Computational complexity of symbolic dynamics at the onset of chaos

Porus Lakdawala*
Tata Institute of Fundamental Research, Homi Bhabha Road, Bombay 400005, India
(Received 7 August 1995)

In a variety of studies of dynamical systems, the edge of order and chaos has been singled out as a region of complexity. It was suggested by Wolfram, on the basis of qualitative behavior of cellular automata, that the computational basis for modeling this region is the universal Turing machine. In this paper, following a suggestion of Crutchfield, we try to show that the Turing machine model may often be too powerful as a computational model to describe the boundary of order and chaos. In particular we study the region of the first accumulation of period doubling in unimodal and bimodal maps of the interval, from the point of view of language theory. We show that in relation to the ''extended'' Chomsky hierarchy, the relevant computational model in the unimodal case is the nested stack automaton or the related indexed languages, while the bimodal case is modeled by the linear bounded automaton or the related context-sensitive languages.

PACS number(s): 05.45.+b

I. INTRODUCTION

The complex systems that we often observe, both in nature and otherwise, are characteristically poised in a delicate balance between the dullness of order and the randomness of disorder. In recent years a lot of effort has been expended in obtaining quantitative measures which would provide suitable definitions for this complexity.

In a study of the qualitative behavior of cellular automata, Wolfram [1] noticed that there was a class of automata, which he called class 4, whose members displayed complex dynamical behavior. In particular, they seemed to exhibit transients which were arbitrarily long-lived. Wolfram suggested that this behavior was reminiscent of the undecidability which characterized the halting problem for Turing machines. He conjectured that class 4 automata then might have a universal Turing machine embedded within them.

The model of the Turing machine has served, in the past, to provide a deeper understanding of phenomena which had been investigated through other means. The most notable among these is the development of the idea of algorithmic information which is the computational counterpart of the traditional Shannon entropy. These ideas have served to provide a quantitative basis for the qualitative notion of randomness. Conceived originally through the efforts of a number of people, it has shed light on some deep issues related to the foundations of mathematics, mainly through the work of Chaitin [2].

In [3] we tried to show that cellular automata, with Turing machines embedded within them, may not always display complex behavior. In our present work we try to demonstrate that the model of the Turing machine may at times be too powerful to describe the computational complexity at the edge of order and chaos. Crutchfield [4] has argued that instead of looking only at the Turing machine, it might be wiser to look at the entire hierarchy of machines that computation theory provides us with. Classical computation theory provides a beautiful framework in which models of machines or automata are related to the languages they recognize, which are in turn generated by their respective grammars. This hierarchy is known as the Chomsky hierarchy. The Turing machines form the top of this hierarchy.

The dynamical systems we study are the rather well understood, iterated maps of the interval. We investigate the onset of chaos which is exhibited by the first accumulation of period doubling for the case of the unimodal and bimodal families of maps. The symbolic dynamics and the kneading theory for these cases are well known [5,6]. We demonstrate that the language generated by the kneading sequences in these two cases can be recognized by machines which lie lower down in the Chomsky hierarchy.

II. THE SYMBOLIC DYNAMICS OF MAPS OF THE INTERVAL

In this section we set up the notation used by Mackay and Tresser [6] to describe the onset of chaos at the first accumulation of period doubling.

A. The unimodal case

A unimodal map is, by definition, a continuous map f of the interval $I=[0,1]$ into itself, which possesses a single turning point c. Normally f is chosen so that it monotonically increases in the interval $[0,c)$ and decreases monotonically in the interval $(c,1]$. With every point in I we associate a symbol L or R depending on whether it lies on the left or on the right of c. c is identified with the symbol C. In this way the orbit of any point in I under iteration by the map f can be associated with a sequence (as is done conventionally, an eventually periodic sequence of symbols will be described as a finite sequence by enclosing the periodic part in brackets with the symbol ''∞'' at the end as in $\cdots(\cdots)^{\infty}$) of symbols from the set $\Lambda=L,C,R$. The kneading sequence, which is defined to be the symbol sequence corresponding to the orbit of the point $f(c)$, is of particular importance in the symbolic dynamics. It has been shown in the pioneering work of Milnor and Thurston [7] that the kneading sequence controls the possible symbol sequences that can occur for a

*Electronic address: porus@theory.tifr.res.in

given map and that it determines important ergodic properties of the map, such as the topological entropy.

In order that we can define the symbol sequences of interest to us, we need mention here only a single detail from symbolic dynamics. Symbolic sequences can be ordered in such a way that they respect the ordering of the interval I, i.e., if $x,y\in I$ have symbol sequences $s(x)$ and $s(y)$, respectively, then

$$x\leqslant y \text{ if } s(x)\leqslant s(y).$$

This ordering is defined as follows.

First we define an ordering on the symbols as $L<C<R$. Now, if $A=Xa\ldots$ and $B=Xb\ldots$ are two sequences, for which X is the common prefix sequence and $a,b\in\{L,C,R\}$, $a\neq b$, then

$$A<B \text{ if } a<b \text{ and } X \text{ contains an even number of } R'\text{s}$$
$$\textbf{or } a>b \text{ and } X \text{ contains an odd number of } R'\text{s},$$

$$A>B \text{ otherwise}$$

It is well known that families of unimodal maps (like the well-known logistic family) exhibit the route to chaos through period-doubling transitions. The renormalization group theory pioneered by Feigenbaum [8] provides a beautiful description of this phenomenon. For our purposes it suffices to describe the kneading sequences that arise through successive applications of the renormalization group transformation. The limit sequence can then be thought of as the symbolic description of the onset of chaos, obtained through an accumulation of period-doubling transformations. The kneading sequences are most conveniently described by the so called $*$-operation, originally discovered by Derrida, Pomeau, and Gervois [9]. Given a word X on the symbols $\{L,C,R\}$, $R*X$ is obtained by applying the following transformations simultaneously on every symbol of X:

$$L\rightarrow RR,$$

$$R\rightarrow RL,$$

$$C\rightarrow RC.$$

To help us motivate the bimodal case that is treated in the next sub-section, however, we use a different algorithm, due to Mackay and Tresser [6], for generating these sequences. Consider words $X^{(i)}$, $i=1,2,\ldots$ defined as follows:

$$X^{(0)}=\varnothing,$$

$$X^{(n+1)}=X^{(n)}\overline{U}^{(n)}X^{(n)},$$

where the $U^{(n)}$ and $\overline{U}^{(n)}$ are symbols chosen from $\{L,R\}$ so that

$$X^{(n)}U^{(n)}<X^{(n)}C<X^{(n)}\overline{U}^{(n)}.$$

The words $K_n=(X^{(n)}C)^{\infty},n\geqslant 0$ are, then, the kneading sequences of maps, having a superstable periodic orbit of period 2^n. Moreover, the set of lengths of the periodic orbits of the map with the kneading sequence K_n is given by $\{1,2,4,\ldots,2^n\}$. Each of these maps have a finite, stable attracting set and hence display ordered dynamics.

We also define another set of kneading sequences which would be useful for later discussion,

$$K_n'=(X^{(n)}\overline{U}^{(n)}X^{(n)}U^{(n)}X^{(n)}\overline{U}^{(n)})^{\infty},\quad n\geqslant 0.$$

The sequence K_n' represents a map which contains an orbit of period 3×2^n. These maps have positive topological entropy and can, in this sense, be considered as displaying chaotic dynamical behavior.

The sequence $K_\infty=K_\infty'$ is the kneading sequence of the map at the onset of chaos. The set of periods of the periodic points of this map is given by $\{1,2,4,\ldots,2^n,\ldots\}$. The topological entropy in this case is zero.

For the details of these results, we refer the reader to the literature [5,6].

B. The bimodal case

We now describe the accumulation of period doubling in bimodal maps through their symbolic dynamics. Once again we refer the interested reader to [6] for the details.

A bimodal map of the interval I is a map f from I into itself with two turning points, c and k. In what follows we consider the case of the $+-+$ maps, i.e., maps which are monotonic increasing on $(0,c)$ and $(k,1)$, monotonic decreasing on (c,k). Each point in I is now associated with a symbol from the set $\{L,R,B\}$ depending on whether it belongs to the interval $(0,c)$, (c,k), or $(k,1)$, respectively. The points c and k are assigned the symbols C,K, respectively. The symbol sequence corresponding to the orbit of a point consists of a string of symbols from the set $\Sigma=\{L,C,R,K,B\}$. In this case, the kneading data of a map correspond to a pair of kneading sequences, which correspond to the symbol sequences of the pair of points $(f(c),f(k))$. As mentioned before, the kneading data uniquely determine some important properties of the map.

The definition of the ordering on symbol sequences for the unimodal case from the preceding section can be retained verbatim for the bimodal case, if we define the ordering on the elementary symbols as $L<C<R<K<B$. The parity of the common prefix of two symbol strings is still determined by the number of R's in it. This is due to the fact that R corresponds to the symbol for that portion of the interval I, on which f is monotonic decreasing. Thus defined, the ordering on symbol sequences respects the ordering on the interval as before.

We now give the description of the onset of chaos, corresponding to the first accumulation of period doubling for bimodal maps. We define operations l and r on pairs (X,Y) of finite (possibly empty) sequences of $\{L,R,B\}$ by

$$(X,Y)\ l=(X,Y\overline{V}XUY),\qquad (X,Y)\ r=(X\overline{U}YVX,Y),\tag{1}$$

where $U,\overline{U}$ are symbols chosen from $\{R,B\}$ so that $XU<XK<X\overline{U}$ and $V,\overline{V}$ are symbols chosen from $\{L,R\}$ so that $YV<YC<Y\overline{V}$.

Given a finite sequence s of l's and r's, we define

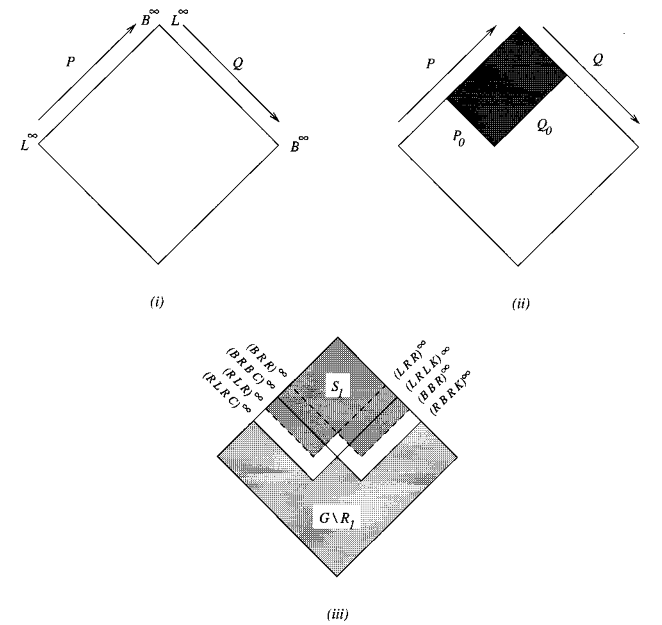

FIG. 1. (i) The space G of pairs (P,Q) of sequences on $\{L,C,R,K,B\}$. (ii) The wedge $\omega(P_0,Q_0)$. (iii) The regions $G\backslash R_1$ and S_1. Taken from Mackey and Tresser [6].

$$(X_s,Y_s)=(\varnothing,\varnothing)s \tag{2}$$

and write $U_s,\bar{U}_s,V_s,\bar{V}_s$ for the $U,\bar{U},V,\bar{V}$ corresponding to X_s,Y_s.

Pairs (P,Q) of sequences (possibly infinite) over the symbol set Σ can be put into a one-to-one correspondence with points in the unit square [see Fig. 1(i)], which we denote by G. For $(P_0,Q_0)\in G$, we define a *wedge* [see Fig. 1(ii)]

$$\omega(P_0,Q_0)=\{(P,Q)\in G : P\geqslant P_0, Q\leqslant Q_0\}.$$

Then define

$$R_n=\bigcup_{\|s\|=n}\omega((X_sU_sY_sC)^\infty,(Y_sV_sX_sK)^\infty) \quad \text{for} \quad n\geqslant 0,$$

where $\|s\|$ is the length of s.

All maps on the boundaries of the wedges defining R_n have a singly superstable, period 2^{n+1} orbit passing through C or K. For example, the kneading sequences in the cases $n=0,1$ are as follows:

$$s=\varnothing:\quad X_sU_sY_sC=RC,\quad Y_sV_sX_sK=RK,$$

$$s=l:\quad X_sU_sY_sC=RLRC,\quad Y_sV_sX_sK=LRLK,$$

$$s=r:\quad X_sU_sY_sC=BRBC,\quad Y_sV_sX_sK=RBRK.$$

As can be seen from the symbol sequences, these two period-doubling cascades are inside each of the hump regions, i.e., around the points C and K.

We also define regions $S_n, n=1,2,3\ldots$, such that S_n contains maps which have certain period $3\times 2^{n-1}$ orbits. The definitions of these S_n is cumbersome. We reproduce it here for completeness.

$$S_1=\omega((RLR)^\infty,(LRR)^\infty)\cup\omega((BRR)^\infty,(RBR)^\infty)$$

while for $n\geqslant 2$,

$$\begin{aligned}S_n=\bigcup_{\|s\|=n-2}&\{\omega(((XUY\bar{V}XUYVXUY\bar{V})_s)^\infty,\\&((Y\bar{V}XUYVXUY\bar{V}XU)_s)^\infty)\\&\cup\omega(((X\bar{U}Y\bar{V}XUY\bar{V}XUY\bar{V})_s)^\infty,\\&((Y\bar{V}XUY\bar{V}X\bar{U}Y\bar{V}XU)_s)^\infty)\\&\cup\ \omega(((X\bar{U}YVX\bar{U}Y\bar{V}X\bar{U}YV)_s)^\infty,\\&((Y\bar{V}X\bar{U}YVX\bar{U}YVX\bar{U})_s)^\infty)\\&\cup\omega(((X\bar{U}YVXUYVX\bar{U}YV)_s)^\infty,\\&((YVX\bar{U}YVXUYVX\bar{U})_s)^\infty)\},\end{aligned}$$

where $(ABC\ldots)_s\equiv(A_sB_sC_s\ldots)$.

The regions R_1 and S_1 are shown in Fig. 1(iii).

With these definitions, following [6], we define subsets of G, which will serve as the regions of order and chaos.

$$\bar{R}_\infty=\bigcup_n\bar{R}_n=\bigcup_n(G\backslash R_n),$$

$$S_\infty=\cup_n S_n,$$

$$D=G\backslash(\bar{R}_\infty\cup S_\infty).$$

Theorem 1 of [6] tells us that the set of periods of a map contained in the region $\bar{R}_\infty$ is given by $\{1,2,4,\ldots,2^n\}$ for some n. Moreover, all such maps are contained within it. Thus maps within this region exhibit ordered dynamics (and have zero topological entropy).

On the other hand, a map contained in the region S_∞ has at least one periodic orbit with a period, which is not a power of 2. Moreover, all such maps belong to this region and have positive topological entropy. Hence, we consider S_∞ as the region of chaos.

The set of periods of a map in the boundary D is given by $\{1,2,4,\ldots,2^n,\ldots\}$. Again, all maps with such a structure of periodic orbits belong to D. We will consider D as the "boundary" of order and chaos. Whether it is a boundary, in a precise topological sense, of the regions $\bar{R}_\infty$ and S_∞ is a long-standing conjecture.

III. THE "EXTENDED" CHOMSKY HIERARCHY

The classical theory of automata and formal languages has its roots, on the one hand, in the work of Turing, Church, and others who developed the foundations of the theory of computation and, on the other, in the work of formal linguists, notably Chomsky. In the present section we present a brief outline of this very large body of work, with an emphasis on that part which will be relevant for our purpose. For details the reader is referred to the classic book by Hopcroft and Ullman [10].

The entities that automata theory deals with are formal descriptions of devices which can perform "mechanical" tasks of varying degrees of difficulty. These devices usually have an input tape containing symbols from an alphabet, a memory unit, and an auxiliary tape which is used to perform

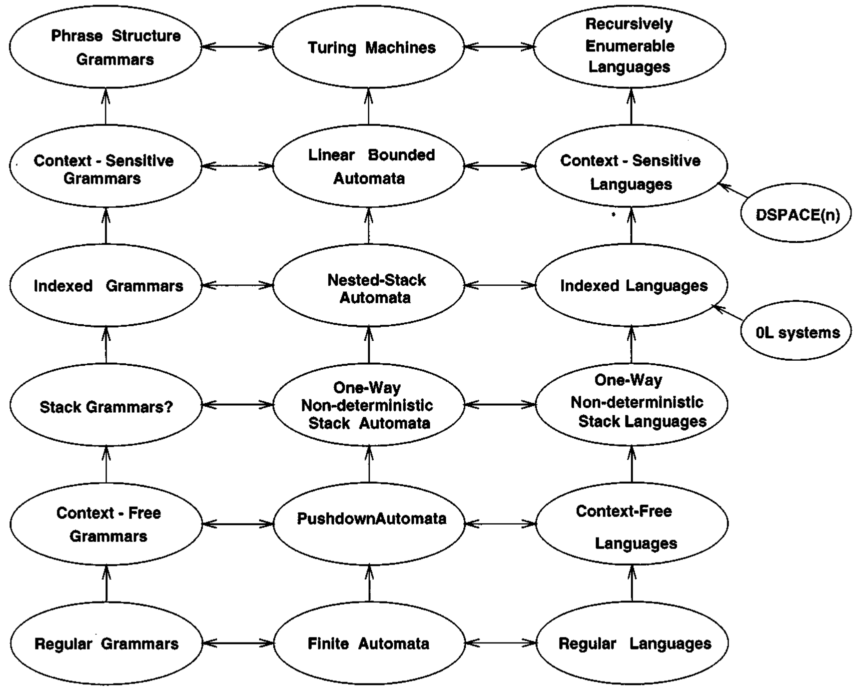

FIG. 2. The "extended" Chomsky hierarchy. The single-headed arrows denote strict inclusion. The double-headed arrows denote an eqivalence. The central column consists of the hierarchy of automata, the left column of the hierarchy of generative grammars, and the right column of the hierarchy of formal languages. In the conventional Chomsky hierarchy, the stack and the indexed languages do not occur.

storage and output. The dynamics of the automaton is described by defining the action(s) to be performed, depending on the symbol being currently read on the input tape, the present memory state, and the current symbol on the storage tape. These actions might entail a change of the memory state, alteration of the current symbol on the storage tape, and motion along the input tape in either direction. The Turing machine is probably the best known example of an automaton. The thesis of Church and Turing states that it is equivalent to any other system that can perform a task that can be defined algorithmically. The central column in Fig. 2 shows various classes of automata that form a hierarchy of machines in terms of their computational power.

Formal languages are sets, containing finite strings (called words) of symbols from some (finite) alphabet. The most common examples of formal languages are the programming languages that are used today. These are contained in the class of formal languages called the context-free languages. One way of describing formal languages is through the grammars that generate them. A grammar contains not only the symbols which form the words in the language that it generates (these are called terminal symbols), but also an auxiliary set of symbols, called the nonterminal symbols. The "productions" of the grammar form the rules by which a given string of terminals and nonterminals can generate another string of terminals and nonterminals. A special nonterminal (usually denoted by S) is designated the "start" symbol. Any string formed from the terminal symbols only, which can be obtained from the "start" symbol, by applying the productions successively, is a valid word in the language. The columns on the left and right in Fig. 2 show a hierarchy of the classes of languages and their generative grammars. It is a beautiful fact of computation theory that the languages generated by the different grammars are also "recognized" by the various classes of automata shown in Fig. 2. This hierarchy of languages, grammars, and machines goes under the name of the Chomsky hierarchy. (Conventionally the Chomsky hierarchy does not contain the indexed and the stacked languages and the automata that recognize them. We call this the "extended" Chomsky hierarchy.)

A very natural problem that arises in computation theory is the determination of the class in which a given language belongs. For this purpose, the theory provides a powerful set of results which are called intercalation theorems or sometimes pumping lemmas. These are necessary conditions for a language to belong to a particular class. The usual application of a pumping lemma is to exclude a given language from a certain class. Most commonly, the form of a pumping lemma is as follows: Given any word belonging to a language (in a certain class), it is possible to find a subword(s) of this word, of length not greater than k (an integer that depends only on the language and not the word) such that all the words obtained by "pumping" (i.e., adding another copy of the subword at its position in the given word) the subword finitely often would all belong to the language. For example, the pumping lemma for the class of regular languages which lie at the bottom of the Chomsky hierarchy is as follows.

Pumping lemma for regular languages. Given a regular language $\mathscr{L}$, there exists an integer n (depending only on $\mathscr{L}$) such that for any word $x=uvw$ in $\mathscr{L}$ with $\|x\|\geqslant n$, $\|uv\|\leqslant n$, v nonempty, all the words $x_i=uv^iw$ also belong to $\mathscr{L}$.

It might be useful to note that in the case when the word represents a periodic orbit, the new words obtained by pumping a subword would correspond to generating new periodic orbits which would generically be unstable. In the next section we use the pumping lemmas for the one-way, nondeterministic stacked languages and the indexed languages.

IV. THE RESULTS AND PROOFS

In this section we present the main results of this paper and their proofs. We first define formal languages corresponding to the onset of chaos in the unimodal and bimodal cases and then obtain the classes in the Chomsky hierarchy where these languages occur. The key ingredient in each proof is a relevant pumping lemma. As the statements of these lemmas are rather involved, we will use the lemma and refer the reader to the literature for its precise statement.

A. The unimodal case

In the notation of Sec. II A, we introduce a sequence of languages $\mathscr{A}_n=\{K_r:0\leqslant r\leqslant n\}$. The words in these languages are formed from the alphabet $\Lambda'=\{L,C,R\}\cup\{(,)^\infty\}$. Moreover for each n, $\mathscr{A}_n$ is finite and is contained in $\mathscr{A}_{n+1}$. Thus each $\mathscr{A}_n$ is a regular language. The limit language $\mathscr{A}_\infty$ can be thought of as a symbolic description of the onset of chaos corresponding to the sequence of period doublings of unimodal maps. Our concern here is to describe the place that this language has in the ''extended'' Chomsky hierarchy. This is summarized by the following theorem.

Theorem 4.1

(i) $\mathscr{A}_\infty$ is not a one-way, nondeterministic stack language.

(ii) $\mathscr{A}_\infty$ is an indexed language (it, in fact, belongs to the more restricted class D0L).

Proof.

(i) We use the pumping lemma for one-way, nondeterministic stack languages, which can be found in [11]. Suppose $\mathscr{A}_\infty$ is a one-way, nondeterministic stacked language. Consider the integer k in theorem 1 (henceforth referred to as OG1) of [11]. Let ξ_0 be a word in $\mathscr{A}_\infty$ of length greater than k. The pumping lemma then guarantees the existence of a string of words $\xi_i, i=1,2,3\ldots$ obtained by intercalating strings within ξ_0 as described in OG1. If $\|\xi\|$ denotes the length of the word ξ, we have from OG1, for all $i>0$,

$$\|\xi_i\|-\|\xi_{i-1}\|=\|\rho_i\|+\|\sigma_i\|+\|\tau_i\|-\|\sigma_{i-1}\|=a+ib, \quad (3)$$

where a is the sum of the lengths of all the α_j's, γ_j's, δ_j's, χ_j's, ψ_j's, and b is the sum of all β_j's that appear in the definitions of ρ_i,σ_i, and τ_i in OG1.

(3) is easily solved recursively, to give

$$P(i)\equiv\|\xi_i\|=\|\xi_0\|+ai+bi(i-1)/2. \quad (4)$$

Now observe that for a word λ in $\mathscr{A}_\infty$, $\|\lambda\|=2^m+2$ for some $m\geqslant 0$. Since each ξ_i is in $\mathscr{A}_\infty$ we have

$$P(i)=2^m+2 \quad (5)$$

for every $i\geqslant 0$, for some $m\in\mathbf{N}$. That this is impossible is most easily seen as follows.

$P'(x)\equiv P(x)-2$ is a polynomial taking integer values on the set of integers. Hence the set $\{P'(n):n\in\mathbf{N}\}$ has an infinite number of prime divisors (this is rather easy to prove). But from (5) $P'(i)$ is contained in $\{2^n:n\in\mathbf{N}\}$. This leads to a contradiction and completes the proof.

(ii) The simplest proof of the fact that $\mathscr{A}_\infty$ in an indexed language is probably obtained by observing that $\mathscr{A}_\infty$ is a D0L language (see [10], p. 390ff, for the definition) and hence is also an indexed language. We will, however, explicitly define the indexed grammar which generates $\mathscr{A}_\infty$.

Consider the indexed grammar defined as follows: The nonterminal symbols are $\{S,T,X,U,\overline{U}\}$, with S as the start symbol. The terminal symbols are $\{L,R,C,(,)^\infty\}$. The index productions are (here ε is the empty string)

$$Xf\to X\overline{U}X,$$

$$Uf\to\overline{U},$$

$$\overline{U}f\to U,$$

$$Xg\to\varepsilon,$$

$$Ug\to L,$$

$$\overline{U}g\to R.$$

The productions are

$$S\to(C)^\infty,$$

$$S\to(TgC)^\infty,$$

$$T\to Tf,$$

$$T\to X\overline{U}X.$$

The derivation of the words in $\mathscr{A}_\infty$ using the indexed grammar proceeds as follows. As is conventional, $A\Rightarrow B$ means that B is derived from A using a single production (or index production), while $A\overset{*}{\Rightarrow}B$ means that there exist $C_1,C_2,\ldots,C_n$ such that $A\Rightarrow C_1\Rightarrow C_2\Rightarrow\cdots\Rightarrow C_n\Rightarrow B$. Here A,B,C_i are words formed from the terminal and nonterminal symbols

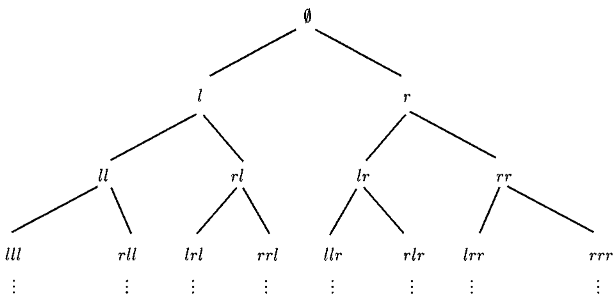

FIG. 3. The tree $\mathcal{T}_s$.

$$S \Rightarrow (C)^{\infty}$$

$$S \Rightarrow (Tg)^{\infty} \Rightarrow (XgU\overline{g}XgC)^{\infty} \overset{*}{\Rightarrow} (RC)^{\infty}$$

$$\Downarrow$$

$$(TfgC)^{\infty} \Rightarrow (XfgU\overline{f}gXfgC)^{\infty}$$

$$\Downarrow$$

$$\vdots \qquad \Downarrow *$$

$$\Downarrow$$

$$\vdots \qquad (XgU\overline{g}XgUgXgU\overline{g}XgC)^{\infty} \overset{*}{\Rightarrow} (RLRC)^{\infty}$$

$$\Downarrow$$

$$\vdots$$

$$(\underbrace{Tff\cdots f}_{\text{n times}}gC)^{\infty} \quad \cdots$$

$$\vdots$$

B. The bimodal case

We define the language which would describe the first accumulation of period doubling for the bimodal maps. In the terminology of Sec. II B, we describe each wedge $\omega(P,Q)$ by the word $\langle P:Q\rangle$. The region R_n is obtained as a union of wedges, each of which corresponds to a unique finite string s of l's and r's of length n.

To describe the word corresponding to R_n we first introduce the lexicographical ordering on strings s, induced by the ordering $l<r$ on the symbols of s, i.e., if $s_1 = xa\ldots$ and $s_2 = xb\ldots$ are two strings on $\{l,r\}$, with a common prefix sequence x and $a,b\in\{l,r\}, a\neq b$, then $s_1<s_2$ if $a<b$, or else $s_1>s_2$. Note that this ordering is different from the ordering on words we had defined in Sec. II A.

Remark (1). The words s can be generated as the nodes of a binary tree. The lexicographical ordering is then the conventional ordering of nodes from left to right of a binary tree (see Fig. 3). We will refer to this tree as $\mathcal{T}_s$.

With this we now define the word corresponding to R_n as follows:

(a) Arrange the wedges occurring in the definition of R_n, in increasing lexicographical order of the strings s that each wedge corresponds to.

(b) Concatenate the words corresponding to each wedge in the same order to form the word corresponding to R_n.

In exactly the same fashion we can obtain the words which describe S_n. Let $w(R_n)$ and $w(S_n)$ be the words which describe R_n and S_n, respectively.

Remark (2). The length of the word representing a wedge in the definition of R_n is of length $2^{n+2}+7$. This follows from the fact that if s has length n, then the lengths of X_s and Y_s add up to $2^{n+1}-2$. Similarly the length of a word that describes a wedge in S_n is $3\times 2^n+7$. In particular, note that the number of symbols that occurs between two consecutive occurrences of the symbol "$\langle$" in the definitions of R_n and S_n is a (nonconstant) function of n.

We define a sequence of languages $\mathcal{B}_i, i=1,2\ldots$ on the alphabet

$$\Sigma' = \Sigma \cup \{\langle,\ \rangle,\ :,\ ;,\ (,\ )^{\infty}\}$$

as follows:

$$\mathcal{B}_i = \{w(R_j): j=1,2\ldots i\}.$$

This language can be thought of as describing the region $\cup_{j=1}^{i}\overline{R}_i$.

As an example, we construct the language $\mathcal{B}_1$.

$$X_{\varnothing}=\varnothing,\ Y_{\varnothing}=\varnothing,\ U_{\varnothing}=R,\ \overline{U}_{\varnothing}=B,\ V_{\varnothing}=R,\ \overline{V}_{\varnothing}=L,$$

$$X_l=\varnothing,\ Y_l=LR,\ U_l=R,\ \overline{U}_l=B,\ V_l=L,\ \overline{V}_l=R,$$

$$X_r=BR,\ Y_r=\varnothing,\ U_r=B,\ \overline{U}_r=R,\ V_r=R,\ \overline{V}_r=L,$$

$$w(R_1)=\langle(RLRC)^{\infty}:(LRLK)^{\infty}\rangle\langle(BRBC)^{\infty}:(RBRK)^{\infty}\rangle,$$

$$\mathcal{B}_1=\{\langle(RLRC)^{\infty}:(LRLK)^{\infty}\rangle\langle(BRBC)^{\infty}:(RBRK)^{\infty}\rangle\}.$$

The limit language $\mathcal{B}_{\infty}$ can be thought of as describing the "boundary" D of order and chaos. The following theorem characterizes $\mathcal{B}_{\infty}$ with respect to the Chomsky hierarchy.

Theorem 4.2

(i) $\mathcal{B}_{\infty}$ is not an indexed language.

(ii) $\mathcal{B}_{\infty}$ is a context-sensitive language [it, in fact, belongs to the more restricted class DSPACE(n)].

Proof.

(i) The pumping lemma for indexed languages is given in [12]. Our proof closely follows theorem 5.3 (henceforth referred to as HA5.3) of [12]. We urge the reader to refer to [12] for a description of the notation that we use in this proof.

Suppose $\mathcal{B}_{\infty}$ is an indexed language. Choose "$\langle$" as a special symbol of Σ'. With every word of an indexed language, we can associate a derivation tree (see [12]). A node, p, of a derivation tree, γ, is said to be a P node, if there exist at least two distinct subtrees under it, each of which contains at least one node with the label "$\langle$." A pair of nodes p_1,p_2 of γ are said to be CF-like if (a) p_2 is a descendent of p_1, (b) p_1 and p_2 have the same labels, (c) there exists a P node p, such that p is a descendent of p_1 and p_2 is a descendent of p. If γ contains no CF like pair of nodes, it is said to be non-CF-like.

We now show that if ψ is a word in $\mathscr{B}_\infty$ of large enough length, then parts of it can be intercalated in such a way that the resulting words would not belong to $\mathscr{B}_\infty$. Choose ψ to be any word in which the number of occurrences of the symbol "$\langle$" is more than k', where k' is an integer which depends only on $\mathscr{B}_\infty$ (and is defined in HA5.3). Say γ is the derivation tree of ψ. We consider two cases.

Case(a). γ is non-CF-like. In this case, following the proof of HA5.3 we can establish that there is a decomposition $\gamma=\alpha\cdot\beta\cdot\delta\cdot\tau\cdot\nu$ such that either α or ν has at least three P nodes. Then by intercalating parts of γ we obtain a sequence of trees θ_n, $n=1,2,3\ldots$ each of which has α and ν as its (first and last, respectively) components. Moreover the word $g(\theta_n)$, of which θ_n is a derivation tree, belongs to $\mathscr{B}_\infty$ and $\|g(\theta_n)\|<\|g(\theta_{n+1})\|$. This means that there exist words of increasing length in $\mathscr{B}_\infty$ such that each of them contains a *fixed* subword of the form $\langle\cdots\langle$ (because either α or ν has at least three P nodes). This contradicts *Remark (2)*.

Case(b). γ has a CF-like pair of nodes. In this case the proof of lemma 2.1 of [12] guarantees a decomposition $\gamma=\alpha\cdot\beta\cdot\delta$ such that β contains a P node. Moreover for each $n=1,2,3\ldots$, there exists a sequence of trees

$$\gamma_n=\alpha\cdot\underbrace{\beta\cdot\beta\cdots\beta}_{\text{n times}}\cdot\delta$$

such that each $g(\gamma_n)\in\mathscr{B}_\infty$ and $\|g(\gamma_n)\|<\|g(\gamma_{n+1})\|$. This implies that for every $n\geqslant 1$, $g(\gamma_n)$ contains a subword of the form $(\cdots\langle\cdots)^n$ (note that by $\cdots\langle\cdots$ we mean a *fixed* subword of that form). Since the γ_n's belong to $\mathscr{B}_\infty$ we have a contradiction with *remark (2)*.

(ii) We prove that $\mathscr{B}_\infty$ is a context-sensitive language by showing that it in fact belongs to the complexity class DSPACE(n). The proof that the class of context-sensitive languages is equivalent to the complexity class NSPACE (n) [which contains DSPACE(n)], can be found in [10].

To show that a language is DSPACE(n), we need to show that the words in the language can be recognized by an off-line, multitape Turing machine such that the size of every tape (including the read-only tape) is limited to the size of the input word (buffered on either side by end-marker symbols). We will informally describe the steps in the algorithm required to recognize the input string. It will be clear from our description that each subroutine in this algorithm can be implemented on a bounded tape (or sometimes a pair of tapes) of our machine. The number of subroutines involved in the description will thus determine the number of tapes of the machine. We buffer each tape of the machine with end-marker symbols. If for a given input string the head on any output tape reaches an end marker, the machine halts in a nonfinal state and thus fails to accept the string. It will be clear that for a string in $\mathscr{B}_\infty$ this will never happen.

In what follows we denote the tapes of the Turing machine by $t_i, i=0,\ldots,N$. t_0 is the input tape which is read-only. Auxiliary tapes necessary to perform computations at *step (1)* will be denoted as t_{ai}, t_{bi}, etc.

Step (1). Determine n for the given input word ψ.

We wish to place n 1's on t_1 to determine the depth of the binary tree [see *remark (1)*] where ψ could occur. This is easily possible using *remark (2)*. Since the first wedge (in what follows we will use the terms "wedge" and "the word representing the wedge" interchangeably when there is no danger of confusion) in any word of $\mathscr{B}_\infty$ has length $2^{n+2}+7$ for some n, we can use two auxiliary tapes t_{a1}, t_{b1} to evaluate n.

Step (2). Determine the possible strings s on $\{l,r\}$ of length n, in lexicographical order.

We now want to obtain the nodes of the tree $\mathscr{T}_s$ [see *remark (1)*] at depth n. These nodes will be written on t_2 in the lexicographical order from left to right and will be separated by commas. To achieve this we need two auxiliary tapes t_{a2}, t_{b2}. For every 1 encountered on t_1, the tapes t_{a2} and t_{b2} can be used alternately to generate the nodes of $\mathscr{T}_s$ at successive depths by copying and prefixing. This is continued until the first blank is encountered on t_1, when the process stops and the contents of the last auxiliary tape written on are copied to t_2. As the formal description of the entire procedure is cumbersome, we leave it to the reader to check it. It is easy to see that the limited size of the tapes available, in fact, suffices for this purpose.

Steps (3)–(8). Determine $X_s, Y_s, U_s, \overline{U_s}, V_s, \overline{V_s}$ for an s on t_2.

For an s that has been written on tape t_2, we wish to determine the corresponding $X,Y,U,\overline{U},V,\overline{V}$. These will be stored on tapes t_3 to t_8, respectively. As the process is recursive we can use auxiliary tapes t_{ai}, t_{bi} $i=3,\ldots,8$ for each main storage tape. While the determination of X,Y requires mere copying from one tape to another, the determination of $U,\overline{U},V,\overline{V}$ will require the determination of the parity of the number of R's in X,Y.

Step (9). Determine the word corresponding to R_n in $\mathscr{B}_\infty$.

For each s the wedges that occur in the definition of R_n are easily obtained by copying the $X,Y,U,\overline{U},V,\overline{V}$ in the relevant order to t_9. Control then returns to *steps (3)–(8)* where $X,Y,U,\overline{U},V,\overline{V}$ corresponding to the next s are obtained, and so forth.

After this the word in t_9 is compared symbol-by-symbol with that in t_0. If it is the same the machine halts in a final state or else it halts in a nonfinal state.

We observe that if $\psi\in\mathscr{B}_\infty$, then each step in the process described above could be carried out on the bounded set of tapes that was available. Thus if during any of these steps a head reaches the end of a particular storage tape, we are guaranteed that $\psi\notin\mathscr{B}_\infty$ and the machine would halt in a nonfinal state, rejecting the word. This completes the proof.

V. APPROACHING THE ONSET OF CHAOS

Let us first consider the description of the onset of chaos in the unimodal case. It might seem at first sight that the description of this set by means of the language $\mathscr{A}_\infty$ as being rather *ad hoc*. In fact, instead of choosing to approach the accumulation point through a sequence of superstable bifurcations, we might as well have chosen any of the other sequences available. However, it is easy to see that the kneading sequences within a periodic window are very simply related to each other. We could in fact have included every kneading sequence less than X_∞ to form a new language $\mathscr{C}_\infty$. This new language would be described simply as

$$\mathscr{C}_\infty = \mathscr{A}_\infty^< \cup \mathscr{A}_\infty \cup \mathscr{A}_\infty^>,$$

where $\mathscr{A}_\infty^<$ ($\mathscr{A}_\infty^>$) denotes the language containing the kneading sequences to the left (resp. right) of the superstable sequences in each periodic window. A proof almost identical to the one given for $\mathscr{A}_\infty$, however, shows that both $\mathscr{A}_\infty^<$ and $\mathscr{A}_\infty^>$ (and hence $\mathscr{C}_\infty$) are indexed languages.

A more pertinent question is what happens if we are to approach the onset of chaos from the chaotic side. This question is rather tricky. For the sake of discussion let us fix the family of maps to be the logistic family, described by the equation $f(x)=\mu x(1-x)$. It will be clear that everything would go through for a much larger class of maps as well. We could approach the accumulation point μ_∞ from the right through a sequence of band-merging points. We begin with the region of well-developed chaos at $\mu=4$. The kneading sequence at this point is $K_1''=R(L)^\infty$. The band-merging points are then obtained through successive applications of the $*$ operator, mentioned in Sec. II A, as follows:

$$K_2''=R*K_1'',$$

$$K_3''=R*R*K_1'',$$

$$\vdots$$

$$K_n''=(R*)^n K_1'',$$

$$\vdots$$

Now consider the language $\mathscr{A}_\infty''=\{K_n'':n=1,2,\ldots\}$. We could think of this language as another description of the edge of chaos, this time from the region greater than μ_∞. In fact the definition of the $*$ operator immediately confirms that this is a D0L language and hence is also an indexed language (see [10]). Note that we could have discussed the languages $\mathscr{A}_\infty$ and $\mathscr{A}_\infty''$ using the $*$ operation as well (see, for example [5],).

We could now ask the same question that we asked before. What about other descriptions of the onset of chaos from the chaotic part of the spectrum? We do not at present have any good answer to this question. As a preliminary observation we might note that we could have chosen to approach the onset of chaos through the kneading sequences K_n' described in Sec. II A and could have described yet another language $\mathscr{A}_\infty'=\{K_n':n=1,2,3\ldots\}$. Of course this language is also an indexed language.

However, there is in fact a crucial difference between the language $\mathscr{A}_\infty$ and the languages $\mathscr{A}_\infty'$ or $\mathscr{A}_\infty''$, which, though obvious, might be well worth pointing out: each word in $\mathscr{A}_\infty$ describes a stable periodic point, whereas that in $\mathscr{C}_\infty$ describes an unstable periodic point. In the chaotic regime the attracting sets would be described by aperiodic symbol sequences which are infinitely long. Unfortunately, classical computation theory does not consider within its domain languages whose words might be infinitely long. In fact the behavior of classical computational devices on words of infinite length is very different from their behavior on words of finite length (see [13]). Thus, we would have to have to find appropriate representations for such sequences so as to be able to deal with this problem within the context of classical computation theory. Alternatively, we speculate that a description of computation over the field of reals, when sufficiently elaborate so as to provide an analog of the Chomsky hierarchy, might throw light upon these issues. The recent developments in this direction due to Blum, Shub, and Smale [14] might provide the seeds of such a theory.

The drawbacks mentioned above also apply to our discussion of the bimodal maps. We could, as in the unimodal case, choose to approach the edge from the chaotic side. In the notation of Secs. II B and IV B, the language $\mathscr{B}_\infty'=\{w(S_n):n=1,2,3\ldots\}$ can be considered as describing the region D, thought of as the boundary of the region S_∞. The proof given in Sec. IV B goes through almost unaltered even for this case. Thus $\mathscr{B}_\infty'$ is a context-sensitive language.

VI. ON COMPLEX DESCRIPTIONS

In conclusion we would like to place our results in perspective. What paradigm do these results suggest for a definition of complexity? In order to address this question we must first inquire into the process of description. The scientific description of phenomena normally involves two aspects. The first is the specification of the *model class*. The second aspect involves the description of the phenomenon at hand, with respect to this model class. Let us call this process, *interpretation*. For a description to be ''useful'' we must ensure that both the model class and the interpretation have been specified in finite terms. Given this rather simplistic picture of the modeling process, we now ask how complexity arises or, more specifically, why are some phenomena more complex than others? Consider a phenomenon which resists finite interpretation with respect to a certain model class. In order to describe it, we would then have to construct a ''larger'' model class. This could be regarded as signalling complexity.

To illustrate this in the context of our results, consider the classes of automata, or equivalently, the classes of grammars in the Chomsky hierarchy as representing model classes. Let the behavior of maps (unimodal or bimodal) represent the entire class of phenomena to be described. The symbolic dynamics, giving rise to a language, and the explicit construction of a grammar (corresponding to a given model class) generating that language, at each value in the parameter space, can serve as an interpretation for the phenomenon. Together these constitute a description of the behavior of the map in question. Now, at the onset of chaos we observe that we are forced to change our model class (in the unimodal case, for example, from the regular grammars to the indexed grammars). In fact, in our case we have proved that no finite interpretation can be obtained of the onset of chaos in terms of the older model class. This describes the complexity at the onset of chaos.

The idea that complexity and emergence must necessarily be defined with respect to models has been considered before (see [15,16] and references therein). Finally we would like to point out that, unlike conventional statistical mechanics, which relies on numerical or quantitative classes (say using critical exponents), the paradigm suggested above favors a more ''descriptive'' definition of complexity. What one is tempted to observe is that complexity is not so much a matter of number as it is of mechanism.

Note added. After most of this work was completed, we discovered that Crutchfield and Young [17,18] have also shown that the formal language corresponding to the unimodal case is an indexed language. Their approach, based on the ε-machine reconstruction, is more general but different from the one (based on symbolic dynamics) pursued in this paper.

ACKNOWLEDGMENTS

I would like to acknowledge Spenta R. Wadia for having introduced me to the subject of complex systems and for his constant guidance during the course of my work. I thank Gautam Mandal, Avinash Dhar, and Spenta R. Wadia for their encouragement and support at all stages. I am grateful for the endless useful discussions that I have had with them which have benefited me enormously. I also wish to thank Jim Crutchfield for having communicated to me details regarding his work and for pointing out an error in the references.

[1] S. Wolfram, Physica D **10**, 1 (1984).
[2] G. Chaitin, *Algorithmic Information Theory* (Cambridge University Press, Cambridge, 1987).
[3] A. Dhar, G. Mandal, P. Lakdawala, and S. R. Wadia, Phys. Rev. E **51**, 3032 (1995).
[4] J. Crutchfield, Physica D **75**, 11 (1994).
[5] P. Collet and J.P. Eckmann, *Iterated Maps of the Interval as Dynamical Systems* (Birkhauser, Boston, 1980).
[6] R.S. Mackay and C. Tresser, J. London Math. Soc. **37**, 164 (1988).
[7] J. Milnor and W. Thurston, in *On Iterated Maps of the Interval*, edited by J.C. Alexander, Lecture Notes in Mathematics Vol. 1342 (Springer-Verlag, Berlin, 1988).
[8] M.J. Feigenbaum, Physica D **7**, 16 (1983).
[9] B. Derrida, A. Gervois, and Y. Pomeau, Ann. de l'Inst. Henri Poincare **29**, 305 (1978).
[10] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation* (Addison-Wesley, Reading, MA, 1979).
[11] W. Ogden, *Intercalation Theorems for Stack Languages*, in *Proceedings of the First Annual ACM Symposium on the Theory of Computing*, edited by M.A. Harrison (ACM, New York, 1969), Vol. 31.
[12] T. Hayashi, Publ. RIMS, Kyoto Univ. **9**, 61 (1973).
[13] S. Eilenberg, *Automata, Languages and Machines* (Academic Press, New York, 1974), Vol. A.
[14] L. Blum, M. Shub, and S. Smale, Bull. Am. Math. Soc. **21**, 1 (1989).
[15] R. Badii, *Complexity and Unpredictable Scaling of Hierarchical Structures*, in *Chaotic Dynamics: Theory and Practice*, edited by T. Bountis (Plenum Press, New York, 1992).
[16] P. Cariani, *Emergence and Artificial Life*, in *Artificial Life II*, edited by C.G. Langton, C. Taylor, J.D. Farmer, and S. Rasmussen (Addison-Wesley, Redwood City, CA, 1991).
[17] J. Crutchfield and K. Young, Phys. Rev. Lett. **63**, 105 (1989).
[18] J. Crutchfield and K. Young, *Computation at the Onset of Chaos*, in *Complexity, Entropy and the Physics of Information* (Addison-Wesley, Redwood City, CA, 1990).