The Observers' Paradox: Apparent Computational Complexity in Physical Systems

John F. Kolen and Jordan B. Pollack

To appear Summer 1994 in *The Journal of Experimental and Theoretical Artificial Intellignce*

Running Head: The Observers' Paradox

August 15, 1993

Laboratory for Artificial Intelligence Research
Department of Computer and Information Sciences
The Ohio State University
Columbus, OH 43210
kolen-j@cis.ohio-state.edu and pollack@cis.ohio-state.edu

Abstract

Many researchers in AI and cognitive science believe that the complexity of a behavioral description reflects the underlying information processing complexity of the mechanism producing the behavior. This paper explores the foundations of this complexity argument. We first distinguish two types of complexity judgements that can be applied to these descriptions and then argue that neither type can be an intrinsic property of the underlying physical system. In short, we demonstrate how changes in the method of observation can radically alter both the number of apparent states and the apparent generative class of a system's behavioral description. From these examples we conclude that the act of observation can suggest frivolous computational explanations of physical phenomena, up to and including cognition.

> The daily warmth we experience, my father said, is not transmitted by Sun to Earth but is what Earth does in response to Sun. Measurements, he said, measure measuring means.[1]

Introduction

Cognitive science has worked under the general assumption that complex behaviors arise from complex computational processes. Computation lends a rich vocabulary for describing and explaining cognitive behavior in many disciplines, including linguistics, psychology, and artificial intelligence. It also provides a novel method for evaluating models by comparing the underlying generative capacity of the model. The generative enterprise in linguistics, for example, maintains that the simplest mathematical models of animal behavior - as finite state or stochastic processes - are inadequate for the task of describing language. Descriptions (or explanations) of language structure require at least a context-free or context-sensitive model:

> There are so many difficulties with the notion of linguistic level based on left-to-right generation, both in terms of complexity of description and lack of explanatory power, that it seems pointless to pursue this approach any further.[2]

Even Newell and Simon's Physical Symbol System Hypothesis (Newell and Simon, 1976) identifies recursive computation of a physical symbol system as both a necessary and sufficient condition for the production of intelligent action. Such claims are important as they focus our research attention on particular classes of solutions which we know *a priori* to have necessary mechanistic power to perform in our desired contexts. Newell and Simon emphasize that:

> The Physical Symbol System Hypothesis clearly is a law of qualitative structure. It specifies a general class of systems within which one will find those capable of intelligent action.[3]

Since the publication of this hypothesis, the consensus of cognitive science has held that the mind/brain is computing something; identifying exactly what it *is* computing has emerged as the goal of the field.

Computational complexity, often used to separate cognitive behaviors from other types of animal behavior, will be shown to be dependent upon the observation mechanism as well as the process under examination. While Putnam (1988) has proved that all open physical system can have *post hoc* interpretations as arbitrary abstract finite state machines and Searle (1990) claimed that WordStar must be running on the wall behind him (if only we could pick out the right bits), neither considered the effects of the observer on the complexity *class* of the behavior.

The rest of the paper is organized as follows. We will first examine the role that discrete measurements play in our studies of complex systems. Over the years, methods of complexity judgement have separated into two orthogonal approaches, namely *complexion* and *generative class*. The former is a judgment related to the number of moving parts (or rules, or lines of code) in a system, while the later may be viewed as a measure of the generative capacity of the chosen descriptive framework. Then we review research on the problem of identifying complexity in physical systems, emphasizing the recent work of Crutchfield and Young (1989). Once we recognize that descriptive frameworks apply to *measurements* of a system's state rather than the state itself, we can demonstrate how simple changes of the observation method or measurement granularity can affect either the system’s complexion or its generative class. For instance, a shift in measurement granularity can promote the apparent complexity of a system from a context-free language to a context-sensitive language. Finally, we discuss the meaning of these results as they pertain to the cognitive science community.

Measurements and Complexity

Cognitive science expends most of its effort describing and explaining human and animal behavior. To construct these behavioral descriptions, one must first collect measurements of the observed behavior. Descriptions take many forms. A list of moves describes the behavior of the chess player, a transcript records linguistic behavior of a conversation, a protocol of introspected states during problem solving describes deliberative means-ends analysis, and a sequence of (x,y) locations over time records eye movement in a study of reading. Each of these examples serves as a description of behavior. The measurement may be simple, as in the case of the Cartesian coordinates, or it may be more involved, like the transcript or protocol. We assume that these measurements are necessarily discrete since we must be able to write them down.[4] To emphasize the creation of discrete symbolic representations of the physical events in the world, we will identify this process as *symbolization*. Transcription of continuous speech, for example, is a symbolization of speech production. It is impossible to avoid symbolization; there is simply too much information inherent in any physical process that is irrelevant to our needs. Imagine trying to describe the conversation between two people on a street corner. The information generated by such an encounter is infinite due to a large number of real dimensions of movement, sound, time, etc. We avoid these complications by symbolizing the physical action into sequences of measurable events, such as phonemes, words, and sentences.

Information is clearly lost during symbolization. A continuous real value is a “bottomless” source of binary digits, yet only a small number of distinctions are retained through the transduction of measurements. Of course, shuffling high precision real numbers is a waste of resources when only a few bits suffice. It is wrong, however, to believe that the information loss is merely modeling error if, as we show below, it often confuses our efforts at understanding the underlying system.

One way of understanding a system is by gauging its complexity. We have some good intuitions about certain relative comparisons: a human being is generally considered more complex than a rock. What does this ordering entail? Although judgements of system complexity have no uniformly accepted methodology, the existing approaches are sharply divided into two groups. The first appeals to the common sense notion that judges the complexity of a system by the number of internal moving parts. Thus, a system is more complex if it has a larger number of unique states induced by the internal mechanisms generating its behavior. Others (see Aida et al., 1984) have adopted the term *complexion*. We specifically use this term to refer to a measure of complexity based upon the number of unique moving parts within a system.

The second approach to judging complexity is more subtle. Imagine a sequence of mechanisms, specified within a fixed framework, with ever increasing complexion. As the complexion of a device increases, its behavior eventually reaches a limit in complexity determined by the framework itself. This limit, or generative class, can only increase or decrease with modifications to the framework. These classes are not unique; many frameworks share the same generative limit. Chomsky's early work (1957, 1965) in linguistics contributed to the foundations of computer science. Followers of this work have enshrined four classes of formal languages, each with a different computational framework. The regular, context-free, context-sensitive, and recursive languages are separated by constraints on their underlying grammars of specification, and form an inclusive hierarchy. In addition, they correlate quite beautifully with families of mechanisms, or automata, operating under memory constraints. Of course, we now know that many other classes are possible by placing different constraints on how the changeable parts interact (see many of the exercises in Hopcroft and Ullman, 1979). We use the term "generative class" out of respect to the fact that this theory of complexity arose in formal languages (automata) and the questions of what kinds of sentences (behaviors) could be generated.

Computation, Cognitive Science, and Connectionism

Computation offers a way to describe and manipulate the resulting measurements once we have symbolized a sequence of measurements. In this respect, computation can be thought of as one of the most powerful tools of cognitive science during its explosive growth over the last forty years. We can see that the rise of the generative enterprise in linguistics, information processing in computational psychology, and the symbolic paradigm of artificial intelligence all benefited from both the discrete measurement of the cognitive behavior of physical systems and the ability to universally simulate symbolic systems.

The rise of the "generative enterprise" over other descriptive approaches to linguistics can be attributed in part to its affinity with computation, since it initially appeared computationally feasible to generate and recognize natural languages according to formal grammars. A formal grammar is a collection of rewrite rules involving terminal symbols (those appearing in target language) and nonterminal symbols (temporary symbols used during the rewrite process). Chomsky (1957) proved that various constraints on rewrite rules produced sets of strings that could not be produced under other constraints. For instance, a grammar composed of rewrite rules involving nonterminals on the left-hand side of the productions and no more than a single nonterminal on the rightmost part of the productions (i.e., $A \rightarrow aB$, where A and B are nonterminal symbols and a is a terminal symbol) has less generative capacity than a grammar with rules whose right-hand sides can contain arbitrary sequences of terminal and nonterminals (such as $A \rightarrow aBb$). Generative capacity refers to the ability to mechanically produce more strings. The generative capacity of regular grammars is strictly less than that of context-free grammars, and both are strictly less than that of context-sensitive grammars. English readily provides several examples of center embedding that eliminate regular grammars from the possible set of mechanical descriptions of natural language grammaticality. Descriptions of natural languages based upon varieties of context-free phrase structure grammars, while very easy to work with, could not stand as correct models under such phenomena as featural agreement or crossed-serial dependencies. From a linguistic standpoint, any system capable of understanding and generating natural language behavior must exhibit context-sensitive generative capacity, though it is widely held that a class known as "indexed context free" is consistent with the weak generative capacity of natural languages (Joshi et al. 1989).

Psychologists, discouraged by behavioristic accounts of human performance, could now turn discretized protocols into working models of cognitive processing driven by internal representations. For example, Newell and Simon’s (1962) Generalized Problem Solver (GPS) implemented means-ends analysis and could model intelligent behaviors like planning. The operators used in GPS were nothing more than productions and the entire system could be viewed as a production system. Production systems are computationally equivalent to Turing machines and other universal computational mechanisms. Based upon this, Newell and Simon (1976) concluded that intelligent action is best modelled by systems capable of universal computation. This claim is now known as the Physical Symbol System Hypothesis and specifically states that a physical system implementing a universal computer is both necessary and sufficient for the modeling of intelligent behavior. Thus, context-sensitivity is not enough for intelligence, as in the case of linguistic competence, but the full computational power of recursive systems must be engaged for the production of intelligent behavior.

These behaviors are easily generated by universal frameworks such as unrestricted production systems or stored program computers. This unconstrained flexibility fueled the explosion in AI, where bits and pieces of cognitive action, such as chess playing, or medical diagnosis, could be converted into effective symbolic descriptions simulated by algorithms in computers. While flexibility is an asset in the development of computer software, unconstrained computational descriptions can not help us develop a theory of cognitive processing. More often than not, we are left with a fragile model which overfits the data and fails to generalize.

The scientific problem regarding the lack of constraints offered by general information processing systems has fueled the recent “back to basics” reaction in using more “neurally plausible” means of modeling. Connectionism has been a vigorous research area expanding in its scope throughout the 1980’s, bounded both by the underconstraints of artificial intelligence and information processing psychhology and by the overconstraints of computational neuroscience. Connectionism seeks to understand how elements of cognition can be based on the physical mechanisms that can be in the brain, without being constrained by the overwhelming detail of neuroscience. As such, it is constantly battered both from below (e.g., Grossberg, 1987), on actual biologically plausibility of the mechanisms, and from above (e.g., Fodor & Pylyshyn, 1988), on the adequacy of its mechanisms when faced with normal tasks of high-level cognition requiring structured representations (Pollack, 1988) and generative complexity.

Our earlier work tried to address the generative capacity issue raised long ago by Chomsky. In this work we examined biologically plausible iterative systems and found that a particular construal, the “dynamical recognizer,” resulted in recurrent neural network automata that had finite specifications and yet infinite state spaces (Pollack, 1991). From this observation we hypothesized that yet another mapping might be found between the hierarchy of formal languages and increasingly intricate dynamics of state spaces (implemented by recurrent neural networks). We were encouraged by Crutchfield and Young’s (1989) paper (summarized below) and similar conjectures regarding the emergence of complex computation in cellular automata as reported in the work of Wolfram (1984) and Langton (1990). After many attempts to reconcile our recurrent neural network findings with both the dynamical systems results and a traditional formal language view, we came to believe that the difficulty of our endeavor lay in the traditional view that a particular

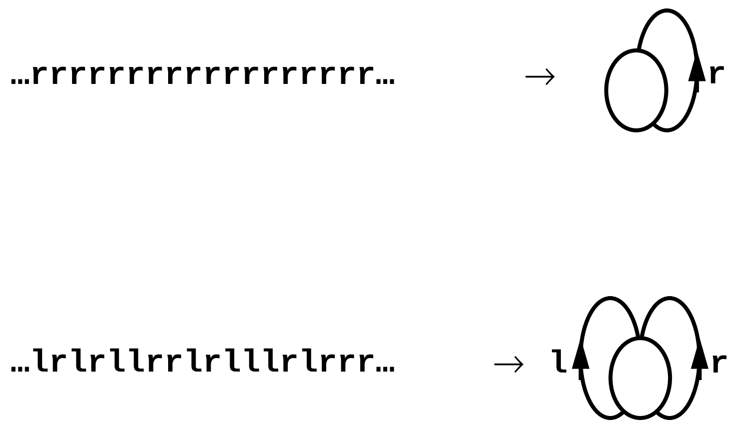

FIGURE 1. Finite state descriptions of equivalent complexity. The first subsequence is from the sequence of all r’s. The second subsequence is from a completely random sequence. Both sequences could be generated by a single state generator since each new symbol is independent from all other preceding symbols.

mechanical framework adheres to a particular generative class.

The Emergence Of Complex Behavior In Physical Systems

The two notions of complexity–complexion and generative class–have been traditionally applied only to computational systems. However, recent work by Crutchfield and Young (1989) suggests that one may be able to talk similarly about the generative class of a physical process. Their work focuses on the problem of finding models for physical systems based solely on measurements of the systems’ state. Rather than assuming a stream of noisy numerical measurements, they explore the limitations of taking very crude measurements. The crudest measurement of state is periodic sampling with a single decision boundary: either the state of the system is to the left or to the right of the boundary at every time step. Unlike numerical measurements that can be described mathematically, the binary sequence they collect requires a computational description (i.e. what kind of automaton could have generated the sequence?) They claim that the minimal finite state automaton induced from this sequence of discrete measurements provides a realistic assessment of the intrinsic computational complexity of the physical system under observation. To test this claim, Crutchfield and Young generated binary sequences from nonlinear dynamical systems such as the iterated logistic and tent maps. These systems have the property that infinitesimally small changes in a single global parameter can cause qualitative behavioral changes, known as period doubling bifurcations, where the behavior of the system in the limit moves from a period n oscillation to a period $2n$ oscillation. In addition to the claim stated above, their paper provides three key insights into the problem of recognizing complexity as it arises in nature.

First, the minimality of the induced automaton is important. Crutchfield and Young propose that the minimal finite state generator induced from a sequence of discrete measurements of a system provides a realistic judgment of the complexity of the physical system under observation. Minimality creates equivalence classes of descriptions based on the amount of *structure* contained in the generated sequence. Consider two systems–the first constantly emits the same symbol, while the second generates a completely random sequence of two different symbols. Both systems can be described by one-state machines that can generate subsequences of the observed sequences (Figure 1). In the constant case, the machine has a single transition. The random sequence, on the other hand, has a single state but two stochastic transitions. The ability to describe these sequences with single state generators is equivalent to saying that any subsequence of either sequence will provide no additional information concerning the next symbol in the sequence. Thus, total certainty and total ignorance of future events are equivalent in this framework of minimal induced description.

Second, they show that physical systems with limit cycles produce streams of bits that are apparently generated by minimal finite state machines whose complexion increases with the period of the cycle. A system with a cycle of period two is held to be as complex as a two-state machine. Systems with constrained ergodic behavior exhibit similar levels of complexion; the number of induced states is determined by the regularities in the subsequences that cannot be generated. These are shown schematically in Figure 2.

Third, Crutchfield and Young proved that the minimal machines needed to describe the behavior of simple dynamical systems *when tuned to criticality* had an infinite number of states. At criticality, a system displays unbounded dependencies of behavior across space and/or time (Schroeder, 1991). The spread of forest fires at the percolation threshold of tree density (Bak et al., 1990) and sand pile avalanches (Bak and Chen, 1991) both produce global dependencies at critical parameters of tree density and pile slope. Even simple systems, such as the iterated logistic function $x' = rx(1-x)$, exhibit criticality for an infinite set of r parameter values between zero and four. Crutchfield and Young proved that the computational behaviors at these parameter settings are not finitely describable in terms of finite state machines, but are compactly described by indexed context-free languages.

Apparent Complexion

The number of states in the systems studied by Crutchfield and Young can be selected by an external control parameter as the system bifurcates through various dynamical regimes. The task of merely increasing the number of apparent states of a system seems uniteresting because the simplest solution lies in being more sensitive to distinct states. Since we can arbitrarily zoom into any physical system, any object, including a rock, can simultaneously have a description requiring only a single state and descriptions with high complexion driven by atomic level motions. Figure 3 shows effects of increasing measurement granularity on the finite state machines induced from a dynamical system. We have selected the iterated mapping $x_{t+1} = 2x_t(\mathrm{mod}\ 1)$, also known as the Baker’s shift, for this demonstration. The behavior of this iterated system is to shift the bits of a binary encoding of the state, x_t at time t to the left by one place and then discard the bits to the left of the decimal point ($x_1 = 0.110101..._2$, $x_2 = 0.10101..._2$.) The first automaton

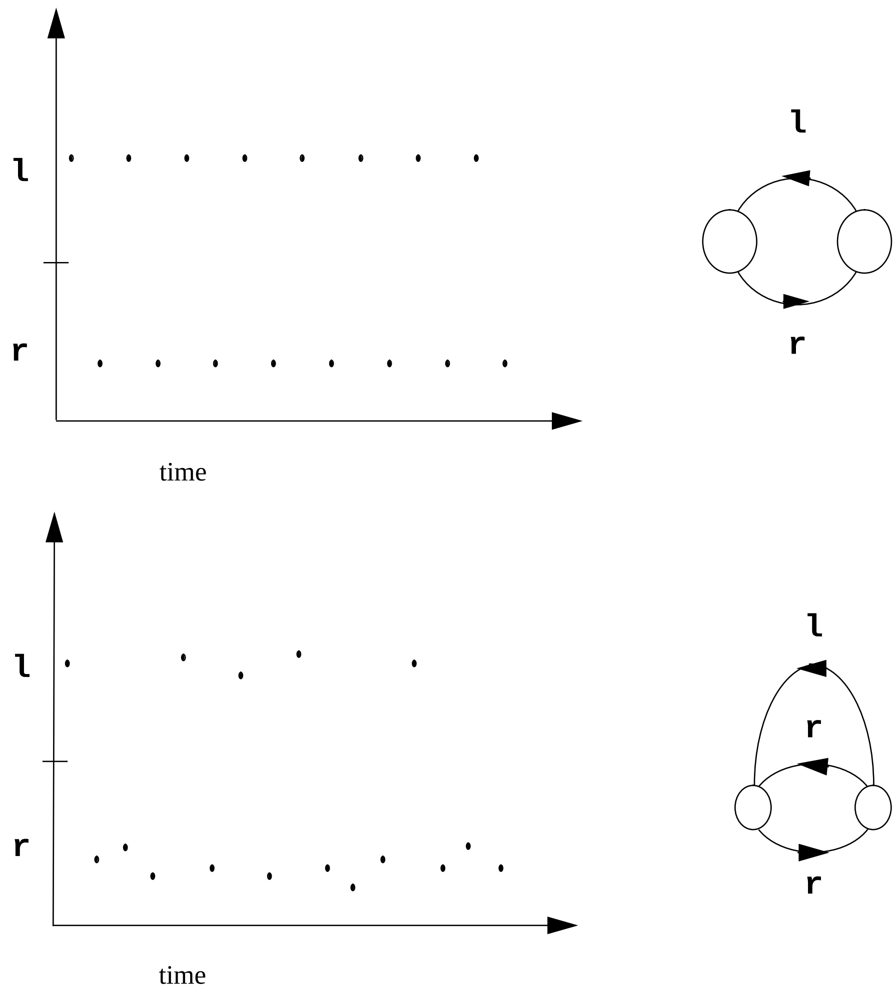

FIGURE 2. The state machines induced from periodic and chaotic systems. Note that the lower system does not produce **ll** pairs. This omission is the reason for the increase in number of states over the random generator in Figure 1.

was constructed by dividing the state space into two equal regions. This division results in a one-state machine that stochastically emits **0**'s and **1**'s with equal probability. The same trajectory, subjected to a measurement mechanism sensitive to three disjoint regions induces a three-state automaton. When four measurement regions are imposed on the state space, the resulting symbol sequence could be generated by the two-state machine at the bottom of Figure 3. Since an odd number of divisions will induce state machines with a corresponding number of states, an infinite number of finite state automata can be induced from the Baker's shift dynamical system.

Other scientists and philosophers have explored this route to complexity. Putnam (1988) has proved that an open system has sufficient state generative capacity to support arbitrary finite state interpretations. His core argument relies on *post hoc* labeling of state space to accommodate an

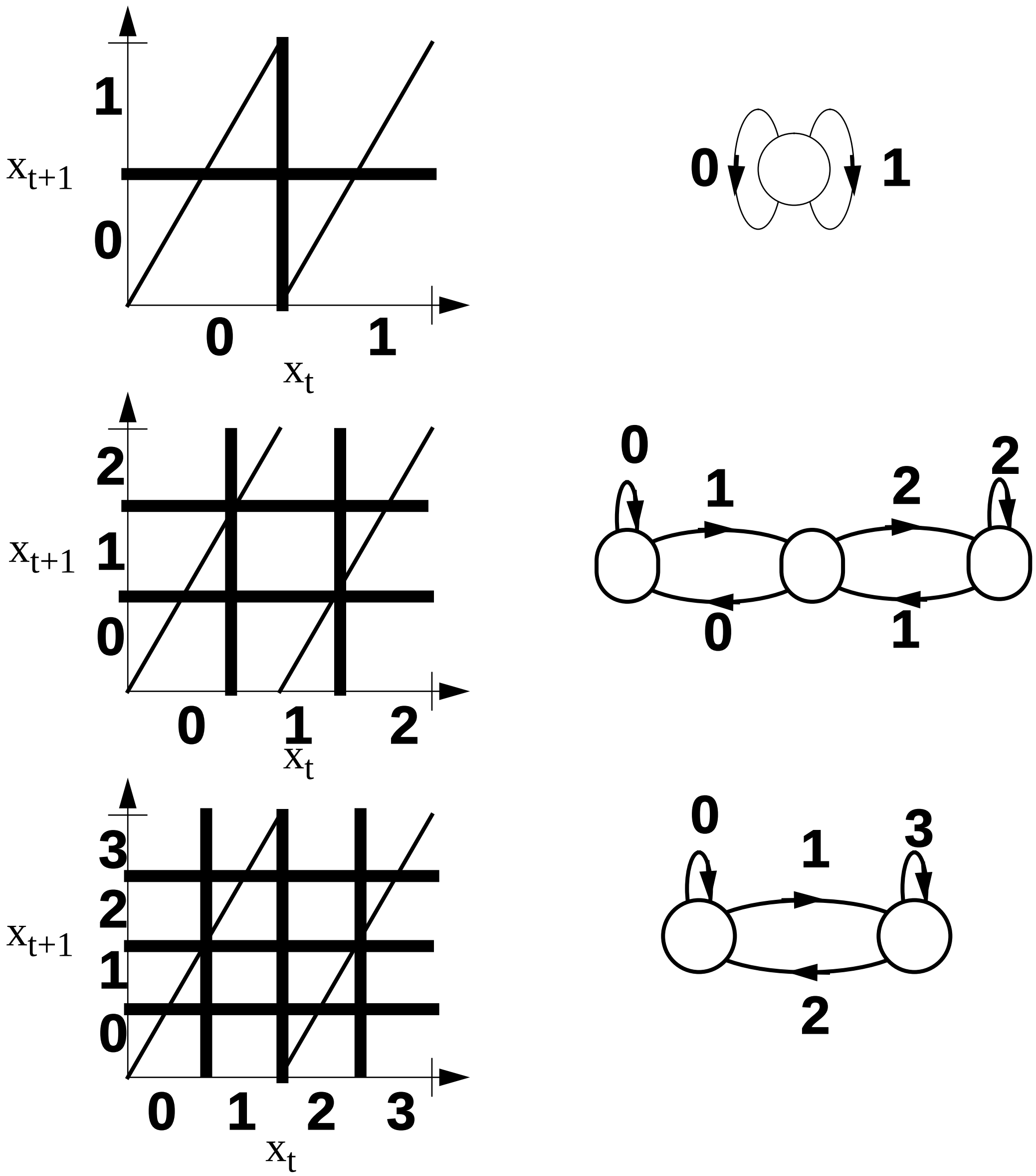

FIGURE 3. An illustration of an increase in the number of internal states due to explicit symbolization. The underlying mapping is $x_{t+1} = 2x_t \mathrm{mod}\ 1$. The x_t and x_{t+1} axes in the graphs range from 0 to 1.

arbitrary mapping between trajectories in state space and automaton states. Searle (1990) questions the relevance of computational models to cognition by claiming that a properly tuned perceptual mechanism could experience the WordStar word processing program in operation on the surface of the wall behind him. Based upon this observation, Searle concludes that causality, blatantly missing from the state transitions in his example, is the key component of cognition (Searle, 1993). Fields (1987), on the other hand, suggests that the arbitrary nature of state labellings is only a problem for classical systems, i.e., systems unaffected by observations of their state. He claims that when observations are made of nonclassical systems, the interaction between observer and system limits the granularity of observation and thus prevents the observer from drawing arbitrary computational inferences from the behavior of the system.

Finally, recall that Ashby (1956) points out that variable selection, which underlies the notion of a system, is critical since any physical process can provide an infinitude of variables that can be combined into any number of systems. This applies on the one hand to arguments of Searle and Putnam. An attribution of computational behavior to a process rests on an isomorphism between states of the physical process and the information processing states of the computational model. The intended computational model, thus, guides the selection of measured quantities of the physical process. In addition, a modeler must measure the current state of a process with sufficient resolution to support the isomorphism. In models capable of recursive computation, the information processing state can demand unbounded accuracy from the modeler's measurements.

On the other hand, the work in dynamical systems, by Crutchfield and Young, relates the problem with unbounded accuracy to the issue of sensitivity to initial conditions. In other words, significant "state" information can be buried deep within a system's initial conditions, and become widely distributed in the state. This implies that often the best way to "measure" a system's complexity is to simply observe its behavior over long periods of time and retroactively determine the critical components of the state. For instance, Takens (1981) method of embedding a single dimensional measurement into a higher dimensional space can provide a judgment of the underlying system's dimensionality which is independent of the dimension of the observable. Crutchfield and Young extend this philosophy to the computational understanding of systems, and infer the generative complexity of a process from long sequences of measurements.

Apparent Complexity

Both Putnam and Searle avoided the bulwark engineered by Chomsky, namely the issue of generative complexity classes. Is generative complexity also sensitive to manipulation of the observation method? The answer, surprisingly, is yes. To support this claim, we will present some simple systems with at least two computational interpretations: as a context-free generator and a context-sensitive generator. A rigorous explanation of the following arguments can be found in the appendix.

Consider a point moving in a circular orbit with a fixed rotational velocity, such as the end of a rotating rod spinning around a fixed center, a white dot on a spinning bicycle wheel, or an electronic oscillator. We measure the location of the dot in the spirit of Crutchfield and Young, by periodically sampling the location with a single decision boundary (Figure 4). If the point is to the left of the boundary at the time of the sample, we write down an "**l**". Likewise, we write down an

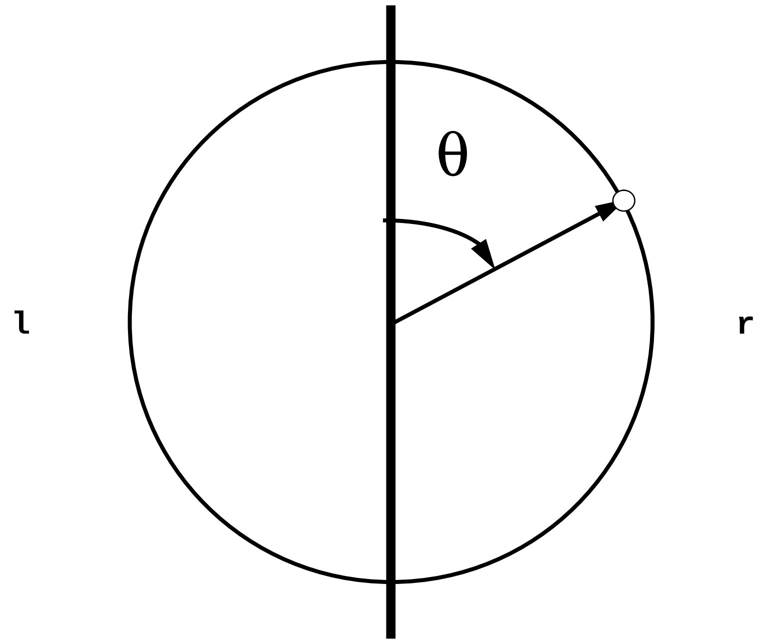

FIGURE 4. Decision regions that induce a context-free language. θ is the current angle of rotation. At the time of sampling, if the point is to the left (right) of the dividing line, an **l** (**r**) is generated.

“**r**” when the point is on the other side.[5] In the limit, we will have recorded an infinite sequence of symbols containing long sequences of **r**’s and **l**’s.

The specific ordering of symbols observed in a long sequence of multiple rotations is dependent upon the initial rotational angle of the system. The sequence does, however, possess certain recurring structural regularities which we call *sentences*. A sentence in this context is a run of **r**'s followed by a run of **l**'s. For a fixed rotational velocity (rotations per time unit) and sampling rate, the observed system will generate sentences of the form $\mathbf{r}^n\mathbf{l}^m$ ($n, m > 0$). (The notation $\mathbf{r}^n$ indicates a sequence of n **r**’s.) That is to say, each rotational velocity specifies at most three sentences, and the number of **r**’s and **l**’s in each sentence differ by at most one. These sentences repeat arbitrarily according to the initial conditions of the rotator. Thus, a typical subsequence of a rotator that produces sentences $\mathbf{r}^n\mathbf{l}^n$, $\mathbf{r}^n\mathbf{l}^{n+1}$, $\mathbf{r}^{n+1}\mathbf{l}^n$ looks like the line below. Individual sentences have been underlined for clarity.

$$\ldots\underline{\mathbf{r}^n\mathbf{l}^{n+1}}\,\underline{\mathbf{r}^n\mathbf{l}^n}\,\underline{\mathbf{r}^n\mathbf{l}^{n+1}}\,\underline{\mathbf{r}^{n+1}\mathbf{l}^n}\,\underline{\mathbf{r}^n\mathbf{l}^n}\,\underline{\mathbf{r}^n\mathbf{l}^{n+1}}\ldots.$$

A language of sentences may be constructed by examining the families of sentences generated by a large collection of individuals, much like a natural language is induced from the abilities of its individual speakers. In this context, a language could be induced from a population of rotators with different rotational velocities where individuals generate sentences of the form $\{\mathbf{r}^n\mathbf{l}^n, \mathbf{r}^n\mathbf{l}^{n+1}, \mathbf{r}^{n+1}\mathbf{l}^n\}$, $n > 0$. The resulting language can be described by a context-free grammar and has unbounded dependencies; the number of **l**’s is related to the number of preceding **r**’s. These two constraints on the language imply that the induced language is context-free.

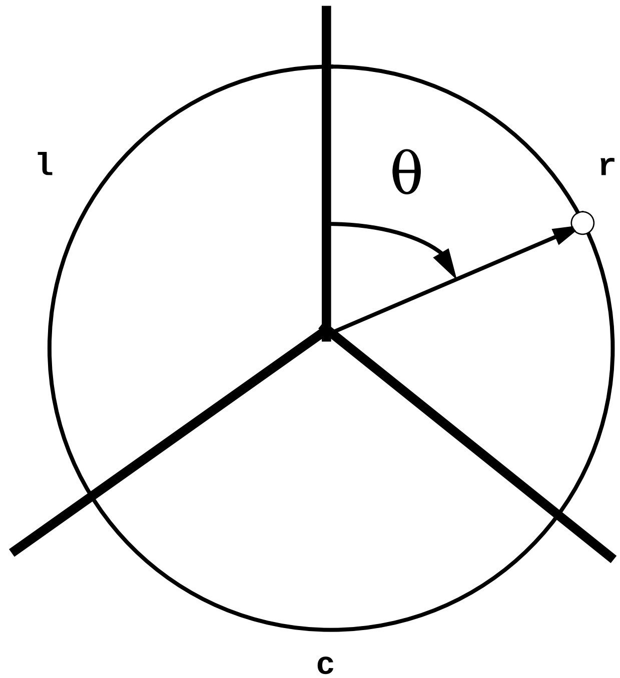

FIGURE 5. Decision regions that induce a context-sensitive language.

Now we show that this complexity class assignment is an artifact of the observational mechanism. Consider the mechanism that reports three disjoint regions covering equal angles of rotation: **l**, **c**, and **r** (Figure 5). Now the same rotating point will generate sequences of the form

rr…rrcc…ccll…llrr…rrcc…ccll…ll….

For a fixed sampling rate, each rotational velocity specifies no more than seven sentences, $\mathbf{r}^n\mathbf{c}^m\mathbf{l}^k$, where n, m, and k can differ no by no more than one. Again, a language of sentences may be constructed containing all sentences where the number of **r**'s, **c**'s, and **l**'s differs by no more than one. The resulting language is context-sensitive since it can be described by a *context-sensitive grammar* and cannot be context-free because it is the finite union of several context-sensitive languages related to $\mathbf{r}^n\mathbf{c}^n\mathbf{l}^n$.

Therefore, a single population of rotators with different rotational velocities exhibited sentential behavior describable by computational models from different generative classes, and the class depended upon the observation mechanism. The two languages observed in the family of rotators can also be observed in the dynamics of a single deterministic system. A slow-moving chaotic dynamical system controlling the rotational velocity parameter in a single system can express the same behavior as a population of rotators with individual rotational velocities. The equations below describe a rotating point with Cartesian location (x, y) and a slowly changing rotational angle θ controlled by the subsystem defined by w, z, and ω.

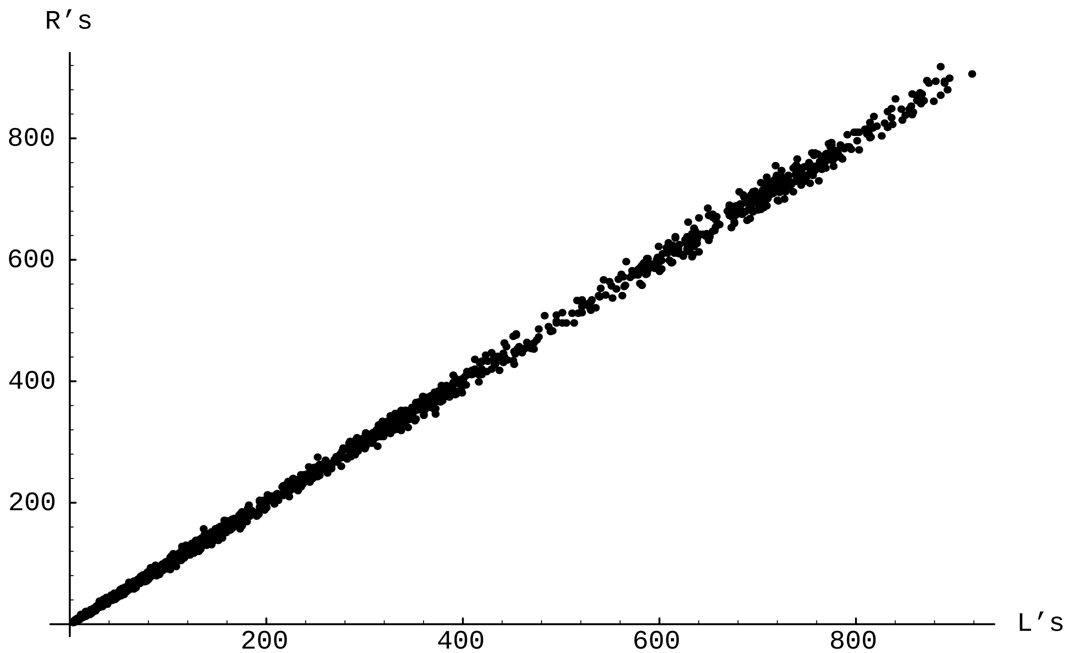

FIGURE 6. The two symbol discrimination of the variable rotational speed system.

$$x = \tanh(x - \theta \tanh y)$$
$$y = \tanh(y + \theta \tanh x)$$
$$w = 4w(1 - w)$$
$$z = \frac{4}{5}y + \frac{1}{5}w$$
$$\omega = \omega\left(1 + \omega\left(w - \frac{1}{2}\right)\left(\frac{1}{2} - \frac{1}{2}\tanh 5z^2\right)\right)$$
$$\theta = \frac{1}{5}\theta + \frac{4}{5}\omega$$

This system slowly spirals around the origin of the (x, y) plane. The value of w is a chaotic noise generator that is smoothed by the dynamics of z and rotational velocity of ω.

As before, we construct two measurement mechanisms and examine the structures in the generated sequences of measurements. The first measurement device outputs an **r** if x is greater than zero, and an **l** otherwise. From this behavior, the graph in Figure 6 plots the number of consecutive **r**'s versus the number of consecutive **l**'s. The diagonal line is indicative of a context-free lan-

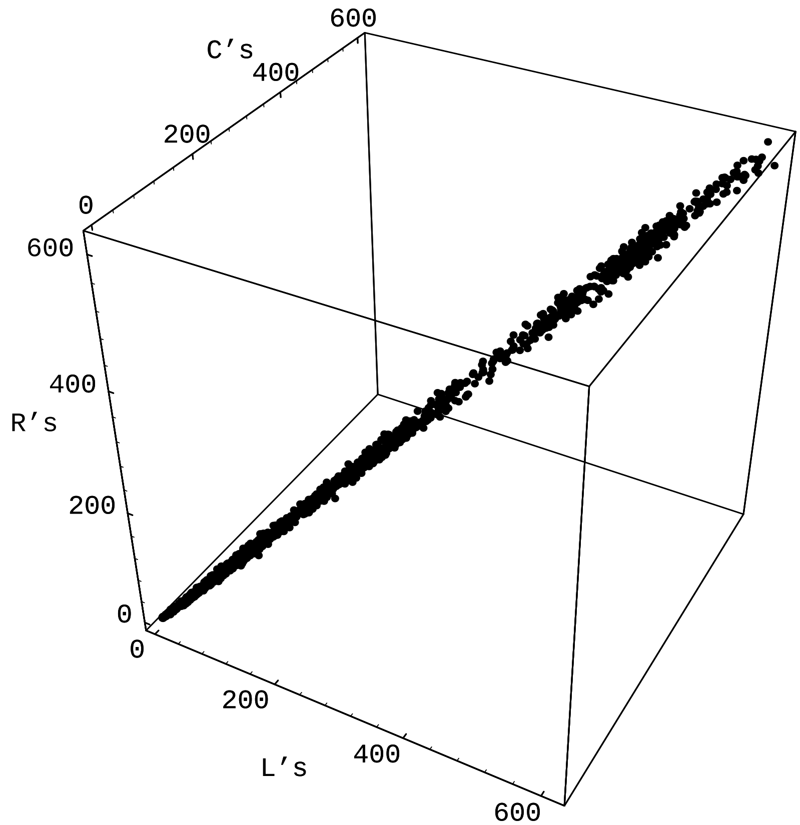

FIGURE 7. The three symbol discrimination of the variable system.

guage as a simple corollary to the pumping lemma for regular languages (Hopcroft and Ullman, 1979).

If the underlying language is regular then according to the pumping lemma one would expect to find pumped revisions of $\mathbf{r}^n\mathbf{l}^n$, i.e., there exists some assignment of u, v, and w such that $uvw = \mathbf{r}^n\mathbf{l}^n$ that indicates that the set of strings uv^iw, for $i > 0$, is also in the language. Since the graph records the number of consecutive **r**’s versus the number of consecutive **l**’s, the uv^iw relationship constrains straight lines in the graph to be either vertical, as in the case of v being all **l**’s, or horizontal, as in the case of v being all **r**’s. If v is a string of the form $\mathbf{r}^a\mathbf{l}^b$, then the graph would not contain any straight lines. A formal proof appears in the appendix.

When the granularity of the measurement device changes from two regions to three, we see a parallel change in the class of the measurement sequence from context-free to context-sensitive. Figure 7 shows the relationship between the number of consecutive **r**'s, consecutive **c**'s, and consecutive **l**'s. As in the previous case, one can interpret the diagonal line in the graph as the footprint of a context-sensitive generator.

Discussion: The Observers’ Paradox

The preceding example suggests a paradox: the variable speed rotator has interpretations both as a context-free generator and as a context-sensitive generator, depending upon measurement granularity. Yet how can this be if the computational complexity class is an inherent property of a physical system, like mass or temperature? What attribute of the rotator is responsible for the generative capacity of this system? There is no pushing and popping nonterminal symbols from an explicit stack. The rotator, however, does have a particular invariant: equal angles are swept in equal time. By dividing the orbit of the point into two equal halves we have ensured that the system will spend almost the same amount of time in each decision region. This constraint “balances the parentheses” in our measurement sequence. One may argue that the rotational velocity and the current angle together implement the stack and therefore claim that the stack is really being used. Such an argument ignores the properties of the stack, namely the ability to arbitrarily push and pop symbols. Claims regarding an internal Turing machine tape are similarly misguided.

The decision of the observer to break the infinite sequence of symbols into sentences can also affect the complexity class. Similar arguments for sentences of the form $\mathbf{r}^{n}\mathbf{l}^{n+a}\mathbf{r}^{n+b}$, $(|a|, |b| < C)$ gives rise to a context-sensitive language. From this perspective, we can see that Crutchfield and Young biased the languages they found by assuming closure under substrings. The assumption “if string x is in language L then all substrings of x are also in L” affected the induced minimal automata and criticality languages.

In other words, **the computational complexity class cannot be an intrinsic property of a physical system**: it emerges from the interaction of system state dynamics and measurement as established by an observer. The complexity class of a system is an aspect of the property commonly referred to as computational complexity, a property we define as the union of observed complexity classes over all observation methods. This definition undermines the traditional notion of system complexity, namely that systems have unique well-defined computational complexities. Consider the case of a sphere painted red on one side and blue on the other. Depending upon the viewing angle, an observer will report that the ball is either red, blue, or dual colored. It is a mistake, however, to claim that “redness”, in exclusion of “blueness”, is an intrinsic property of the ball. Rather, “color” is a property of the ball and “redness” and “blueness” are mere aspects of this property.

An observation such as the one described in this paper should not be surprising considering the developments in physics during the first half of this century. The observation methods described above can select computational complexity in the same manner that observation of a quantum phenomenon collapses its wave function of possible values. Specifically, the wave/particle duality of radiation is an example of how observation can affect the apparent properties of a system. Depending upon experimental setup, a beam of light can either display wave-like diffraction or particle-like scattering.

As shown above, strategic selection of measurement devices can induce an infinite collection of languages from many different complexity classes. For a single physical system, the choice of method and granularity of observation also “selects” the computational complexity of a physical system.

Conclusion

The goal of most modelers in cognitive science has been to build computational models that can account for the discrete measurements of input to and output from a target system. The holistic combination of the organism and symbolizing observer can create apparent computational systems independent of the actual internal behavior producing processes. Our examples show that the resulting computational model embodies an apparent system that circumscribes the target processes and the measurement devices. The apparent system has apparent inputs and outputs given by the symbolic inputs to and outputs from the computational model. For both input and output, the one-to-many mappings entailed by symbolization are not unique and can have simultaneous multiple symbol reassignments depending upon the observer. As the multiple interpretations of the rotator shows, these reassignments can change the computational complexity class of this apparent system.

We believe the results described above have relevance for cognitive science. Recall that both the Physical Symbol System Hypothesis and generative linguistics rest on an underlying assumption of the intrinsic nature of computational complexity classes. It suggests, on the surface, the irrelevancy of the hierarchy of formal languages and automata as accounts of complexity in physical systems. At a deeper level, it implies that we cannot know the complexity class of the brain's behavior without establishing an observer since the brain itself is a physical system. Thus the Physical Symbol System Hypothesis relies on an unmentioned observer to establish that an ant following a pheromone trail is not computational while problem-solving by humans is. The necessary and sufficient conditions of universal computation in the Physical Symbol System Hypothesis provide no insight into cognitive behavior; rather, it implies that humans can write down behavioral descriptions requiring universal computation to simulate.

Even the computational intractability of models of linguistic competence (e. g., Barton, et al., 1987) is dependent on a particular symbolization of human behavior, not an underlying mechanical capacity. This highlights the groundless nature of rejections of mathematical models solely on claims of insufficient computational complexity. Our work suggests that alternative mechanisms and formalisms that exhibit apparent complexities of the sort attributed to the "language module" should also be explored.

As our ability to establish good measurements has increased, we now know that there are many areas in nature where unbounded dependencies and systematic forms of recursive structuring occur. The genome code, the immunological system, and botanical growth are but a few examples that are proving as complex as human languages. Physics was able to accept wave/particle duality as a product of observation. It is only cognitive science that presumes the "specialness" of language and human mental activity to justify a different set of scientific tools and explanations based upon the formal symbol manipulation capacity of computational models. To truly understand cognition, we may have to stop relying on symbolic models with fixed complexity classes and turn to explanations whose apparent complexity matches our observations.

Acknowledgments

A earlier version of this paper appears in *The Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society.* This work was supported by Office of Naval Research grant N00014-92-J-1195. We thank all those who have made comments and suggestions on the multiple drafts of this paper, including the various members of the LAIR (Angeline, Becker, Large, Nguyen, Parekh, Saunders, and Stucki), ConnectFest III participants, and Paul Smolensky. We also appreciated the constructive suggestions of the reviewers, especially the colored ball foil supplied by Eric Deitrich.

Notes

1 Cage (1969, p. 7)
2 Chomsky (1957, p. 24).
3 Newell and Simon (1976, p. 116).
4 This becomes crucial when trying to measure an apparent continuous quantity like temperature, velocity, or mass. Recording continuous signals simply postpones the eventual discretization. Rather than measuring the original event, one measures its analog.
5 The probability of the point landing on the boundary is zero and can arbitrarily be assigned to either category without affecting the results below.

References

Aida, S., et al. (1984) *The Science and Praxis of Complexity* (Tokyo: United Nations University).

Ashby, W. R. (1956) *An Introduction to Cybernetics.* (London: Chapman and Hall).

Bak, P., & Chen, K. (1991) Self-organized criticality, *Scientific American*, 46-53.

Bak, P., Chen, K., & Creutz, M. (1990) A forest-fire model and some thoughts on turbulence, *Physics Letters*, 147:297-300.

Barton, G. E., Jr., Berwick, R. C., and Ristad, E. S. (1987) *Computational Complexity and Natural Language* (Cambridge, Mass.: MIT Press).

Cage, J. (1969) *A Year From Monday: New Lectures and Writings by John Cage*. (Middletown, Connecticut. Wesleyan University Press).

Chomsky, N. (1965) *Aspects of the Theory of Syntax* (Cambridge, Mass.: MIT Press).

Chomsky, N. (1957) *Syntactic Structures* (The Hague: Mounton & Co.).

Crutchfield, J., & Young, K. (1991) Computation at the onset of chaos. In W. Zurek (Ed.) *Entropy, Complexity, and the Physics of Information (*Reading: Addison-Wesely).

Fields, C. (1989) Consequences of nonclassical measurement for the algorithmic description of continuous dynamical systems. *Journal of Experimental and Theoretical Artificial Intelligence*, 1:171-178.

Fodor, J. A., & Pylyshyn, Z. W.(1988) Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28:3-71.

Grossberg, S. (1987) Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11:23-63.

Hopcroft, J. E., and Ullman, J. D. (1979) *Introduction to Automata Theory, Languages, and Computation* (Reading: Addison-Wesely).

A. K. Joshi, K. Vijay-shanker, & D. J. Weir, (1989) Convergence of mildly context-sensitive grammar formalisms. In T. Wasow, P. Sells (eds) *The Processing of Linguistic Structur*e.

Cambridge: MIT Press.

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

Newell, A., and Simon, H. A. (1962) GPS--A program that simulates human thought. In E. A. Feigenbaum & J. Feldman (Eds.) Computers and Thought. (New York: McGraw-Hill).

Newell, A., and Simon, H. A. (1976) Computer Science as Empirical Inquiry: Symbols and Search. *Communications of the ACM*, 19:3.

Pollack, J. B. (1991) The induction of dynamical recognizers, *Machine Learning*, 7:227-252.

Putnam, H. (1988) *Representation and Reality* (Cambridge, Mass.: MIT Press).

Schroeder, M. (1991) *Fractals, Chaos, Power Laws* (New York: W. H. Freeman and Company).

Searle, J. (1990) Is the brain a digital computer? *Proceedings and Addresses of the American Philosophical Association*, 64:21-37.

Searle, J. (1993) *The Rediscovery of the Mind*. (Cambridge, Mass.: MIT Press).

Takens, F. (1981) Detecting strange attractors in turbulence. *Lecture Notes in Mathematics 898*. (Berlin: Springer Verlag).

Wolfram, S. (1984) Universality and complexity in cellular automata, *Physica*, 10D:1-35.

Appendix

This appendix contains both the formal derivation of the existence of context-free and context-sensitive interpretations of the rotator, and a discussion regarding the inductions of language classes from graphs of the relationship between runs of symbols.

The Rotator

The derivation of the languages observed in the rotator are described here. First, we show that the two region case produces a context-free language. The derivation of the three region case parallels this derivation.

The specific ordering of symbols in a long sequence of multiple rotations is dependent upon the initial, assumed random, rotational angle of the system. For a fixed rotational velocity (rotations per time unit), ω, and sampling rate, s, $(\omega < 2s)$ the observed system will generate sentences of the form $\{\mathbf{r}^n \mathbf{l}^{m-n}\}$, where $n = \left\lfloor \frac{s}{2\omega} - \phi \right\rfloor$, $m = \left\lfloor \frac{s}{\omega} - \phi \right\rfloor$, and ϕ is a random variable such that $0 \le \phi < 1$. The value of ϕ embodies the slippage of n and m due to incommensurate rotational velocities and sampling rates. If s is an integer multiple of ω, then no matter the value of ϕ then values of n and m will be constant. If s is an irrational multiple of ω, then the current value of ϕ will produce minor (no more than 1) variation in the value of n and m.

For a fixed sampling rate, each rotational velocity specifies up to three sentences, $L(\omega) \subseteq \{\mathbf{r}^n \mathbf{l}^n, \mathbf{r}^n \mathbf{l}^{n+1}, \mathbf{r}^{n+1} \mathbf{l}^n | n = \left\lfloor \frac{s}{2\omega} - \phi \right\rfloor\}$ that repeat in an arbitrary manner according to the divisibility of s and ω. We can then induce a language from a *population* of rotators with different rotational velocities, thus $L_2 = \bigcup_{\omega \in (0,s)} L(\omega)$. L_2 contains all sentences of the form $\{\mathbf{r}^n\mathbf{l}^n, \mathbf{r}^n\mathbf{l}^{n+1}, \mathbf{r}^{n+1}\mathbf{l}^n\}$, $n > 0$. The resulting language can be described by the *context-free grammar*

$$(\{S\},\{\mathbf{r},\mathbf{l}\},S,\{S\text{->}\mathbf{r},S\text{->}\mathbf{l},S\text{->}\mathbf{rl},S\text{->}\mathbf{r}S\mathbf{l}\}).$$

No regular grammar can describe this language due to the unbounded dependency between the number of r's and l's. Therefore L_2 is a context-free language.

The three region case is similar. For a fixed rotational velocity, ω, and sampling rate, s, $(\omega < 3s)$ the observed system will generate sentences of the form $\{\mathbf{r}^n\mathbf{c}^{m-n}\mathbf{l}^{k-n-m}\}$, where $n = \left\lfloor \frac{s}{3\omega} - \phi \right\rfloor$, $m = \left\lfloor \frac{2s}{3\omega} - \phi \right\rfloor$, $k = \left\lfloor \frac{s}{\omega} - \phi \right\rfloor$ and $0 \le \phi < 1$. For a fixed sampling rate, each rotational velocity specifies up to seven sentences, $\mathbf{r}^n\mathbf{c}^n\mathbf{l}^n$, $\mathbf{r}^n\mathbf{c}^n\mathbf{l}^{n+1}$, $\mathbf{r}^n\mathbf{c}^{n+1}\mathbf{l}^n$, $\mathbf{r}^{n+1}\mathbf{c}^n\mathbf{l}^n$, $\mathbf{r}^n\mathbf{c}^{n+1}\mathbf{l}^{n+1}$, $\mathbf{r}^{n+1}\mathbf{c}^n\mathbf{l}^{n+1}$, $\mathbf{r}^{n+1}\mathbf{c}^{n+1}\mathbf{l}^n$. Let L_3 equal the union of all sentences generated from the rotational velocities in $(0, s)$. As before, $L_3 = \bigcup_{\omega \in (0,s)} L(\omega)$. L_3 contains all sentences of the form $\mathbf{r}^n\mathbf{c}^n\mathbf{l}^n$, $\mathbf{r}^n\mathbf{c}^n\mathbf{l}^{n+1}$, $\mathbf{r}^n\mathbf{c}^{n+1}\mathbf{l}^n$, $\mathbf{r}^{n+1}\mathbf{c}^n\mathbf{l}^n$, $\mathbf{r}^n\mathbf{c}^{n+1}\mathbf{l}^{n+1}$, $\mathbf{r}^{n+1}\mathbf{c}^n\mathbf{l}^{n+1}$, and $\mathbf{r}^{n+1}\mathbf{c}^{n+1}\mathbf{l}^n$, where $n > 0$. The resulting language can be described by the *context-sensitive grammar*

$$(\ \{X, Y, Z\}\ ,\ \{r, c, l\}\ , X$$
$$\{X \to aXY, X \to aZ, ZY \to bZc, cY \to Yc, Z \to bc\}\ \ )$$

Since L_3 is the finite union of several context-sensitive language related to $\mathbf{r}^n\mathbf{c}^n\mathbf{l}^n$, no context-free grammar can describe this language. Therefore, L_3 is a context-sensitive language.

An Application of the Pumping Lemma

Determining if a language is context-free is hard enough when you have a mathematical description in front of you, but what do you do when you have to answer this question about a set of strings defined by a set of examples? To solve this problem in the context of this paper, we have focused on a particular structure found in a sample of strings, namely pairs of run lengths for symbols in the language. The regularities present in these strings allows us to rule out classes of automata capable of generating the observed strings.

The languages we examine in this paper possess two important properties:

Property 1: Each string has one run, or subsequence of the same symbol, for each symbol in the language.

Property 2: The runs are always in the same order.

Consider, for example, the string **rrrrlll**. This string is in the language due to Property 1 since it contains one run of **r**'s and one run of **l**'s, while **rrrrllllrr** is not in the language since it has two runs of **r**'s. According to Property 2, if **rrrrlll** is in the language, then **llll-**

rrrr is not.

The first language we are interested in comes from measurements based on two region. The strings in this language are from the strings defined by $\mathbf{r}^{n}\mathbf{l}^{m}$. One way of representing this language is by plotting points on a Cartesian grid where the (x,y) location is determined by the number of **r**'s and the number of **l**'s in each string. With help from the pumping lemma for regular languages, a few predictions can be made about this graph when $n + m$ is large, if the underlying language is regular. (The quantity $n + m$ is large when it is greater than the number of states in the minimal finite state generator for the language.) In this case, one would expect to find pumped revisions of $\mathbf{r}^{n}\mathbf{l}^{m}$; there exists some assignment of u, v, and w such that $uvw = \mathbf{r}^{n}\mathbf{l}^{m}$ in that the set of strings $uv^{i}w$, for $i > 0$, is also in the language. Since the graph plots number of consecutive **r**'s versus the number of consecutive **l**'s, the $uv^{i}w$ relationship constrains straight lines in the graph to be either vertical, as in the case of v being all **l**'s, or horizontal, as in the case of v being all **r**'s. The partition v can not be a string of the form $\mathbf{r}^{a}\mathbf{l}^{b}$, because $uv^{i}w$ would violate the general properties of our languages. Even in the general case, such a partition would lead to graphs without straight lines.

The straight line seen in the empirical data is *diagonal*. Because neither horizontal or vertical lines are present in the graph, we are forced to conclude that the underlying finite state generator for this language either has a very large number of states, or it simply does not exist. The former is ruled out if we assume that the diagonal structure extends to strings of every length.

A similar argument can be made against the context-freeness of the language produced by the three region measurement system. As before, the elements of the language are represented on a lattice according to the run lengths of their substrings. A three dimensional lattice, however, replaces the two dimensional lattice of the previous case. Likewise, constraints on straight lines emerge from the intersection of the set of strings predicted by the pumping lemma for context-free languages and set of strings described by $\mathbf{r}^{n}\mathbf{c}^{m}\mathbf{l}^{k}$. If the language is the product of a push down automata, one could see straight lines in which one or two dimensions are varied. But the major diagonal exhibited by the data from the rotator system implies either a large number of states in the push down automata or one does not exist. The former is ruled out again if we assume that the diagonal structure extends to strings of every length.