Grammatical complexity for two-dimensional maps

2004 J. Phys. A: Math. Gen. 37 10545

(http://iopscience.iop.org/0305-4470/37/44/006)

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND GENERAL

J. Phys. A: Math. Gen. **37** (2004) 10545–10559 PII: S0305-4470(04)80537-7

Grammatical complexity for two-dimensional maps

Ryouichi Hagiwara and Akira Shudo

Department of Physics, Tokyo Metropolitan University, Minami-Ohsawa, Hachioji, Tokyo 192-0397, Japan

E-mail: ryouichi@comp.metro-u.ac.jp and shudo@phys.metro-u.ac.jp

Received 11 May 2004, in final form 5 August 2004
Published 20 October 2004
Online at stacks.iop.org/JPhysA/37/10545
doi:10.1088/0305-4470/37/44/006

Abstract

We calculate the grammatical complexity of the symbol sequences generated from the Hénon map and the Lozi map using the recently developed methods to construct the pruning front. When the map is hyperbolic, the language of symbol sequences is regular in the sense of the Chomsky hierarchy and the corresponding grammatical complexity takes finite values. It is found that the complexity exhibits a self-similar structure as a function of the system parameter, and the similarity of the pruning fronts is discussed as an origin of such self-similarity. For non-hyperbolic cases, it is observed that the complexity monotonically increases as we increase the resolution of the pruning front.

PACS numbers: 05.45.−a, 05.45.Ac

1. Introduction

Characterizing and measuring complexity is one of the central issues in the study of complex systems. In particular, much work has been devoted to understanding the complexity of dynamical systems situated between completely regular and fully random states. Various definitions of complexity have so far been proposed in many fields such as information theory and computer science, and they are employed for quantifying complexity. A series of research is surveyed, for example, in [1].

Wolfram exploited for the first time the theory of formal languages, which was developed by Chomsky, to study dynamical systems in his paper about cellular automata [2]. Since then, this idea has been applied to the study especially on the complexity of one-dimensional unimodal maps on an interval [3–13]. The Chomsky hierarchy consists of four levels of languages: regular languages, context-free languages, context-sensitive languages and recursively enumerable languages, from the bottom. The lower levels are contained in the upper ones. It was shown in [5, 6] that if the kneading sequence of a unimodal map is either periodic or eventually periodic, the language generated from the map is regular (i.e. in the

lowest level of the Chomsky hierarchy [14]). After that, Xie and Wang proved that the converse is also true [10, 11]. For unimodal maps, it was also proved that the language generated at the period-doubling accumulation point is one in the level located between context-free and context-sensitive [8, 9, 12]. Moreover, Xie *et al* have proposed a conjecture that unimodal maps never generate proper context-free languages [12, 13].

On the other hand, it is a difficult task to carry out an analogous programme in higher dimensional dynamical systems in general. This is because the construction of a proper symbolic dynamics, which is necessary to apply the theory of languages, is not easy except for the trivial horseshoe dynamics. The difficulty lies in the fact that there are no critical points in generic higher dimensional maps. The itinerary of the critical point, i.e. the kneading sequence, characterizes all the topological characters of a one-dimensional map. As mentioned, the classification of complexity based on the theory of languages is done by referring the character of the kneading sequences.

However, there are several recent examples of steady progresses made even in higher dimensional cases: the idea of the pruning front, which was proposed in order to construct a two-dimensional analogy of the kneading theory [15, 16], has not only been mathematically formulated [17–19], but also a concrete algorithm to perform the proposed idea for the Lozi map [20]:

$$L_{a,b} : \begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} 1 + by - a|x| \\ x \end{pmatrix}$$

has been presented [17]. A heuristic algorithm to generate the pruning front for the Hénon map [21]

$$H_{a,b} : \begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} a + by - x^2 \\ x \end{pmatrix}$$

has also been presented [22]. We note here that the latter algorithm can be applied to the area-preserving case, $|b| = 1$.

These two maps are the simplest possible examples of higher dimensional problems. Thus these results provide us with a promising opportunity to see how the complexity of higher dimensional dynamics behaves and how the class of formal languages is related to the structure of the pruning front in the symbol plane. The purpose of the present paper is to compute the so-called grammatical complexity (GC) of such simple two-dimensional maps by employing an explicit algorithm to construct the pruning front for the Hénon map [22] and the one for the Lozi map [17]. GC is a quantity measuring the difficulty in describing the topological structure of a map.

In particular, we will focus on the behaviour of GC as a function of the system parameter. Our motivation originates from the numerical studies suggesting that there are infinitely many parameter intervals where the map recovers hyperbolicity even after the destruction of the horseshoe [23, 24]. In such a case, the language has only finitely many rules, and is regular. By the definition of Wolfram [2], the complexity of a regular language is the number of states of the minimal finite automaton equivalent to the language. So GC is a computable object if a map has hyperbolic structure.

The paper is organized as follows. In the next section, we will give some definitions and notation for regular languages together with finite automata. Section 3 provides a method of obtaining a language from the Hénon and the Lozi maps, and calculating its GC. In section 4, the results of calculation are presented and their implications are discussed. Finally in section 5, we summarize the paper and mention some open and related problems.

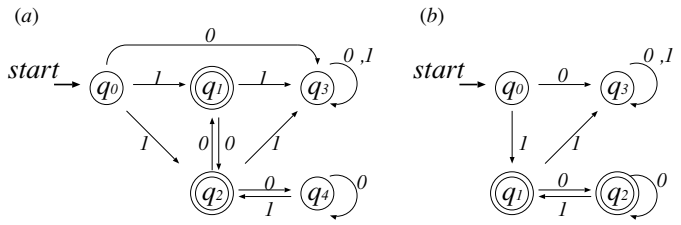

Figure 1. (*a*) Example of a non-deterministic finite automaton. (*b*) Example of a deterministic finite automaton. In each figure, a node of a graph represents a state. A transition between two states is possible if an input symbol is the same as the one labelled on an arrow connecting the two states. The initial state is q_0, and the final states are denoted by double circles. In (*a*), if the initial input is 1, the transition from q_0 is allowed to either q_1 or q_2, and if the initial input is 0, the transition is allowed only to q_3.

2. Regular-language complexity

In this section, we briefly explain some basics of regular languages and finite automata, and introduce the definition of regular-language complexity. For the readers who are interested in the details, a standard textbook [25] is helpful.

A language is a set of finite strings. In this paper, we hereafter refer to a finite string by the word 'string', and an infinite one by 'sequence'. A regular language is a set of strings expressed by three kinds of operations:

$$\begin{cases} \text{concatenation} & : \quad L_1 L_2 := \{xy \mid x \in L_1,\, y \in L_2\} \\ * & : \quad L^* := \bigcup_{j=0}^{\infty} L^j \\ + & : \quad L_1 + L_2 := L_1 \cup L_2 \end{cases}$$

where L^j is the j-fold concatenation of L. For example, the set of all strings over the alphabet $\Sigma = \{0, 1\}$ is a regular language expressed as $L = (0 + 1)^*$. This is equal to Σ^*. The string of length 0 is denoted by ϵ.

A regular language is closely related to a finite automaton. A finite automaton is a mathematical model $M = (Q, \Sigma, \delta, q_0, F)$, which represents transitions between states with a certain input string:

$$\begin{cases} Q = \{q_0, q_1, \ldots, q_n\} & : \quad \text{set of states} \\ \Sigma = \{0, 1\} & : \quad \text{input alphabet} \\ \delta : Q \times \Sigma^* \to Q & : \quad \text{transition function} \\ q_0 \in Q & : \quad \text{initial state} \\ F \subset Q & : \quad \text{set of final states} \end{cases}$$

Here $\delta(q, x)$ denotes a function of a state $q \in Q$ and an input string $x \in \Sigma^*$.

We say that a finite automaton is a deterministic finite automaton (DFA) if the transition function $\delta(q, x)$ is uniquely determined by the present state q and an initial input string x, and a non-deterministic finite automaton (NFA) otherwise. Figure 1 shows examples of a DFA and a NFA.

For a finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ and a string $x \in \Sigma^*$, if $\delta(q_0, x) \in F$, then we say that x is accepted by M. For a NFA, it suffices that there exists a transition by which the NFA reaches one of the final states among the possible ones with an input. The set of strings

$$L(M) = \{x \in \Sigma^* \mid \delta(q_0, x) \in F\}$$

is the language accepted by M.

It is equivalent that a language L is regular and that L is accepted by a finite automaton. That is, for any regular language, there exist a NFA and a DFA accepting it, and a language accepted by a NFA or a DFA is regular [25]. For example, figures 1(a) and (b) are a NFA and a DFA accepting $L = 1(0 + 01)^*$ (i.e. a language without consecutive 1s), respectively.

In general, there may exist several different automata representing the same language. However, as is known as the Myhill–Nerode theorem, there exists a unique DFA up to permutation of labelling such that the number of states is minimal among DFAs accepting a given regular language. Therefore, minimal DFAs can be used for comparing different languages. A concrete algorithm minimizing a redundant DFA is presented in [25].

Now we can state the definition of complexity proposed by Wolfram [2]. The complexity of a regular language L is defined as

$$C(L) := \log N(L)$$

where $N(L)$ is the number of states of the minimal DFA accepting L. We call $C(L)$ the grammatical complexity (GC) of the language L.

3. Method

In this section, we first explain how to obtain the symbolic dynamics for the area-preserving Hénon map. Then we describe the construction of a finite automaton accepting the language made of it.

3.1. Calculation of transition rules

A crucial task to apply the theory of formal languages to a dynamical system is to find a proper discrete coding. This can be achieved if there exists a symbolic dynamics which is conjugate or semi-conjugate to the original dynamics. For a one-dimensional unimodal map, the critical point divides the interval, and a symbol is assigned to each subinterval thus determined. Furthermore, as is well known as the Milnor–Thurston's kneading theory, the itinerary of the critical point controls admissibility of the orbits [26].

A straightforward extension of the kneading theory to two-dimensional maps cannot be made due to the lack of critical points, but a possibility of analogous construction has been explored by Cvitanović *et al* [15, 16]. An idea of *pruning* is first to suppose the two-dimensional symbol plane for the horseshoe dynamics, and then to observe which orbits become nonadmissible in the dynamics as the system parameter is varied. Nonadmissible orbits are considered pruned from the horseshoe symbol plane. The forbidden regions in the symbol plane are called pruned regions. Note that the pruned regions cannot be determined uniquely because, by definition, forward and backward iterations of some pruned region yield other pruned regions. So it may have redundancy unless additional conditions are specified. The *primary pruned region* is introduced to avoid such ambiguity, and its border is the *pruning front*. A spirit of the pruning front is to completely specify the admissible orbits in the symbol plane. It has indeed been expressed as the *pruning front conjecture*, which states that all forbidden orbits are specified solely by the pruning front and that there are no other independent pruning mechanisms [15, 16].

An idea of pruning would be certainly natural and has been formulated even rigorously [17–19], but we need a concrete algorithm to provide the pruning front for a given map in order to perform an explicit computation of the grammatical complexity introduced above. A recipe proposed for the Lozi map was the first complete solution to such a programme [17]. For an arbitrary parameter value of the Lozi map, the pruning front can explicitly be written using the piecewise linearity of the Lozi map.

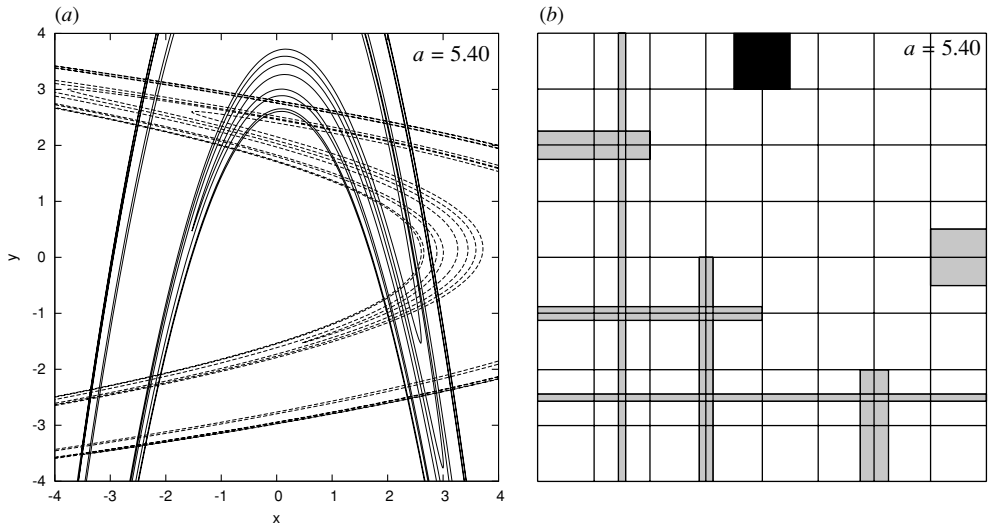

Figure 2. The stable and unstable manifolds for the area-preserving Hénon map at $a = 5.4$ and the pruned regions. The primary pruned region is coloured black, and its forward and backward images are grey.

On the other hand, it would be difficult to find such a simple algorithm for the Hénon map. An attempt was made in our previous paper to give a prescription for constructing the pruned regions of the area-preserving Hénon map [22]. Our algorithm assumes hyperbolicity of the map, otherwise the procedure will not stop within finitely many steps. Crucial information required in the construction is a bifurcation diagram of homoclinic orbits on a fundamental segment of the stable (or unstable) manifold. Our basic idea is to convert information of homoclinic points aligned on a fundamental segment into that of the two-dimensional symbol plane in which the pruning front is drawn. A bifurcation diagram—that is, information describing in what order homoclinic points on a fundamental segment disappear as a function of the system parameter—is obtained by continuation of orbits from the horseshoe limit using, for example, a technique developed in [27, 28].

The numerical studies made in [23, 24] strongly suggest that a set of parameter values for which the Hénon map has hyperbolic structures has a positive Lebesgue measure. Since our algorithm terminates within finitely many steps for hyperbolic parameter intervals, it has also been indicated that there exist infinitely many intervals to which our algorithm can be applied. Figure 2(*a*) shows the stable and unstable manifolds for the area-preserving Hénon map at $a = 5.4$, and (*b*) shows the corresponding pruned regions constructed using our algorithm. This parameter value is in the longest hyperbolic interval, which is henceforth called the *longest plateau*.

On the other hand, as implied in [24], a set of parameter values for which the system is not hyperbolic also has a positive Lebesgue measure. Our algorithm may not terminate and precise primary pruned regions cannot be determined in such cases. In the following, we will cope with non-hyperbolic cases by calculating *approximate* pruned regions. They are obtained by introducing a certain resolution in the symbol plane, which means that we limit the length of forbidden strings and ignore the finer structures of the pruning fronts. Therefore, the resulting pruned regions and the corresponding automata depend on the size of resolution. Indeed as observed in the next section, the number of steps of the pruning fronts increases with the increase of resolution.

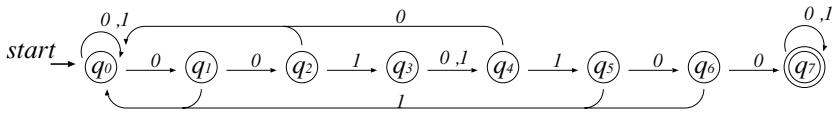

Figure 3. A non-deterministic finite automaton accepting $\bar{L}$.

3.2. Conversion of a transition rule into a finite automaton

Now taking the longest plateau ($a \simeq 5.4$), we present an example of actual procedures by which a finite automaton accepting all the admissible strings is constructed.

The forbidden strings of the longest plateau are 0010100 and 0011100, which correspond to the blocks shown in figure 2. Let L be the set of all strings which do not contain these forbidden ones. The complement $\bar{L} = \Sigma^* \backslash L$ is easily obtained, and is expressed as

$$\bar{L} = (0+1)^*(0010100 + 0011100)(0+1)^*.$$

This is regular, and is accepted by the NFA shown in figure 3. This NFA accepts all sequences which contain the forbidden strings. So it does not depend on how to decide the initial digit (i.e. the position of '·' in a symbol sequence) as an input, although we consider the set of doubly infinite sequences for a two-dimensional map.

If a language L is regular, then the complement $\bar{L} = \Sigma^* \backslash L$ is also regular [25]. Correspondingly for automata, if $M = (Q, \Sigma, \delta, q_0, F)$ is a DFA accepting L, then $M' = (Q, \Sigma, \delta, q_0, Q \backslash F)$ is a DFA accepting $\bar{L}$. This is because $x \in \bar{L}$ and $\delta(q_0, x) \in Q \backslash F$ are equivalent. Thus, we first construct a DFA accepting $\bar{L}$, and then exchange the roles of the final states and the others.

We denote the NFA in figure 3 by $M = (Q, \Sigma, \delta, q_0, F)$. Now we construct a DFA $M' = (Q', \Sigma, \delta', q_0', F')$ equivalent to M. First, the initial state is set as $q_0' = \{q_0\}$. Since it is possible to move from q_0 to either q_0 or q_1 with an input 0, let $\delta'(q_0, 0) = \{q_0, q_1\}$. This is one of the elements of Q'. Note that Q' consists of the subsets of Q. Next, since the NFA can move from q_0 to $\{q_0, q_1\}$ and from q_1 to q_2, respectively, with 0, let the state to which M' moves from $\{q_0, q_1\}$ be $\{q_0, q_1, q_2\}$. That is, to complete Q' it suffices to add (the sets of) states which can be reached from q_0 one after another. The set of the final states F' consists of all the elements of Q' including one of the final states of M. If all the nodes have two arcs labelled 0 and 1, then the DFA M', which is shown in figure 4, is completed.

To calculate the GC, we have to reduce M' to the minimal DFA. Note that all the accessible states from $\{q_0, q_1, q_2, q_7\}$ are the final states of M', since they contain the final state q_7 of the NFA. That is, once it reaches $\{q_0, q_1, q_2, q_7\}$, all strings are accepted by this DFA, no matter what the subsequent input symbols are. Therefore, it is unnecessary to distinguish these final states. Performing the reduction algorithm of [25] confirms that there is no other state which can be reduced besides these states. Hence, the minimal DFA equivalent to M' is the one shown in figure 5.

Exchanging the roles of the final states and the others gives the minimal DFA accepting L. If a string containing one of the forbidden strings 0010100 and 0011100 is given as an input, the minimal DFA reaches the state $\{q_0, q_1, q_2, q_7\}$, implying that the string is never accepted. In other words, admissible symbol sequences on the non-wandering set of the Hénon map do not change the state of the DFA to $\{q_0, q_1, q_2, q_7\}$. We let the logarithm of the number of states other than this one be the GC. Consequently, for the longest plateau, $C(L) = \log 8$. The whole procedure is summarized as shown in figure 6.

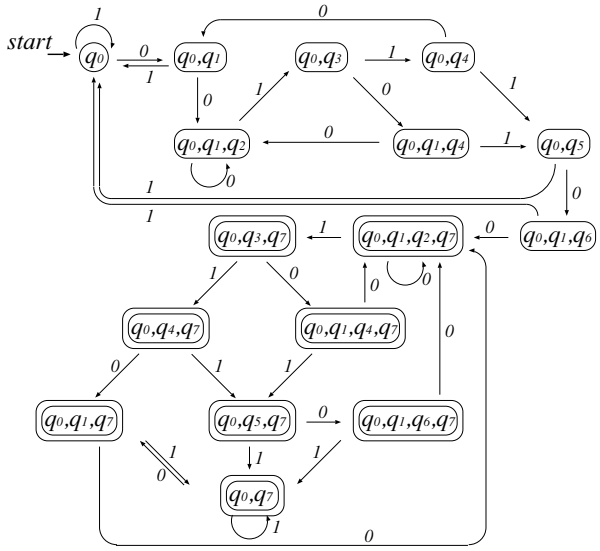

Figure 4. A deterministic finite automaton accepting $\bar{L}$.

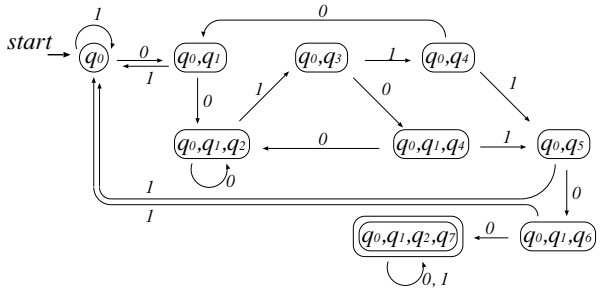

Figure 5. The minimal deterministic finite automaton accepting $\bar{L}$. Exchanging the role of the final states and the others gives the minimal one accepting L.

Figure 6. Procedure for obtaining the grammatical complexity. The notation $\overline{\text{NFA}}$ means a non-deterministic finite automaton accepting $\bar{L}$, and so on.

Now we should discuss whether or not the GC obtained by this procedure described surely characterizes a given dynamical system. This is because a primary pruned region can be expressed by apparently different sets of forbidden strings, and therefore can give different automata. For example, if the forbidden string is 001, it can be expressed by two strings 0010 and 0011. As shown in figure 7, several different NFAs are constructed depending on the difference of forbidden strings. It should be noted that these NFAs are different in the ordinary sense: for example, the lower automaton in figure 7 does not accept 001, even though this is

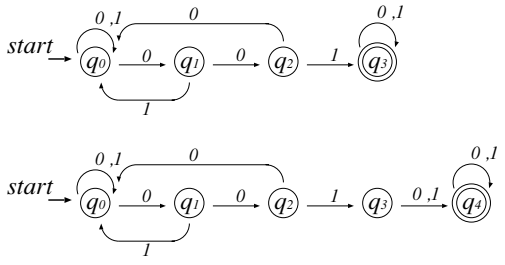

Figure 7. Two equivalent non-deterministic finite automata under the condition that infinite sequences are inputed.

a forbidden string. However, since the length of input sequences considered here is infinite, these NFAs represent the same dynamics. So here we regard these automata as *equivalent*. After a NFA is obtained, converting it into the minimal DFA is performed uniquely. In general, a minimal DFA may have states reached only once. Since we consider an infinite sequence for an input, we again remove these states from the minimal DFA. We further remove the final states of the DFA accepting $\bar{L}$. (In case of figure 5, the final state is just $\{q_0, q_1, q_2, q_7\}$.) This is because the sequences by which the DFA reaches those states are not admissible. We call these operations 'pruning of automata'. For all the examples studied so far, the pruned DFA is uniquely obtained from different NFAs, and no exceptions have been found. So we conjecture that the pruned DFA describes the dynamics on the non-wandering set of the system, and that it does not depend on which NFA we take at the first step. If this is true, the pruned DFA gives the GC of a system and certainly characterizes it.

The transition rule of the pruned DFA can be written in the form of a matrix. As for the longest plateau of the area-preserving Hénon map, this is

$$\begin{array}{r} \\ q_0 \\ q_0q_1 \\ q_0q_1q_2 \\ q_0q_3 \\ q_0q_1q_4 \\ q_0q_4 \\ q_0q_5 \\ q_0q_1q_6 \end{array} \begin{array}{c} \begin{array}{cccccccc} q_0 & q_0q_1 & q_0q_1q_2 & q_0q_3 & q_0q_1q_4 & q_0q_4 & q_0q_5 & q_0q_1q_6 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

This just represents a structure matrix of the map, and the number of states of the pruned DFA also gives the order of the minimal structure matrix.

4. Results

Here we present the results of our numerical calculations. In figure 8, we plot the GC for the Hénon map as a function of the system parameter a. We only consider the area-preserving case, so the other parameter is taken as $b = -1$ throughout the following calculations. Each figure is a magnification of the upper one. Above $a = 5.699\ldots$ at which the first tangency between the stable and unstable manifolds occurs [27], the symbolic dynamics forms the binary full shift.

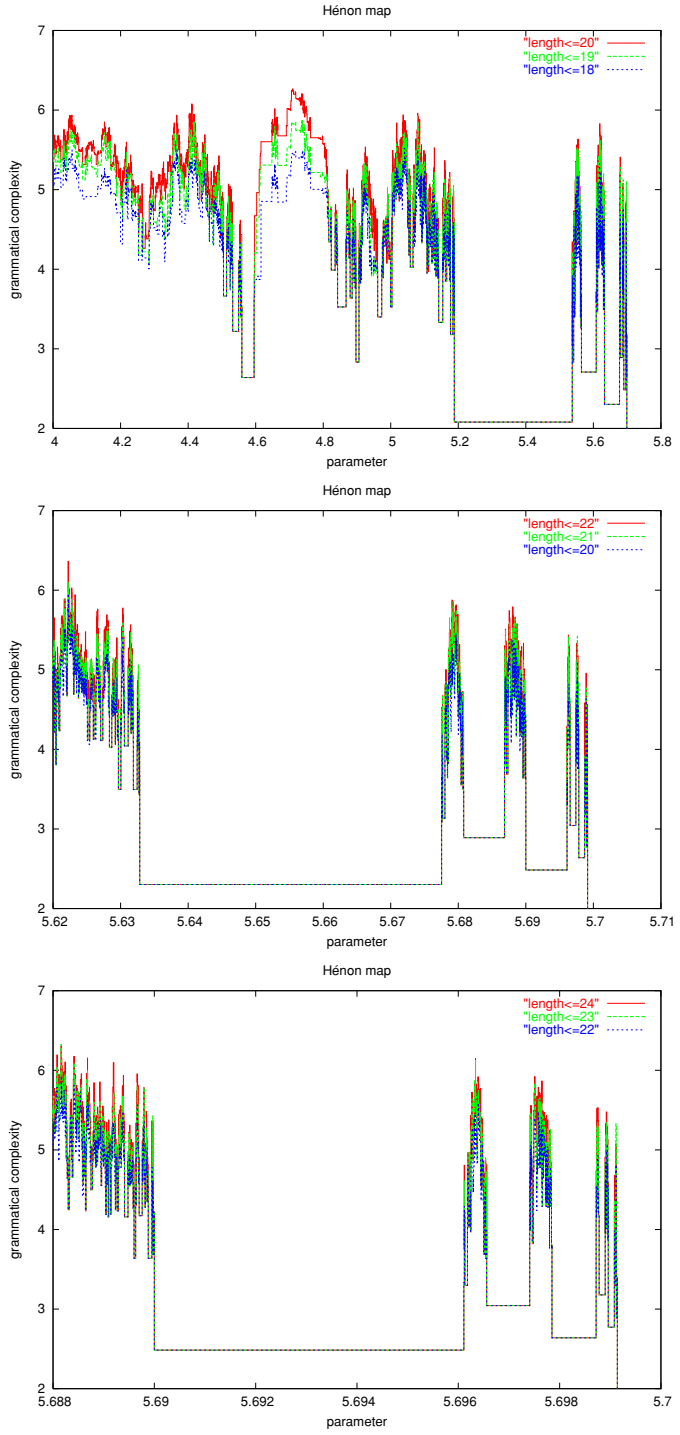

Figure 8. Grammatical complexity for the area-preserving Hénon map. Forbidden strings shorter than the specified length are included in the calculations.

Table 1. Parameter values in the hyperbolic intervals where the Hénon map and the Lozi map give the pruning fronts shown in figure 11, the forbidden strings, and the grammatical complexity. A symbol X stands for both 0 and 1.

Labels	Parameter		Forbidden strings		Grammatical complexity
	Hénon map	Lozi map			
a1	5.4	–	$0^2 1X10^2$		$\log 8$
a2	5.65	–	$0^3 1X10^3$		$\log 10$
a3	5.693	2.875	$0^4 1X10^4$		$\log 12$
a4	5.6983	2.9015	$0^5 1X10^5$		$\log 14$
	...	...			
b1	5.59	2.69	$0^2 1X10^3$,	$0^3 1X10^2 1$	$\log 15$
b2	5.684	2.832	$0^3 1X10^4$,	$0^4 1X10^3 1$	$\log 18$
b3	5.697	2.886	$0^4 1X10^5$,	$0^5 1X10^4 1$	$\log 21$
b4	5.6988	2.9057	$0^5 1X10^6$,	$0^6 1X10^5 1$	$\log 24$
	...	...			

As mentioned in the previous section, the order of the minimal structure matrix may not be finite when the system is not hyperbolic. We can see that this is indeed the case: according to the increase of resolution, that is, the increase of length of strings used to represent the pruned regions, the GC gradually becomes larger, which implies that it does not converge. In each figure, we only report three different resolution cases, but all the results thus examined tell us that this is a generic feature in non-hyperbolic situations. We can see, in addition to such divergent intervals, that there are many parameter intervals in which the GC already converges. Such intervals correspond to the ones in which the system has hyperbolic structures [23, 24].

As shown in figure 9, qualitatively the same behaviour is found for the Lozi map. We have performed the calculation of the GC by the same method as that used for the Hénon map, and also compared the result with that based on the method using the continued fractions [17]. We see from figure 10 that both give almost the same results, although there is a slight difference in the intervals where the GC does not seem to converge. This confirms the validity of our present calculations.

In either case, the figures demonstrate that, as a whole, the GC gradually increases as the parameter decreases from the first tangency point, and self-similar peaks are clearly noticed. In particular, in the vicinity of the first tangency point, they are remarkable. We can explain the reason for the appearance of such self-similarity by extracting the pruning fronts for a series of plateaux that are located at the same positions in each magnified figure. For the Hénon map, similar pruning fronts such as figures 11(a) and (b) are observed; these are sequences asymptotic to the first tangency point. These pruning fronts are obtained using the procedure of [22].

On the other hand, for the Lozi map, though the largest two pruning fronts in figure 11(a) are missing, the others are found. Such a difference may exist, because the shapes of the pruning fronts depend on the ordering of bifurcations and it is connected to the system-specific combinatorial properties of a given map. The two series shown in the figures are the simplest possible ones which we can easily extract manually, but it is natural to expect that self-similar peaks reflect such underlying structures of the primary pruned regions. Table 1 shows the parameter values at which the pruning fronts in figure 11 appear, together with the forbidden strings and the corresponding GC.

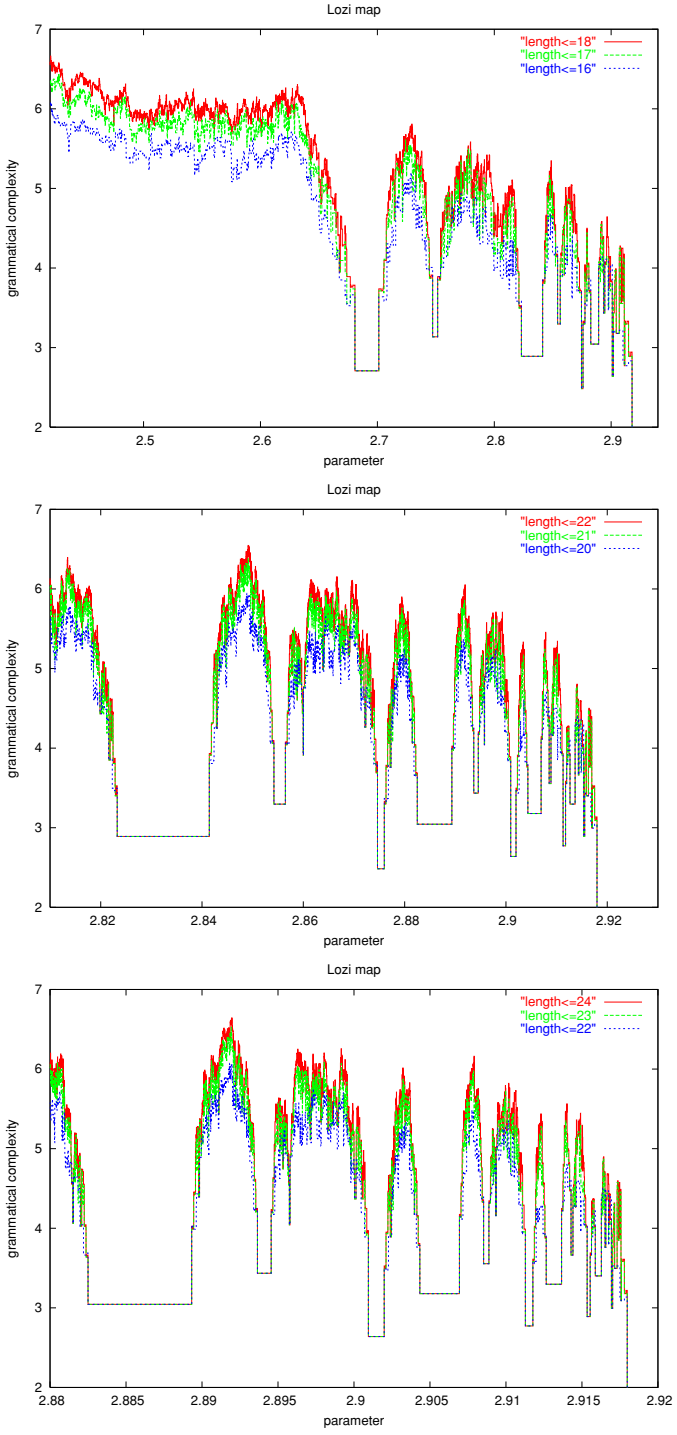

Figure 9. Grammatical complexity for the area-preserving Lozi map.

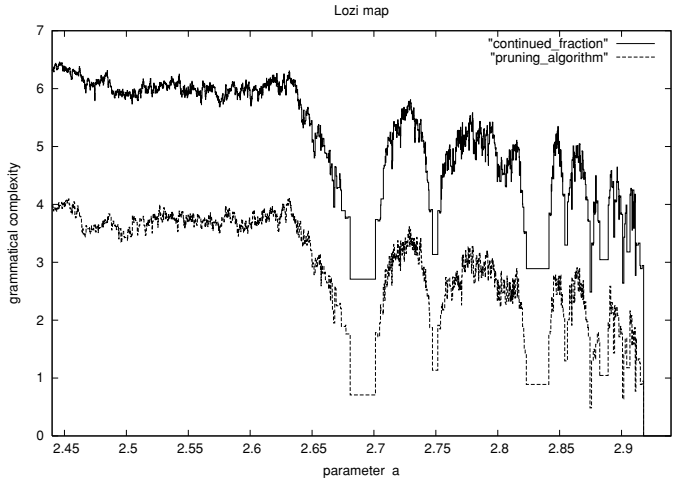

Figure 10. Comparison between the result using the continued fractions [17] and that using our pruning algorithm [22]. Although the latter, shifted by two to distinguish the results, tends to be slightly smaller than the former, they coincide at relatively long plateaux. The maximal forbidden strings taken in the calculations is 18.

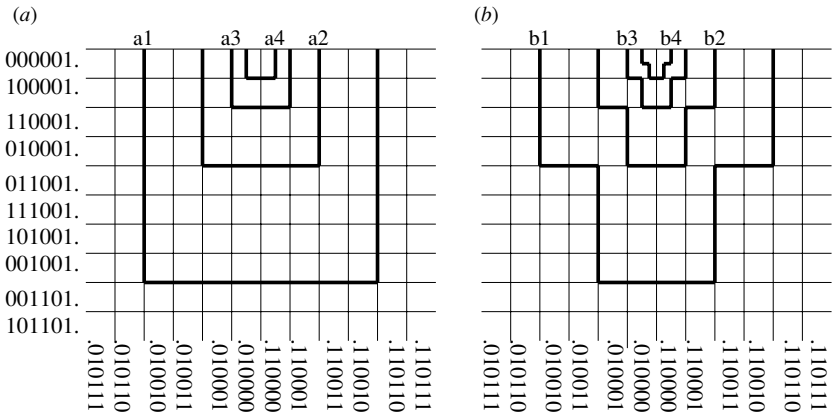

Figure 11. Similar pruning fronts observed near the first tangency point. The Hénon map has all the fronts in both (*a*) and (*b*), while the Lozi map does not have (a1) and (a2). The labels in the figure correspond to those in table 1.

5. Conclusion and discussion

In this paper, we have investigated the grammatical complexity (GC) of the symbol sequences generated from the Hénon and the Lozi maps. Explicit algorithms to give the pruning fronts enabled us to determine a minimal deterministic finite automaton (DFA) and thus to compute the GC. For both maps, there exist ranges of the system parameter where the horseshoe dynamics is realized. The GC in these regions is trivially zero. However, when the horseshoe structure breaks, the invariant sets of both maps become quite complicated. We are interested in how the complexity behaves as a function of the system parameter. What we have found is that there exist remarkable self-similar peak structures, whereas a gradual increase of the GC is observed as a whole. By presenting several series of simple pruning fronts, we have

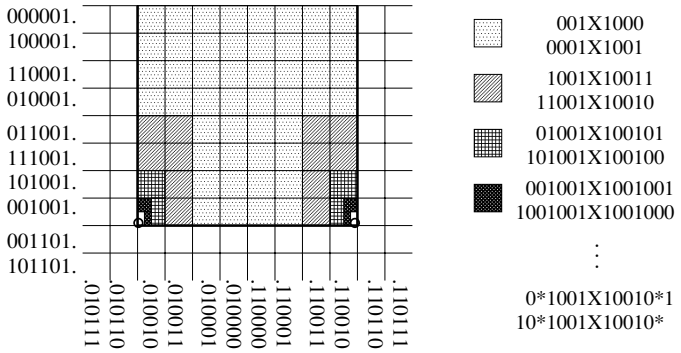

Figure 12. Pruning front for the upper endpoint of the longest plateau of the area-preserving Hénon map, $a = 5.537\ldots$. There exists a homoclinic tangency between $(0^\infty 1001 \cdot X10010^\infty)$ denoted by circles. However, the primary pruned region, which is the block $001 \cdot X100$ other than these two points, is specified by a regular expression.

argued that the origin of self-similar peaks can be attributed to similar-shaped primary pruned regions.

Obviously, hyperbolicity of the dynamics makes a language regular, and then the GC takes a finite value. On the other hand, one may expect that the GC diverges for non-hyperbolic parameter intervals, but it is not so evident. This is because a language can be regular even if a homoclinic tangency exists. For example, at the upper endpoint of the longest plateau of the area-preserving Hénon map, $a = 5.537\ldots$, two homoclinic points denoted by $(0^\infty 1001 \cdot X10010^\infty)$ $(X = 0, 1)$ degenerate. Note that they are the last pruned points in the block $001 \cdot X100$. Nevertheless, as shown in figure 12, we can easily specify the corresponding pruned region only by identifying the missing points as

$$001X1000 + 0001X1001 + 0^*1001X10010^*1 + 10^*1001X10010^*.$$

The language at this parameter value is obviously regular.

According to the works of Xie and Wang [10, 11], the language generated from a unimodal map is regular if and only if the kneading sequence of the map is either periodic or eventually periodic. Furthermore, Xie *et al* conjecture that a unimodal map may not generate a proper context-free language [12, 13]. A similar question arises for two-dimensional maps. It is clear that one of the sufficient conditions for two-dimensional maps is hyperbolicity. However, as noted in a brief argument made above, the necessary conditions are far from trivial. For non-hyperbolic parameter regions, as pointed out in [15], the pruning fronts may have infinitely many fractal steps, unlike those with finitely many steps (see for example figure 11). Nevertheless, the language can still be regular, because one has a finite automaton representing such a pruning front if the rule to construct self-similar steps is explicitly specified. Here we assume that there are ranges of the system parameter in which self-similar pruning fronts are realized, but checking this assumption itself is not a trivial issue. In the area-preserving case, a recent result is suggestive of our problem: the Hénon map cannot have finite-to-one symbolic representation by subshift if there exists an elliptic point [29]. It is interesting to study whether or not such a situation is beyond the category of regular languages.

Finally, we remark on the relation of self-similar peaks found here to the fractal diffusion coefficient discovered in deterministic diffusion [30–32]. In [30, 31], parameter-dependent deterministic diffusion in a one-dimensional array of a piecewise linear map has been studied. The authors found that the diffusion coefficient exhibits a fractal structure as a function of the control parameter. They have analysed the origin by developing a way to

calculate the parameter-dependent Markov partitions and to solve eigenvalue problems for the corresponding transition matrices. In the map considered there, the slope of the map function is greater than 1, so the system is always hyperbolic. An interesting fact is that the diffusion coefficient, which is a typical macroscopic quantity, depends sensitively on the change of the control parameter. Since the diffusion coefficient can be connected with the eigenvalue of transition matrices, it is natural to expect that underlying Markov partitions change sensitively as a function of the control parameter.

Similar fractal structures are also observed for two-dimensional billiard problems [32], where small elliptic islands may be partly responsible for irregular oscillations of the diffusion coefficient as a function of the energy. The situation is quite analogous to what was found in our analysis. For the Hénon and the Lozi maps, the shape of the primary pruned region, which gives the corresponding Markov partitions as well, changes in a self-similar manner. Although we have not calculated macroscopic quantities, the underlying mechanism generating self-similar patterns must be common.

Acknowledgments

The authors would like to thank Y Ishii and A Sannami for stimulating discussions and helpful comments.

References

[1] Badii R and Politi A 1997 *Complexity—Hierarchical Structures and Scaling in Physics* (Cambridge: Cambridge University Press)
[2] Wolfram S 1984 Computation theory of cellular automata *Commun. Math. Phys.* **96** 15–57
[3] Grassberger P 1986 Toward a quantitative theory of self-generated complexity *Int. J. Theor. Phys.* **25** 907–38
[4] D'Alessandro G and Politi A 1990 Hierarchical approach to complexity with applications to dynamical systems *Phys. Rev. Lett.* **64** 1609–12
[5] Auerbach D and Procaccia I 1990 Grammatical complexity of strange sets *Phys. Rev.* A **41** 6602–14
[6] Hao B-L 1991 Symbolic dynamics and characterization of complexity *Physica* D **51** 161–76
[7] Crutchfield J P and Young K 1989 Inferring statistical complexity *Phys. Rev. Lett.* **63** 105–8
[8] Crutchfield J P and Young K 1990 Computation at the onset of chaos *Complexity, Entropy, and Physics of Information* ed W Zurek (New York: Addison-Wesley) pp 223–69
[9] Lakdawala P 1996 Computational complexity of symbolic dynamics at the onset of chaos *Phys. Rev.* E **53** 4477–85
[10] Xie H-M 1993 On formal languages in one-dimensional dynamical systems *Nonlinearity* **6** 997–1007
[11] Wang Y and Xie H-M 1994 Grammatical complexity of unimodal maps with eventually periodic kneading sequences *Nonlinearity* **7** 1419–36
[12] Xie H-M 1996 *Grammatical Complexity and One-Dimensional Dynamical System* (Singapore: World Scientific)
[13] Wang Y, Yang L and Xie H-M 1999 Complexity of unimodal maps with aperiodic kneading sequences *Nonlinearity* **12** 1151–76
[14] Chomsky N 1959 On certain formal properties of grammars *Inf. Control* **2** 137–67
[15] Cvitanović P, Gunaratne G and Procaccia I 1988 Topological and metric properties of Hénon type strange attractors *Phys. Rev.* A **38** 1503–20
[16] Cvitanović P 1991 Periodic orbits as the skeleton of classical and quantum chaos *Physica* D **51** 138–51
[17] Ishii Y 1997 Towards a kneading theory for Lozi mappings I: a solution of the pruning front conjecture and the first tangency problem *Nonlinearity* **10** 731–47
[18] de Carvalho A 1999 Pruning fronts and the formation of horseshoes *Ergod. Theor. Dynam. Syst.* **19** 851–94
[19] de Carvalho A and Hall T 2002 How to prune a horseshoe *Nonlinearity* **15** R19–68
[20] Lozi R 1978 Un attracteur étrange(?) du type attracteur de Hénon *J. Phys. (Paris)* **39** Colloq. C5 9–10
[21] Hénon M 1976 A two-dimensional mapping with a strange attractor *Commun. Math. Phys.* **50** 69–77
[22] Hagiwara R and Shudo A 2004 An algorithm to prune the area-preserving Hénon map *J. Phys. A: Math. Gen.* **37** 10521–43
[23] Davis M J, MacKay R S and Sannami A 1991 Markov shifts in the Hénon family *Physica* D **52** 171–8

[24] Lai Y C, Grebogi C, Yorke J A and Kan I 1993 How often are chaotic saddles nonhyperbolic? *Nonlinearity* **6** 779–97
[25] Hopcroft J E and Ullman J D 1979 *Introduction to Automata Theory, Languages, and Computation* (Reading MA: Addison-Wesley)
[26] Milnor J and Thurston W 1988 On iterated maps of the interval *Dynamical Systems* (Berlin: Springer) pp 465–563
[27] Sterling D, Dullin H R and Meiss J D 1999 Homoclinic bifurcations for the Hénon map *Physica* D **134** 153–84
[28] Sterling D 1999 Anti-integrable continuation and the destruction of chaos *PhD Thesis* University of Colorado
[29] Sannami A 2003 Nonexistence of symbolic representation for discrete dynamical systems *Preprint*
[30] Klages R and Dorfman J D 1995 Simple maps with fractal diffusion coefficients *Phys. Rev. Lett.* **74** 387–90
[31] Klages R and Dorfman J D 1999 Simple deterministic dynamical systems with fractal diffusion coefficients *Phys. Rev.* E **59** 5361–83
[32] Harayama T and Gaspard P 2001 Diffusion of particles bouncing on a one-dimensional periodically corrugated floor *Phys. Rev.* E **64** 036215-1