*International Journal of Theoretical Physics, Vol. 36, No. 4, 1997*

# Exact Complexity of the Logistic Map

**W.-H. Steeb[1,2] and R. Stoop[1]**

*Received August 7, 1996*

The relation between chaotic behavior and complexity for one-dimensional maps is discussed. The one-dimensional maps are mapped into a binary string via symbolic dynamics in order to evaluate the complexity. We apply the complexity measure of Lempel and Ziv to these binary strings. To characterize the chaotic behavior, we calculate the Liapunov exponent. We show that the exact normalized complexity for the logistic map $f$: $[0, 1] \to [0, 1], f(x) = 4x(1 - x)$ is given by 1.

Since the discovery of chaotic attractors, chaos has become an important concept in nearly all branches of the natural sciences. The difference and differential equations which are believed to govern our natural world and may exhibit chaotic attractors are widely discussed in literature (see, for example, Steeb 1992a,b, 1996). The simplest systems showing chaotic behavior are one-dimensional maps $f$: $I \to I$, where $I$ is an interval. A definition for chaos is as follows: Let $X$ be a set. The mapping $g$: $X \to X$ is said to be chaotic on $X$ if (1) $g$ has sensitive dependence on initial conditions, (2) $g$ is topological transitive, and (3) periodic points are dense in $X$. Here we use the Liapunov exponent to characterize chaos. The exponent measures the sensitive dependence on initial conditions.

Many different definitions of complexity have been proposed in the literature. Among them are: algorithmic complexity (Kolmogorov–Chaitin) (Chaitin 1987), Lempel–Ziv complexity (Lempel and Ziv, 1976), the logical depth of Bennett (Bennett, 1988), the effective measure complexity of Grassberger (Grassberger, 1986), the complexity of a system based in its diversity (Huberman and Hogg, 1986), the thermodynamic depth (Lloyd and Pagels, 1988), and a statistical measure of complexity (Lopez-Ruiz *et al.*,

[1] Department of Applied Mathematics and Nonlinear Studies, Rand Afrikaans University, Auckland Park 2006, South Africa; e-mail: WHS@RAU3.RAU.AC.ZA.
[2] National University of Singapore, Computational Science Programme, Republic of Singapore.



0020-7748/97/0400-0949$12.50/0 © 1997 Plenum Publishing Corporation

1995). In the following we consider the Lempel–Ziv complexity. A definition of the complexity (Chaitin, 1987) of a binary string (a string of zeros and ones) is given by the number of bits of the shortest computer program which can generate this string. A general algorithm which determines such a program cannot be given. Lempel and Ziv have chosen from all possible programs one class that allows only two operations: copying and inserting. For the reconstruction of the given binary string of length $n$, using these two operations, they have introduced a complexity measure $c(n)$. Here an algorithm can be given. Of course, a binary string consisting only of 0's (or 1's) must have the lowest complexity, namely 2. A string consisting of a sequence of 01's, i.e., ``010101...01,'' has complexity 3.

Let us now describe how the complexity (Lempel and Ziv, 1976; Steeb, 1996) is evaluated. Given a binary string $S = s_1 s_2 \ldots s_n$ of finite length $n$, let $A^*$ denote the set of all finite length sequences (strings) over a finite alphabet $A$ (in our case $\{0, 1\}$). The quantity $S(i, j)$ denotes the substring $S(i, j) := s_i s_{i+1} \ldots s_j$. A vocabulary of a string $S$, denoted by $v(S)$, is the subset of $A^*$ formed by all the substrings, or words, $S(i, j)$ of $S$. The complexity in the sense of Lempel and Ziv of a finite string is evaluated from the point of view of a simple self-delimiting learning machine, which, as it scans a given $n$-digit string $S = s_1 s_2 \ldots s_n$ from left to right, adds a new word to its memory every time it discovers a substring of consecutive digits not previously encountered. Thus the calculation of the complexity $c(n)$ proceeds as follows. Let us assume that a given string $s_1 s_2 \ldots s_n$ has been reconstructed by the program up to the digit $s_r$ and that $s_r$ has been newly inserted, i.e., it was not obtained by simply copying it from $s_1 s_2 \ldots s_{r-1}$. The string up to $s_r$ will be denoted by $R := s_1 s_2 \ldots s_r \circ$, where the $\circ$ indicates that $s_r$ is newly inserted. In order to check whether the rest of $R$, i.e., $s_{r+1} s_{r+2} \ldots s_n$, can be reconstructed by simple copying or whether one has to insert new digits, we proceed as follows: First, one takes $Q \equiv s_{r+1}$ and asks whether this term is contained in the vocabulary of the string $R$ so that $Q$ can simply be obtained by copying a word of $R$. This is equivalent to the question of whether $Q$ is contained in the vocabulary $v(RQ\pi)$ of $RQ\pi$, where $RQ\pi$ denotes the string which is composed of $R$ and $Q$ (concatenation) and $\pi$ means that the last digit has to be deleted. This can be generalized to situations where $Q$ also contains two (i.e., $Q = s_{r+1} s_{r+2}$) or more elements. Let us assume that $s_{r+1}$ can be copied from the vocabulary of $R$. Then we next ask whether $Q = s_{r+1} s_{r+2}$ is contained in the vocabulary of $RQ\pi$ and so on until $Q$ becomes so large that it can no longer be obtained by copying a word from $v(RQ\pi)$ and one has to insert a new digit. The number $c$ of production steps to create the string $S$, i.e., the number of newly inserted digits (plus one if the last copy step is not followed by inserting a digit), is used as a measure of the complexity of a given string. Of course, a binary string consisting only of

0's (or 1's) must have the lowest complexity, namely 2, since `0∘000....` A string consisting of a sequence of `01`'s, i.e., "`010101...01`," has complexity 3, since `0∘1∘0101...01`. In order to obtain a complexity measure which is independent of the string length, we use a normalized complexity measure. For very large strings it makes sense to normalize them. To normalize them, we consider the interval [0, 1]. The rational numbers in this interval are of Lebesgue measure zero. For almost all numbers in the interval [0, 1] (the irrational numbers) the string of zeros and ones which represents their binary decomposition is not periodic. Therefore almost all strings which correspond to a binary representation of a number $x \in [0, 1]$ should be random and have maximal complexity. The complexity tends to the same value for $n \to \infty$, namely $n/\log_2 n$. We use this quantity to normalize the complexity $c(n)$. Thus the largest value the complexity can have is equal to 1.

Next we have to find a binary string from the one-dimensional map. This is done by using symbolic dynamics (Steeb, 1992a,b; Fang, 1994). To construct the symbolic dynamics of a dynamical system, the determination of the partition and the ordering rules for the underlying symbolic sequences is of crucial importance. In the case of one-dimensional maps, the partition is composed of all the critical points. We consider maps $f$: $[0, 1] \to [0, 1]$ with one critical point at $x = 1/2$. Thus we divide the phase space into two intervals [0, 1/2) and [1/2, 1]. If the iterate is in the interval [0, 1/2), we map into 0, and if the iterate is on the right-hand side, we map into 1. When we consider, for example, the logistic map $x_{t+1} = 4x_t(1 - x_t)$ and $x_0 = 1/3$ we find the sequence "`01010110....`"

The rational numbers in the interval [0, 1] are of Lebesgue measure zero. Thus for almost all numbers in [0, 1] (i.e., for all irrationals) the string of zeros and ones which gives their binary representation is not periodic. Therefore, almost all strings which correspond to a binary representation of a number in the interval [0, 1] are random and have maximal complexity. Lempel and Ziv (1976) have shown that for almost all $x \in [0, 1]$ the complexity $c(n)$ of the binary string which gives the binary representation tends to the same value

$$\lim_{n\to\infty} c(n) \equiv b(n) = \frac{n}{\log_2 n}$$

Thus $b(n)$ gives the asymptotic behavior of $c(n)$. Thus we normalize $c(n)$ with respect to $b(n)$. A C++ program that finds the complexity $c(n)$ for a given string of length $n$ is available on request from the authors (http://zeus.rau.ac.za/steeb/steeb.html).

The logistic map is the most studied equation with chaotic behavior. All quantities of interest in chaotic dynamics can be calculated exactly. Examples are the fixed points and their stability, the periodic orbits and their stability, the moments, the invariant density, topological entropy, the metric entropy, Liapunov exponent, autocorrelation function, and the exact solution. The Liapunov exponent for almost all initial conditions is given by ln(2). The map is also ergodic and mixing.

We now show that the normalized complexity of the logistic map is given by 1. The tent map $g$: $[0, 1] \to [0, 1]$

$$g(x) := \begin{cases} 2x & \text{if} \quad x \in [0, 1/2] \\ 2(1 - x) & \text{if} \quad x \in (1/2, 1] \end{cases}$$

is chaotic if the initial value $x_0$ is an irrational number in the interval [0, 1]. Thus, as described above, the symbolic dynamic leads to a binary string with maximal complexity if the initial value is an irrational number. Thus the normalized complexity is 1 for $n \to \infty$. Moreover, we notice that the Liapunov exponent is equal to ln 2 for a chaotic orbit.

Next we show that the tent map and the logistic map are topologically conjugate. Let $f$: $A \to A$ and $g$: $B \to B$ be two maps. $f$ and $g$ are said to be topologically conjugate if there exists a homeomorphism $h$: $A \to B$ such that, $h \circ f = g \circ h$. The homeomorphism $h$ is called a topological conjugacy. Mappings which are topologically conjugate are completely equivalent in terms of their dynamics. For example, if $f$ is topologically conjugate to $g$ via $h$, and $p$ is a fixed point for $f$, then $h(p)$ is a fixed point for $g$. The Liapunov exponent and the complexity are preserved under the homeomorphism $h$. Now the tent map and the logistic map are topologically conjugate. The homeomorphism (which is even a diffeomorphism) $h$: $[0, 1] \to [0, 1]$ is given by $h(x) = (2/\pi) \arcsin \sqrt{x}$. Since the complexity is preserved under the diffeomorphism $h$, we proved that the normalized exact complexity of the logistic map is equal to 1.

Finally, a comment is in order in calculating the Lempel and Ziv complexity. We assumed that the information needed in Lempel and Ziv coding is one unit for each new word. Only then do we obtain complexity 3 for an alternating string "`0101010...`," and only then would a random string have complexity $n/\log_2 n$. One is of course free to count complexity in any units one wants. The usual units are bits. In this case the information needed to specify one among $n$ words is $\approx \log_2 n$ instead of 1. Thus the Lempel and Ziv complexity of a fully random binary string is 1 bit/symbol and not $n/\log_2 n$ units. The average Lempel and Ziv complexity per symbol for random strings coincides with the Shannon entropy, and hence the Lempel and Ziv complexity per symbol for the logistic map at fully developed chaos is equal to 1 bit/symbol. This assumes already the limit $n \to \infty$. For finite $n$ there are several sources of logarithmic corrections, one of them being that the Lempel and Ziv complexity is only defined for finite strings in this picture, whence one has to specify also the string length. Thus the Lempel and Ziv complexity for the alternating string "`0101010...`" of length $n$ is $\geq \log_2 n$ in this picture, and not finite.

## REFERENCES


Bennett, C. H. (1988). In *Emerging Syntheses in Science,* D. Pines, ed., Addison-Wesley, Reading, Massachusetts.

Chaitin, G. J. (1987). *Information, Randomness and Incompleteness,* World Scientific, Singapore.

Fang, H. P. (1994). *Journal of Physics A: Mathematical and General,* **27**, 5187.

Grassberger, P. (1986). *International Journal of Theoretical Physics,* **25**, 907.

Huberman, B. A., and Hogg, T. (1986). *Physica D,* **22**, 376.

Lempel, A., and Ziv, J. (1976). *IEEE Transactions on Information Theory,* **IT-22**, 75.

Lloyd, S., and Pagels, H. (1988). *Annals of Physics,* **188**, 186.

Lopez-Ruiz, R., Mancini, H. L., and Calbet, X. (1995). *Physics Letters A,* **209**, 321.

Steeb, W.-H. (1992a). *A Handbook of Terms Used in Chaos and Quantum Chaos,* BI-Wissenschaftsverlag, Mannheim.

Steeb, W.-H. (1992a). *Chaos and Fractals: Algorithms and Computations,* Spektrum Akademischer Verlag, Heidelberg.

Steeb, W.-H. (1994). *Chaos und Quanten Chaos in Dynamischen Systemen,* Spektrum Akademischer Verlag, Heidelberg.

Steeb, W.-H. (1996). *Problems and Solutions in Theoretical and Mathematical Physics,* Volume II, *Advanced Problems,* World Scientific, Singapore.