. They even apply to the logical universe inside computers. The software laws, if they exist, would probably recognize information rather than energy as a fundamental quantity. One of the software laws might resemble the second law of thermodynamics, except that it would imply that the complexity of some sub-systems (e.g., the biosphere) is non-

decreasing with time and would explain how our diverse world could have evolved from a featureless, totally symmetric early Universe, even though the disorder (entropy) of the entire Universe is also non-decreasing with time.

Systems composed of many objects with nonlinear interactions defy traditional analytical techniques. Computer simulation of discrete models of physical systems has gained acceptance as a valid technique. A class of cellular automata (cf. sec. 2.3) known as lattice gasses have been successfully used to model a variety of physical systems traditionally modelled by partial differential equations, such as the Navier-Stokes equations from hydrodynamics [Doolen, Manneville]. Cellular automata consist of arrays of sites with simple local dynamics and are ideally suited to parallel computer simulation.

Before delving into the details of the information variable method, a brief survey of related subjects and relevant earlier complexity research is given in chapter 2. The method is presented in chapter 3. It is then applied to cellular automata in chapters 4 and 5 and to one-dimensional maps in chapter 6. Chapter 7 contains conclusions and suggestions for further research.

2. Background

Complexity research is truly interdisciplinary. A brief survey of subjects most closely related to the primary topic of this dissertation is presented below. These subjects are nonlinear dynamics, complex systems, cellular automata, computation theory, information theory and the theory of randomness. Following that is a brief survey of relevant earlier complexity research.

2.1. Nonlinear dynamics [Campbell, Gleick]

Perhaps the most significant relevant historical event was the rediscovery of deterministic chaos attributed to Lorenz in 1963. (Poincaré noticed its existence at the turn of the century, but he lacked an essential resource that Lorenz possessed – a digital computer.) However, it wasn't until the early 1980's that the full significance of the discovery began to be realized, primarily by a team of graduate students at UC Santa Cruz: Crutchfield, Packard, Farmer and Shaw. Since then there has been an explosive growth of interest in chaos. Some have even suggested that the discovery of chaos ranks in importance with the discoveries of relativity and quantum phenomena as being the major revolutions in physical understanding of this century.

The field of nonlinear dynamics blossomed in the 80's as the implications of chaos were incorporated. The jargon of this subject: transients, attractors, repellers, fixed points, limit cycles, strange attractors, bifurcations, period doubling, fractal dimension, Lyapunov exponents, sensitivity to initial conditions, etc., grew and spread to other fields. Some techniques of nonlinear dynamics, such as portraying behavior in phase space, shed light on previously misunderstood phenomena ranging from weather patterns to brain waves. Underlying order could now be seen where only 'noise' was apparent before. Scientists

became aware that very simple dynamical models can produce complex behavior if they have enough nonlinearity to allow for chaos.

2.2. Complex systems [Stein-2, Jen]

'Complex system' is still a loosely defined term. It does not, as its name suggests, cover the entire realm of systems that defy our ability to understand them in neat analytical terms. A distinction is made between arbitrarily complicated systems and those that have simple, easily understood and often identical components, but that display emergent or *self-organized* behavior – behavior that results from interaction between the components and cannot be understood by consideration of the components in isolation. These conditions can be met when the components are inherently nonlinear.

Examples of frequently studied complex systems are: fluids, spin glasses, neural networks, lattice gasses and populations (genetic and social). Results from these studies show a surprising amount of universality in such matters as transitions from order to disorder and response to perturbations. There is a movement toward an abstract theory of complexity – one that is equally applicable to organisms in a population and neurons in a brain. Components of a system are viewed as having known behavior in the context of their immediate environment. System-level behavior emerges as a result of information flow between components as constrained by the topology of interconnections and boundary conditions.

2.3. Cellular automata [Wolfram-5]

In an effort to isolate the essential mechanisms of complex behavior, cellular automata (CA) are studied. They are extremely simple in construction yet capable of highly complex behavior. Composed of arrays of identical cell sites connected in uniform spatial patterns, they may be infinite in extent or limited by boundaries. Each cell has a finite number of possible discrete states. The next

state of a cell is determined by the present states of cells in a surrounding neighborhood. All cells in an array are updated simultaneously. Thus, CA are discrete in both time and space, and the deterministic rules governing their dynamical evolution are local in both time and space.

CA were first created by Von Neumann and Ulam in 1948 as a model to study self-reproduction. A self-reproducing automaton with 29 states per cell was subsequently discovered by Von Neumann. Interest in CA research was renewed by two key developments. In 1970 Conway introduced the Game of Life CA [Berlekamp, Poundstone]. It is much simpler than Von Neumann's CA, with only two states per cell and a nine cell neighborhood, but displays genuinely complex behavior, including self-reproduction. A fascinating aspect of Life is its mixture of order and disorder. It has frequently recurring patterns yet is very difficult to anticipate. The other milestone was Wolfram's extensive study of one- and two-dimensional CA with a definite mathematical and physical rather than biological flavor [Wolfram-5].

In a series of papers from 1982-86 Wolfram established a new line of complexity research based on CA. In analogy with continuous nonlinear dynamical systems, Wolfram introduced four classes of CA based on their evolutionary behavior: Class I behavior resembles attraction to a fixed point, class II to a limit cycle, and class III to a strange attractor. Class IV CA, believed to be rare, are those CA not found in the first three classes. They are characterized by long transients and long correlation lengths, complex localized structures, and propagating structures.

The discrete nature of CA permits an entirely different analogy. By viewing CA as computing machines with their initial states as the 'programs', Wolfram was able to invoke the results of computation theory, thereby connecting two apparently distant areas of research. He speculated that class IV CA were

capable of universal computation (cf. sec. 2.4). One of the properties of universal computers, undecidability, inspired his notion of computational irreducibility. A system has this property if its future behavior cannot be predicted by any algorithm that is significantly shorter than an actual simulation of the system. In other words, there is no analytical short-cut of the type we have come to expect from traditional physics. For example, we take for granted that we can compute the time until the next solar eclipse long before the event takes place.

2.4. Computation theory [Hopcroft, Dewdney]

What is a universal computer? It is an idealized machine that can carry out any algorithm – a step-by-step procedure involving the manipulation of discrete units of information. In fact, according to the Church-Turing thesis, universal computers can compute anything that is computable! There is no concern whatever with speed or efficiency and no limit on memory storage capacity. Ordinary digital computers have the power of universal computers – or would have if additional expansion memory could be plugged in, without limit, as needed to complete computations. However, in computation theory, universal computers are usually discussed in terms of the simplest implementations in hardware, namely universal Turing machines.

A Turing machine is a finite automaton (to be discussed) which controls the operation of a read/write head. The head reads or writes symbols on an infinite memory tape, one at a time. The tape is used to store initial symbols, temporary results, and the final result of a computation. The simplest Turing machines have only one semi-infinite tape and two symbols. A universal Turing machine has a finite automaton designed to read some initial tape symbols as a program to reproduce the behavior of another arbitrary machine.

The power of Turing machines has an unavoidable consequence, *undecidability*, related to Gödel's arithmetic undecidability. Of the many instances of

computational undecidability the best known is the undecidability of the halting problem. The halt state is a designated state in the finite automaton of a Turing machine that is only entered upon completion of a computation. It is possible that a computation may never finish if the machine gets caught in an infinite loop. One can ask whether there is any single Turing machine that can determine, when given a description of another arbitrary machine with an arbitrary input, whether it will ever halt. Turing showed in 1937 that no such machine exists. Thus there is no single algorithm one can use that will always indicate the ultimate fate of an arbitrary computation. One must face the proposition of observing the computation in question for an indefinite period. It may halt in a reasonable time, but if it doesn't it may be impossible to determine whether it will eventually halt or never halt.

The implications of undecidability in physics are staggering. In 1990, Moore constructed a model physical system whose dynamics are undecidable, not merely unpredictable [Moore]. Even with exact knowledge of the initial condition, its future behavior is formally undecidable. It may eventually permanently settle into either ordered or chaotic behavior, but there is no way of determining if or when this might happen. For further, more speculative implications, see Penrose [Penrose].

If class IV CA are indeed capable of universal computation, they would share this property; but Culik and Yu demonstrated in 1988 that determining the class membership of an arbitrary CA is itself an undecidable problem [Culik]. This obstacle does not necessarily apply to every specific CA. Computational universality was proven for the Game of Life by explicit construction of a universal computer within the CA. Various objects – some found by chance – were combined to form basic computing elements such as logic gates, memory

elements and 'wires'. The construction depends heavily on the existence of gliders – propagating objects used to transmit information.

The temptation to equate complexity with computational power is strong. Vaguely analogous to Wolfram's four classes are the four levels of the Chomsky hierarchy of computing machines. Each level contains all lower levels as special cases and can perform some computations that they cannot. In order from most to least powerful are Turing machines, linear bounded automata, stack automata and finite automata. The latter three have restrictions on memory tape access. Finite automata are not allowed to use the tape for temporary memory storage.

The Chomsky hierarchy is rigorously defined, more so than Wolfram's classes, but lacks the resolution one hopes for in a quantitative scale. Also, it may be inherently inadequate as a measure of complexity because it does not take into account speed and efficiency. Nevertheless, it provides a reference with which any other measure based on computation must at least be consistent.

2.5 Information theory

This is a vast subject; here it is only mentioned that in 1981 Shaw applied information theory to chaotic dynamics [Shaw]. He showed how the unpredictability associated with deterministic chaos is a byproduct of a continuous chaotic system's ability to magnify microscopic information. If the microscopic information happens to be random, then random behavior appears on a macroscopic scale. Order, in contrast, results from an orderly system's ability to absorb information that might otherwise influence it. Shaw's approach can be adapted to systems with a spatial aspect. For example, in CA chaos is manifested as information transmission rather than magnification.

Information is a fundamental quantity [Landauer]. It is useful in both dynamical and computational modes of description and applies naturally to physical systems when statistical behavior is emphasized. In fact, there is a

somewhat mysterious equivalence between the mean value of information over the states a system and the thermodynamic quantity *entropy* [Zurek]; the two descriptions are often used interchangeably.

2.6 Randomness

An important development in complexity research, emphasized by Grassberger, Huberman, Bennett and others [Grassberger-1, Huberman, Bennett-1] is the recognition that pure randomness is simple from a statistical point of view. According to algorithmic complexity (or information) theory [Chaitin], in which the complexity of a string of N symbols is equated to the length of the shortest program that could direct a universal computer to reproduce it, a truly random string is most complex. By virtue of its randomness, the string has no hidden pattern. There is no shorter way to direct the computer than to explicitly list all N symbols. However, if one is interested only in reproducing the random quality of the string, in a statistical sense, rather than the exact sequence of symbols, then one needs merely to instruct the computer to read a register with randomly changing contents, equivalent to flipping a coin, N times. (The register might be connected to a simple quantum device.)

Consider the program consisting of the instruction PRINT CONTENTS OF X, N TIMES, where X is the address of a memory register. The minimal length of the program is log(N) plus a constant. If the register contains a fixed symbol then the computed string is perfectly ordered, but if it contains a randomly changing symbol then the string is random. The computational effort required to generate either a generic random string or a perfectly ordered string of the same length is identical.

Therefore, algorithmic complexity is often referred to as algorithmic randomness. It is closely related to entropy; the average algorithmic randomness of any concisely defined ensemble of strings is nearly equal to the entropy of the

ensemble [Bennett-3]. Complexity or randomness might be considered properties of the ensemble of 'similar' strings to which a particular string belongs [Grassberger-1].

Another important result from the theory of randomness is due to Martin-Löf [Martin-Löf]. He showed that nearly all infinite binary sequences or, equivalently, binary representations of real numbers on the unit interval, are random. In terms of algorithmic randomness, there aren't enough short stings available to encode longer strings, so most longer strings don't have a shorter description. In the limit of infinite strings, 'most' becomes 'nearly all'.

2.7. Brief survey of relevant earlier complexity research

The complexity research of Wolfram, Grassberger, Lindgren and Nordahl, Crutchfield and Young, Langton, Bennett, and their collaborators is briefly discussed to provide a context for my original research.

2.7.1. Wolfram

Wolfram showed that the sets of spatial patterns generated by CA (from random initial patterns) after a finite number of time steps form regular languages [Wolfram-3]. Regular languages are sets of strings of symbols acceptable by finite automata [Hopcroft]. A finite automaton accepts a language if there is a path in the automaton leading from a designated starting state to one of a set of final (halt) states for every possible string in the language. Wolfram defined the *regular language complexity* of a CA to be (the logarithm of) the size of the minimal finite automaton that can accept the language generated by the CA. He found that this complexity is usually non-decreasing as time evolves. In the limit of infinite time, some CA correspond to more complicated languages than regular languages; the regular language complexity itself becomes infinite. The more complicated languages are accepted only by more powerful machines on the Chomsky hierarchy: context-free languages by stack automata, context-sensi-

tive languages by linear bounded automata, and recursively enumerable languages by Turing machines.

2.7.2. Grassberger

Wolfram's regular language complexity measure considers all possible spatial patterns that can be generated by a CA. Grassberger's *set complexity* measure weights the states of the accepting automaton by their probability [Grassberger-1]. This *metric,* rather than *topological,* emphasis is preferred for physical complexity measures since it discounts possible but unlikely events.

Grassberger also defines *effective measure complexity* as the amount of information that must be stored in order to make an optimal prediction about the next symbol in a very long string. It is computed from probabilities of contiguous blocks of symbols of all lengths and takes into account that the information stored in a sequence of symbols can be more than the sum of the information stored in the individual symbols of the sequence. It is called 'effective' because is effectively computable. It can be computed directly from a pattern even when the underlying generating mechanism is not known.

Zambella and Grassberger define *forecasting complexity* as the minimal weighted size of a *deterministic* finite automaton that can generate a given block probability distribution [Grassberger-4]. It is more sophisticated than effective measure complexity because it quantifies the amount of information needed to predict *probabilities* of symbols rather than symbols themselves. However, it is not effectively computable and an approximation scheme may be necessary; also it is infinite in most situations of interest [Grassberger-5].

2.7.3. Lindgren and Nordahl

Another variation on the general approach initiated by Wolfram, where attention is focused on the complexity of spatial patterns produced by CA, is due to Lindgren and Nordahl [Lindgren-1]. Their measures, *finite state source complex-*

ity (topological) and *generating complexity* (metric), are the minimal size of the *nondeterministic* finite automaton that can generate a given block probability distribution. The use of a nondeterministic automaton (cf. sec. 3.2) ensures finiteness and makes these measures more useful than Grassberger's forecasting complexity.

2.7.4. Crutchfield and Young

Crutchfield and Young adapted some of the methods discussed above to the task of inferring the complexity of a dynamical system by analyzing the data time series that it produces [Crutchfield-1, Crutchfield-2]. Time series take the place of spatial patterns. The method begins by constructing a tree of possible data sequences of a given length. Probabilities are added in the metric version of the method. The 'diversity' of the tree is minimal for a purely ordered or disordered source in the sense defined by Huberman and Hogg [Huberman]. The next step is to construct a finite automaton that has a separate state for each unique sub-tree. Complexity is defined as the minimal size of the automaton. If the complexity diverges with data sequence length, then a more powerful computing machine on the Chomsky hierarchy must be invoked. Thus an iterative process is required involving increasing the sequence length and possibly revising the computational model. The process ends when the observed data can be statistically reproduced within a specified tolerance.

Their complexity measure was applied to the logistic map (cf. chapter 6) and it was found that complexity is greatest at the boundary between order and disorder [Crutchfield-3]. When complexity is graphed vs. entropy, there is a sharp peak in complexity at an intermediate value of entropy. At the peak the complexity based on a finite automaton model is infinite and a stack automaton model is required to regain finiteness. Crutchfield and Young interpret this transition from low entropy order to high entropy disorder as a phase transition.

2.7.5. Langton et al.

Langton and his collaborators also observed a phase transition between order and disorder, but they used an entirely different approach [Langton-3, Li-2, Wooters]. Langton equates complexity to computational power, but at a more basic level than the Chomsky hierarchy. The hierarchy organizes computing machines according to their ability to implement specific algorithms, without regard for efficiency. (Although not a feature of the hierarchy, the size of machines within a given level is considered significant by complexity researchers attempting to define computational power.) Computational power can also be defined in terms of the basic elements of computation – namely transmission, storage and occasional modification of information – which can be measured directly.

Langton suggests that the empirical observation of high levels of mutual information (cf. sec. 3.3) is an indication of computational activity and hence complexity. Mutual information measures the correlation between information at sites separated by time or space. A correlation in space may imply information transmission (due to chaos) while one in time may imply information storage (due to order). Both are needed for computation but too much of one would inhibit the other.

Langton applied mutual information to CA. He introduced a parameter that modifies CA rules so that they change from class I to II to III. He found that temporal mutual information peaks as the rules change from class II to III. The CA in this small region have the features associated with class IV. He proposed the idea that class IV CA lie in the region of a phase transition in CA rule space. Rules on one side exhibit primarily ordered behavior, those on the other side chaotic behavior. For the rare rules found in the transition region, order and chaos are balanced. This allows for the existence of marginally stable structures

that can preserve information indefinitely yet may participate in exchanges of information with other structures. The implication is that such structures and interactions are necessary ingredients for higher levels of computation.

2.7.6. Bennett

Bennett introduced *logical depth* as a measure of complexity consistent with intuitive notions [Bennett-1, Bennett-2]. It is based in part on algorithmic complexity, which considers the complexity of a string to be the shortest program that can direct a universal Turing machine to reproduce it. Algorithmic complexity can be generalized to the reproduction of arbitrary objects by universal computers of any type. Logical depth is defined as the execution time of the shortest program that can reproduce a given object. Provisions can be made to make the measure machine independent. Thus logically deep objects contain internal evidence of being the result of unavoidably (except by remote chance) long processes. This measure agrees intuitively with biological complexity. In a sense it quantifies Wolfram's notion of computational irreducibility.

Logical depth is conceptually simple and sound. However, it is practically difficult to apply. It shares the limitation of undecidability with algorithmic complexity, although approximation methods exist to deal with this problem.

3. The Information Variable Method

The nonlinear dynamical concepts of order and chaos are integrated into a computational model applicable to discrete dynamical systems by using information as the unifying language of description. Measures of complexity based on information flow follow naturally.

3.1. Applicability of the method

The method is directly applicable to dynamical systems discrete in both time and space. Continuous systems can be discretized to any desired resolution, given enough resources. Discretization is always necessary when digital computer analysis is involved and for many complex systems this may be the only effective approach.

Infinite (or even large) systems have infinite (or vast) numbers of states. These systems must be localized to make them manageable. An essential feature of the method is to treat systems as having open boundaries across which information can flow. Thus, by defining imaginary boundaries containing relatively small regions embedded within large systems, localization is achieved. An important question is whether the complete dynamics of systems can be grasped by studying such small regions. The answer probably depends on the homogeneity and degree of locality of the systems. In the case of CA, which are homogeneous and have locally acting rules, the results are encouraging.

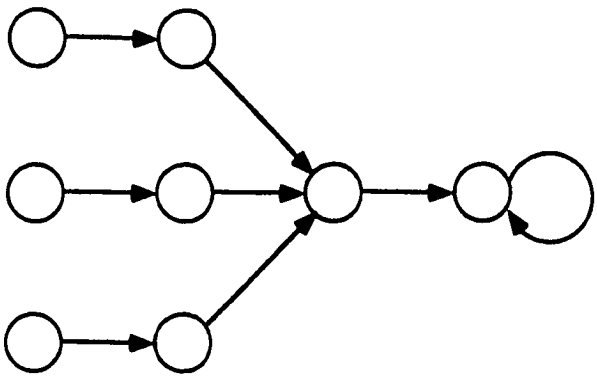
Although not necessarily a restriction on the method, all dynamical systems considered are assumed to be deterministic; their evolution is governed by rules that have no element of chance.

3.2. State space description of closed and open systems

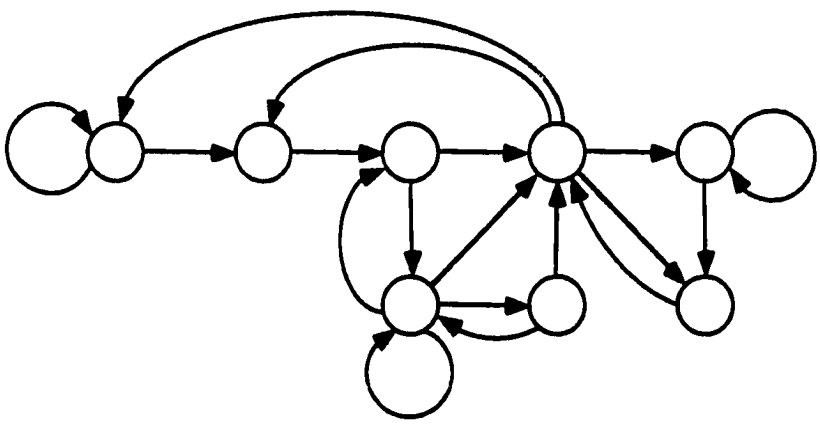
The information variable method relies on state space descriptions of discrete dynamical systems, analogous to phase space descriptions of continuous dynamical systems. At some point in time, a given system is in a well-defined state. As a single discrete unit of time passes, the system makes a transition to another state. The dynamics can be completely captured in graphical form if the system is finite.

The state graph of a particular discrete, finite *closed* system is depicted in figure 3.1a, with nodes representing the possible states and directed links the allowed transitions between states. Graphs of closed systems always have one or more basins of attraction that consist of transient states leading to an attractor. In this case there is one basin of attraction with seven transient states leading to a single-state 'fixed point' attractor. The system is deterministic, so one and only one link leaves each node. Quantities such as number of basins, attractor size and transient length may vary erratically with system size (see the appendix of [Wolfram-5]). Internal information flow, the relative roles of chaos and order, and complex behavior are obscured.

These deficiencies are largely remedied by considering the system to have *open* boundaries; it is allowed to exchange information with its environment. The next state of the system depends on external information. Thus, the open version of the same system, shown in figure 3.1b., can have several links leaving a node. Sensitivity to external information is the discrete analog of sensitivity to initial conditions in continuous systems: the hallmark of deterministic chaos. Therefore, chaos takes the form of diverging links in the graph. Order, the tendency of attraction to a preferred subset of states, takes the form of converging links. I propose that complexity is



(a)



(b)

Figure 3.1. Elementary CA rule 110: (a) State graph of a 3-cell closed array with periodic boundary conditions, (b) NFA graph for a 3-cell region.

manifested as variability in the convergence and divergence of links throughout the graph, reflecting a mixture of order and chaos.

Models of the type shown in figure 3.1b are properly called finite automata. Although the systems being modelled are deterministic, they will be treated as if they are not. Predictable systems that depend on unpredictable information themselves become unpredictable. When several links leave a node, there is no way to know which one applies until external information is made available. The term nondeterministic finite automata (NFA) will be used to emphasize this point.

3.3. Quantifying the description using information variables

The chaos, order, and complexity inherent in the topology of an NFA graph can be quantified by stimulating the system with an information-rich source in the same way that an electric circuit is studied by driving it with a frequency-rich impulse or square wave. In this sense information is used as a tool to probe (and somewhat refine) the topological structure of the NFA.

The richest source of information is one in which the possible discrete values have equal probabilities. Given such an unbiased random source, the graph links can be labeled with forward conditional transition probabilities, as in figure 4.1, according to the probability that the information source produces the input value(s) that corresponds to the allowed transition. The NFA now becomes a Markov chain, but the term 'NFA' is preferred because of the reference to computation.

The Shannon information associated with an event is log(1/p), where p is the probability of the event. In an NFA an event can be the occurrence of a state or a transition between states. Given unbiased random stimulation, forward conditional transition probabilities can be derived if the dynamics of the deterministic system are known. If i is the present-state, then j will be the

next-state with probability $p_{i\to j}$. Let p_i denote the probability that state i is the present-state and p_{ij} the probability that a transition from state i to state j occurs. Then

$$p_{ij} = p_i\, p_{i\to j}. \tag{3.1}$$

Since the probabilities are normalized,

$$p_j = \Sigma_i\, p_{ij} = \Sigma_i\, p_{ji}, \tag{3.2}$$

where the sum is over all states. Therefore,

$$p_j = \Sigma_i\, p_i\, p_{i\to j}, \tag{3.3}$$

and the p_i can be computed from the $p_{i\to j}$ by matrix methods. Note that multiple attractors are possible and transient states have vanishing probability.

Ignoring transient states while considering only attractor states is appropriate in the information variable method, which considers large systems to be composed of sub-systems with relatively few states. Long-term global transient behavior important in complexity considerations is attributed to information flow between sub-systems, which may persist long after local transients are gone. Steady-state (asymptotic) behavior of NFA models is implicitly assumed throughout this dissertation.

Pertinent to the issue of information flow are the reverse conditional transition probabilities, $p_{i\leftarrow j}$. If state j is the present-state, then state i was the prior-state with probability $p_{i\leftarrow j}$. Since

$$p_{ij} = p_{i\leftarrow j}\, p_j, \tag{3.4}$$

the $p_{i\leftarrow j}$ can be computed easily.

Knowledge of the state and transition probabilities permits the computation of several information variables. The information needed to specify the present-state of the NFA, the *state information*, is

$$I_i = \log(1/p_i), \tag{3.5}$$

the mean of which is the *entropy*,

$$< I > = \Sigma_i \, p_i \, I_i. \tag{3.6}$$

$< I >$ is a measure of state utilization. It has a maximum value of log(n), where n is the number of states, when all states are equally probable. The *information gain* G_{ij} associated with a transition from state i to state j is defined as

$$G_{ij} = \log(1/p_{i\rightarrow j}). \tag{3.7}$$

$G > 0$ when links diverge. The *information loss* L_{ij} is similarly

$$L_{ij} = \log(1/p_{i\leftarrow j}). \tag{3.8}$$

$L > 0$ when links converge. *Net information gain* Γ_{ij} is just

$$\Gamma_{ij} = G_{ij} - L_{ij} = \log(p_{i\leftarrow j}/p_{i\rightarrow j}). \tag{3.9}$$

The latter variables can be averaged over all transitions in the NFA. For example,

$$< \Gamma > = \Sigma_{ij} \, p_{ij} \, \Gamma_{ij} \, . \tag{3.10}$$

It is easily verified that $< \Gamma > = 0$ and $< G > = < L >$. However, the mean-square deviation of Γ,

$$\sigma_\Gamma^2 = < (\Gamma - < \Gamma >)^2 > = < \Gamma^2 >, \tag{3.11}$$

need not be zero. σ_Γ captures the degree of variability in link convergence and divergence in the graph and hence the fluctuation in the relative dominance of chaos or order as the system evolves in time. Fluctuation in Γ indicates temporary memory storage capacity. In extremely chaotic, periodic or ordered systems (systems with extremes of entropy, cf. sec. 3.5), $\Gamma_{ij} = 0$ for every transition and therefore $\sigma_\Gamma = 0$. Such systems have no temporary memory storage capacity and are unable to compute.

Combining equations (3.1), (3.4), (3.5) and (3.9) gives

$$\Gamma_{ij} = \log(p_i/p_j) = I_j - I_i. \tag{3.12}$$

Thus, Γ can be computed from state probabilities alone, implying that the *cumulative net information gain* for a sequence of transitions is path independent: only the initial and final states are relevant. To avoid confusion, the composite symbol $\Sigma\Gamma$ will be used for sequences of transitions:

$$\Sigma\Gamma_{in} = \Gamma_{ij} + \Gamma_{jk} + \ldots + \Gamma_{lm} + \Gamma_{mn}. \tag{3.13}$$

It is easy to show that

$$\Sigma\Gamma_{in} = \log(p_i/p_n) = I_n - I_i. \tag{3.14}$$

$I_n - I_i$ is a measure of the rarity of state n relative to state i and $\Sigma\Gamma_{in}$ is the cumulative net information gain needed to reach state n from state i. Note that $\Sigma\Gamma_{ii} = 0$; the cumulative net information gain around any closed path of an NFA is zero.

The properties of Γ and $\Sigma\Gamma$ permit a physical analogy. $\Sigma\Gamma$, or relative I, is an *information potential,* while its discrete derivative (single time-step difference), Γ, is an *information force.* External information in the form of unlikely (sequences of) input values is needed to force a system 'uphill' into a rare state with large potential. From there it may spontaneously 'fall' into a common state, dissipating stored information in the process.

Using (3.14) and designating a reference state 0,

$$\Sigma\Gamma_{0i} = I_i - I_0. \tag{3.15}$$

Averaging over all states

$$<\Sigma\Gamma_0> = <I - I_0> = <I> - I_0, \tag{3.16}$$

$$<\Sigma\Gamma_0^2> = <(I - I_0)^2> = <I^2> - 2\,I_0 <I> + I_0^2, \tag{3.17}$$

and thus

$$\sigma_{\Sigma\Gamma_0}{}^2 = <\Sigma\Gamma_0^2> - <\Sigma\Gamma_0>^2 = <I^2> - <I>^2 = \sigma_I{}^2. \tag{3.18}$$

The fluctuation in $\Sigma\Gamma_0$ is equal to the fluctuation in I. The latter description will be used to emphasize the path-independent nature of the quantity. Although derivable from net information gain associated with transitions

between states, σ_I applies to models of dynamical systems where only state probabilities are known, including non-Markov chain models. σ_I, like σ_Γ, is zero for systems with extremes of entropy.

Information variables have been introduced to quantitatively describe chaos and order in state space, but they also have significance in the time domain. σ_Γ is the long-time rms deviation, or fluctuation, of the Γ time series of a randomly stimulated NFA, assuming the state and transition probabilities have been established. Similarly, σ_I is the fluctuation of the $\Sigma\Gamma$ time series about its mean.

The difference between σ_Γ and σ_I is one of duration in the time domain or scope in state space. A suggestive computational analogy can be made. σ_Γ may indicate whether a system has the basic components of computation: transmission, temporary storage and occasional modification of information – analogous to wires, memory elements and logic gates in a computer; σ_I may indicate whether the components are 'connected' appropriately to achieve higher-level multi-step computational tasks – analogous to the computer itself.

Mutual information is not an essential variable in the method, but is mentioned because of its frequent use as a complexity measure [Li-3]. Temporal mutual information is defined as

$$M_{ij} \equiv \log_2(p_{ij}/p_i\, p_j). \tag{3.19}$$

It is a measure of the correlation between state j and state i or vice versa. If the knowledge that the present-state is i does not provide any information about what the next-state j will be, then $p_{ij} = p_i\, p_j$, $M_{ij} = 0$, and the two states are considered independent. On the other hand, if $p_{i\to j} = 1$ so that state j is certain to follow state i, then $p_{ij} = p_i$, $M_{ij} = I_j$ (its maximum value) and all the infor-

mation needed to specify the next-state j is already contained in the knowledge of the present-state i.

The mean mutual information, $< M >$, is defined as in equation (3.10) and can be expressed in terms of $< G >$ and $< I >$:

$$< M > = < I > - < G > \tag{3.20}$$

M is sometimes a misleading measure. $< M > = 0$ normally implies unpredictability: all states are independent and $< G > = < I >$. However, if $< I > = 0$, corresponding to a single-state attractor, $< G >$ and $< M >$ also must be zero but the system is totally predictable.

3.4. Simple examples

Before applying the method in earnest, we first test it on very simple models to gain an intuitive sense of how information variables measure order, chaos and complexity. Figure 3.2a shows the most general two-state NFA. The states are numbered zero and one. Only two of the four forward conditional probabilities are independent:

$$p_{0\to0} + p_{0\to1} = p_{1\to0} + p_{1\to1} = 1. \tag{3.21}$$

Taking $p_{0\to1}$ and $p_{1\to0}$ to be independent and solving equation (3.3), the state probabilities are

$$p_0 = p_{1\to0}/(p_{1\to0} + p_{0\to1}) \tag{3.22}$$

and

$$p_1 = p_{0\to1}/(p_{1\to0} + p_{0\to1}) \tag{3.23}$$

The joint and reverse conditional probabilities can be obtained using equations (3.1) and (3.4).

Consider the special cases of two-state NFA shown in figures 3.2b, c, d and e. The various information variables for these NFA are displayed in table 3.1. Each case will be discussed in detail. It is difficult to illustrate the

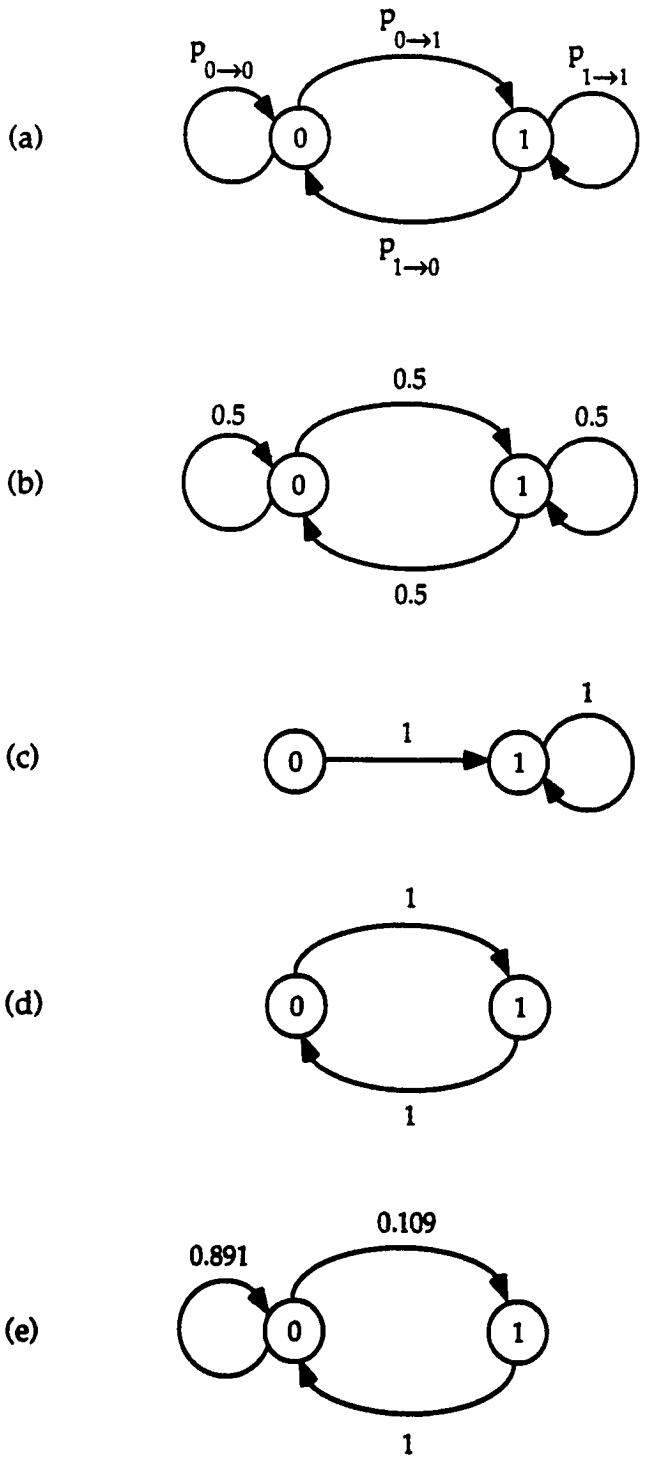


Figure 3.2. Two-state NFA: (a) general, (b) chaotic, (c) ordered, (d) periodic, (e) complex.

difference between σ_Γ and σ_I in systems with only two states, so little distinction will be made in this section.

Table 3.1
Analysis of two-state NFA

Figure	Class	$<I>$	$<M>$	$<G>$	σ_Γ	σ_I
3.2b	chaotic	1	0	1	0	0
3.2c	ordered	0	0	0	0	0
3.2d	periodic	1	1	0	0	0
3.2e	complex	0.46	0.015	0.45	1.42	0.95

3.4.1 Chaotic NFA

For the NFA of figure 3.2b, $< I > = 1$ bit, indicating full and equal state utilization. $M_{ij} = 0$; knowledge of the present-state provides no information about what the next-state will be. The bit of information needed to select the next-state originates external to the NFA. $G_{ij} = L_{ij} = 1$ bit, indicating that one bit is gained by the NFA upon leaving state i for state j, only to be lost upon arriving in state j. Thus $\Gamma_{ij} = 0$ and $\sigma_\Gamma = 0$, indicating there is never any net information gain during the NFA's evolution. It acts as a trivial *information pump*, or *information pipe*.

This NFA is *purely* chaotic (cf. sec. 6.2). As it evolves, an observer might witness a random sequence of state values. The information contained in them originates external to the NFA, the NFA merely acting as a conduit. Whenever two or more next-states follow the same present-state, inherently unpredictable external information is needed to select the next-state. If the external information is thought of as a flow of bits, most possible bit sequences are random (cf. sec. 2.6), resulting in an unpredictable sequence of state values and hence a large entropy. This does not mean that periodic

behavior is impossible. Though relatively rare, many of the possible input bit sequences are periodic and the evolution of purely chaotic NFA, being totally dependent on their input, would reflect this.

3.4.2. Ordered NFA

For the NFA of figure 3.2c, $< I > = 0$, indicating minimal state utilization (one state). $< M > = 0$, but in this case one cannot conclude that knowledge of the present-state gives no information about the next-state. In fact, the opposite is true: knowledge of the present-state entirely determines the next-state. $< G > = < L > = 0$, indicating that information is neither gained nor lost as a result of the only possible state transition. Of course, $\sigma_\Gamma = 0$, indicating there is never any net information gain during the NFA's evolution.

This NFA is opposite in character to that of figure 3.2b. It represents order rather than chaos. It shows a preference for a subset of states even when exposed to disrupting influences. Any noise or perturbation that might cause a transition to state zero would be quickly absorbed. No external information is needed during evolution. In fact, information about the initial state of the NFA is quickly lost; its asymptotic behavior is independent of its initial state. Instead of being sensitive to external information, the NFA is insensitive.

3.4.3. Periodic NFA

For the NFA of figure 3.2d, $< I > = 1$ bit, indicating full and equal state utilization. $< M > = 1$ bit, implying a completely predictable succession of states. $< G > = < L > = < G > = \sigma_\Gamma = 0$. Information is neither gained nor lost during evolution of the NFA. This NFA is periodic. Its behavior has features in common with both chaotic and ordered NFA. It shares full state utilization with the former but has the extreme predictability of the latter. It is on the boundary between order and chaos, yet its behavior is not complex.

As this case illustrates, periodicity is not equivalent to order. The term 'order' means predictability even under the influence of non-deterministic disruptive influences. In an ordered NFA, there is an active mechanism providing stability against noise or perturbations. A periodic NFA may exhibit periodicity in a noise-free environment, but its behavior is easily disrupted. A noise-induced change of state would cause a permanent phase shift in its state sequence.

3.4.4. Complex NFA

The NFA of figures 3.2b, c and d have $\sigma_\Gamma = 0$ and therefore are unable to temporarily store information. Maximizing σ_Γ for the two-state case yields the conditions

$$p_{1\to0} = 1 \tag{3.24}$$

and

$$2\,(1 + p_{0\to1}) = -\log_e(p_{0\to1}), \tag{3.25}$$

the solution to which is $p_{0\to1} = 0.109$. (The designation of states zero and one is arbitrary.) Thus, the NFA of Figure 3.2e is the two-state NFA with maximum fluctuation in net information gain.

For this NFA $< I > = 0.463$ bit, indicating unequal state utilization. $< M > = 0.015$ bit indicating low predictability. It can usually be found in state zero, but about once in nine time-steps external information causes a transition to state one. A transition back to state zero necessarily follows, but until then the information that caused the first transition is 'remembered' by the NFA. Since the event is rare, it represents a significant amount of information. In fact, $G_{01} = 3.20$ bits of information are needed to select the $0\to1$ transition if the present-state is zero. But no information is lost as a result of the transition: $L_{01} = 0$. Therefore, the net information gained is $\Gamma_{01} = G_{01} - L_{01} = 3.20$ bits. On average $< G > = < L > = 0.448$ bit and of course $< \Gamma > = 0$, but Γ fluctuates so $\sigma_\Gamma =$

1.42 bits. I also fluctuates and $\sigma_I = 0.952$ bits. σ_I is sensitive to accumulation of information and the duration of storage. It is less than σ_Γ in this case because there is no accumulation and the duration is short.

The two-state NFA with maximum σ_I satisfies condition (3.24) and

$$2\,(1 + p_{0\to1}) \,/\, (1 - p_{0\to1}) = -\log_e(p_{0\to1}), \tag{3.26}$$

the solution to which is $p_{0\to1} = 0.091$. $< I > = 0.413$ bits and $\sigma_I = 0.956$ bits, only slightly different from the values associated with maximum σ_Γ.

Two-state NFA with greatest information fluctuation occur at intermediate values of entropy, about halfway between pure order and pure chaos or periodicity. They exhibit the basic feature of complex behavior: a mixture of order and chaos resulting in occasional information modification as well as information storage, albeit only for a single time-step.

3.4.5. General two-state NFA

Consider again the general two-state NFA of figure 3.2a. If $p_{0\to1}$ and $p_{1\to0}$ are designated as the two independent forward conditional transition probabilities, information variables can be plotted along the third direction of a 3-D graph. Figures 3.3, 3.4, and 3.5 show how $< I >$, σ_Γ and $< M >$ vary for different NFA.

Figures 3.6 and 3.7 show how $< M >$ and σ_Γ are correlated with $< I >$. These figures are provided for comparison with similar figures found in the literature and presented with the interpretation that the $< I >$ axis represents a continuum from order to disorder or chaos. Typically, complexity is greatest at intermediate values of entropy where order and chaos are both present. Clearly, $< M >$ is totally uncorrelated with $< I >$; but figure 3.7 shows that the largest value of σ_Γ is permitted (but not required) at an intermediate value of $< I >$. Extremes of $< I >$ permit only null complexity.

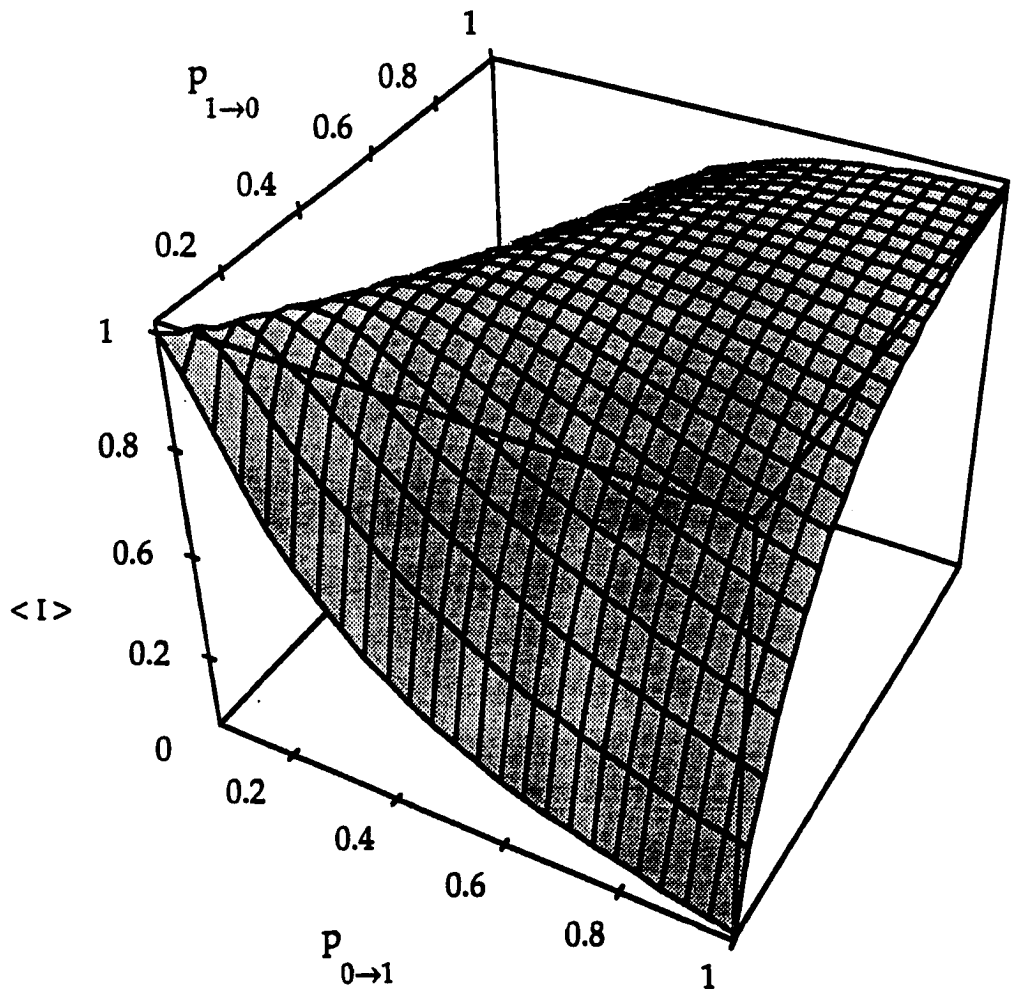


Figure 3.3. < I > vs. $p_{0\to1}$ and $p_{1\to0}$ for two-state NFA.

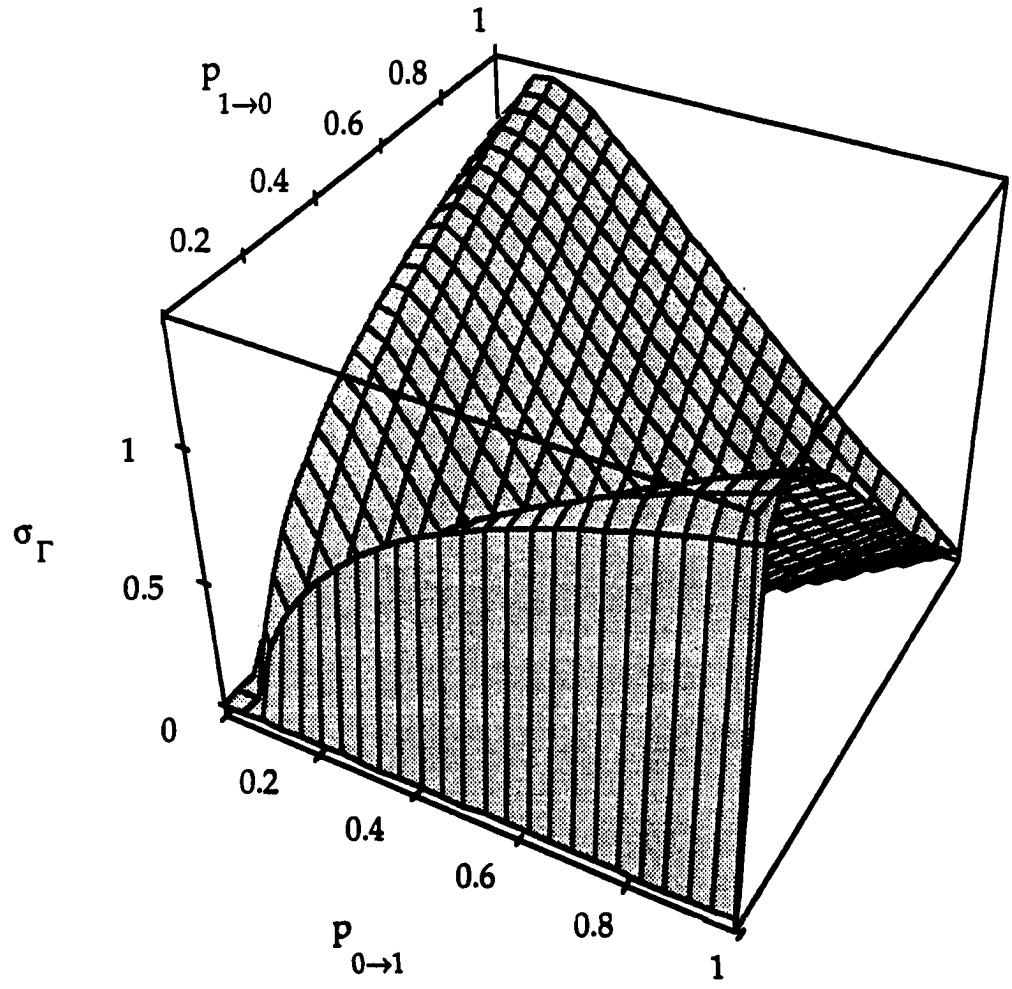


Figure 3.4. σ_Γ vs. $p_{0\to1}$ and $p_{1\to0}$ for two-state NFA.

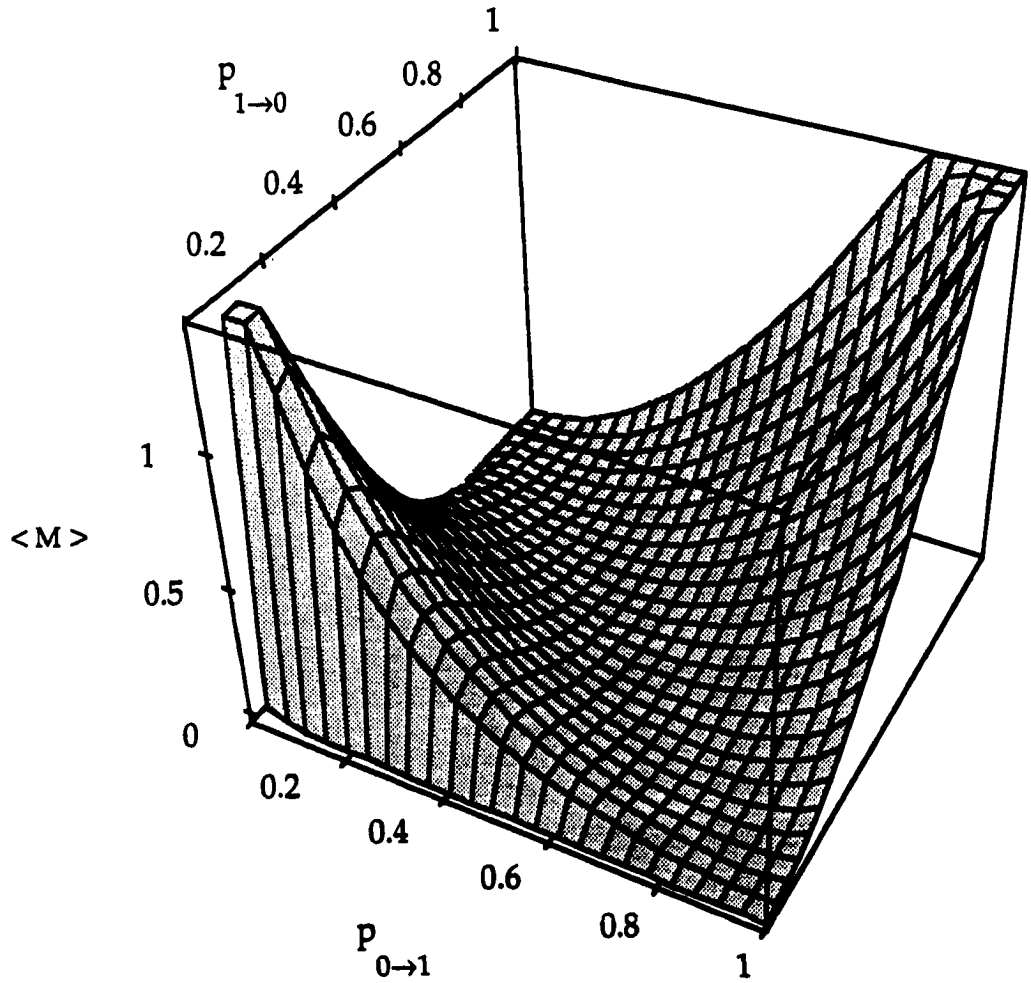


Figure 3.5. < M > vs. $p_{0\to1}$ and $p_{1\to0}$ for two-state NFA.

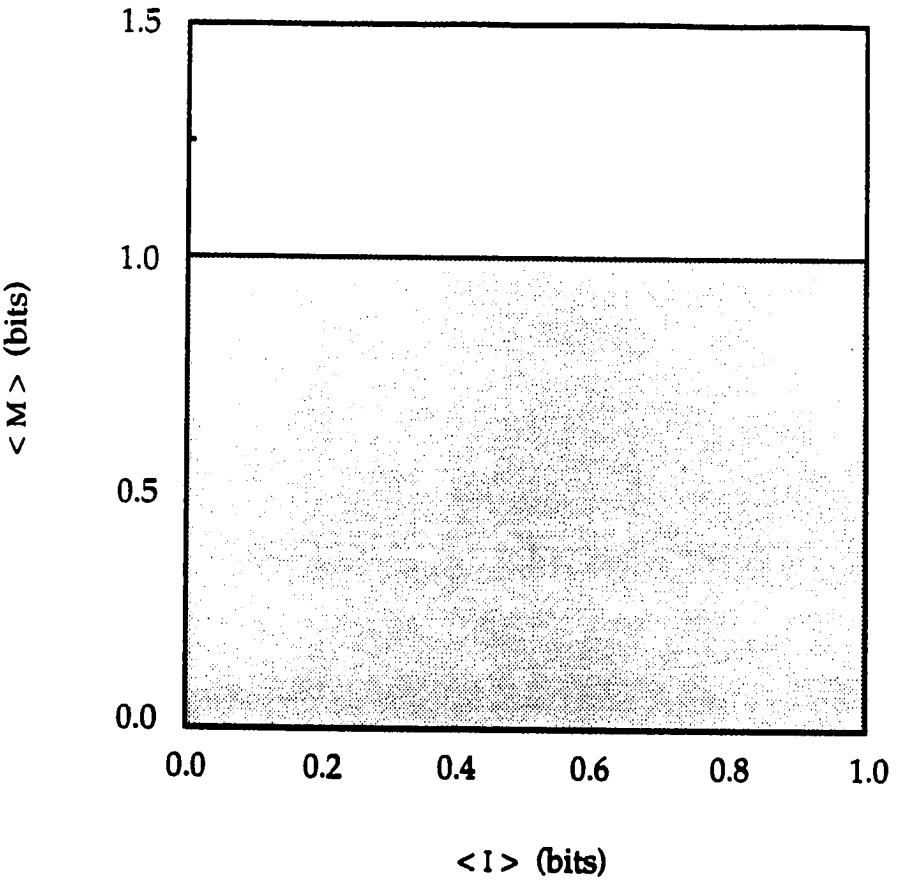


Figure 3.6. < M > vs. < I > for two-state NFA.

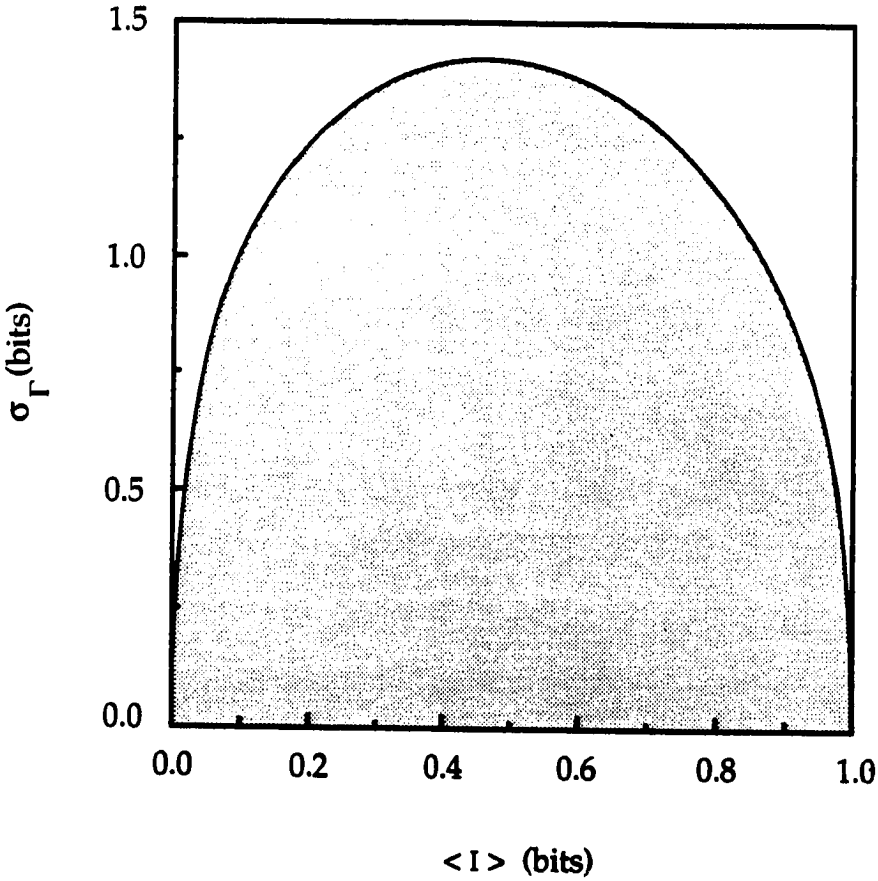


Figure 3.7. σ_Γ vs. < I > for two-state NFA.

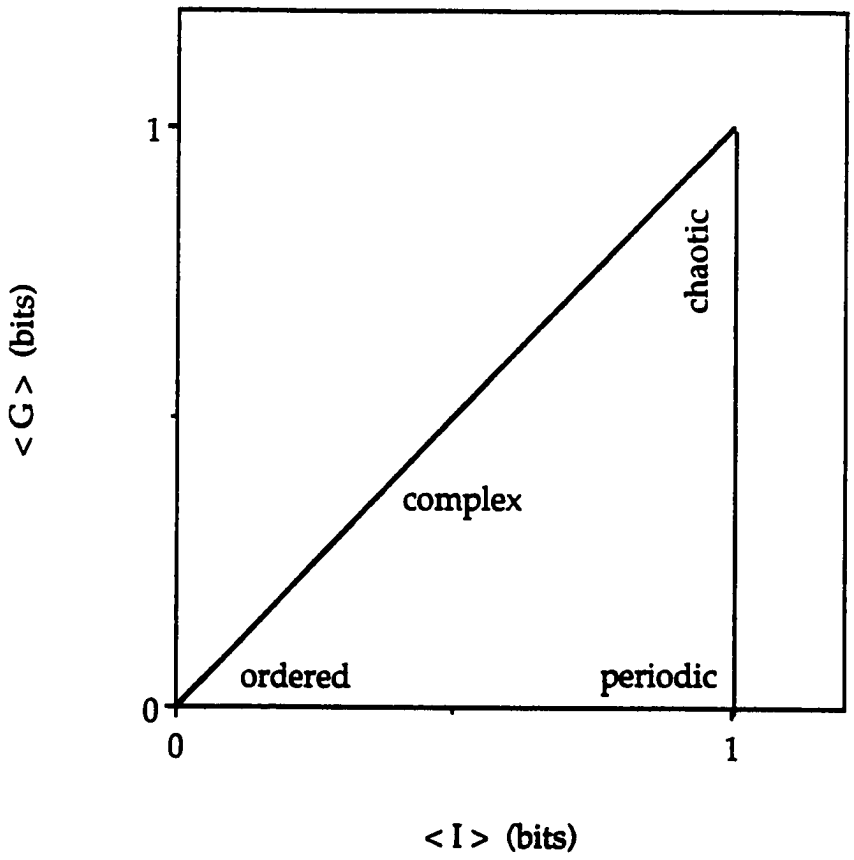


Figure 3.8. Dynamical behavior vs. < G > and < I > for two-state NFA.

Figure 3.8 shows how system behavior varies with < G > and < I >. Both variables are needed for complete classification. < I > alone does not resolve the difference between extreme chaos and periodicity. The area in the plane representing the most complex behavior is near the middle of the < G > = < I > (or < M > = 0) line. The order-chaos 'continuum' appears to be an oversimplified concept.

3.5. Classification of behavior using information variables

The discussion of section 3.4, although based on two-state NFA, has more general implications. Table 3.2 summarizes the classification of NFA. The qualitative visual impressions of link convergence and divergence in NFA graphs are considered the fundamental basis of the classification scheme. Information variables quantify these impressions and are considered to reflect rather than cause the various types of dynamical behavior.

Table 3.2
Classification of behavior

Class	Cause	Symptoms
chaotic	diverging links	large < G >, < I >; small < M >, σ
ordered	converging links	small < G >, < I >, σ; < M > ~ < I >
periodic	few converging or diverging links	large < I >, < M >; small < G >, σ
complex	variability in link convergence and divergence	medium < G >, < I >; small < M >, large σ

Table 3.3 shows how the information variables can be used as indicators of system behavior. < M >, although useful, is redundant. It certainly is not useful as a complexity measure, at least as applied here. (*Spatial* mutual information has not been considered here. As with temporal mutal informa-

Table 3.3 shows how the information variables can be used as indicators of system behavior. < M >, although useful, is redundant. It certainly is not useful as a complexity measure, at least as applied here. (*Spatial* mutual information has not been considered here. As with temporal mutal information, it is useful in some cases but can give misleading indications [Bennett-3].) < G > indicates chaos, but there is no single variable that indicates order, as distinct from predictability. The finer points of using σ_Γ, σ_I and their rates of change as complexity measures will be examined in the next chapters.

Table 3.3
Summary of information variables

Quantity	Indicates
< I >	state utilization
< M >	state utilization and predictability
< G >	sensitivity to external influences, chaos
σ	complexity

4. Application to Cellular Automata – General

Cellular automata provide a convenient subject for the information variable method. They are discrete, homogeneous, deterministic systems with local dynamics. Only one-dimensional CA with rule-range r = 1 (the next state of a cell depends on itself and its two nearest neighbors) are considered in detail. Attention is also restricted to CA with k = 2 (elementary CA) or k = 3 (trinary CA) states per cell. Localized regions of adjacent cells with open boundaries are modelled as NFA (cf. sec. 3.2). The states of the NFA are the set of all possible region-states. Allowed transitions between states are governed by the particular CA rule in effect in conjunction with the possible states of the *driver cells* – the cells bordering the region within range of the rule.

4.1. CA application example – elementary CA rule 110

Figure 1b shows the NFA model of elementary CA rule 110 for region size N = 3 cells. (Fig. 1a is a state graph of rule 110 for a 3-cell closed array with periodic boundary conditions.) There are $2^N = 8$ NFA-states and as many as four different next-states for each present-state, depending on the rule. The rule code '110' follows the standard Wolfram convention. It can be decoded by converting it from decimal to binary, 01101110, and aligning the bits with the binary triplet sequence

(111)	(110)	(101)	(100)	(011)	(010)	(001)	(000)
0	1	1	0	1	1	1	0

Each triplet represents the states of three adjacent cells. The next cell-state of the middle cell is the corresponding bit of the code. The use of binary digits is convenient when there are two possible states per cell.

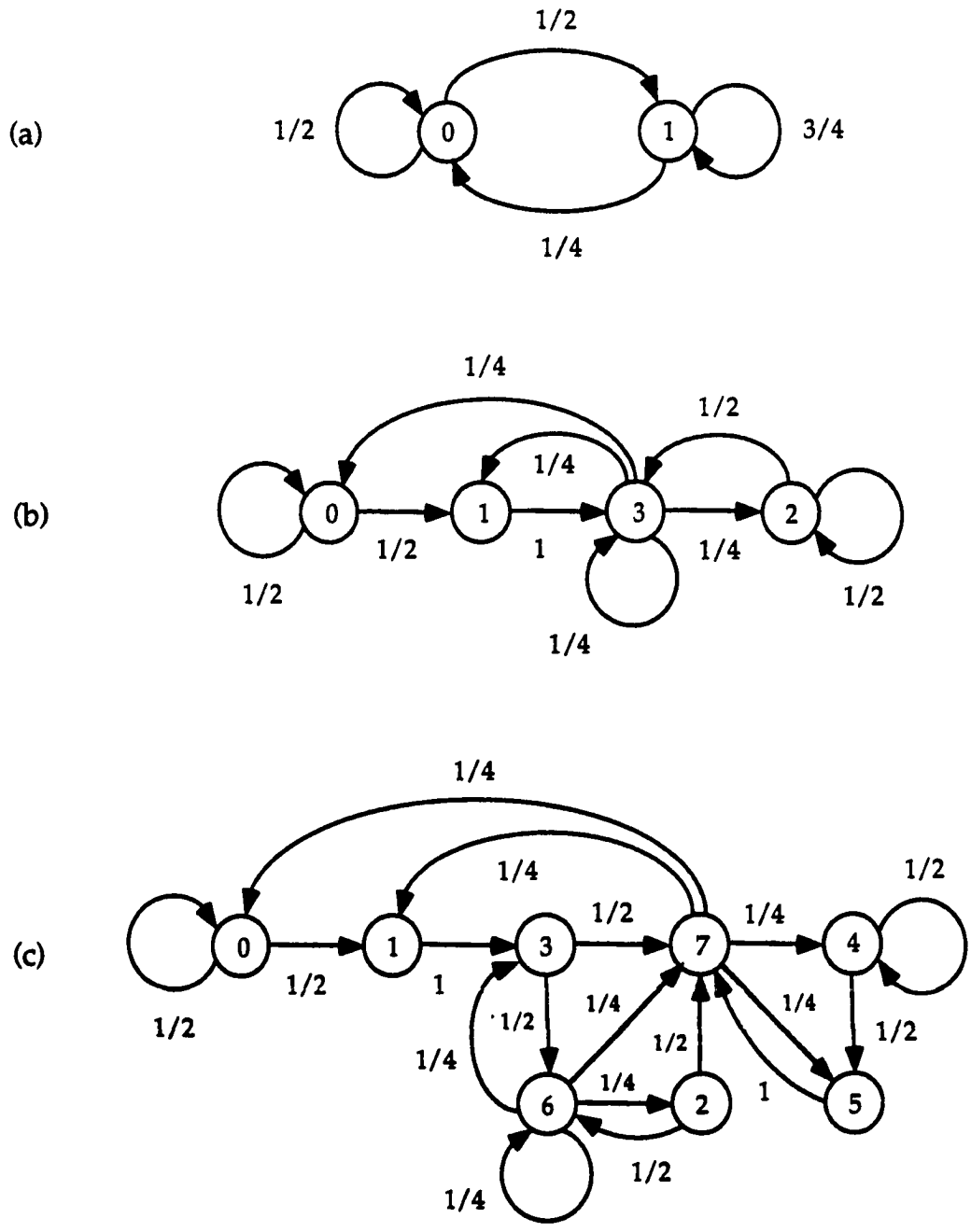


Figure 4.1. NFA for elementary CA rule 110 for (a) N = 1, (b) N = 2, (c) N = 3.

The first step in NFA construction is to assign nodes to every possible region-state. Nodes can be labeled with state codes obtained by treating the string of cell-states that constitute the region-state as a binary number, then (optionally) converting to decimal, as in figure 4.1. For N = 2, the region-states are 00, 01, 10 and 11, or in decimal 0, 1, 2 and 3.

Transition links are added by determining the four next-states associated with each of the four driver cell combinations for each present-state. For example, if the present-state is $11_2 = 3_{10}$ then the four transitions are $0|11|0 \rightarrow 11$, $0|11|1 \rightarrow 10$, $1|11|0 \rightarrow 01$ and $1|11|1 \rightarrow 00$. The next-states are 3, 2, 1 and 0 in decimal.

The next step is to evaluate the forward conditional transition probabilities assuming the four driver cell combinations are equally probable, consistent with unbiased random stimulation (cf. sec. 3.3). The transition links can be labeled with these probabilities as in figure 4.1b. Note that when several driver cell combinations cause a transition to the same next-state, the several probabilities are added and only one link is shown.

Information variables can then be computed (cf. sec. 3.3). The explicit steps are shown here for rule 110, N = 2. The forward conditional transition probabilities $p_{i \rightarrow j}$ are

i\j	0	1	2	3
0	0.5	0.5	0	0
1	0	0	0	1
2	0	0	0.5	0.5
3	0.25	0.25	0.25	0.25

The information gains G_{ij} are, in units of bits (base 2 logarithms),

i\j	0	1	2	3
0	1	1	–	–
1	–	–	–	0
2	–	–	1	1
3	2	2	2	2

The state probabilities are

i	0	1	2	3
p_i	0.2	0.2	0.2	0.4

The state informations I_i are, in bits,

i	0	1	2	3
I_i	2.32	2.32	2.32	1.32

The transition probabilities p_{ij} are

i\j	0	1	2	3
0	0.1	0.1	0	0
1	0	0	0	0.2
2	0	0	0.1	0.1
3	0.1	0.1	0.1	0.1

The reverse conditional transition probabilities $p_{i\leftarrow j}$ are

i\j	0	1	2	3
0	0.5	0.5	0	0
1	0	0	0	0.5
2	0	0	0.5	0.25
3	0.5	0.5	0.5	0.25

The information losses L_{ij} are, in bits,

i\j	0	1	2	3
0	1	1	–	–
1	–	–	–	1
2	–	–	1	2
3	1	1	1	2

The net information gains Γ_{ij} are, in bits,

i\j	0	1	2	3
0	0	0	–	–
1	–	–	–	–1
2	–	–	0	–1
3	1	1	1	0

$< I > = 1.92$ bits and $\sigma_I = 0.490$ bits. $< G > = < L > = 1.20$ bits, $< \Gamma > = 0$ and $\sigma_\Gamma = 0.775$ bits. These values are difficult to appreciate in isolation. How they vary with region size and how they compare with those of other rules determines their significance.

4.2. Dependence of information variables on region size

Figure 4.2 shows how σ_Γ and σ_I depend on region size for rules 110 and 22. There is generally a characteristic length scale Λ_Γ for each rule such that for $N > \Lambda_\Gamma$ there is little further change in σ_Γ and therefore in the variability of link convergence and divergence in the NFA graph. The progression in the 'variety' of the graph of rule 110 for N = 1, 2 and 3 is quite noticeable in figure 4.1. If the 'variety' of the topology is directly related to the variability of link convergence and divergence one would expect the graph to become topologically repetitive for N greater than about 5-6 cells, where σ_Γ begins to level off in figure 4.2a. This relationship has been confirmed for rules with small values of Λ_Γ but is more difficult to test for rules such as 110 due to the exponential increase in numbers of states with N. In contrast with σ_Γ, σ_I does not necessarily approach an asymptotic value as N increases.

The state-space structures associated with cooperating groups of adjacent cells are constrained by dimensionality, number of cell-states, and rule-range. It is reasonable to expect that these constraints prohibit complex, non-repetitive structures beyond a certain size. It is possible that the global complexity of such systems arises from the interaction of localized regions of cells of order Λ_Γ in

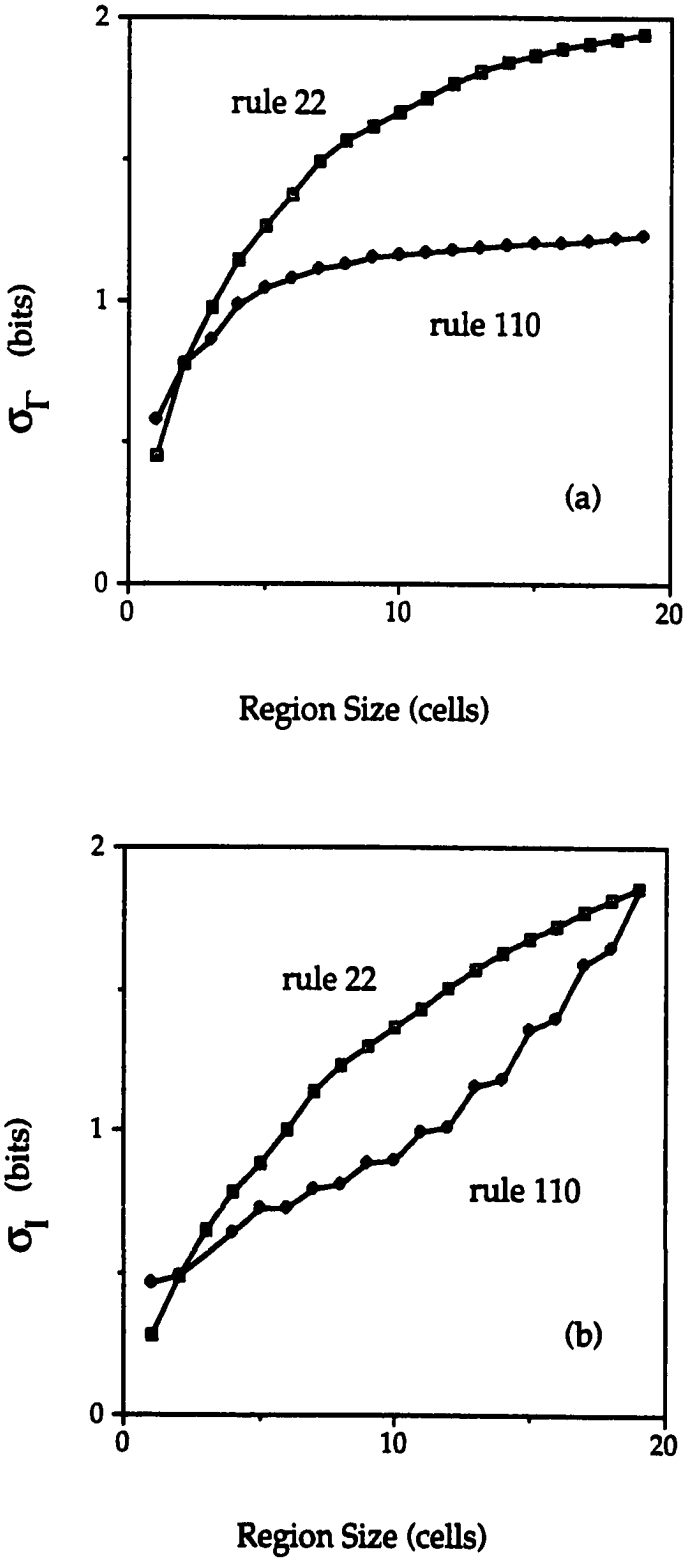
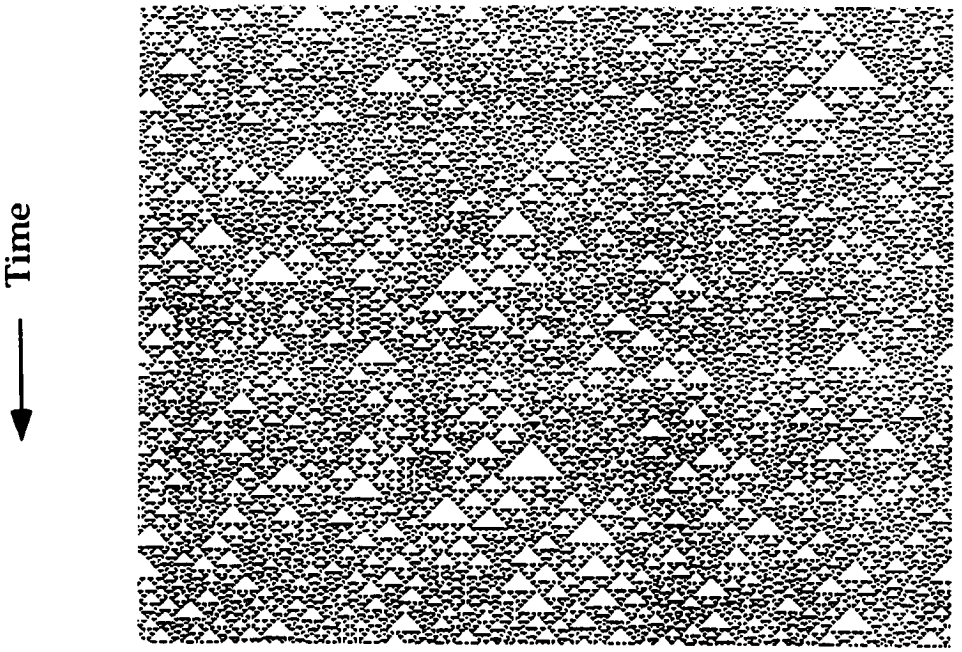
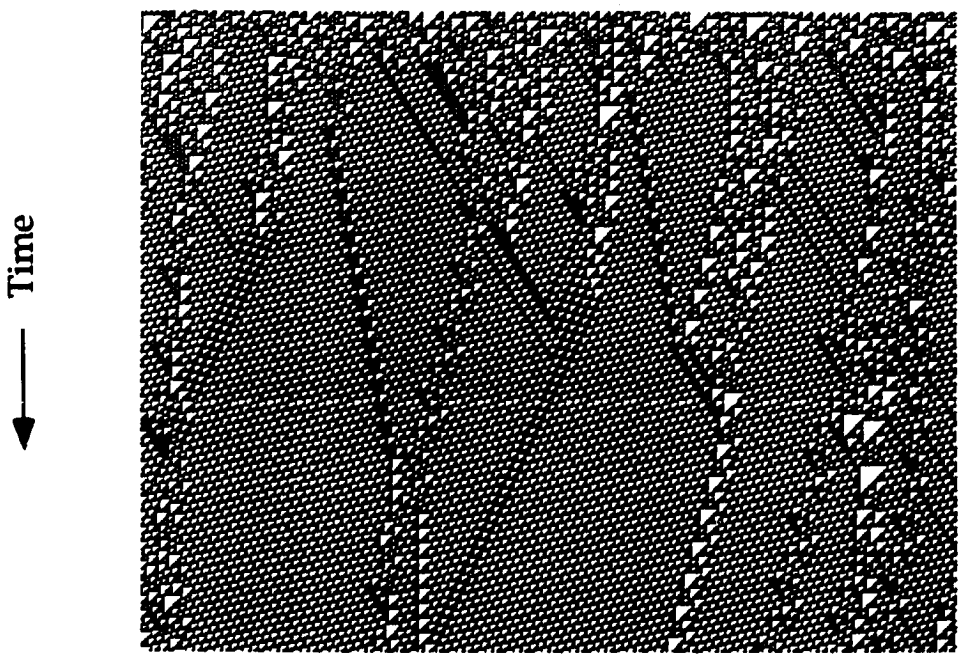


Figure 4.2. (a) σ_Γ and (b) σ_I vs. N for elementary CA rules 22 and 110.



(a)



(b)

Figure 4.3. Evolution of elementary CA rules (a) 22 and (b) 110

size. If these regions possess certain computational abilities, then multiple regions can interact to achieve greater computational power, and hence complexity, on a larger scale. (Consider the universal Turing machine (cf. sec. 2.4), which has a repetitive state-space structure and is composed of relatively basic computing elements, but is capable of universal computation.) σ_I may detect this interaction when it continues to increase with N beyond $N = \Lambda_\Gamma$.

Thus, rule 110 should be considered complex, and it is in the literature (cf. sec. 4.3). Rule 22 is also considered complex, but the two rules are traditionally difficult to compare because their behavior is qualitatively quite different. This is evident in figures 4.3a and b, which show the evolution of rules 22 and 110 from random initial states on the top rows. The arrays have 500 cells with periodic boundary conditions. Dark cells represent state 1. Information variables provide a direct quantitative means of comparison as in figure 4.2. Rule 22 has larger values of σ_Γ and Λ_Γ than rule 110, but σ_I of the latter overtakes that of the former. For rule 110, the single time-step fluctuations in Γ add constructively to produce significantly larger fluctuations in $\Sigma\Gamma$ (or I) for large region sizes. This can be seen in figure 4.4, which is a graph of Γ and $\Sigma\Gamma$ vs. time for a randomly stimulated 18 cell region (cf. sec. 3.3). Rule 110 seems to have the interactive property discussed above to such a degree that perhaps it is more complex than rule 22 on a global scale.

One obvious mechanism for interaction is propagating structures. In figure 4.3, rule 110 has several propagating structures (gliders) while rule 22 has none that can easily be recognized. In chapter 5, large, long-term fluctuations in $\Sigma\Gamma$ time series, such as the one that dominates figure 4.4, are identified with memory-loops in state space. These in turn are shown to be essential for the existence of slow-moving gliders.

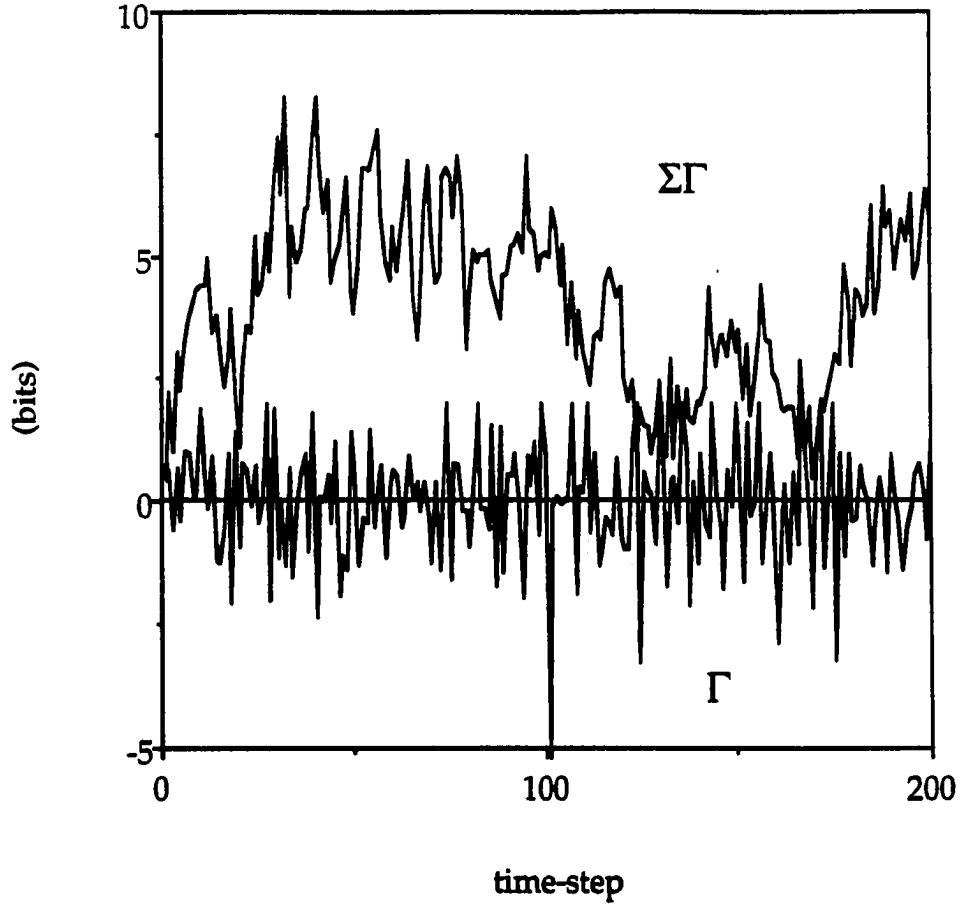


Figure 4.4. Γ and ΣΓ vs. time for a randomly stimulated 18 cell region, elementary CA rule 110.

4.3. Further comments on elementary CA rules 22 and 110

Rule 22 has generally been recognized to be among the most complex of the elementary CA rules and has been studied in detail by several researchers. Grassberger (cf. sec. 2.7.2) has performed extensive simulations of many rules, including 22. By studying the decay of the spatial and temporal block entropies, he has argued that rule 22 has very long and complex 'hidden' correlations [Grassberger-2, Grassberger-3]. In a recent work, Eisele has presented another analysis of rule 22 [Eisele]. Approximating the stationary state by a Markov chain which describes the set of possible spatial sequences at a fixed time, he also finds evidence for long-range spatial correlations. He notes that "The slow decay of spatial correlations at long distances is caused by far-reaching fluctuations in the local irreversibility." Fluctuation in irreversibility is closely related to fluctation in net information gain. Converging links in an NFA graph indicate irreversibility, and σ_Γ includes their variability. However, the NFA model presented in this dissertation is different from Eisele's and Grassberger's, my emphasis being on temporal behavior, with spatial dependence determined by varying the size of the cell region.

Rule 110 has also been given special attention by several researchers. The multiple propagating structures supported by this rule make it an obvious candidate for Wolfram's class IV. Rule 54 is the only other elementary rule with class IV-like behavior [Li-1]. Lindgren found that rule 110 exhibits an increase in spatial correlation information while evolving from a random initial pattern, corresponding to the appearance of the background pattern in figure 4.3b [Lindgren-3]. Li, Packard and Langton classified rule 110 as complex based on poor convergence of statistical properties, such as difference pattern spreading rate, with number of samples [Li-2].

4.4. Comparison of the elementary CA for large region size

There are 256 elementary rules, but accounting for rules that are related by reflection or complementation ($0 \leftrightarrow 1$) leaves only 88 distinct rules. Table 4.1 lists the values of $< I > / N$, σ_Γ and σ_I for region size $N = 19$, for all 88 rules, sorted by σ_I. When there is more than one attractor, the largest values of σ_Γ and σ_I and the corresponding (largest) value of $< I > / N$ are listed. The glider/object column is discussed in section 5.5.

Table 4.1
Elementary CA information variables for $N = 19$

Rules	$< I > / N$	σ_Γ (bits)	σ_I (bits)	Glider/object
7	0.45	1.31	3.09	S
62	0.48	0.85	2.84	ST
35	0.76	1.18	2.54	SF
54	0.68	1.24	2.49	FFT
37	0.75	1.59	2.33	
14	0.70	1.22	2.21	FF
3	0.74	0.95	2.14	S
25	0.72	1.32	2.12	SSF
9	0.61	1.43	1.90	SFT
22	0.86	1.94	1.86	
110	0.87	1.23	1.85	SSSSST
74	0.72	1.16	1.72	
146	0.74	1.39	1.65	
41	0.82	1.48	1.63	
18	0.75	1.47	1.63	
134	0.68	1.18	1.61	
6	0.66	1.27	1.59	
46	0.56	0.85	1.55	
2, 130	0.52	1.03	1.53	
24, 152	0.53	0.85	1.51	
184	0.80	1.20	1.51	FF
122	0.79	1.25	1.49	
43, 142	0.82	1.19	1.48	FF
26	0.84	0.95	1.47	
126	0.74	1.33	1.42	
56	0.72	1.16	1.41	FF
58	0.48	1.10	1.37	
11	0.72	0.94	1.28	

Table 4.1 (continued)

Rules	$<I>/N$	σ_Γ (bits)	σ_I (bits)	Glider/object
10	0.69	1.00	1.26	
38	0.77	1.03	1.18	
34, 162	0.68	0.67	1.17	
42	0.87	0.62	1.08	
57	0.55	1.26	1.07	FF
27	0.83	0.89	1.03	S
1, 5	0.11	2.00	1.00	
138	0.81	0.66	0.91	
33	0.18	1.49	0.82	
29	0.15	1.15	0.67	
28	0.15	0.82	0.47	
44	0.05	0.82	0.47	
15, 30, 45, 60, 90, 105	1	0	0	
106, 150, 154, 170	1	0	0	
73	0.36	0	0	
94	0.19	0	0	
108, 156	0.16	0	0	
13, 19, 23, 50, 51	0.05	0	0	
78, 172, 178	0.05	0	0	
0, 4, 8, 12, 32, 36, 40	0	0	0	
72, 76, 77, 104, 128	0	0	0	
132, 136, 140, 160	0	0	0	
164, 168, 200, 204, 232	0	0	0	

Ideally, similar tables for many values of N would be given. Both σ_I and its dependence on N should be considered when judging the complexity of rules. N = 19 was the largest region size achievable with the computational resources available. If only one value of N is considered, the choice of a large value allows for rules with small initial values of σ_I but large rates of increase, such as rule 110, to be near the top of a list sorted by σ_I. Therefore, the order of table 4.1 is intended only as an approximation of the relative complexity of the elementary CA. Rule 110 is expected to move further up the list from its eleventh position as N increases beyond 19, since it has the greatest rate of increase in σ_I at N = 19.

Among the rules with $\sigma_I = 0$, rules with $< I > = 0$ have single-state attractors while rules with $< I > / N = 1$ have full and equal state utilization (all $n = 2^N$ states in the NFA are equally probable). The zero entropy rules are ordered ($< G > = 0$) while the maximum entropy rules are either chaotic or periodic. Purely periodic ($< G > = 0$, $< I > = \log(n)$) or purely chaotic ($< G > = < I > = \log(n)$) behavior is possible in general but the constraints mentioned in section 4.2 prohibit these extremes for large N.

The elementary CA with $< I > / N = 1$ fall into three classes. The simple *shift* rules, 15, 60 and 170, have $< G > = < L > = 1$ bit. With each time step, they gain a bit of information as a bit shifts into one end of a region (of any size) while simultaneously losing a bit out the other end. The *exclusive-or* rules, 90, 105 and 150, have $< G > = 2$ bits. These rules can shift information unhindered in both directions simultaneously. Finally there are the *mixing* rules, 30, 45, 106 and 154, with $< G > = 1.5$ bits. The average value of 1.5 bits is due to an equal mixture of $G_{ij} = 1$ bit and $G_{ij} = 2$ bits. Thus, exclusive-or rules are the most chaotic while shift rules are the most periodic, but for large region sizes, all three classes are closer to the periodic than the chaotic extreme.

Mixing rules are unique in their ability to produce random behavior from ordered initial conditions. They are analogous to the logistic map with r = 4 (cf. chapter 6). Both produce random behavior starting from most initial conditions, including ordered, although both can produce periodic behavior starting from certain rare initial conditions. Shift rules are analogous to the shift map (cf. chapter 6). Both can behave randomly, but the randomness or lack of it is entirely a reflection of the initial condition, which is most probably random (cf. sec. 2.6). Figure 4.5 shows the evolution of rule 30 from an initially ordered state on the top row. Vertical sequences of cells have been shown to be high quality pseudo-random sequences [Wolfram-4].

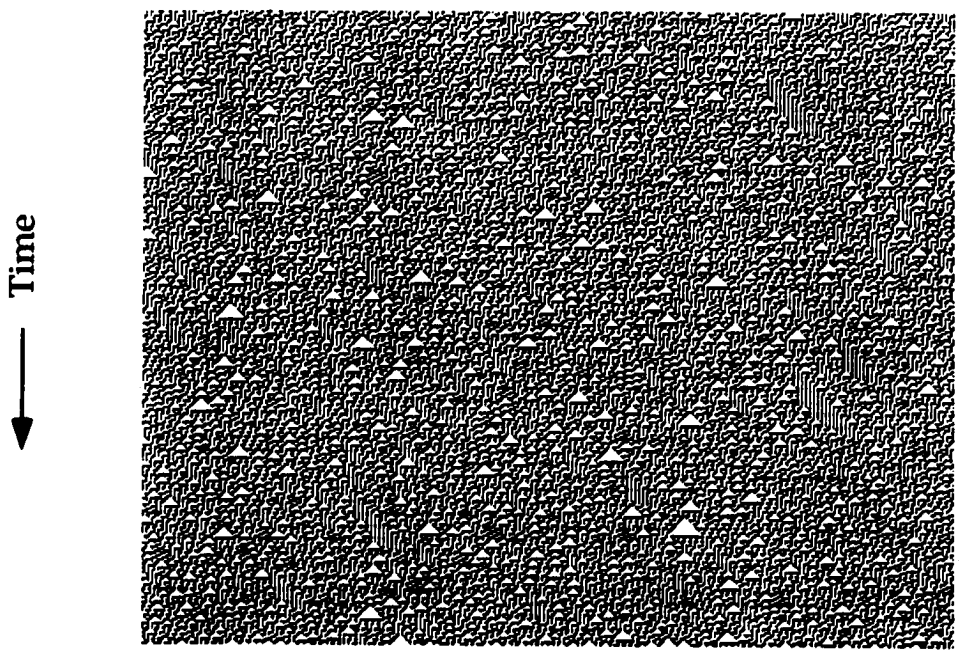


Figure 4.5. Evolution of elementary CA rule 30

Other, more complex rules can have a mixing quality. Rule 22 has $< G >$ = 1.62 bits for N = 19. ($< G >$ changes only slightly for N > 3.) It visually resembles rule 30 because of its significant disordered aspect (cf. fig. 4.3a), but it also has an ordered aspect. In contrast, rule 110 has $< G >$ = 1.11 bits for N = 19. After an initial transient phase it develops a periodic background (spatial period 14, temporal period 7) (cf. fig. 4.3b). There may be a correlation between half-integer values of $< G >$ and the phenomenon of mixing.

4.5. Elementary CA equivalence classes

Table 4.1 indicates that some of the 88 distinct elementary CA rules are indistinguishable from an information variable standpoint. I define an equivalence class as a group of rules that have the same values of $< I >$ and σ_I for a given region size.

When N = 1 all 88 rules merge into just eight classes of two-state NFA. As N increases, new classes form. Certain rules remain in the same class for all values of N. Some split away from the original eight to form new classes either singly or with others. When N = 2 there are 28 equivalence classes. When N = 3 there are 42, but for the first time some rules have non-identical multiple attractors. Due to their multiple attractors, rules 13, 28 and 78 can be found in more than one class. They are assigned their own classes, making a total of 45 classes for N = 3. For N = 4 six more rules develop non-identical multiple attractors: rules 5, 29, 44, 108, 156 and 172. One of these already constitutes a separate class, so five new classes are added. However, rule 73 which had constituted a separate class, merges with another class, so the new total is 49 classes. For N = 5 rules 1 and 94 develop non-identical multiple attractors. Rule 1 already constitutes a distinct class so the number of classes reaches 50. Finally, for N = 6, rule 73 develops non-identical multiple attractors bringing the total number of equivalence classes to 51.

As N continues to increase, intermediate entropy rules tend to utilize a successively smaller fraction of the total available states. This tendency might encourage the development of multiple attractors. For example, this occurs for rule 33 when N = 7. However, all intermediate entropy rules (or rule-pairs) already constitute their own equivalence class. Thus, we expect that no new equivalence classes will form as N increases. 51 is evidently the maximum number of equivalence classes for the elementary CA.

Figure 4.6 is a plot of σ_Γ vs. $< I > / N$ for N = 19. There are five broad classes of rules relative to σ_Γ and $< I >$. The two least complex are the extremely ordered rules such as rule 0, with $< I > = 0$, and the rules such as 15, 30 and 90, with $< I > / N = 1$. These two classes are opposite in terms of state utilization but identical in their simplicity with $\sigma_\Gamma = 0$. They are completely established by N = 3 and remain unchanged for larger N.

Then there is the class of rules such as rule 19 for which $< I > = 1$ and $\sigma_\Gamma = 0$, established by N = 3. These show simple period-two behavior.

Next is the group of low entropy rules such as rules 1 and 5 with non-identical multiple attractors. These rules can have wide ranges of values of σ_Γ. This class begins forming at N = 3 and is nearly complete by N = 6 but may add to its number as N increases.

The fifth class consists of rules such as rules 22 and 110 with intermediate entropy and moderate to high complexity. Virtually all have a single attractor through N = 19. There is a noticeable gap in entropy between this class and the prior one, and between this one and the chaotic class. The former gap decreases while the latter increases with N.

Although rules 1 and 5 have the largest values of σ_Γ, their entropies and characteristic lengths are much smaller than that of the next one, rule 22, which is often cited as one of the most complex elementary CA (cf. sec 4.3).

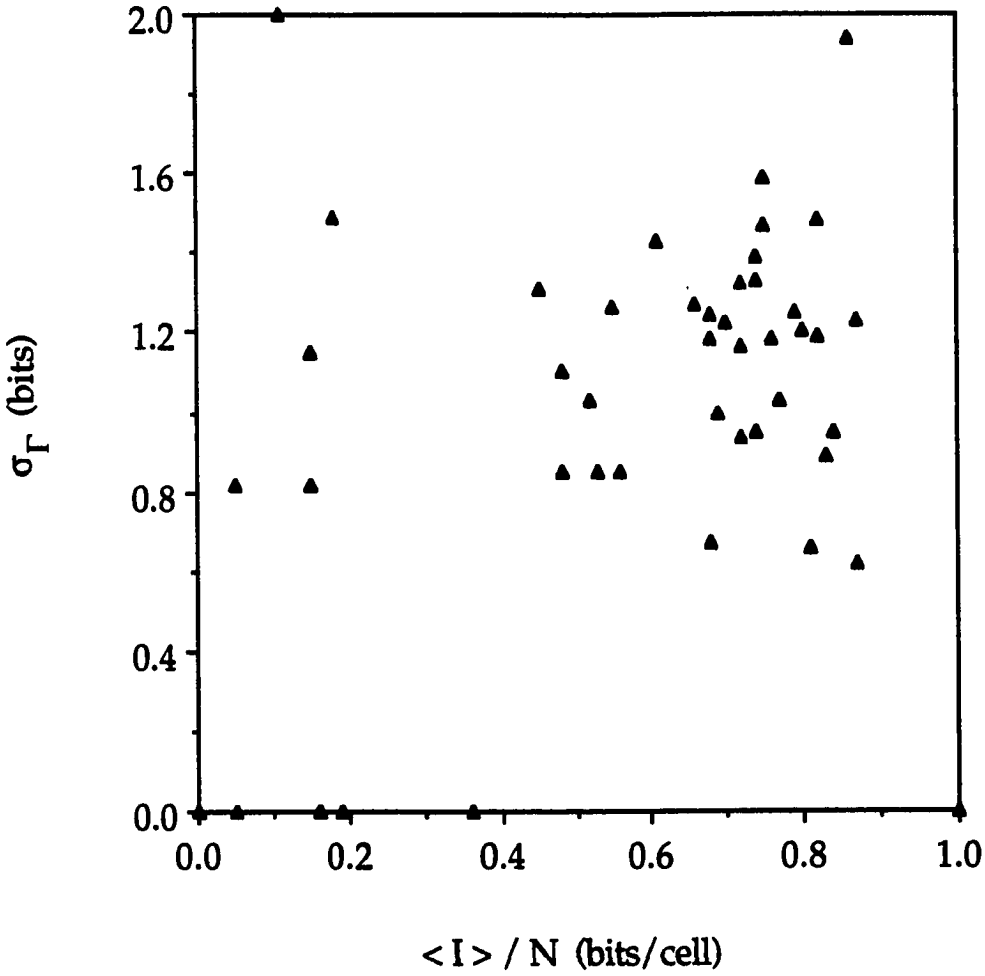


Figure 4.6. σ_Γ vs. $< I > / N$ for the elementary CA rules with N = 19.

However, nine rules with lower values of σ_Γ have larger values of σ_I, and therefore can store, on average, greater amounts of information for longer periods in a 19 cell region.

4.6. A statistical study of the trinary CA

There are more than 7×10^{12} trinary (k = 3 states per cell, rule-range r = 1) CA, too many to study exhaustively. A statistical study based on a random sample of one million trinary CA was conducted. The maximum values of σ_I and σ_Γ were computed for each rule in the sample and truncated to the tenths decimal place. Relative frequencies are plotted for $1 \leq N \leq 5$ in figure 4.7. Logarithms are evaluated in base 3 so the units of σ_I and σ_Γ are trits. It should be noted that the distributions are actually highly irregular; only the relatively coarse discretization (0.1 trits) gives the appearance of smoothness.

There are some similarities in the distributions of the two variables. In both cases the most probable value of σ increases with N, but the rate of this increase appears to decrease with N. Both distributions have a significant number of rules with $\sigma \approx 0$ for all values of N, but a region of scarcity forms for $N > 2$ as the main part of the distribution moves away from zero.

The distributions are noticeably different in the dependence of their height and width on N. The σ_I distribution broadens as N increases resulting in an extended exponential tail. In contrast, the σ_Γ distribution narrows as N increases. There is a smaller deviation about the mean and a vanishing exponential tail.

The extended exponential tail of the σ_I distribution is an important feature. It shows that σ_I can be highly selective. It would have to be if it is to select the most complex rules (class IV), since they are believed to be rare (cf. sec. 2.3). The maximum observed (truncated) values of σ_I as N increases from 1 to 5 are 0.7, 1.0, 1.4, 1.7 and 2.1 trits. This steady rate of increase is significantly

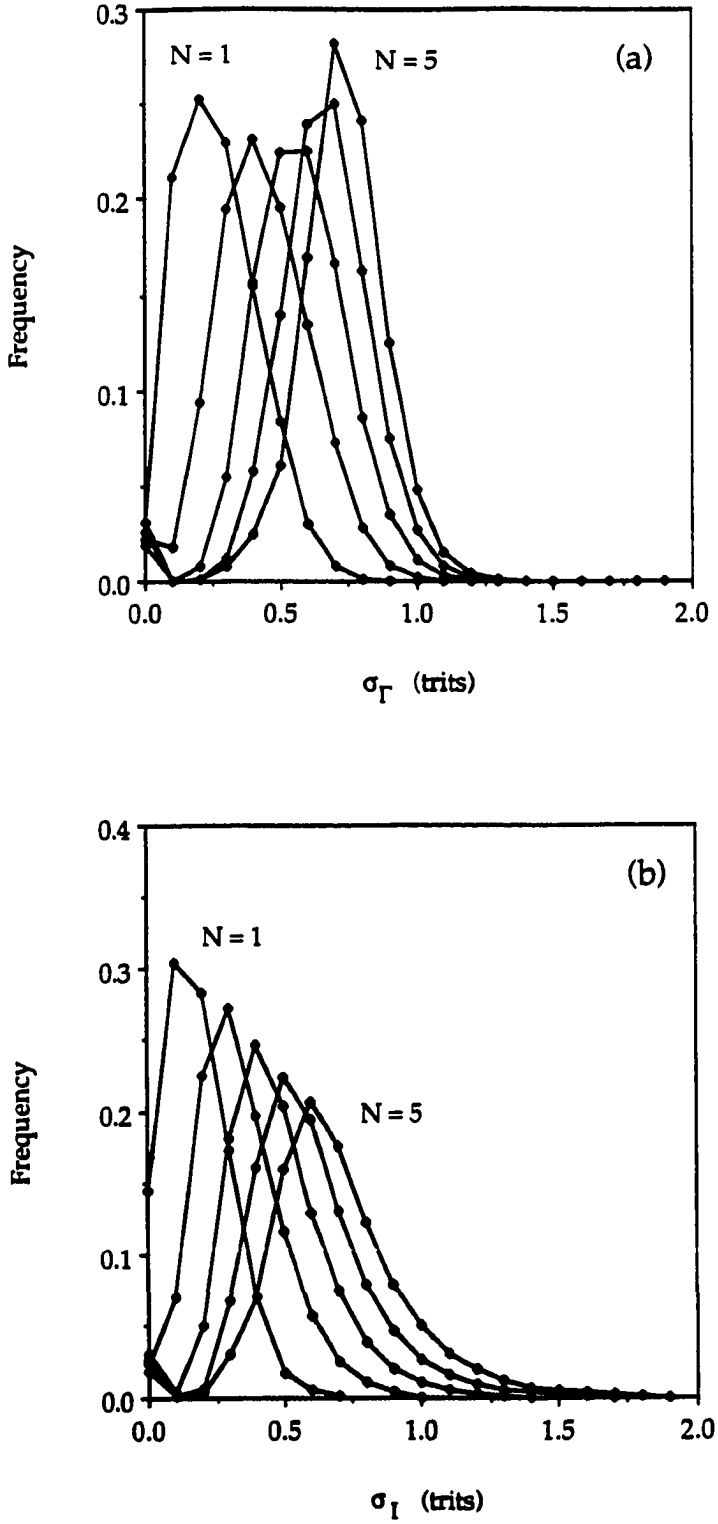


Figure 4.7. Relative frequencies of trinary CA rules relative to (a) σ_Γ and (b) σ_I for $1 \leq N \leq 5$.

greater than that of most rules. The rare rules with this behavior have a large capacity for cumulative information storage and presumably are capable of high levels of computation. σ_I might be practically useful in searching vast rule spaces for rare complex rules.

Figure 4.8 shows the correlation between the two distributions for N = 5. The vertical axis is the logarithm of the number of rules (plus one) with a given (σ_I, σ_Γ) pair. Note the small 'foothills' to the right of the figure. These rare rules have values of σ_I well above typical values, but only have moderately large values of σ_Γ, much like elementary rule 110. Curiously, the largest values of σ_I and σ_Γ are mutually exclusive, while the smallest values are roughly correlated.

Figure 4.9 shows the correlation between σ_I and $< I >$ for N = 5. The vertical axis is the logarithm of the number of trinary CA rules (plus one) with a given (σ_I, $< I >$) pair. It resembles figure 3.7, σ_Γ vs. $< I >$ for two-state NFA. σ_I is maximum at an intermediate value of entropy. An interesting feature is the dearth of rules with high (but not maximal) entropy and very low values of σ_I.

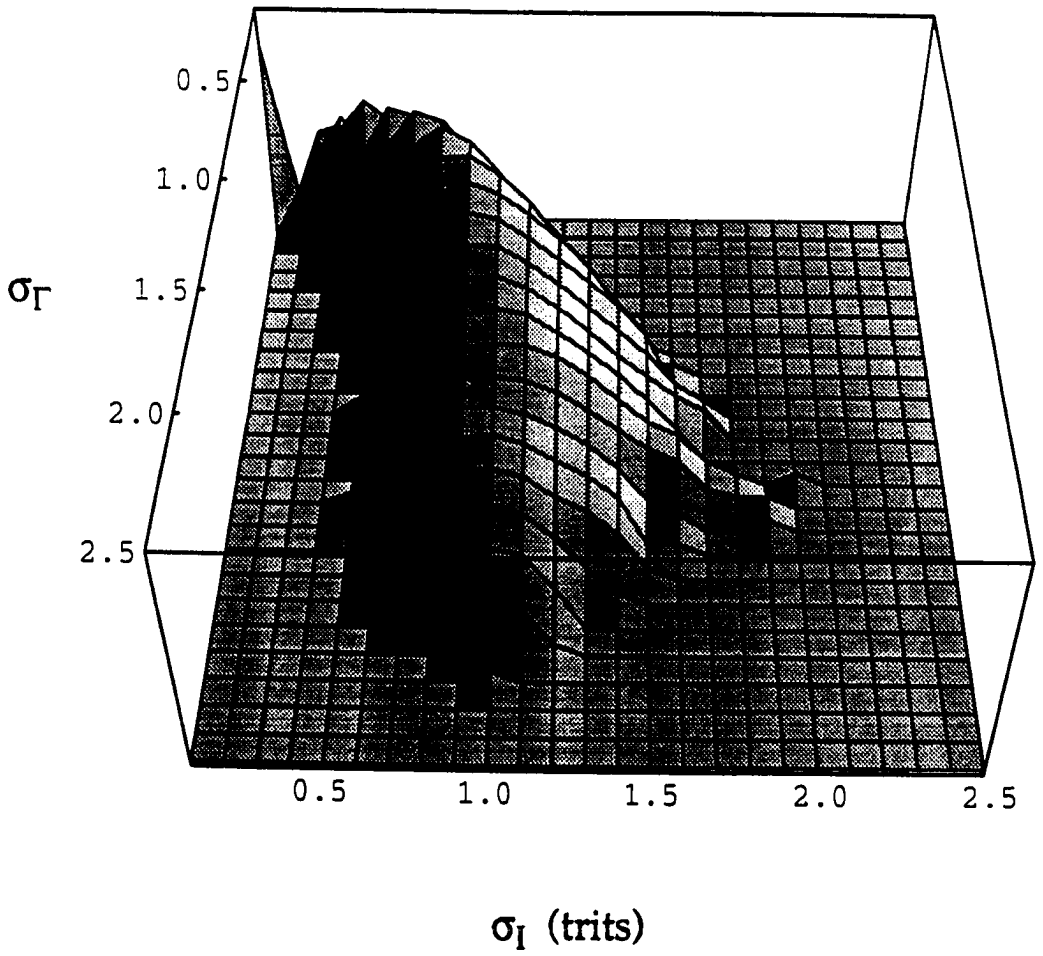


Figure 4.8. Correlation between σ_I and σ_Γ for N = 5.

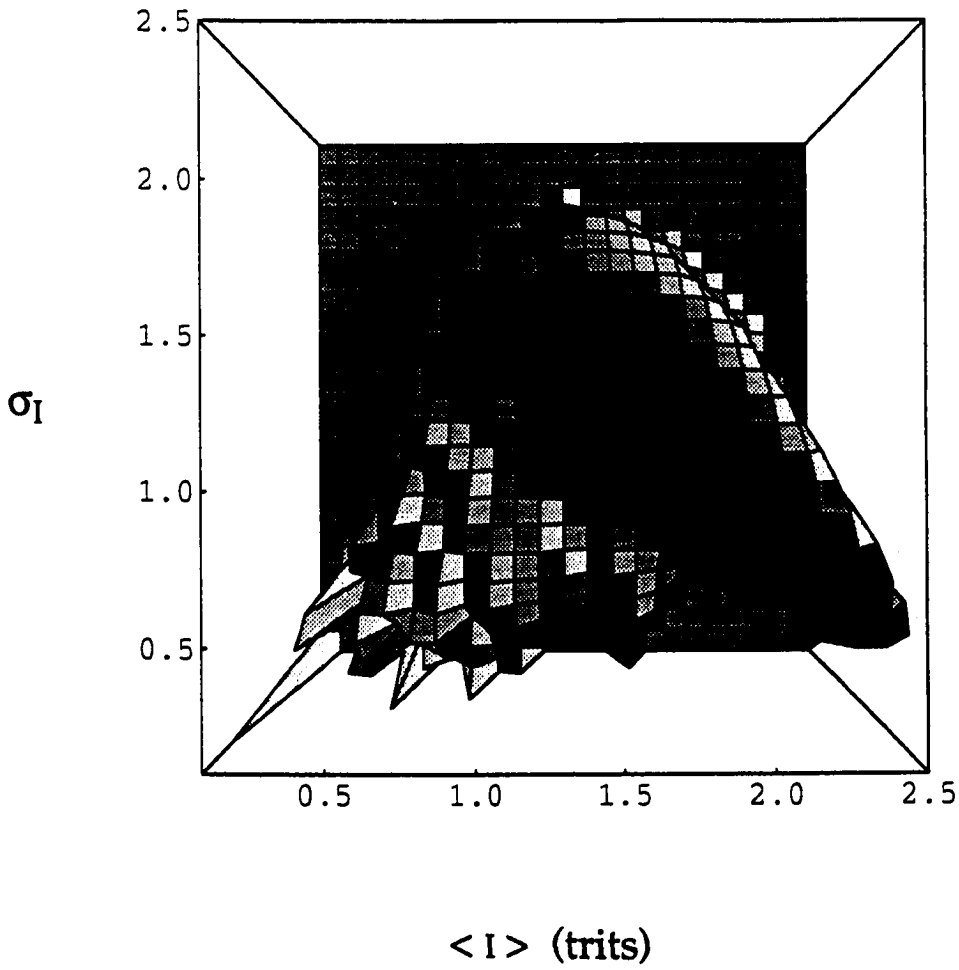


Figure 4.9. Correlation between σ_I and < I > for N = 5.

5. Application to Cellular Automata – Gliders

One mechanism that might enhance inter-regional interaction in CA (presumably permitting higher levels of computation and greater complexity) is the existence of propagating structures or *gliders*[Berlekamp], especially several types of gliders and other objects that can interact. Gliders owe their existence to the underlying dynamics of CA rules and can be viewed as complex objects in their own right [Langton-1, Langton-2]. However, they can also function as elements of computing machines embedded in a CA universe [Lindgren-2]. Thus, gliders have both dynamical and computational aspects. Successful application of the information variable method to them would support its usefulness as a general technique capable of connecting these two aspects.

5.1. Glider definition

For the purposes of this dissertation, a glider is defined as an object that: resides in a one-dimensional CA array, moves through a well-defined, consistent background, has a fixed period of pattern repetition, T, and a constant average speed, v. The background can be periodic or quiescent (spatial and temporal period one). A glider may oscillate in size as it moves, but must maintain its identity in the absence of collisions with other objects. (Its pattern is invariant in spacetime under a time displacement T combined with a space displacement v T in the direction of motion.) λ is the maximum glider size and assumes that it is not a union of smaller gliders. This definition and the following discussion can easily be extended to multi-dimensional gliders that move in a fixed direction and have some constant maximum size in directions transverse to their motion.

5.2. Glider variable relationships

Maximum glider speed is v = r (rule range) cells per time-step. Gliders moving at this speed will be referred to as *fast gliders. Slow gliders* have speeds in the range $0 < v < r$. *Stationary objects* have all the properties of gliders, except that $v = 0$. Fast gliders are trivial: they are supported by simple shift rules that have no memory capacity (in the sense of net information gain). However, if a slow-moving glider passes through a region of cells, those cells must be able to remember that the glider is present for several time-steps. The information pertaining to the glider cannot merely be shifted along as soon as it arrives. While the glider is lingering in the region, the memory of it is retained in a sequence of (mostly) distinct region-states. These states form a *glider memory-loop* in state space, beginning and ending with the background state(s). The slower the glider, the longer the memory-loop. However, a single cell has only k states. If more are needed to form the memory-loop, then several adjacent cells must cooperate to form a collective memory. The glider must necessarily cover a region of several adjacent cells.

An approximate inequality between glider speed and size can be derived. Consider a glider traversing a fixed region of cells of size λ that initially contains the background pattern. The time required for a complete traverse is at most $2\lambda / v$. During this time the region sequences through a succession of region-states equal in number to $2\lambda / v$. Now, this sequence may be non-repeating, each state in the sequence being unique. However, there are only k^{λ} region-states available. Therefore

$$2\lambda / v \leq k^{\lambda} \qquad (5.1)$$

or

$$v \geq 2\lambda / k^{\lambda}. \qquad (5.2)$$

The inequality becomes stronger when many other distinct objects (in addition to the background) are supported by the region. In this case fewer region-states are available for the exclusive use of the glider and ≥ should be replaced by >> in equation (5.2).

The inequality is weakened if a glider uses region-states more than once. In general, repetition of region-states is unlikely because when a region-state recurs with the same boundary conditions, the region is forced to repeat a previous pattern, and the glider is unable to advance. However, repetition can occur, for example, when the leading edge of the glider enters the region but then temporarily retreats. If the background is periodic, repeated use of region-states is more likely, since the boundary conditions may change and allow a different succession of states to follow the repeated state. We can modify equation (5.2) to include the effect of a periodic background:

$$v \geq 2\lambda / T_b k^{\lambda}, \tag{5.3}$$

where T_b is the temporal background period.

Equation (5.3) gives a minimum λ for a given v and is based on the simple necessity of having enough states available. The maximum λ (or minimum v) depends on how many of the available states can be connected appropriately to form a glider memory-loop. This is a difficult theoretical question. The computer experiment discussed in section 5.5 provides some empirical clues to the answer for a limited case.

As a simple application of equation (5.3), consider the elementary CA (k = 2, r = 1). For these CA

$$v \geq 2\lambda / T_b 2^{\lambda}. \tag{5.4}$$

Small gliders ($\lambda \leq 2$) must move at the 'speed of light' v = r = 1 when the background is quiescent ($T_b = 1$). An elementary CA rule that supports fast gliders can have no stationary objects since it must be a simple shift rule. No

computation occurs within these gliders and no computing machine can be constructed from them. Information transmission is present, but information storage and modification are absent. One might as well try to build a computer with only wires. Slow gliders are supported by some elementary rules, but in every observed case $T_b > 1$ (cf. sec. 5.5).

Next, consider the trinary CA (k = 3, r = 1) rule specified by $000000210011101012000201020_3$ in the Wolfram convention, base 3. This rule supports a glider in a quiescent background with $\lambda = 5$ and $v = 1/6$. The glider is shown in spacetime in figure 5.1. Its speed is well above the theoretical minimum speed of 10/243 according to equation (5.3). Its actual speed indicates that it would use $2\lambda / v = 60$ region-states out of $3^\lambda = 243$ available in a five-cell region if it was fixed in size and used no region-state more than once. However, its size does oscillate and some states do repeat. 55 time-steps are needed to completely traverse the region, but only 36 distinct states, including the background state, are actually used. Thus, about 15% of the available region-states are used.

5.3. Glider memory-loops

In this section the glider memory-loops introduced in section 5.2 are examined in detail at the levels of dynamics, information flow, and computation, emphasizing the unifying role of information.

A memory-loop is a particular closed path through the NFA of a region of cells. Consider the idealized glider memory-loop shown in figure 5.2. The sequence of states in the loop are those that occur when a glider enters, traverses and then exits the region. The remaining region-states that are part of the entire NFA but not of the memory-loop are not shown. The loop begins and ends with a quiescent background state.

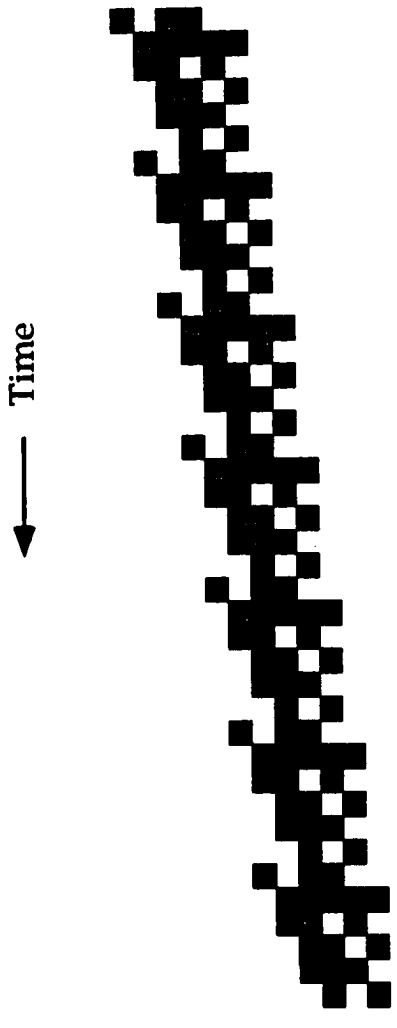


Figure 5.1. Spacetime trajectory of a v = 1/6 trinary CA glider.

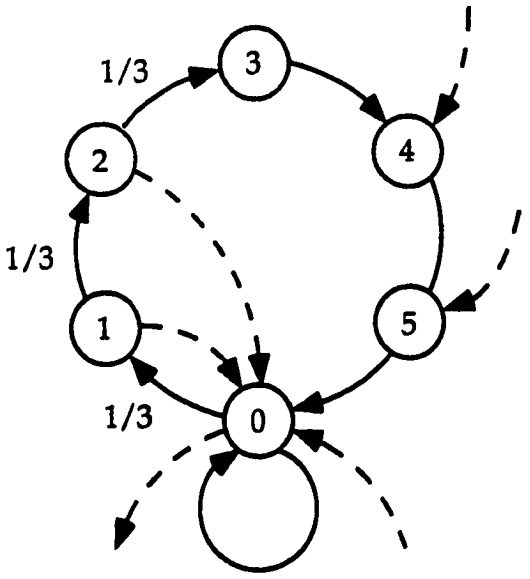


Figure 5.2. Idealized glider memory-loop.

The loop has two primary phases: *acceptance* and *restoration*. The acceptance phase is characterized by the chaotic influence of diverging links and the consequent sensitivity to external information. However, the glider is accepted only if a certain sequence of driver-cell values occurs. Thus, the region is only *selectively* sensitive to *sequences* of external information. Any deviation from the acceptable sequence results in a prompt return to the attracting background state as indicated by the dashed lines in figure 5.2, where order is restored by the converging links. The acceptable sequence triggers a self-sustaining process that results in the glider traversing and then exiting the region. In the orderly restoration phase, converging links dominate, ensuring that no debris are left trailing behind the glider. If the region is longer than the glider, a third, intermediate phase (not shown in fig. 5.2) occurs while the glider propagates across the interior of the region.

The dynamics of the glider memory-loop – its topological structure and the roles of chaos and order – have been described in a qualitative manner. The description is quantified by introducing information variables. For illustrative purposes, suppose that $p_{i\rightarrow j} = 1/3$ and $p_{i\leftarrow j} = 1$ during the acceptance phase, and that $p_{i\rightarrow j} = 1$ and $p_{i\leftarrow j} = 1/3$ during the restoration phase. Figure 5.3 shows the values of Γ and $\Sigma\Gamma$ vs. time as the glider passes through the region. Γ is positive in the chaotic accepting phase of the loop, indicating that the region stores external information every time-step. There is an accumulating net information gain along the path; $\Sigma\Gamma$ increases steadily, indicating a trend toward rarer states. These states are rare because specific, relatively unlikely driver cell sequences are needed to achieve them. The loop is effectively a multi-step memory of a sequence of external information. Γ is negative in the orderly restoration phase as the region 'cleans up' behind the glider. $\Sigma\Gamma$ returns to zero as the glider exits the region and the memory of it is erased.

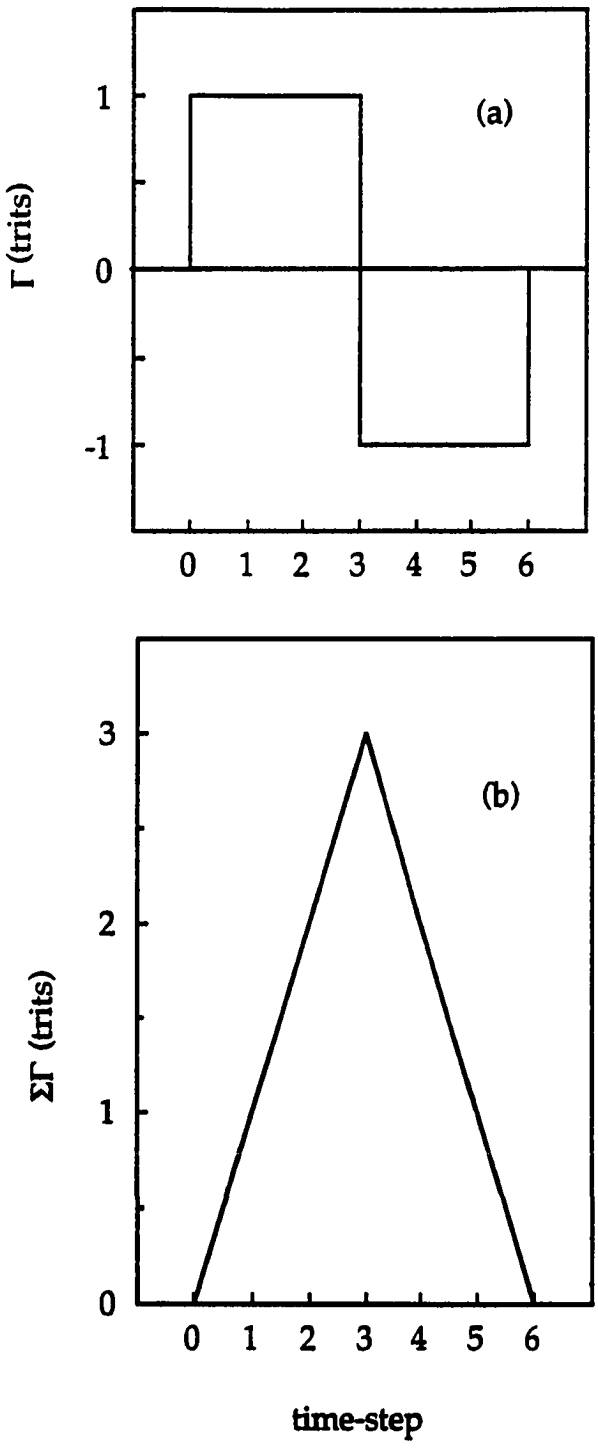


Figure 5.3. (a) Γ and (b) ΣΓ vs. time for the idealized glider memory-loop of figure 5.2.

The memory-loop also can be described in computational terms (cf. sec. 2.4). Actually, use has already been made of such terms: memory, NFA, and acceptance, not to mention information. The region can be viewed as a computing machine. It can perform all three basic computational functions: information transmission, storage and occasional modification. On a higher level the acceptance function is exhibited. If the region supports multiple gliders, each with their own memory-loop, then it could be said to accept a language, namely *glider language,* the set of all driver cell sequences that result in the creation of a glider. The acceptance function depends on the ability of the region to store sequences of information.

In terms of information, a glider memory-loop is characterized by a single major fluctuation in $\Sigma\Gamma$. The duration of the fluctuation relative to the size of the region reflects the glider's (lack of) speed. The maximum value of $\Sigma\Gamma$ is a measure of the memory capacity of the loop and hence the cumulative amount of information needed for the glider to gain acceptance to the region. Meanwhile the discrete derivative of $\Sigma\Gamma$, Γ, fluctuates twice, first positively then negatively. The magnitude of the positive fluctuation in Γ is limited by the maximum amount of information that can cross the region's borders, which is the product of number of driver cells and log(k). The magnitude of $\Sigma\Gamma$ is ultimately limited by the number of region-states that form the chaotic accepting portion of the memory-loop. The magnitude of the negative fluctuation in Γ is limited by the amount of information that can be lost in one-time step, which is greater than the limit on information gain by a factor that can approach the number of region-states. Thus, the restoration phase can be much shorter than the acceptance phase.

Fluctuations in Γ indicate low-level computational activity. To achieve the higher-level acceptance function that glider formation depends upon, the

fluctuations must be of a certain type. Using the language of discrete time-series analysis we could say that the discrete Fourier transform of $\Sigma\Gamma$ must have strong low-frequency components. For example, if Γ changes sign every time-step with no magnitude change, it has no low-frequency components, net information gain does not accumulate, $\Sigma\Gamma$ is zero and sequences of information cannot be stored. However, it seems more direct to say that fluctuations in $\Sigma\Gamma$ (or I) are necessary to achieve the high-level computational activity of acceptance.

The idealized glider memory-loop has been examined at the three levels of dynamics, information flow and computation. Consider again the $k = 3$, $r = 1$ CA rule discussed in section 5.2. In figure 5.4a $\Sigma\Gamma$ is plotted vs. time as the $\lambda = 5$, $v = 1/6$ glider traverses a five-cell region. Six consecutive memory-loops occur. The first two are 'sorties' where the glider penetrates the region only to retreat. These memory-loops are non-accepting and are a result of oscillations in the glider's size. The third loop is accepting. The final three identical memory-loops are due to unacceptable penetrations back into the region after the glider has exited. The accepting glider memory-loop corresponds to the single major fluctuation in $\Sigma\Gamma$ (or I). It differs from the idealized glider memory-loop in its irregularity and asymmetry.

5.4. Conditions for glider memory-loop existence

Describing information flow in a glider memory-loop is one thing, but how does such a loop arise from individual cells obeying a rule? Consider the information flow in a single-cell region as the glider passes through it, as shown by the graph of $\Sigma\Gamma$ in figure 5.4b. The two non-background states 1 and 2 have storage capacities of 0.70 and 0.87 trits respectively. None of the nine memory-loops are accepting since the region is smaller than the glider. At any given point in time, one can compare the information stored in a cell to that

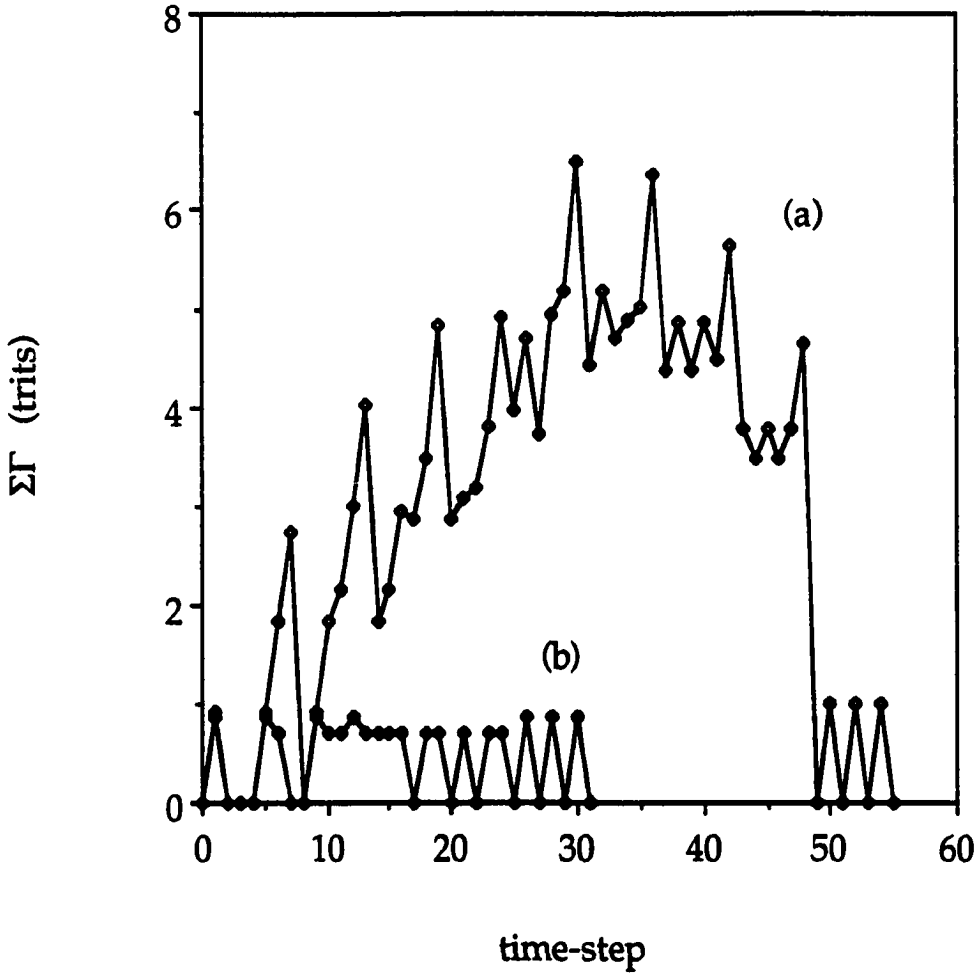


Figure 5.4. ΣΓ vs. time for the v = 1/6 trinary CA glider of figure 5.1 moving through (a) a five-cell region and (b) a one-cell region.

stored in its neighbors by shifting left or right on the graph multiples of six time-steps. One soon discovers that the combined information in all single-cells involved with the glider is not a conserved quantity. Single-cell region state probabilities do not account for information stored in unlikely *sequences* of adjacent cell-states. However, at least in qualitative terms one can see that there are up to five adjacent cells interacting in a complex way. Each cell has its own three cell-states capable of forming memory-loops of variable length. Of the nine memory-loops actually used by the glider, five are distinct, with three different lengths (2, 3 and 9 time-steps). The loops are out of phase and have a changing phase relationship. It is the changing phase that allows larger enveloping regions to cycle through long sequences of distinct states. The changing phase is due to the presence of diverging links that allow for flexibility in the sequence of cell-states. This chaotic influence must be balanced by order, or else unconstrained activity could result.

The basic three-state NFA of a single-cell region must have a variety of converging and diverging links, reflecting a mixture of order and chaos. This is a prerequisite for inter-cell cooperation, but the extent of such cooperation over large regions of cells is difficult (if not impossible in principle – perhaps analogous to the halting problem (cf. sec. 2.4)) to predict by examination of such a small region. Chaos allows for communication between cells over a distance, but too much chaotic influence would prevent any structures from forming and persisting in time and no organized activity requiring a specific sequence of events could occur.

There are still a number of questions to be answered before a complete understanding of inter-cell cooperation is achieved. At this stage a proposed list of requirements for glider existence is ventured.

1. $\sigma_\Gamma > 0$. There must be single time-step fluctuations in net information gain in the smallest region: the essential ingredient for low-level complex dynamical behavior.
2. $\Lambda_\Gamma > 0$. There must be an increase in these fluctuations as region size increases: evidence of inter-cell cooperation to form new structures in state space.
3. $\sigma_I > 0$. Fluctuations in net information gain must constructively add along some multiple-time-step paths: evidence that the structures formed by inter-cell cooperation are memory-loops.

The third requirement is the most important. It is contingent on the first two, but as discussed in chapter 4, σ_Γ and Λ_Γ need not be extremely large to achieve a large value of σ_I.

Inter-cell cooperation and the existence of memory-loops do not ensure glider existence. At least one of the memory-loops must be a glider memory-loop. The three requirements are statistical in nature since they are obtained by randomly stimulating regions of cells. However, a glider is a specific object and requires a specific sequence of events to occur. An NFA may have many memory-loops but no gliders. Some other requirement (as yet unknown) must be met.

5.5. Correlation between glider existence and σ_I

Nevertheless, since σ_I must be non-zero if any slow gliders exist, it is reasonable to expect that rules with large values of σ_I are more likely to support them. To test this expectation, computer experiments were performed in which trinary CA rules were classified according to whether they supported slow gliders, among other things. Classification was done visually as an array of 500 cells with random initial cell-states and periodic boundary conditions evolved in time on a video screen. Obviously, this type of experiment is subjective and

the results should be considered as preliminary indications of the potential value of σ_I.

Table 5.1
Glider observations for trinary CA with N = 5

$\sigma_{I_{min}}$ (trits)	percentile (%)	quiescent background		non-quiescent background	
		slow gliders (%)	multiple objects (%)	slow gliders (%)	multiple objects (%)
0.0	100	1	0	31	16
0.7	51	2	1	38	27
1.1	8.3	9	3	26	22
1.4	1.9	41	17	14	11
1.7	0.34	59	26	1	1
1.8	0.13	64	26	1	0
1.9	0.038	64	23	3	2
2.2					

The results of the experiment are shown in table 5.1. The time and memory required to compute fluctuations increases exponentially with region size, hence the choice of a relatively small region size, N = 5. σ_I was computed for a random sample of one million trinary rules. The distribution of σ_I values has an extended exponential tail (cf. sec. 4.6). For example, only 0.038% of all rules have $\sigma_I \geq 1.9$ trits. Each row of the table corresponds to a range of σ_I values with $\sigma_{I_{min}}$ as the minimum (≥) and $\sigma_{I_{min}}$ of the next row as the maximum (<). A sample of at least 350 rules from each of the ranges was observed and classified.

The fraction of rules that support at least one slow glider in a quiescent background increases dramatically with σ_I. Rules in this class are rare in general, as are rules with large σ_I, but there is a strong correlation between the two groups. The fraction that support multiple objects (a slow glider plus at least one other stable object – a glider or stationary object – of different speed or direction) in a quiescent background also increases with σ_I. However the

increase is not as dramatic and there is a small decrease at the largest values of σ_I. Slow gliders in a non-quiescent background are plentiful among the trinary rules, but it was subjectively observed that many are of low 'quality': the gliders were mere phase shifts in the background or the background was a tiny island momentarily appearing in a sea of disorder. The 'quality' increases with σ_I as does the fraction with multiple objects, but eventually this class of rules becomes rare.

Rules with large values of σ_I tend to evolve through three phases noted by Galas and Herrmann [Gallas]. There is an initial phase in which the random initial state evolves toward a quiescent background. During this phase stable objects may be created. These objects may interact during the second phase, and if the interaction ever ceases or becomes repetitive, or if all objects are annihilated, the third and final phase is realized. For the trinary CA it was observed that the duration of phase 1 is, typically, tens of time-steps while that of phase 2 can be many hundreds or even thousands of time-steps.

In general, phase 2 behavior provides a link between computation and complexity. Object interactions might form the basis of activities in a CA version of a universal computer carrying out some purposeful computation, with phase 3 representing the halt state (cf. sec. 2.4). In such a case the duration of phase 2 is undecidable. In contrast, glider objects might be viewed as propagating structures in a naturally evolving Wolfram class IV (complex) rule (cf. sec 2.3). σ_I encompasses this link in a quantitative way and is more general: it is based on the essential information processing properties of systems but is not directly tied to glider existence and thus allows for complex rules that do not have visually identifiable propagating structures. It is worth recalling that class IV (complex) rules are believed to be rare, consistent with the rarity of rules

with large values of σ_I, which is more pronounced with larger region sizes (cf. sec. 4.6).

The existence of multiple object types may be interpreted as an indication of rule complexity because of the possibility of phase 2 activity. A distinction must be made between rule complexity and glider complexity. A glider is considered internally complex if it is large and slow-moving. However, an isolated glider cannot participate in a higher level computation if there are no other objects with which it can interact. If its available region-states are reallocated to allow for the existence of several object types with less internal complexity, the possibility of complex behavior on a large scale arises if the objects can have non-trivial interactions. It is not yet clear how many objects with what type of interactions can achieve what level of computational ability in the sense of the Chomsky hierarchy (cf. sec. 2.4). The questions of whether σ_I properly accounts for the tradeoff between rule and glider complexity and of whether σ_I correlates with the hierarchy are important matters for future research.

Another question is whether rules with non-quiescent backgrounds are generally less complex than those with quiescent backgrounds, as σ_I seems to indicate. Consideration of the elementary CA provides a clue. The rightmost column of table 4.1 shows glider/objects observed in an experiment similar to that conducted for the trinary CA. 'S' indicates a slow glider, 'F' a fast glider and 'T' a stationary object. When repeated, each 'S' indicates a glider of different speed. 'FF' indicates fast gliders moving in both directions. Cases where only fast gliders were observed moving in only one direction are not shown. None of the designated gliders resides in a quiescent background, consistent with the premise that the elementary CA, as a class, are less complex than other classes, such as trinary CA, that support slow gliders in a quiescent background.

However, σ_I appears to be effective within this class. Of the 8 rules observed to support slow gliders, 7 are among the 11 rules with the greatest values of σ_I. Rule 110, which has the most object types, has the most rapidly increasing value of σ_I near N = 19.

6. Application to One-Dimensional Maps

Maps provide an example of how the information variable method can be applied to dynamical systems with a continuous variable. One-dimensional maps are discrete in time but have a single continuous spatial variable. The continuous variable must be discretized by partitioning the space axis into *bins*. This introduces an approximation that improves as the partition size decreases. Two examples are considered: the shift map and the logistic map. Following this, a comparison is made between information variables and the *Lyapunov exponent* which is traditionally used to quantify chaos and order in continuous dynamical systems.

6.1. The shift map

The shift map [Wolfram-4, Shaw] is described by

$$x_{t+1} = 2\,x_t \pmod 1, \quad 0 \le x_t < 1. \tag{6.1}$$

The smallest NFA model of the map is obtained by partitioning the unit interval into two equally sized bins such that

$$\text{bin 0: } 0 \le x_t < 0.5 \tag{6.2}$$

and

$$\text{bin 1: } 0.5 \le x_t < 1. \tag{6.3}$$

The states of the NFA correspond to x being in one of the two bins. The NFA links are obtained by using the map to determine the possible next-bins for each present-bin. In this case, either bin leads to both bins. Next, each link is assigned a forward conditional probability based on the probability of arriving in the next-bin to which it leads. The probability is equal to the fraction of the total next-bin coverage that overlaps the next-bin under consideration. This approach assumes that any value of x is equally probable within the present-

bin. In this case, there is a 50-50 chance that either bin 0 or bin 1 follows, say, bin 0. Thus all conditional probabilities are 0.5 and the NFA is the purely chaotic two-state NFA shown in figure 3.2b. The corresponding information variables are listed in table 3.1.

The behavior of the actual map can be understood best by representing the initial value x_0 in binary. Each iteration of the map – multiplication of x by two followed by the modulus one operation – is equivalent to shifting the bits (binary digits) of x_n one place to the left and dropping the most significant bit. The most significant remaining bit is the bin number. Thus, the bin sequence is exactly the sequence of bits in the initial value, read from left to right.

The shift map illustrates the essence of chaotic behavior: sensitivity to initial conditions. As time goes on, the bin sequence is determined by initially microscopic information. This is achieved by the stretching effect of the map on the unit interval (the factor of two). The folding of the map (the modulus operation) restricts the trajectories to the unit interval, which would otherwise move toward infinity on the x axis. With each iteration, a bit is brought up from microscopic scales to determine the next-bin, while a bit is lost in order to restrict the trajectory. (It is impossible to determine the prior-bin once the modulus is taken.) The shift map usually behaves randomly since virtually all real numbers on the unit interval have a random bit sequence when expressed in binary (cf. sec. 2.6). However, it is more accurate to say that it simply reflects the randomness (or occasional lack of randomness) of its initial condition. It acts as an *information pipe*, magnifying microscopic information to macroscopic scales (cf. sec. 2.5).

The NFA model faithfully captures the behavior of the shift map. Exactly one bit of information is needed to select the next-state. Instead of coming from the initial condition, this bit comes from an unbiased random source consid-

ered external to the system. The random source simulates the random nature of typical real numbers. Like the shift map, the NFA merely acts as a conduit for information. It is as chaotic as possible for a two-state system.

The NFA model has been described as "purely chaotic". This actually means "as chaotic as possible for a finite model". In a finite model, there is an unavoidable element of order represented by converging links (thus < G > always equals < L >). If the number of bins was increased, the new NFA model would no longer be *purely* chaotic (unless the stretching factor of the map was also increased), since the bins would no longer be fully interconnected by single links.

6.2. The logistic map

The logistic map [Campbell, Shaw] is described by

$$x_{t+1} = r\, x_t (1 - x_t), \quad 0 \le r \le 4, \quad 0 \le x_0 \le 1. \tag{6.4}$$

Like the shift map, trajectories are restricted to the unit interval due to folding. $x_{t+1} = 0$ when $x_t = 0$ or 1, and $x_{t+1} = r/4$ (its maximum) when $x_t = 0.5$. Unlike the shift map, stretching is nonlinear; the degree of nonlinearity is controlled by the parameter r. The asymptotic behavior of the map is shown as a function of r in figure 6.1. After 10,000 iterations beginning with $x_0 = 0.5$, 150 consecutive values of x are plotted for 1,000 values of r in the range $2 \le r \le 4$. Small values of r result in a stable fixed point attractor at x = 0. When r is increased through 1.0 the fixed point at zero becomes unstable and a new stable fixed point appears at $x = 1 - 1/r$. Further increases in r lead to a period doubling cascade beginning at r = 3.0 and extending to $r \approx 3.57$. Just beyond this accumulation point where the period becomes infinite, the trajectory becomes chaotic. As r continues to increase the size of the chaotic attractor increases steadily until at r = 4.0, the entire unit interval is the attractor. However, windows of periodic behavior

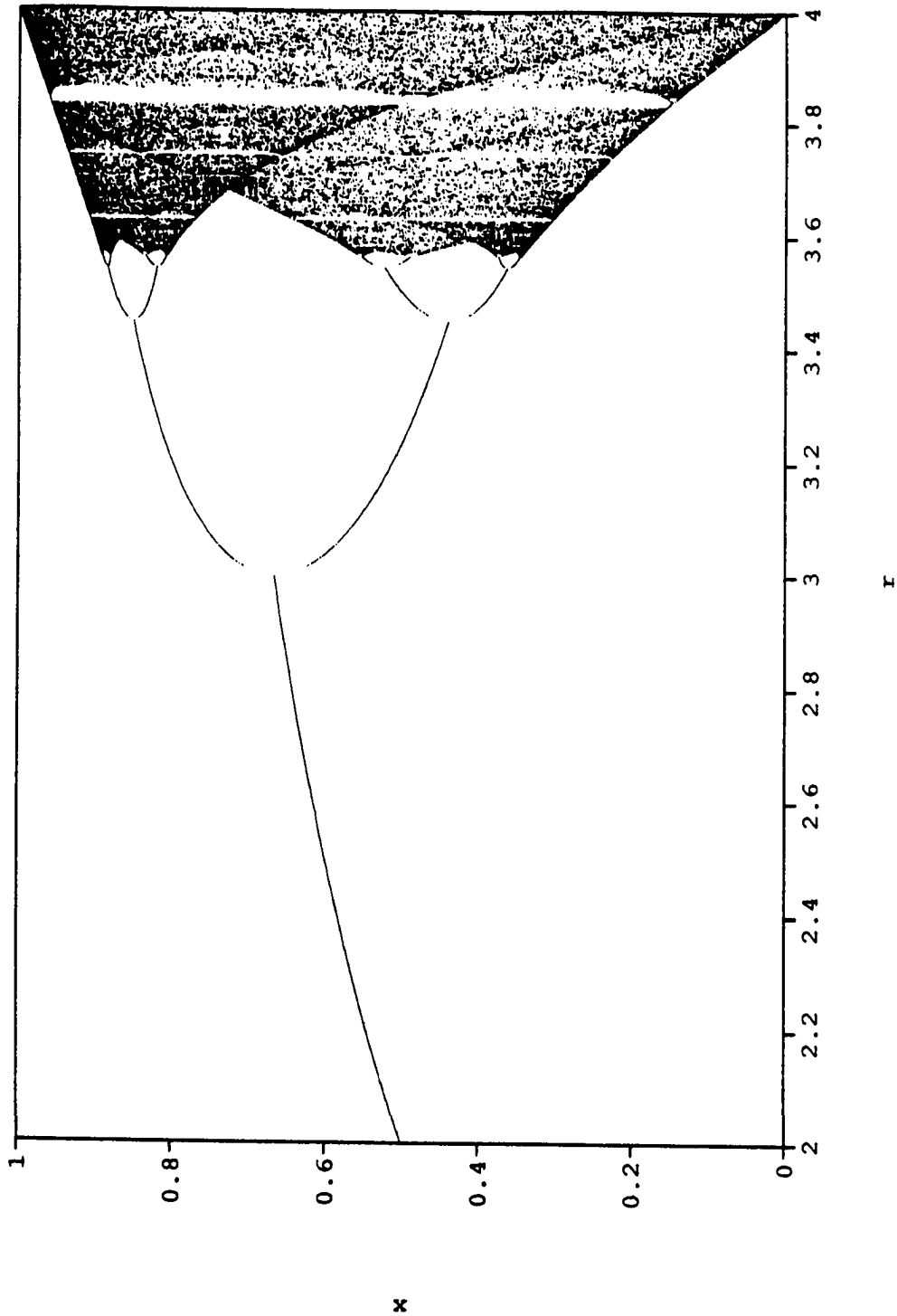


Figure 6.1. Asymptotic behavior of the logistic map as a function of the nonlinearity parameter r.

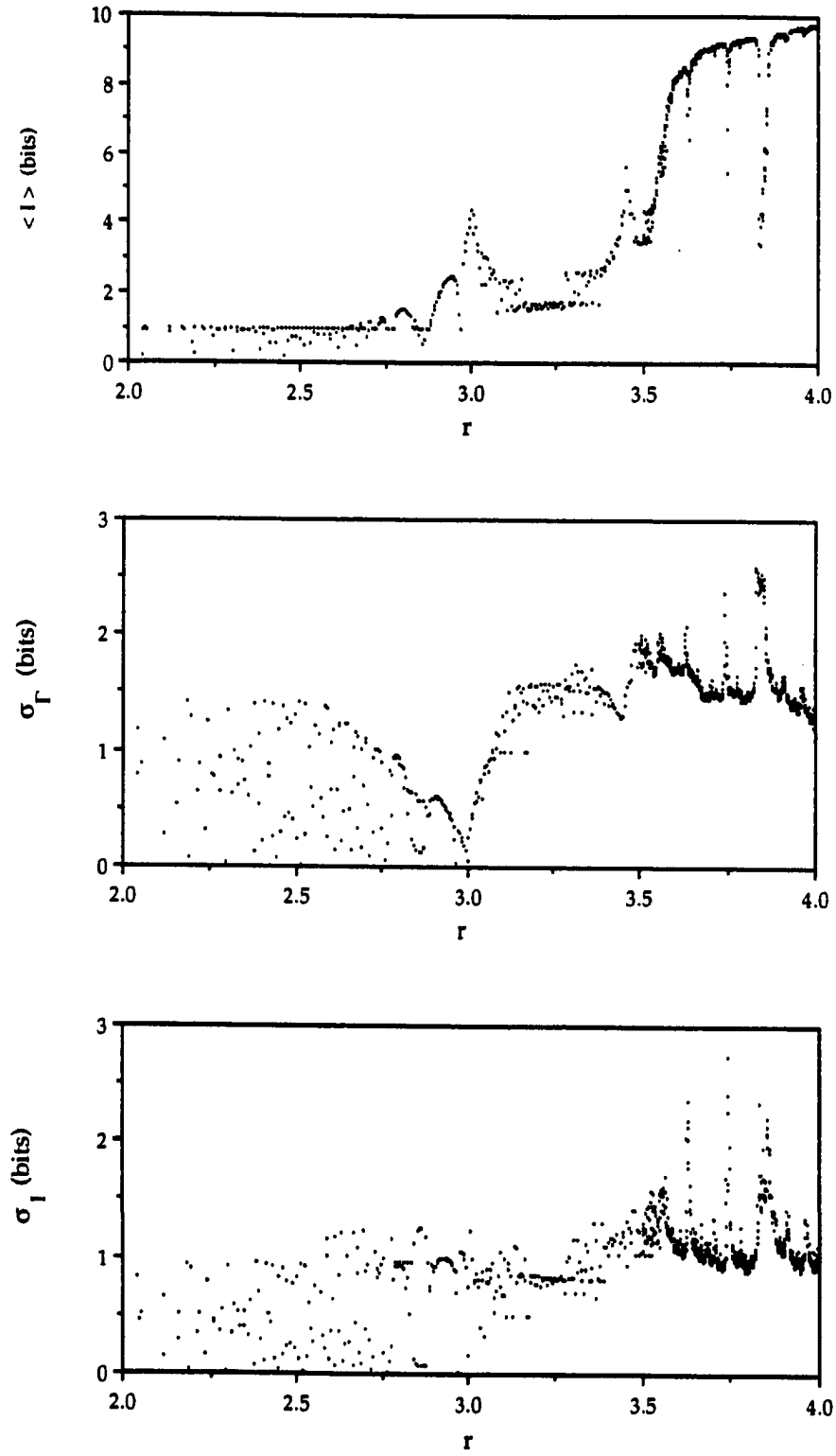


Figure 6.2. Information variables $< I >$, σ_Γ and σ_I vs. r for the logistic map NFA model with the unit interval partitioned into 1024 bins.

appear intermittently, the largest being a period 3, 6, 12,..., cascade beginning at r ≈ 3.83.

The behavior of the logistic map is clearly more complex than that of the shift map for some values of r. Information variables can quantify the level of complexity. NFA construction proceeds as for the shift map, but the determination of forward conditional probabilities introduces a piecewise linear approximation that should improve as the number of bins increases. Figure 6.2 shows the information variables $< I >$, σ_Γ and σ_I vs. r with the unit interval partitioned into 1024 bins of equal width. 500 values of r in the range $2 \leq r \leq 4$ and an extra 500 in the range $3.5 \leq r \leq 4$ are used.

$< I >$ indicates state utilization. Over all, state utilization increases with r, but not uniformly. In chaotic regimes state utilization is high, thus indicating a single large chaotic attractor. At r = 4.0, the attractor includes every possible state but state probabilities are unequal, so the maximum entropy is slightly less than $\log_2(1024) = 10$ bits. In periodic regimes $< I >$ increases as the size of the periodic attractor increases, as expected, but it also peaks at the bifurcation points where the period doubles. This is due to the instability of the map near these points. Transients are much longer and there is greater sensitivity to external noise. In figure 6.1 the only noise present is computer roundoff error, so the bifurcation points appear sharp. However, the NFA attractor includes all the states that can be reached by any value of x within a bin. Thus, the discretization process combined with random stimulation reveals important behavior of the map not observable in a noise-free environment.

The largest periodic window, of period 3^n beginning at $r \approx 3.83$, provides the best resolution for examining the rapidly changing σ_I and σ_Γ. For this window the largest observed value of σ_I is 2.34 bits at r = 3.828, just before the transition from chaos to order. σ_Γ peaks at r = 3.830, just after the transition,

with a value of 2.59 bits. Thus, σ_I and σ_Γ indicate that complexity is greatest at the boundary between order and chaos in close agreement with the work of Crutchfield (cf. sec. 2.7.4). The present results indicate that the complexity at the *beginning* of the period 3^n window is significantly greater than at the end.

At most values of r throughout the chaotic regime, σ_Γ is greater than σ_I, indicating that the complexity is relatively low-level – information storage is relatively short in duration. The exceptions are at some of the local maxima, where the two variables seem to converge, indicating a relative increase in high-level complexity – greater information storage for longer durations. The large values of σ_I and σ_Γ in the first periodic regime ($r < 3.57...$) are rendered less significant by the corresponding low entropies. σ_Γ and σ_I drop sharply at the bifurcation point at $r = 3.0$, indicating that the map exhibits complex behavior *near*, but not *at* the point.

Like the shift map, the logistic map can pump information from microscopic to macroscopic scales. However, the logistic map can temporarily store information as it passes through. Complexity is made possible by the nonlinearity of the map. r must be large enough to allow for chaos, but chaos alone is not enough. There must be a variability in the relative amounts of chaos and order in different parts of the map (or NFA); the changing slope of the map (when x_{t+1} is graphed as a function of x_t) provides this. Then, its behavior can depend on its history as well as the current information being gained from the initial condition. Complexity, as measured by fluctuation in net information gain, is locally maximum at the boundaries between order and chaos (to the degree of resolution in fig. 6.2).

Figure 6.3 is a plot of σ_I and σ_Γ vs. $\log_2$(number of bins) for $r = 4$. σ_Γ approaches an asymptotic value, suggesting that the choice of 1024 bins is adequate for an accurate model at this value of r. In general, however, the

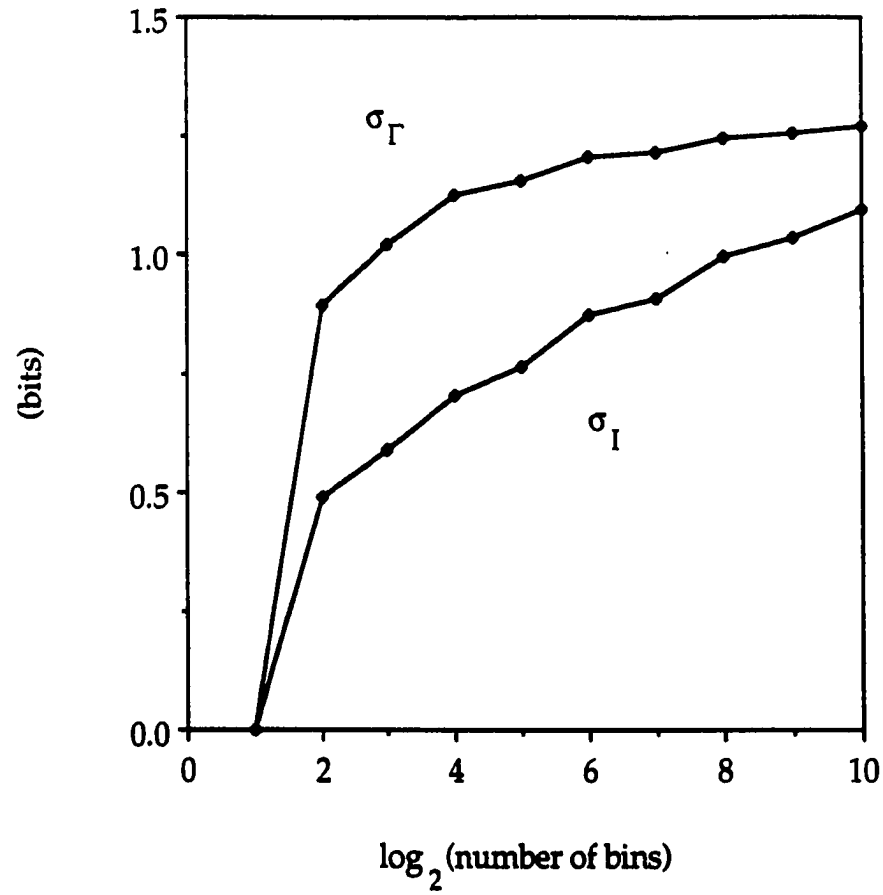


Figure 6.3. σ_I and σ_Γ vs. $\log_2$(number of bins) for the r = 4 logistic map.

discretization process can cause erratic dependence of information variables on partition resolution near sensitive values of r, such as $r \approx 3.83$. Also, as discussed in section 2.7.4, no finite number of bins may be adequate to capture the behavior of the logistic map exactly at the critical points corresponding to the onset of chaos.

6.3. The Lyapunov exponent

The Lyapunov exponent [Campbell, Shaw] is defined for maps as

$$\lambda = (1/n) \log_e(D_n/D_0) \tag{6.6}$$

in the limit of large n, where D_0 is the initial separation (in space) of two distinct trajectories and D_n is the separation after n iterations. For one-dimensional maps, the ratio D_{n+1} / D_n is just the (average) slope of the map at bin n. In terms of the slope

$$\lambda_i = \log_e(|\text{slope}| \text{ at bin i}). \tag{6.7}$$

A useful quantity is the asymptotic mean of λ

$$< \lambda > = \Sigma_i \, p_i \, \lambda_i. \tag{6.8}$$

It has become customary to define chaotic behavior in continuous systems as that which occurs when $< \lambda >$ is positive. Orderly behavior occurs when $< \lambda >$ is negative. For the shift map $< \lambda > = 1$ bit since the slope of the map is two at all the bins. Nearby trajectories separate by a factor of two with each iteration. Therefore, the shift map displays chaotic behavior.

Although $< \lambda >$ refers to the asymptotic (steady-state) attractor, it contains information about transient bins that are close to the attractor. Negative values of $< \lambda >$ imply stability: If a trajectory is perturbed away from a fixed point or limit cycle it will quickly return. Noise is damped. Positive values imply instability: Perturbations are amplified over subsequent time-steps. Initially close trajectories diverge exponentially. When the exponent is negative but near zero, perturbations result in long excursions away from periodic attractors. For

example, this occurs in the logistic map as r is increased through 3.0 as shown in figure 6.4. The dynamic behavior undergoes an abrupt qualitative change (a bifurcation) from period-one to period-two. The period-one attractor becomes unstable as the Lyapunov exponent approaches zero. Any noise present will no longer be damped out. Noise-induced excursions away from the attractor are sustained in time.

Long transients and marginal stability are considered indications of complex behavior. The Lyapunov exponent would seem to indicate complexity when $< \lambda >$ is near zero and order and chaos are balanced. This occurs both near bifurcation points and at the beginnings and ends of periodic windows, where $< \lambda >$ changes sign, in agreement with the results discussed in the previous section. Nevertheless, $< \lambda >$ qualifies as a complexity measure only in the crudest sense. There is no way to compare the relative complexity associated with different values of r where $< \lambda > = 0$. There is another serious flaw in that $< \lambda >$ would be zero for a purely periodic system. As discussed in section 3.4.3 purely periodic systems lie on the boundary between order and chaos and yet should not be considered complex.

The shortcomings of the Lyapunov exponent are apparent when compared with information gain. λ_i, as defined, equals G_{ij} when the |slope| is greater than one but equals $1 - L_{ij}$ when the |slope| is less than one if the present-states leading to the next-state j are equally probable. It emphasizes information gain in chaotic situations but information loss in ordered situations. Having only one subscript, it takes no account of non-uniform probabilities and considers only one end of the $i \rightarrow j$ transition. In the shift map, probabilities are uniform (for typical initial conditions) and G_{ij} and L_{ij} are equal, so information variables provide no advantage. However, in the logistic map probabilities can be non-uniform and G_{ij} and L_{ij} can be different. The net

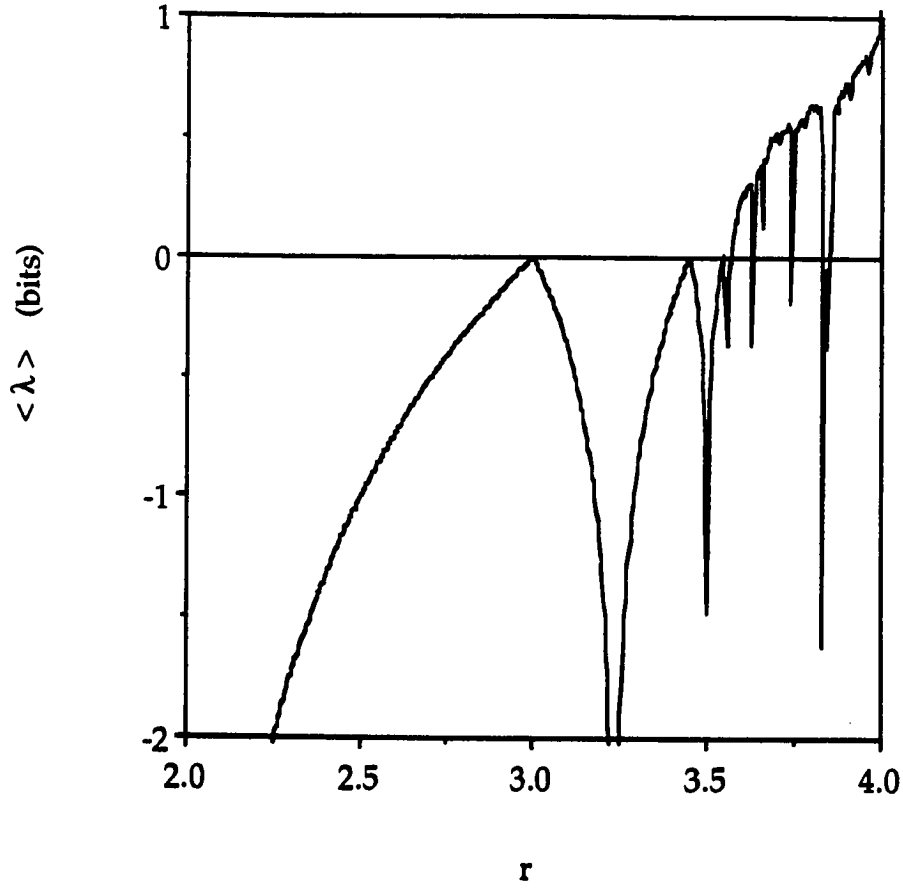


Figure 6.4. < λ > vs. r for the logistic map, 2 ≤ r ≤ 4, 1024 bins.

information gain, $\Gamma_{ij} = G_{ij} - L_{ij}$, can be thought of as a generalization of λ_i. However, even Γ misses the mark as a complexity measure. As described above, it is the fluctuation in net information gain over one or more time-steps that best measures complexity.

7. Conclusions

A general method for analyzing finite discrete dynamical systems has been described and applied to some one-dimensional cellular automata and maps. The method provides a quantitative description of order and chaos in terms of information variables. Complexity has been identified with the *variability* in order and chaos and is manifested in the variance of net information gain over one or more time-steps. Indirectly, complexity is identified with computational power since the information variables directly measure the basic components of computation: information storage, transmission and occasional modification. However, this connection has not yet been firmly established.

The information variable method begins with construction of a nondeterministic finite automaton model of a given system. The system must be discrete and finite. If it is not, it must be discretized and localized as necessary. Whether the resulting model accurately captures the dynamics of the system depends on the resolution of the discretization and on the homogeneity and locality of the dynamics. Localization is not merely a device to contain the number of states to a manageable level. It is congruous with the method's focus on information flow between the system and its environment. The manner in which the flow changes with system size reveals vital characteristics of the system.

In the applications discussed in the previous chapters it was assumed that the dynamics of the systems being modelled were known. It is also possible to construct NFA models based on empirical observation of systems whose dynamics are not known. If a system has inputs that can be controlled,

then it can be randomly stimulated. The internal states of the system may be hidden, but state space structure can be inferred by observation of output sequences [Packard, Broomhead, Fraser-1, Fraser-2]. A model can be constructed that can reproduce the statistical behavior of the system within certain tolerances (cf. sec. 2.7.4). Complex systems require longer stimulation, observation of longer sequences and construction of larger models to achieve a given tolerance.

To the extent that an NFA model is accurate, it captures the dynamics of the system being modelled. When the NFA is small enough, it can be comprehended visually as a graph of nodes and links. Order, chaos and complexity appear as convergence, divergence and the variability of these throughout the graph. Why not stop with this state space description? One reason is that only the smallest NFA can be displayed visually. A 19-cell elementary CA region has more than a half-million states. The other reason is that scientists always look for the most compact description of systems adequate for a particular purpose. For example, in mechanics an irregular object might normally require a lengthy description, but for most purposes only its mass, center of mass, and moments of inertia are important. For these purposes it can be considered identical to all other objects with the same values of these parameters.

The information variable method offers a compact, quantitative description of the order, chaos and complexity in a model. The basic information variables are entropy I, information gain G, information loss L and net information gain Γ. They apply locally in state space but a compact description is achieved by global averaging. Variability in link convergence and divergence translates to variance in Γ. Application to CA suggests that the most complex systems are those for which the fluctuation in I, σ_I, increases

characteristic length Λ_Γ. Localized regions have a capacity to mutually interact such that combining them results in greater information processing capability or computational power. Further study is needed to verify this conjecture, particularly by applying the method to other classes of systems.

Also needed is a detailed comparison of the information variable method with alternative methods, some of which were discussed briefly in section 2.7. At this stage, only a few tentative remarks will be made. Of the alternatives discussed, my method most resembles Crutchfield and Young's because of its primary emphasis on temporal behavior and NFA models, although their NFA states represent historical contexts whereas mine represent actual system states. However, their complexity measure is based on the size of the NFA model and identifies directly with the Chomsky hierarchy, whereas mine is based on variability of link convergence and divergence in the model. As such, my method is closer in spirit to Langton's in its identification with information flow. Otherwise, Langton's method is totally different. It is specific to CA and uses the suspect quantity mutual information. My results generally agree with both Crutchfield and Young , and Langton et al.. I find that complexity peaks at boundaries between order and chaos where entropy is intermediate. I am reluctant, however, to claim, as they do, that there is a phase transition in the physical sense at these boundaries.

Comparison with the methods of Wolfram, Grassberger, Lindgen and Nordahl on a theoretical level is more difficult, partly because of their primary emphasis on spatial patterns. A superficial comparison can be made with Wolfram's regular language complexity and Grassberger's set complexity (see the appendix of [Wolfram-5]), and Grassberger and Zambella's forecasting complexity [Grassberger-4] based on their published tables of the complexity of elementary CA rules. Comparison of their tables with each other and table 4.1

reveals strong correlations but also significant differences between the various complexity measures. Let us focus on a few representative rules. Rule 22 (cf. sec. 4.3) is at or near the top of all the complexity lists except the one based on σ_I where it is ranked tenth. Rule 110 is far down all the lists except the σ_I list where it is ranked eleventh and is moving up. Rule 30 has null complexity in every case. Rule 37 is in the top ten of every list. Rules 7 and 62, which top the σ_I list, are far down the others. Rule 164, which is near the top of the forecasting complexity list, is far down the set complexity list but at the bottom of the σ lists. A tentative conclusion is that the measures are quite different from one another (except regular language and set complexity, which are closely related). σ_I is the only one that seems to give a higher rating to rule 110, which was observed to have, by far, the greatest number of slow-moving gliders of different speeds. However, it would be premature to draw any definite conclusions about which complexity measure is superior.

Another matter for future research is to investigate the behavior of information variables, particularly σ_I, for larger CA region sizes than previously considered. This might require access to a supercomputer. Also, investigation of CA with more than three states per cell and two- and three-dimensional CA is warranted.

A stronger connection with computation theory needs to be established. This could be accomplished by using information variables to search for CA rules with multiple interacting objects and then attempting to construct computing devices such as logic gates, counters, carry generators, stack automata, etc.. Several interesting trinary rules have already been found: one that does a kind of parity operation and another that can divide-by-two. Others come close to being able to accept palindromes (which would make them stack automata), but more states-per-cell are probably needed. As

discussed in section 5.5, the whole question of whether a CA rule's computational power can be judged by the number and interaction properties of particle-like objects is unresolved.

There are still other questions that invite further investigation. Chief among them in the mind of the author is the question of scales and hierarchies. Gliders are examples of coherent systems with internal complexity that can become components on a larger scale. Examples of this phenomenon are ubiquitous in nature. Consider the hierarchical organization that begins on the subatomic scale and proceeds through the levels of atoms, molecules, cell structures, cells, organs, organisms, families, societies, species, ecosystems, Gaia, and beyond. There is a remarkable consistency in this scheme, independent of scale. The tendency of systems to organize into coherent subsystems with well-defined boundaries might be understandable in terms of information flow. In systems with a dimensional aspect, there are unavoidable constraints imposed by geometry. The volume of a region always increases in size faster than its surface, the boundary across which information flows. Also, the number of states increases exponentially with system size, making full interconnection rapidly unfeasible.

It is true that the information variable method is limited to consideration of relatively small systems, but this may be a limitation of nature as well. Even the vast amounts of time associated with evolutionary development are inadequate to allow natural selection to explore the state space of systems beyond a certain size. The characteristic length Λ_Γ, or some similar measure derived from basic information processing capabilities, may determine the scale at which a system naturally organizes into subsystems. At each level of a hierarchy, subsystems can be modelled in terms of their input-output behavior and how they interact with other subsystems. The material basis and

internal structure of systems becomes unimportant compared to their capacity to process information. The subsystems of even the most complex hierarchies might be manageable subjects for the information variable method.

LIST OF REFERENCES

[Anderson] P. Anderson, Is Complexity Physics? Is It Science? What Is It?, Physics Today, July 1991, 9.

[Bak] P. Pak, C. Tang and K. Wiesenfeld, Self-organized criticality, Physical Review A 38 (1988), 364-374.

[Bennett-1] C. Bennett, Information, Dissipation, and the Definition of Organization, in D. Pines, ed., Emerging Syntheses in Science (Addison-Wesley , Reading, Massachusetts, 1985), 297-313.

[Bennett-2] C. Bennett, Computational Measures of Physical Complexity, in D. Stein, ed., Lectures in the Sciences of Complexity (Addison-Wesley , Reading, Massachusetts, 1989), 787-798.

[Bennett-3] C. Bennett, How to Define Complexity in Physics, and Why, in W. Zurek, ed., Complexity, Entropy, and the Physics of Information (Addison-Wesly, Reading, Massachusetts, 1990), 137-148.

[Berlekamp] E. Berlekamp, J. Conway, and R. Guy, Winning Ways for Your Mathematical Plays, Vol. 2 (Academic Press, 1982), Chapter 25.

[Broomhead] D. Broomhead and G. King, Extracting qualitative dynamics from experimental data, Physica D 20 (1986) 217-236.

[Campbell] D. Campbell, An Introduction to Nonlinear Dynamics, in D. Stein, ed., Lectures in the Sciences of Complexity (Addison-Wesley , Reading, Massachusetts, 1989), 3-105.

[Chaitin] G. Chaitin, Algorithmic Information Theory (Cambridge University Press, Cambridge, 1987).

[Crutchfield-1] J. Crutchfield and K. Young, Inferring Statistical Complexity, Phys. Rev. Letters 63 (1989), 105-108.

[Crutchfield-2] J. Crutchfield, Inferring the Dynamic, Quantifying Physical Complexity, in N. Abraham, ed., Measures of Complexity and Chaos (Plenum Press, New York, 1989).

[Crutchfield-3] J. Crutchfield and K. Young, Computation at the Onset of Chaos, in W. Zurek, ed., Complexity, Entropy, and Physics of Information (Addison-Wesley, Reading, Massachusetts, 1989), 223-269.

[Culik] K. Culik and S. Yu, Undecidability of CA classification schemes, Complex Systems 2 (1988), 177-190.

[Davies] P. Davies, The Cosmic Blueprint (Simon and Schuster, New York, 1988).

[Dewdney] A. Dewdney, The Turing Omnibus (Computer Science Press, Rockville, Maryland, 1989).

[Doolen] G. Doolen, et al., eds., Lattice Gas Methods for Partial Differential Equations (Addison-Wesley, Reading, Massachusetts, 1990).

[Eisele] M. Eisele, Long-range correlations in chaotic cellular automata, Physica D 48 (1991), 295-310.

[Fraser-1] A. Fraser, Phase Space Reconstructions, Center for Nonlinear Dynamics, University of Texas at Austin preprint (1988), 58-72.

[Fraser-2] A. Fraser, Reconstructing attractors from a scalar time series: a comparison of singular system and redundancy criteria, Physica D 34 (1989), 391-404.

[Gallas] J. A. C. Gallas and H. J. Herrmann, Investigating an Automaton of "Class 4", International Journal of Modern Physics C 1 (1990), 181-191.

[Gleick] J. Gleick, Chaos (Viking, New York, 1987).

[Grassberger-1] P. Grassberger, Toward a Quantitative Theory of Self-Generated Complexity, Intl. J. Theor. Phys. 25 (1986), 907-938.

[Grassberger-2] P. Grassberger, Long-Range Effects in an Elementary Cellular Automaton, J. Stat. Phys. 45 (1986), 27-39.

[Grassberger-3] P. Grassberger, Some more exact enumeration results for 1D cellular automata, J. Phy. A 20 (1987), 4039-4046.

[Grassberger-4] D. Zambella and P. Grassberger, Complexity of Forecasting in a Class of Simple Models, Complex Systems 2 (1988), 269-303.

[Grassberger-5] P. Grassberger, Information and complexity measures in dynamical systems, in H. Atmanspacher and H. Scheingraber, eds., Information Dynamics (Plenum Press, New York, 1991), 15-34.

[Hopcroft] J. Hopcroft and J. Ullman, Introduction to Automata Theory, Languages, and Computation (Addison-Wesley, Reading, Massachusetts, 1979).

[Huberman] B. Huberman and T. Hogg, Complexity and adaptation, Physica D 22 (1986), 376-384.

[Jen] E. Jen, ed., 1989 Lectures in Complex Systems (Addison-Wesley , Reading, Massachusetts, 1990).

[Landauer] R. Landauer, Information is Physical, Physics Today, May 1991, 23-29.

[Langton-1] C. Langton, Studying artificial life with cellular automata, Physica D 22 (1986), 120-149.

[Langton-2] C. Langton, Virtual State Machines in Cellular Automata, Complex Systems 1 (1987), 257-271.

[Langton-3] C. Langton, Computation at the edge of chaos: phase transitions and emergent computation, Physica D 42 (1990), 12-37.

[Li-1] W. Li and N. Packard, The Structure of the Elementary Cellular Automata Rule Space, Complex Systems 4 (1990), 281-297.

[Li-2] W. Li, N. Packard, and C. Langton, Transition Phenomena in Cellular Automata Rule Space, Physica D 45 (1990), 77-94.

[Li-3] W. Li, A Relation Between Complexity and Entropy for Markov Chains and Regular Languages, Complex Systems 5 (1991), 381-399.

[Lindgren-1] K. Lindgren and M. Nordahl, Complexity Measures and Cellular Automata, Complex Systems 2 (1988), 409-440.

[Lindgren-2] K. Lindgren and M. Nordahl, Universal Computation in Simple One-Dimensional Cellular Automata, Complex Systems 4 (1990), 299-318.

[Lindgren-3] K. Lindgren, Entropy and Correlations in Dynamical Lattice Systems, in P. Manneville, et al., eds., Cellular Automata and Modeling of Complex Physical Systems (Springer-Verlag, Berlin, 1990), 27-40.

[Manneville] P. Manneville, et al., eds., Cellular Automata and Modeling of Complex Physical Systems (Springer-Verlag, Berlin, 1990).

[Martin-Löf] P. Martin-Löf, Definition of Random Sequences, Information and Control 9 (1966), 602-619.

[Moore] C. Moore, Unpredictability and Undecidability in Dynamical systems, Physical Review Letters 64 (1990), 2354-2357.

[Newell] A. Newell, The Dynamics and Analysis of Patterns, in D. Stein, ed., Lectures in the Sciences of Complexity (Addison-Wesley , Reading, Massachusetts, 1989), 107-173.

[Packard] N. Packard, J. Crutchfield, J. Farmer, and R. Shaw, Geometry from a Time Series, Physical Review Letters 45 (1980), 712-716.

[Penrose] R. Penrose, The Emperor's New Mind (Oxford University Press, Oxford, 1989).

[Poundstone] W. Poundstone, The Recursive Universe (Contemporary Books, Chicago, 1985).

[Shaw] R. Shaw, Strange Attractors, Chaotic Behavior, and Information Flow, Z. Naturforsch. 36a (1981), 80-112.

[Stein-1] D. Stein, Spin Glasses, Scientific American, July 1989, 52-59.

[Stein-2] D. Stein, ed., Lectures in the Sciences of Complexity (Addison-Wesley , Reading, Massachusetts, 1989).

[Wolfram-1] S. Wolfram, Statistical Mechanics of Cellular Automata, Reviews of Modern Physics 55 (1983), 601-644.

[Wolfram-2] S. Wolfram, Universality and complexity in cellular automata, Physica D 10 (1984), 1-35.

[Wolfram-3] S. Wolfram, Computation Theory of Cellular Automata, Commun. Math. Phys. 96 (1984), 15-57.

[Wolfram-4] S. Wolfram, Origins of Randomness in Physical Systems, Physical Review Letters 55 (1985), 449-452.

[Wolfram-5] S. Wolfram, Theory and Applications of Cellular Automata (World Scientific, Singapore, 1986).

[Wootters] W. Wootters and C. Langton, Is there a sharp phase transition for deterministic cellular automata?, Physica D 45 (1990), 95-104.

[Zurek] W. Zurek, Complexity, Entropy, and the Physics of Information – a Manifesto, in W. Zurek, ed., Complexity, Entropy, and the Physics of Information (Addison-Wesley , Reading, Massachusetts, 1990), vii-x.