International Journal of Theoretical Physics, Vol. 25, No. 9, 1986

Toward a Quantitative Theory of Self-Generated Complexity

Peter Grassberger[1]

Received March 14, 1986

Quantities are defined operationally which qualify as measures of complexity of patterns arising in physical situations. Their main features, distinguishing them from previously used quantities, are the following: (1) they are measure-theoretic concepts, more closely related to Shannon entropy than to computational complexity; and (2) they are observables related to ensembles of patterns, not to individual patterns. Indeed, they are essentially Shannon information needed to specify not individual patterns, but either measure-theoretic or algebraic properties of ensembles of patterns arising in *a priori* translationally invariant situations. Numerical estimates of these complexities are given for several examples of patterns created by maps and by cellular automata.

1. INTRODUCTION

Sciences like biology or information theory have always been confronted with the problem of describing complex systems. Physics has long been able to avoid complex situations and to concentrate on systems that are comparatively simple, either since few degrees of freedom were involved or since in systems with large numbers of degrees of freedom one can apply central limit theorems. But recently it has become clear that both reasons are not sufficient to avoid complex behavior: on the one hand, even very simple systems with few degrees of freedom can show very complex behavior if they are "chaotic" (Schuster, 1984; Guckenheimer and Holmes, 1983); on the other hand, systems with many degrees of freedom, such as cellular automata (Wolfram, 1983), can behave such that central limit theorems need not be applicable (Wolfram, 1984b). Natural situations where these problems appear are, e.g., time series from nonlinear electronic circuits, the pattern of the reversals of the earth's magnetic field, and spatial patterns

[1]Physics Department, University of Wuppertal, D56 Wuppertal 1, Gauss-Strasse 20, West Germany.

0020-7748/86/0900-0907$05.00/0

in Bernard experiments and in not-well-stirred oscillating chemical reactions.

One characteristic common to all these instances is that the complexity is self-generated, in the sense that the formulation of the problem is translationally invariant and the observed structure arises from a spontaneous breakdown of translational invariance.

Confronted with situations that intuitively are judged as "complex," one of the first reactions should be to quantify this judgment by defining an observable. This is what the present paper aims at.

Indeed, there have been several attempts to define complexity formally, although all of them have severe drawbacks when applied to the problems at hand. Also, they are widely ignored by active researchers in the field of self-generated complexity, and the notion of complexity is sometimes used to mean different things. The present paper was most influenced by the seminal work by Wolfram (1984b). But the notion closest to the present approach is described in a small booklet (van Emden, 1976) by a taxonomist interested in finding the least complex scheme to organize living species. A similar approach toward measuring the structure of living beings is due to Chaitin (1979).

Much better known is the concept of computational complexity (Hopcroft and Ullman, 1979), and it might at first seem natural to take over the concepts used there. This was indeed done quite successfully in Wolfram (1984b), but, as we shall show, that approach has certain drawbacks. The main problem is that computation theory deals mainly with the possible and not with the probable (although practitioners of course also apply ad hoc probabilistic concepts there!). It is an algebraic theory and not a measure-theoretic one. This can be seen, e.g., from Hofstadter's (1979) book, which does not once mention the notions of entropy or Shannon information, although its index has about 1500 other entries. For applications to physics, this is disastrous: a theorist of complexity, confronted with the problem of describing an ideal gas, could not use even such basic notions as temperature or pressure. Thus, our first requirement of physically useful measures of complexity is that they be probabilistic.

The other problem is a conundrum probably known for some time to many, although it seems to have appeared in print only recently (Hogg and Huberman, 1985). It is that the intuitive notion of complexity of a pattern does not agree with the only objective definition of the complexity of any specific pattern that seems possible.[2] This latter definition is due to Kolmogorov (Alekseev and Yakobson, 1981). The Kolmogorov complexity of a pattern is essentially the length of the shortest program on a general-purpose computer needed to generate that pattern, divided by the size of

[2]See note added in proof.

the pattern itself. (In order to make this meaningful, one has to take the limit of infinitely large patterns. We assume that we should take this limit anyhow, throughout the following.) Thus, it is some kind of information per "pixel" or per "letter" stored in that pattern, and in the cases in which we are interested it seems to agree with the Shannon information (Shannon and Weaver, 1949) or specific entropy. Kolmogorov complexity seems to be the quantity most closely related to the intuitive notion of randomness, not of complexity. See Section 6 for a further discussion of this point.

Compare now the three patterns shown in Fig. 1. Fig. 1c is made by using a random number generator. Kolmogorov complexity and Shannon entropy are biggest for it, and smallest for Fig. 1a. On the other hand, most people will intuitively call Fig. 1b the most complex, since it seems to have more "structure." Thus, complexity in the intuitive sense is not monotonically increasing with entropy or "disorder." Instead, it is small for completely ordered and for completely disordered patterns, and has a maximum in between (Hogg and Huberman 1985). This agrees with the notion that living mattter should be more complex than both perfect crystals and random glasses, say.

The solution of this puzzle is the well-known ability of humans to make abstractions, i.e., to distinguish intuitively between "important" and "unimportant" features. For instance, when one is shown pictures of animals, one immediately recognizes the concepts "dog," "cat," etc., although the individual pictures showing dogs might in other respects be very different. So one immediately classifies the pictures into sets, with pictures within one set considered as equivalent. Moreover, these sets carry probability measures (since one expects not all kinds of dogs to appear equally often, and to be seen equally likely from all angles). Thus, one actually has ensembles: when calling a random pattern complex or not, one actually means that the ensemble of all "similar" patterns (whatever that means in detail) is complex or not complex. After all, if the pattern in Fig. 1c were made with a good random number generator, the chance of producing precisely Fig. 1c would be exactly the same as that to produce Figs. 1a or 1b (namely 2^{-N}, where N is the total number of pixels). If we call the latter more "complex," it really means that we consider it implicitly to belong to a different ensemble, and it is this ensemble that has different complexity.

Thus it is clear that our measures of complexity will be the Shannon information needed to describe properties of ensembles of patterns. This still does not specify these measures completely.

Before going on, we have to restrict ourselves to a situation typical for the self-generated patterns in which we are interested. Although this need not always be the case (Fig. 1b is a counterexample), we assume that our ensembles (not the individual patterns!) are translationally invariant,

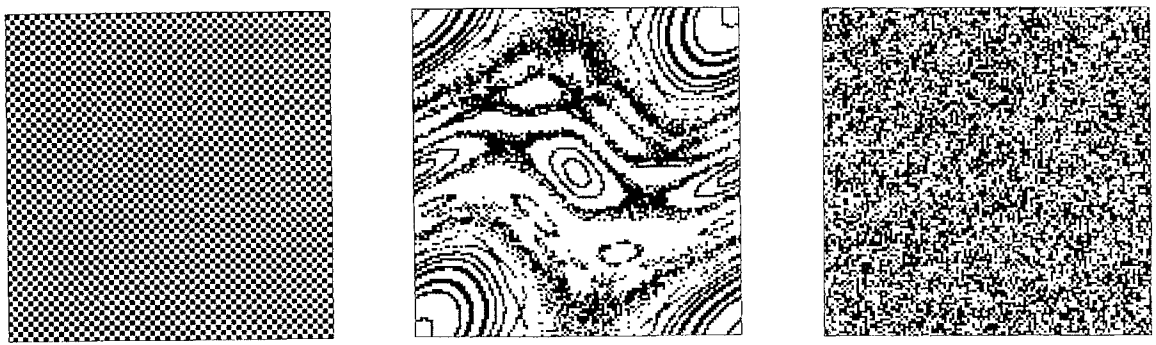

Fig. 1. Three patterns used to demonstrate that the pattern that one intuitively would call the most complex (b) is neither the one with highest entropy [and Kolmogorov complexity; (c)] nor the one with lowest (a).

and the patterns can be extended toward infinity, with everywhere the same average features.

The easiest situation, studied in most of this paper, prevails in the case of one-dimensional patterns, consisting just of (infinite) strings of "letters" (pixels, digits, spins, etc.; higher dimensional patterns could also be translated into strings, but translational invariance would be lost thereby). Let us discuss the one-dimensional case first, with higher dimensions deferred to Section 4.

Also, we shall only be concerned with discrete patterns. Questions related to the discretization of continuous patterns will not be discussed.

In a first approach, we are interested in describing only the sets making up the ensemble, disregarding probabilities. Consider an infinite string $\{s_i | i \in \mathbb{Z}\}$ of "letters." We are then interested in deciding whether or not this string belongs to some set defined by a suitable grammar. Assume that we have checked that the string up to s_i does belong to the set, and we want to check s_{i+1}, s_{i+2}, and so on. For checking s_{i+1}, we of course have to know something about the previous letters. We define as the *set complexity* (SC) the average Shannon information needed to be stored for that. Notice that this is not the information per letter needed to specify the particular sequence considered, since we do not want to actually specify s_{i+1}. Rather, it is a measure of how complicated the grammar was. For the regular grammars (Hopcroft and Ullman, 1979), the SC is easily seen to be a lower bound to the complexity defined by Wolfram (1984b). The latter, which we call *algorithmic complexity* (AC), is related to the SC in a way similar to the relation between topological and metric entropies of dynamical systems (Eckmann and Ruelle, 1985). More details will be given in Section 2.

The other way to define complexity measure-theoretically, discussed in Section 3, again uses the same sequence $\ldots, s_i, s_{i+1}, \ldots$ as above. Now, however, we are not content with verifying that the entire sequence is grammatically correct; we also want it to be "stylistically" correct, i.e., we want it to have the right statistical properties. So we not only have to exclude "wrong" letters, we also want to predict the actual ones as well as possible. If the language has positive entropy, we cannot predict s_{i+1} from $\{s_i, s_{i-1}, \ldots\}$ completely, of course, but we can make optimal predictions in the sense of minimal uncertainty. For such an optimal prediction we again have to store some minimal information about s_i, s_{i-1}, etc. The average of this latter information is called the *true measure complexity* (TMC) in the following.

The last complexity-like observable we shall discuss is the *effective measure complexity* (EMC). It is not the information needed to be stored for an optimal prediction, but it is the "value" of that information in helping to predict. At first sight, one might be tempted to believe that the stored information could always be used so efficiently that it is equal to the decrease

of uncertainty in the prediction. As we shall see in Section 4, this is not the case. The EMC, defined as the minimal information that would have to be stored for optimal predictions if it could be used with 100% efficiency, is sometimes strictly less than the TMC.

Among these four measures of complexity (SC, AC, TMC, and EMC), it is the last that we believe to be the most interesting in physics, since it seems to be the only one observable in general situations where the grammar is not known, i.e., where one does not yet understand the mechanism generating the patterns.

After introducing these concepts more formally in Sections 2–4, we discuss numerical examples in Section 5. These are, on the one hand, symbolic sequences generated by generating partitions in one-dimensional maps, and on the other hand they are time and space patterns generated by one-dimensional cellular automata.

In physics, it appears very often that a problem is characterized by several length scales, and this naturally suggests relevant ensembles. Compare, e.g., the complexities of a silicium crystal with with an array of computer chips on it, and of an equally large piece of glass. On the atomic level, the glass is more random and might well be more complex than the single crystal of the chip. But interest in most cases will not be in the atomic properties, but rather in the complexity of the layout of the chip. Thus, relevant ensembles are those where a "coarse-graining" is done with a resolution of ~ 100 Å. For the glass, the ensemble is then essentially the canonical ensemble of thermodynamics, and it has all complexities zero. In the case of the chips, the ensemble has positive complexities. This is not all, however. Each ensemble is again an element of a set (of all coarse-grained states), and we can consider now ensembles of coarse-grained states (i.e., of different layouts). It is these latter that we mean when when we say that the chips are complex but not random: the set of all functional layouts is much harder to describe than the set of all (random) layouts, although each single functional layout has vanishing entropy, being a periodic array of identical chips.

This example also shows that the most intriguing cases are those with large complexity and small entropy. Our most interesting numerical results in Section 5 thus concern patterns arising from a random input and having zero entropy but infinite complexity, the latter measured by the EMC.

Finally, we should mention that very similar constructs can also be applied to the problem of coding and decoding. The encoding complexity of a code is the average amount of information by which the sent encoded message lags behind the message received by the encoder. It is also the average amount of information that the encoder has to store during the process of encoding. The decoding complexity is obviously defined in an analogous way.

2. ALGORITHMIC AND SET COMPLEXITIES

We consider a set $\sum$ of k symbols s_i ("alphabet") and strings $\ldots, s_i, s_{i+1}, \ldots$ of arbitrary length formed from these symbols. The index i will be called "time" in the following. Not all strings are allowed. Instead, one assumes a set of rules ("grammar") which are to be strictly followed in the allowed strings. In addition, we assume that a probability measure is given on the allowed strings. Notice that we could take the grammatical rules as part of the definition of the probability measure. We shall not do that, and consider the grammar as separate in order to stay as close to algorithmic complexity theory as possible. More strongly, we shall demand that if any string is strictly forbidden, then this is always due to the grammar and not due to a vanishing measure. We shall sometimes call the weighted set of all strings a "style," in order to distinguish it from the "language" defined by the grammar alone. In the terminology of physics, we are dealing with an ensemble of (string) patterns.

Our next assumptions are that both the grammar and the probability measure are invariant under time translations, and that the ensemble is ergodic. By the latter we mean that the probability measures on all finite substrings of any fixed infinite string are the same as the probability measures on all substrings of all infinite strings. Thus, we assume that we can study the ensemble numerically by studying one very long string only. In the examples studied in the next section, this seems to be the case.

Notice that the assumption of ergodicity is much less important and subtle than the assumption of translation invariance (or "stationarity"). In a nonergodic case, the ergodic components are in general enumerable, and we can just study each component by itself. To see the nontriviality of the assumption of stationarity, consider, e.g., a language where in each (infinite) sequence the letter "A" must occur exactly three times. Since these occurrences could be anywhere, one might at first argue that this is a stationary ensemble. Actually, it is not: after "A" has occurred three times, the effective grammar can be simplified from "A should occur three times" to "A is forbidden," and an optimal test for correctness of the sequence changes after the third occurrence. More generally, we demand that the grammar (and the probability measure) are such that they cannot be simplified during the observation of a sequence due to the occurrence of some special "signal."

2.1. Regular Languages

We consider first the case that the grammar is regular. Then, it can be implemented by a deterministic finite automaton (Hopcroft and Ullman, 1979) in the following sense. There is a finite directed graph with N nodes and with at most k arcs, labeled by different letters from the alphabet, leaving each node. As one scans the string, one simultaneously moves from

node to node by the following rule: if the node p has been reached at time i, then the letter s_{i+1} is allowed for the next step if and only if there is an arc labeled "s_{i+1}" starting from p. Depending on the letter actually observed, one follows the corresponding arc to the next node and repeats the same procedure [notice that this way of using an automaton for defining a set of strings is different from that used, e.g., in Christol et al. (1980) and Allouche and Cosnard (1984)].

Among all automata corresponding to a given language, there is one that is minimal in the sense that its graph contains the smallest number N of nodes. It is $\log(N)$ which is defined as the complexity of the language in Wolfram (1984b), and called AC in the present paper. It is easy to see that it is an upper bound on the SC, defined as the smallest average Shannon entropy stored about the past string for verifying the correctness of the future string. Indeed, the only information stored about the past is the actual position in the graph. Denoting the frequency of being at node i ($i=1,\ldots,N$) as $p(i)$, then

$$\mathrm{SC} = -\sum_{\text{nodes}} p(i)\log p(i) \tag{1}$$

If all nodes had an equal occupation probability, this would just be $\log(N)$.

There is a subtlety in this argument. Usually (Hopcroft and Ullman, 1979) one considers only languages of one-sided infinite strings. For these, there exists an algorithm which yields both the minimal automaton and the starting point on the automaton. In many cases, it might happen that this automaton contains a part which contains the starting point and is connected to the bulk of the graph only by arcs leading away from the start. An example representing the set of all strings of "0" and "1" and containing no blocks 010 or 111 is shown in Fig. 2a, while another example corresponding to exclude blocks 111 and $110^*(10^*)^{2^n}11$ is shown in Fig. 2b [Wolfram (1984b); the notation 0^* indicates any number of 0's, and $(\cdots)^n$ indicates a string of n blocks $10\cdots0$]. According to our definition, we would not call the movement in these graphs stationary. Instead, we could cut off the transient parts, and use only the reduced graphs shown in Figs. 3a and 3b. Notice that the reduction was much more severe in Fig. 3b than in Fig. 3a. In the latter, the transient part had to be left immediately, while in the former the cut-off part alone could accept all strings without two 1's in succession. In a similar way, we can have automata where the end point (defined as that point where the string is ultimately rejected as forbidden) is in a transient part which can only be entered but not left again. An example is presented in Fig. 4. Here, we shall again reduce the graph by truncating it, rendering the situation stationary (which, in our strict sense, it was not before reduction).

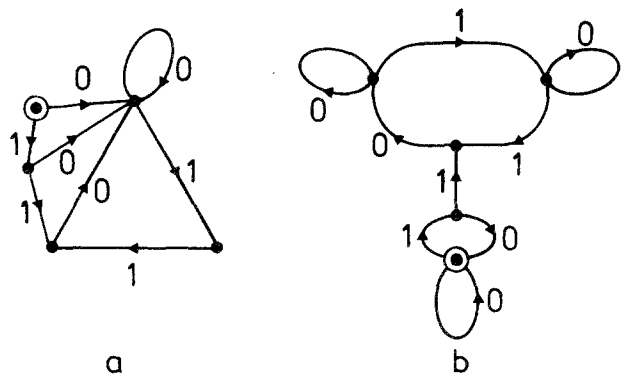

Fig. 2. Deterministic automata accepting strings (a) without sequences ...010... and ...111..., (b) without an odd number of isolated 1's between any two occurrences of neighboring pairs of 1's. Here and in subsequent figures the encircled node represents the starting point.

For information theoretic arguments using doubly-infinite strings, the reduced graphs of Fig. 3 are certainly sufficient. The problem with them is that we don't have a general proof that they are indeed the minimal ones (there might be other automata whose graphs had been bigger before reduction but after reduction have become smaller), and we don't any more have an algorithm telling us where we are in the graph at any given time. But even if we don't have an algorithm for that, we can learn where we are by observing the string, for almost all strings. Anyhow, the logarithm of the number of nodes in the reduced graph is an upper bound on the SC.

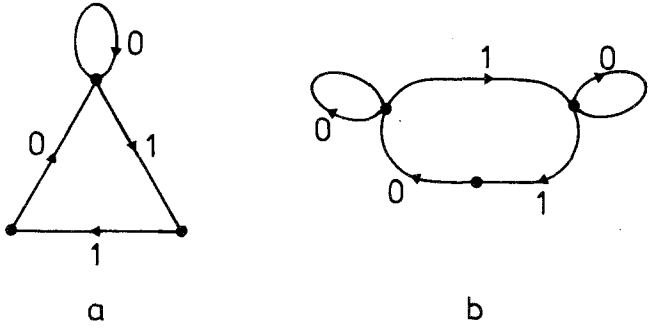

Fig. 3. Reduced automata obtained by truncating the transient parts in Fig. 2. Notice that these automata are no longer strictly deterministic, since the starting point is not defined.

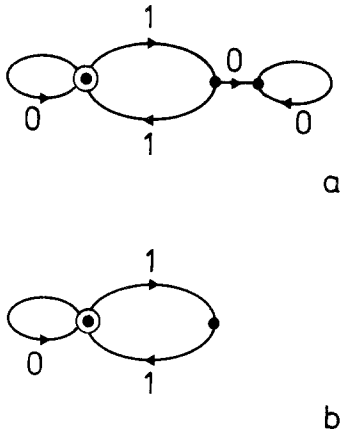

Fig. 4. (a) Automaton accepting all strings with at most one isolated 1. (b) Reduced automaton accepting no isolated 1's at all.

Obviously, the AC is related to the SC in a way similar to the relation between the topological and the metric entropies of dynamical systems (Eckmann and Ruelle, 1985). There, one considers symbolic sequences generated by partitions of state space. While the metric entropy is a measure for the information stored in the sequence, the topological entropy counts just the number of different sequences independently of any probability measure. At least in simple cases like iterated maps of an interval onto itself, there exists always a measure such that both entropies are the same, i.e., for which all sequences are roughly equally probable.

We shall now show that no analogous result can hold for complexities: there exist regular grammars for which there doesn't exist any probability measure for which AC = SC. The proof proceeds by giving counter examples. The simplest counter example is shown in Fig. 5, representing strings with 0's and 1's only occurring in pairs. For this graph, one has

$$\mathrm{SC} = \log 8 < \mathrm{AC} = \log 3 \tag{2}$$

So our next obvious question is whether there exists any grammar and style for which both complexities are equal. We conjecture that this never

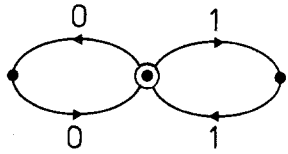

Fig. 5. Automaton accepting only pairs of 1's and zeros.

happens (except for graphs with a single node or a single loop), although both can become arbitrarily close. If we have SC = AC, then each node in the corresponding graph must be visited equally often. There do exist graphs with closed loops visiting every node exactly once, and a path always following this loop (or these loops) would seem to qualify. But in all studied cases such a path corresponds to a string that obeys stricter rules than specified in the grammar, and it is to be excluded for that reason (all strictly obeyed rules are included in the grammar by assumption). Furthermore, due to the assumption of stationarity, excursions from the above loops cannot occur less and less frequently with time. Instead, they must occur with some finite probability, supporting our conjecture.

Numerical simulations of strings created by cellular automata and reported in Section 5.2.1 show indeed that the SC is always considerably lower than the AC.

Estimating the SC is not as straightforward as determining the AC. If the grammar is not known, there does not exist any algorithm to compute either: if some sequence has not yet been observed, one can never be sure that it is indeed strictly forbidden and not just very unlikely (this does not mean of course that one cannot make guesses about the grammar; such guesses ought to become more and more reliable with the length of the observed string). If the grammar is known, there exists an algorithm yielding the minimal automaton and thus also AC (modulo the problem of reduction discussed above). But in general, this minimal graph need not always give the smallest SC. There might exist other automata with larger graphs but where fewer of the nodes are visited frequently.

The last question is, Why should we be more interested in probabilistic quantities at all? One might argue that, after all, the most important problem consists in deciding whether an observed sequence follows a certain grammar or not, and the importance of the complexity is that it tells us the size of the smallest computer able to do that. But the last remark is not really true. Instead of using for each sequence its own computer, one can imagine a large, general-purpose computer (or a network of computers) performing several such tasks in parallel, with arbitrarily much cheap storage available for slow input/output, and with the amount of fast storage attributed to each task depending on its demand. In this case, the verification of an observed string would proceed by storing the entire graph in slow storage, and fetching only those parts presently needed. The cost for that is, in the limit of very complex grammars, proportional to the effort necessary to address the region of slow storage where the relevant portion of the graph is located, times the probability that this portion is required. This is precisely the SC as we defined it, provided the storage of the automata in slow memory is optimal.

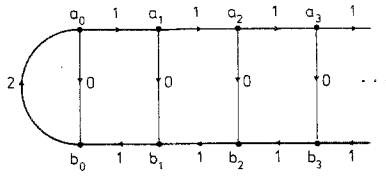

Fig. 6. Infinite automaton accepting only strings of the form $1^n01^n21^m01^m2\ldots$ with $n>0$, $m>0,\ldots,$

2.2. More Complex Languages

Regular languages are the only ones for which a finite automaton can be given that checks any string. Thus, the AC for nonregular languages [Context-free, context-senstive, or type-0 (Hopcroft and Ullman, 1979)] is always infinite. This does not mean that their SC is also infinite. For instance, one can imagine a case of infinite automata with a suitable probability measure defined on their nodes, such that equation (1) gives a finite result.

Consider, e.g., the infinite automaton shown in Fig. 6. It accepts all sequences of the structure $1^n01^n21^m01^m2\ldots$. This is clearly not a regular language, and thus $AC=\infty$. Assume now that the probability measure is such that "1" and "0" appear with equal probability whenever the latter is allowed to appear at all. For the probabilities p(node) this implies

$$\begin{aligned} p(a_k) &= p(b_k) \qquad \text{for all } k \\ p(a_{k+1}) &= p(a_k)/2 \qquad \text{for all } k \end{aligned} \tag{3}$$

This is solved to give $p(a_k)=p(b_k)=2^{-k-2}$, whence

$$SC \leq -\sum_{\text{nodes}} p(i)\log p(i) = \log 8 \tag{4}$$

The infinite ladder in Fig. 6 is of course very reminiscent of the stack of a pushdown automaton, and it was intended to be so: pushdown and Turing automata together with their stacks and tapes can be considered as special infinite automata. Seen from that point of view, the computational complexity of an algorithm running on these machines is very similar to our AC (except that in defining the AC we do not specify in any way the architecture of the machine, whence computational complexity can only be an upper bound to the AC). The reasons (given in the last subsection) for considering the SC as being at least as relevant as the AC are valid also in the present, more general case—provided a stationary probability measure exists.

3. MEASURE COMPLEXITIES

In this section, we shall primarily not be interested in verifying the grammatical correctness of strings. Instead, we are interested in predicting

optimally the conditional probability $p(s_{i+1}|s_i, s_{i-1}, \ldots)$ of occurrence of the letter s_{i+1} at some $i+1$, being given the string up to s_i. We call the minimal average Shannon information about $\{s_i, s_{i-1}, \ldots\}$ needed for that the *true measure complexity* (TMC).

Predicting $p(s_{i+1}|s_i, s_{i-1}, \ldots)$ is certainly not less difficult than predicting which $p(s_{i+1}|s_i, s_{i-1}, \ldots)$ are equal to zero. Assume now that there is no finite substring that is allowed to occur but does so only with zero probability. Then the TMC is at least as big as the information needed for excluding wrong s_{i+1}'s, which by definition is the SC, and we have the inequality

$$\text{TMC} \geq \text{SC} \tag{5}$$

But the assumption that no nonforbidden substring occurs with zero probability is part of our assumption of stationarity: if that would happen, the chance to encounter such a substring would have to decrease with time, and we would not call the probability measure stationary.

Estimating the TMC for an observed string is as difficult as estimating the set and algorithmic complexities. But a more easily obtained lower limit is provided by the *effective measure complexity* (EMC). Its definition is as follows.

We call $p\{S\}$ the probability to observe a substring $S = \{s_i, \ldots, s_{i+N-1}\}$ of length N. It is by assumption independent of i. The Shannon entropy stored in such a substring is

$$H_N = -\sum_S p_N\{S\} \log p_N\{S\} \tag{6}$$

For $N = 0$, we define $H_0 = 0$. Then the additional information needed to predict s_{i+N}, already being given S, is equal to

$$h_N = H_{N+1} - H_N \tag{7}$$

We shall call this the Nth-order block entropy.

It is well known that the block entropies decrease with N. Indeed, it is intuitively obvious that the uncertainty about s_{i+N} cannot increase if more and more of its predecessors are known. More precisely, the difference

$$\delta h_N = h_{N-1} - h_N \tag{8}$$

is just the average amount by which the uncertainty of s_{i+N} decreases due to knowledge of s_i. At least this amount of information about s_i has to be stored in order to make an optimal prediction of s_{i+N}, and can be discarded after s_{i+N} has been observed: its influence in determining all subsequent letters following is taken care of by s_{i+N}.

So we have, at any time i and for every N, at least an amount δh_N of information that we have to store N time steps. The minimal total amount

of information stored at any time for optimal predictions is thus

$$\mathrm{TMC} \geq \sum_N N\delta h_N = \sum_N N(h_{N-1} - h_N) \underset{\text{def}}{=} \mathrm{EMC} \tag{9}$$

This can also be written as

$$\mathrm{EMC} = \sum_{N=0}^{\infty} (h_N - h) \tag{10}$$

with

$$h = \lim_{N\to\infty} h_N \tag{11}$$

being the Shannon entropy per letter. In the case of dynamical systems, the latter is called the metric (Kolmogorov-Sinai) entropy (Eckmann and Ruelle, 1985).

One might believe that the information stored could always be selected optimally such that its amount is equal to the amount of actually needed information, and that thus always EMC = TMC. This is not true. The reason is that for optimal selection one has to code the information properly, and the encoding itself would require additional information to be stored.

As an example, consider the language defined by Fig. 7, with probability q to choose "0" when being at node a. One finds in this case $p_a = 1/(1+q)$ and

$$h_0 = \mathrm{SC} = \log(1+q) - \frac{q}{1+q}\log q \tag{12}$$

and

$$h_1 = h = \frac{1-q}{1+q}\log(1-q) - \frac{q}{1+q}\log q \tag{13}$$

giving $\mathrm{EMC} = h_0 - h < \mathrm{TMC}$ for all q.

As another example, let us discuss the language accepted by Fig. 5. We assume the probability measure such that whenever a choice between "1" and "0" is possible, the chance for either is 1/2. Then, one has SC = 3/2 bits and $h = 1/2$ bit/step, since one bit is needed every second time to fix the string. Measured in bits per time steps, the block entropies are computed

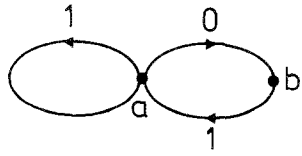

Fig. 7. Automaton accepting all strings without pairs of neighboring zeros.

as

$$h_0 = 1$$

$$h_{2N} = \frac{1}{2} - \frac{1}{2^{N-1}} + \frac{3}{2^{N+1}} \log_2 3, \qquad N \geq 1 \tag{14}$$

$$h_{2N-1} = h - h_{2N} + \frac{1}{2^N}$$

Inserting this into equation (9), we find that EMC = SC = 3/2.

In both examples, the information needed to predict the string optimally is equal to the information needed to exclude wrong strings, and thus TMC = SC. This derives from the fact that in both examples there is only one node where a choice is possible, and whenever this choice is possible, it is made with the same probabilities.

More generally, inequality (5) always is an equality if the probabilities in every node where a choice is possible depend only upon the node. Since such probability measures exist for any grammar, we see that for any grammar there exist measures for which (5) is saturated.

Another feature seen in the second example is that the block entropies h_N converge exponentially toward h. This is expected to be the typical behavior in the case of regular grammars and short-range probability correlations. In cases where the EMC is infinite, we cannot have exponential convergence. The simplest alternative would then be power-law behavior. We shall present numerical data in the next section that indeed suggest power laws with anomalous exponents.

The simplest situation prevails in the case of Markov processes, of which Figure 7 is an example. In a Markov process of order n, the block probabilities $p_{n+1}(s_1, \ldots, s_{n+1})$ satisfy the relations

$$p_{n+1}(s_1, \ldots, s_{n+1}) = p_n(s_{n+1} | s_2, \ldots, s_n) p_n(s_1, \ldots, s_n) \tag{15}$$

As is well known, in a Markov process of order n, all block entropies h_N are equal to h for $N > n$. Thus, Markov processes can be characterized as those processes that for given probabilities $p_n(s_1, \ldots, s_n)$ with fixed n have maximal entropy and minimal effective measure complexity.

Using mutual Shannon information as measure of complexity goes back, to our knowledge, to van Emden (1975). He did not, however, define an observable as we did. Instead, he called a system complex in a qualitative sense if the "interactions" between its parts, measured via mutual informations, were large.

Before leaving this section, let us make some comments about estimates of EMC from experimental data. Equation (9) shows that knowing precisely the value of h is essential. If h is overestimated due to an overlooked decay

of the h_N for large N, the EMC is underestimated. Thus, unless $h=0$, phenomenological estimates of EMC can only represent lower limits. Notice that in the case of the other complexities the situation is worse: there, overlooked small effects can influence estimates of complexities in both directions.

4. HIGHER DIMENSIONAL PATTERNS

In more than one dimension, it is not immediately obvious how the concept of an automaton scanning a pattern is to be implemented. The first attempt might consist in scanning it in some definite way, e.g., along a spiral, as indicated in Fig. 8a (with an arbitrarily chosen starting point). One problem is that the grammar generated in this way will in general not be stationary. More serious (but related to this) is that all nontrivial patterns will have infinite complexity when defined this way. The reason is that whenever correlations between neighboring points are not zero (even in one direction only), it will ultimately take infinitely many steps to go from one neighbor to the next.

In view of this problem, one might give up the idea of scanning the pattern. After all, the notion of Kolmogorov complexity of an ensemble that we want somehow to implement does not seem to require any scanning. Thus, a second way to define measure complexities is the following: after having chosen a random site i, we want to predict the letter at this site optimally, using knowledge about the letters at all other sites. To do this, we consider a sequence of increasing neighborhoods $\{U_k \mid k \in \mathbb{N}\}$ of site i, such that two successive neighborhoods differ by just one site j. We call $h(U_k)$ the uncertainty about site i when knowing all sites in $U_k \backslash i$, and define $\delta h_j(U_k) = h(U_{k+1}) - h(U_k)$. The effective path-independent measure complexity is then defined as in Section 3. It is the average over all i of $\inf[\sum_j r_{ij} \delta h_j(U_k)]$, where r_{ij} is the distance between sites i and j, and the infimum is taken over all sequences $\{U_k\}$.

In many cases, this definition seems to be very natural. But one problem is that it leads to finite complexities for all space-time patterns created by discrete local rules such as cellular automata. Moreover, one can have the somewhat paradoxical case of space-time patterns with smaller complexities than their sections at fixed time. It is not clear whether this represents a drawback of the definition of path-independent measure complexities or not. One might take the point of view that this increase of complexity when considering only part of a pattern illustrates just the way complexity is generated in general: by making unaccessible such information that would make predictions easy. Another problem is that there does not seem to exist

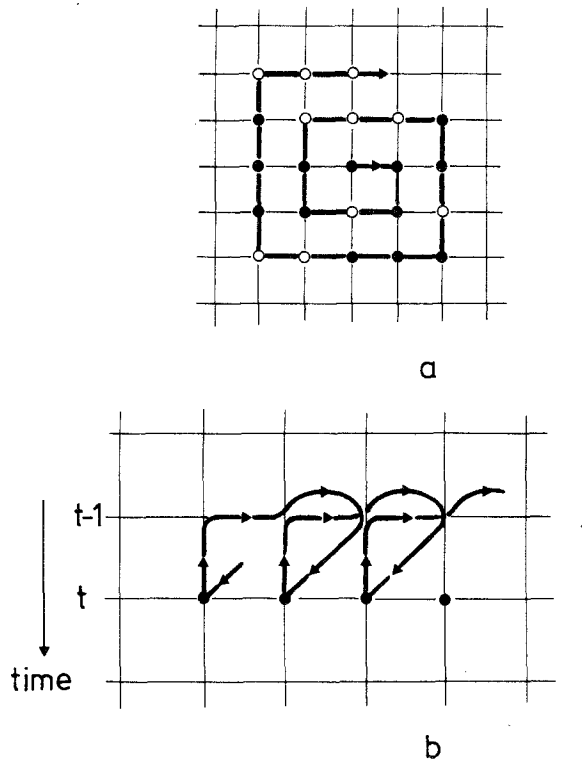

Fig. 8. (a) Possible (bad) way of scanning a two-dimensional pattern. (b) Path for a space-time scan yielding finite complexities for all 1D cellular automata with nearest neighbor rules. During the scan, the information picked up at time $t-1$ is used to predict the state at t.

any natural way to define path-independent algorithmic and set complexities, or true measure complexities.

Finally, a third possibility consists in scanning the pattern as we had tried first, but without prescribing the path along which it should be scanned. Instead, when looking for the minimal information or for the smallest automaton needed to continue the scan, we could also minimize with respect to possible paths (this minimal information must of course include the information needed to continue the path). If we allow multiple visits to a site in order to recall information stored there, this is not so different from the path-independent method, and it also gives finite complexities in the case of discrete local rules. For example, for any cellular automaton in one space dimension with a nearest neighbor rule, the path shown in Fig. 8b gives finite complexities.

Once having decided on how to scan, the rest is straightforward in principle, and the definitions and inequalities of Sections 2 and 3 can be carried over immediately. We shall not go into details, since we shall not study any application. Also, while presenting no problems in principle, finding the most efficient path might be a formidable task in practice. Certainly no algorithm exists for it in general.

5. APPLICATIONS

5.1. One-Dimensional Maps

In this section, we shall study families of maps $x_{n+1}=f_a(x_n)$ of the interval $[-1, 1]$ onto itself, of the type of the logistic map

$$f_a(x)=1-ax^2 \tag{16}$$

More precisely, we demand that (for all considered values of the parameter a) $f_a(x)$ has a quadratic maximum at $x=0$ with $f_a(0)=1$, that df_a/dx is positive (negative) for $x<0$ ($x>0$), and that $f_a(x)$ has negative Schwarzian derivative (Collet and Eckmann, 1980). As a function of a, we assume that $f_a(1)$ is monotonically decreasing such that there is a maximal value $a_{\max}$ at which $x_n=1$ is mapped onto $x_{n+1}=-1$, while $x_n=1$ is mapped onto positive values for sufficiently small values of a.

Such maps have attractors which either are periodic orbits, chaotic attractors with a completely continuous measure, or Cantor sets. The latter, studied in particular by Feigenbaum (1978, 1979), occur at infinitely many values of a, all of which are cumulation points of bifurcation points. The former two both occur on sets of a values of positive measure.

The sequence $\{x_i;\ i\in\mathbb{N},\ x_i\in[-1, 1]\}$ of continuous variables is mapped onto a sequence ("itinerary") of discrete variables $s_i=0, 1$ by

$$\begin{aligned} x_i<0 &\leftrightarrow s_i=0 \\ x_i\geq 0 &\leftrightarrow s_i=1 \end{aligned} \tag{17}$$

and it is this itinerary that we shall study in the following.

The set of all possible strings $\{s_i\}$ thus generated for a fixed map is given by the following theorem [Collet and Eckmann (1980); the simplified version presented here is due to Allouche and Cosnard (1984) and Dias de Deus et al. (1984)]. Starting from any sequence $S=\{s_i\}$, define first a sequence $A.S$ by

$$(A.S)_i=\sum_{k=1}^{i} s_k \bmod 2 \tag{18}$$

and define $y(S)$ as the number $\in[0, 1]$ whose binary representation is just $y(S)=(0.s_1s_2s_3\cdots)_2$. Finally, the "kneading sequence" T is defined as the

sequence $\{t_i\}$ generated by $x_1 = 1$, and starting with $t_1 = 1$. Then, a sequence S is allowed if and only if

$$1 - y(A.T) \leq y(A.\{s_i s_{i+1} s_{i+2} \cdots s_{i+n}\}) \leq y(A.T) \tag{19}$$

for all i and all n.

In the following, we shall study special cases of a.

5.1.1. *Fully Developed Chaos and Band-Merging Points*

The simplest situation—apart from the periodic regime below the first Feigenbaum point—prevails at $a_{\max}$. In this case, called fully developed chaos, the intervals $[-1, 0]$ and $[0, 1]$ are both mapped one-to-one onto $[-1, 1]$. The kneading sequence is $100\ldots$, and hence all sequences are allowed itineraries. Thus the grammar is trivial, the topological entropy is one bit/iteration, and both set complexities are zero. The measure-complexity is not zero in general, but is finite. Indeed, it was shown by Györgyi and Szepfalusy (1985) that the block entropies h_N converge in this case exponentially with N.

A very similar situation prevails at the so-called band-merging points (Grossmann and Thomae, 1977). At these points, a suitable iterate of the map is equivalent to a fully developed chaotic map on some subinterval of $[-1, 1]$.

5.1.2. *Periodic Windows*

Let us next study the case where the attractor is periodic, but where the algorithmic entropy is nonzero. The set of all itineraries is in this case a finite-complement (and thus regular) language (Block et al., 1980).

For the period-3 window, e.g., all sequences with blocks "00" are forbidden after the first occurrence of "1." The graph accepting this can be truncated, and after truncation we have the automaton shown in Fig. 7.

Starting with a random point (with respect to Lebesgue measure), the orbit is attracted toward the periodic orbit with probability 1; thus the itinerary is not a stationary sequence. But there are orbits (with starting point of Lebesgue measure zero) that generate nontrivial itineraries with stationary probability measures. We shall not go further into detail, since the results are well known (Block et al., 1980).

5.1.3. *Typical Chaotic Maps*

At parameter values where the map is chaotic but not fully developed, the block entropies typically converge very irregularly (Crutchfield and

Packard, 1983). Three examples obtained by straightforward simulations are shown in Fig. 9. The values of h used in the figure are obtained from measuring simultaneously the Lyapunov exponents (which are equal to h for 1D maps). Although the convergence is too irregular to make strong statements about asymptotic behavior, we see that an exponential

$$h_N - h = \text{const} \times e^{-Nh/2} \tag{20}$$

provides a reasonable fit, indicating a finite EMC. The same behavior is found for Henon's map (Grassberger and Kantz, 1985), although there the

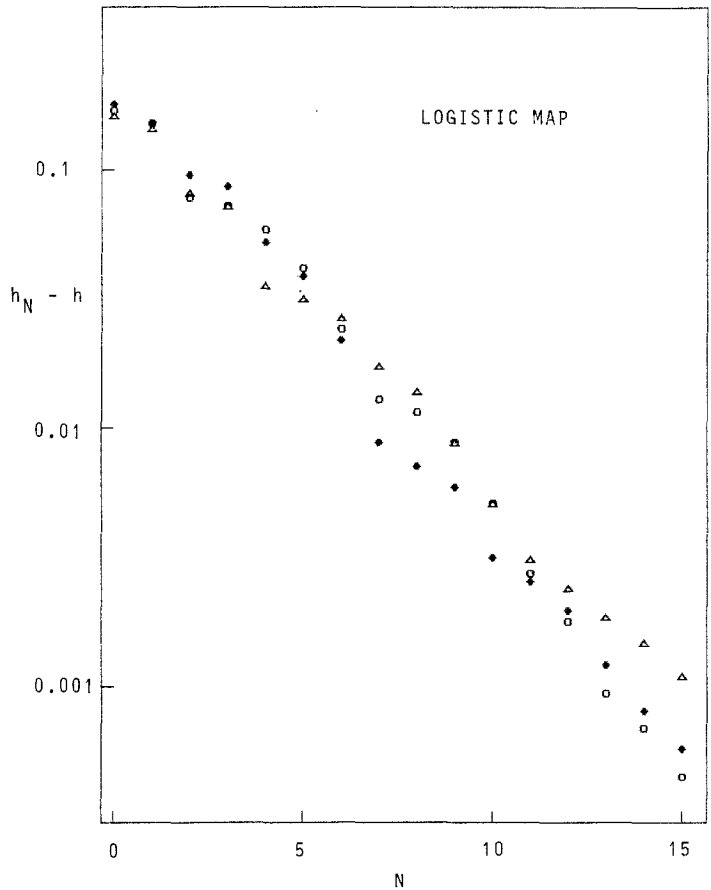

Fig. 9. Differences $h_N - h$ for the logistic map (5.1) with parameter values (●) $a = 1.89$, (△) $a = 1.90$, and (○) $a = 1.91$, each obtained from 5×10^8 iterations and plotted on a logarithmic scale. The entropy h was obtained in all three cases from the Lyapunov exponent. Statistical errors are less than the size of the symbols.

numerical results are less reliable. It was conjectured by György and Szepfalusy (1985) and proven there for Markov partitions, but Markov partitions are not easy to find for typical chaotic maps; our partition is certainly not Markov. Anyhow, we have strong evidence that the EMC is finite for typical chaotic 1D maps, and is approximately proportional to $1/h$.

On the other hand, the algorithmic complexity should be infinite for nearly all chaotic parameter values. This follows simply from the fact that typically the kneading sequence is not periodic, whence one has to test infinitely long strings to verify equation (19) in the worst case.

5.1.4. Feigenbaum Points

The most interesting case is that of the accumulation points of bifurcations studied by Feigenbaum (1978, 1979).

There, we have $h = 0$, i.e., all orbits are nonchaotic. The block entropies h_N for $N = 2^m$ are easily obtained as follows. First, one has $p_2(11) = p_2(10) = p_2(01) = 1/3$, giving $H_2 = \log 3$. Next, if N is even and >2, then due to the Cantor structure of the attractor one finds $H_N = H_{N/2} + 1$, giving

$$h_N \sim \frac{1}{N} \qquad \text{for } N \to \infty \tag{21}$$

and

$$\text{EMC} = \sum_{N=0}^{\infty} h_N = \infty \tag{22}$$

Thus, all itineraries at the Feigenbaum point have zero entropy, but infinite complexity, in agreement with the naive intuition.

In general we might conclude that for 1D maps the EMC seems to agree better with the intuitive concept of complexity than either the AC or the SC.

5.2. Cellular Automata

In this section, we shall discuss one-dimensional cellular automata (CA) with two states per site ("0," "1"), and nearest neighbor rules. Such rules are called "elementary" in Wolfram (1983).

In principle, we could (and should) discuss the two-dimensional patterns created by these CA in space-time. We shall not do this, because of the technical problems discussed in Section 4. Instead, we shall first study set complexities of spatial patterns created by "legal" rules [in the sense of Wolfram (1983)] after finite numbers of iterations. After that, effective measure complexities of spatial, temporal, and more general one-dimensional patterns will be discussed for two different types of rules. In

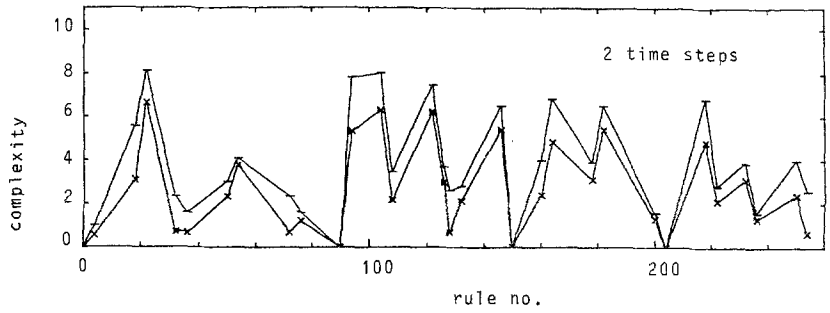

Fig. 10. Algorithmic complexities (horizontal bars) and set complexities (crosses; both measured in bits) of spatial patterns generated after two time steps by "legal elementary" 1D cellular automata, plotted versus the number of the CA in the notation of Wolfram (1983). Starting configurations consisted of completely random strings. Values for the SC are actually upper limits only, since the accepting automata might not be optimal for the SC.

the last case we shall encounter again patterns with zero entropy, but infinite complexity.

Individual CAs will be denoted by numbers following Wolfram (1983).

5.2.1. Algorithmic and Set Complexities

In this subsection, we first follow Wolfram (1984b) in constructing minimal deterministic automata recognizing the spatial strings $\{\cdots s_{i-1}s_is_{i+1}\cdots\}$ generated after two and three iterations. More iterations would of course be extremely useful, since visual inspection indicates that typical behavior is often seen only much later. Unfortunately, for the more complex rules the size of these automata (i.e., the algorithmic complexity in our notation) increases so fast with the number of time steps that at present this seems impossible.

After having obtained these accepting automata, we took very long random strings (length = 5000 × size of accepting automaton) as starting configurations, and estimated from this the set complexity. Results are shown in Fig. 10 (for two time steps) and in Fig. 11 (for three time steps).

Let us make a few comments about these data:

1. In all cases, set complexities are strictly smaller than algorithmic complexities, in agreement with our general statement in Section 2.

2. There are some rules (94, 104, 164, and 218) which have fairly large AC, although they seem to settle on a trival (periodic) time behavior. We find that their SC is suppressed compared to the AC more than the average. We furthermore expect that their complexities will increase less rapidly with time than for the rules showing complex time behavior.

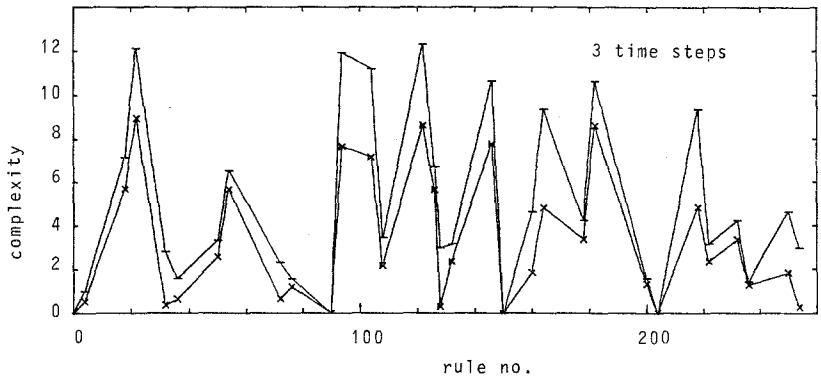

Fig. 11. Same as Figure 9, but after three time steps.

3. For other rules, such as 50, 132, 178, 222, and 232, the AC are much smaller, although the patterns look very similar to those of the previous group. Indeed, their SC are much less suppressed that in the previous group.

4. In cases, e.g., rules 32, 72, 128, and 160, the asymptotic patterns seem to be completely trivial, consisting only of zeros. This triviality is not directly reflected in the AC, which seems to increase with t, but it is seen in a decrease of SC.

5. The biggest complexities are shown by rules with aperiodic behavior, as we should have expected (rules 18, 22, 122, 126, 146, 182, and, to some lesser extent, 54). As shown by Grassberger (1984), rule 22 should be the most complex among these. This is not clearly borne out in Figures 9 and 10, neither by the AC nor by the SC.

Summarizing, we might say that in these cases the SC corresponds better to the naive expectations from visual inspections. But the difference is less than what one might have hoped, presumably due to the small number of time steps.

5.2.2. Measure Complexities; Asymmetric Rules

Next we study EMC for cellular automata for which all random strings $\{s_1, \ldots, s_N\}$ appear equally often in the stationary distributions, i.e.,

$$p_N(s_1, \ldots, s_N) = 2^{-N} \tag{23}$$

For these rules, we shall study both timelike sequences and sequences taken along a diagonal line $i - t = \text{const}$ (Sinai, 1985).

The first class of rules satisfying equation (23) are "additive" rules such as rules 90 and 150. They also have zero complexity for timelike sequences and thus they are of no interest for us.

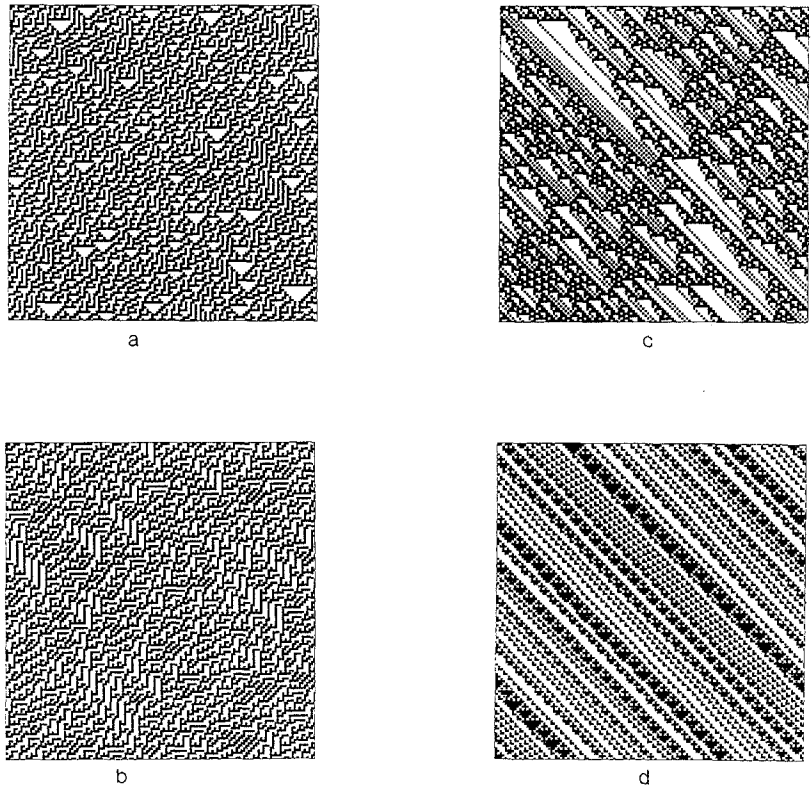

Fig. 12. Patterns generated by CA rules (a) 30, (b) 45, (c) 120, (d) 210. time is increasing from top down.

The other class, studied in the following, consists of rules of the type (Packard, 1983; Wolfram, 1985)

$$s_i(t+1) = s_{i-1}(t) \text{ XOR } f(s_i(t), s_{i+1}(t)) \tag{24}$$

where $f(s, s')$ is a nontrivial mapping from $\{0, 1\} \times \{0, 1\}$ to $\{0, 1\}$, i.e., $f(\cdots)$ is not equal to a constant (0 or 1) nor equal to s, $1-s$, s', or $1-s'$. It is easy to see (Wolfram, 1985) that for all such rules one has equation (23).

The rules we shall study here are rule 30 [$f(s, s') = s$ OR s'), rule 45 [$f(s, s') = s$ OR NOT s'], rule 120 [$f(s, s') = s$ AND s'], and rule 210 [$f(s, s') = s$ AND NOT s']. Patterns generated by these rules are shown in Fig. 12. All other rules satisfying (24) are either "totalistic" [allowing thus a fairly complete treatment (Martin et al., 1984)] or trivial or related to these rules by exchanging 0 and 1.

For these rules, all temporal strings $\{s_i(t), \ldots, s_i(t+T-1)\}$ occur also with equal probability 2^{-T}. But from the dynamical systems point of view,

more interesting than such sequences are rectangular $N \times T$ blocks

$$S_{N,T} = \{[s_i(t), \ldots, s_{i+N-1}(t)], \ldots, [(s_i(t+T-1), \ldots, s_{i+N-1}(t+T-1)]\} \tag{25}$$

or similar trapezoidal blocks with $s_j(t+k)$ replaced by $s_{j+k}(t+k)$. Consider the entropies

$$H_{N,T} = -\sum_S p_{N,T}(S_{N,T}) \log p_{N,T}(S_{N,T}) \tag{26}$$

One expects (Wolfram, 1984a; Sinai, 1985) that these tend toward $T.h$ when first T and then N tend toward infinity, with h being the time and diagonal entropy, respectively. Alternatively, define block entropies

$$h_{N,T} = H_{N,T+1} - H_{N,T} \tag{27}$$

Then one can show generally that all $h_{N,T}$ are nonnegative and decreasing with T, and that

$$h = \lim_{N,T \to \infty} h_{N,T} \tag{28}$$

irrespective of the order in which the limit is taken.

For all rules satisfying equation (24) the limit N need not be taken in equation (28). Instead, it is sufficient to take $N = 2$. This follows simply from the observation that if a strip of width $N = 2$ and of length T is known, further columns of length $T-1$, $T-2, \ldots$ to the left of it can immediately be obtained by inverting equation (24). Thus, the information $H_{N,T}$ cannot grow faster than $H_{2,T}$ with T.

Numerical results for $h_{2,T}$ were obtained by exact enumerations. Results are shown in Fig. 13 for the temporal block entropies and in Fig. 14 for the diagonal entropies.

As seen from Fig. 12, rule 210 leads to periodic stripes of variable width and period. Accordingly, the EMC for this rule is finite both for diagonal and timelike blocks (the latter are not shown in Fig. 13). Also, the diagonal metric entropy is zero, although the diagonal topological entropy is equal to one. This seems to be bigger than the diagonal topological entropies of the other three rules.

For the other rules 30, 45, and 120 we find that temporal metric entropies are exactly one, within the estimated errors, while diagonal metric entropies are less than one (they are indicated by arrows in Fig. 14). For the temporal block entropies we find, more precisely,

$$h_{2,T} = h + \text{const}/T^{\alpha} \tag{29}$$

with $\alpha = 0.6 \pm 0.1$ (rules 30, 45) and $\alpha = 1.0 \pm 0.1$ (rule 120), respectively, and with $h = 1$. The convergence of diagonal entropies is also compatible with this power law, with the same exponents α, although errors are too large to allow a more definite statement.

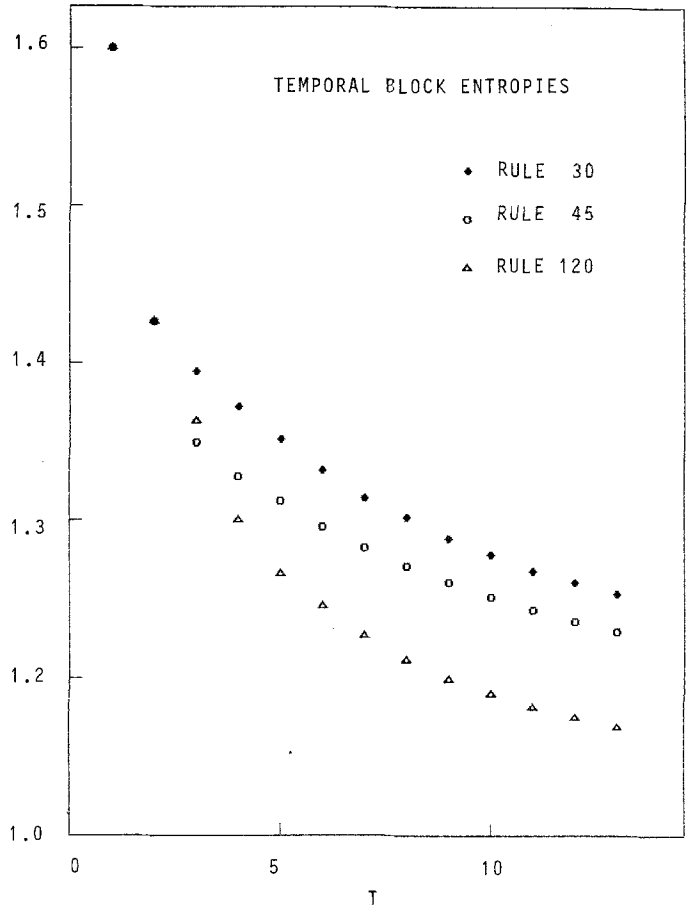

Fig. 13. Temporal block entropies $h_{2,T}$ for the three rules shown in Figs. 12a–12c.

Anyhow, our results show that the EMC is infinite for all three rules 30, 45, and 120, both for temporal and for diagonal strings.

5.2.3. Measure Complexities; Rule 22

The last example that has been studied in detail is rule 22. It was chosen since among all "legal" (in particular, symmetric) rules it seems to show the most complex behavior. It can also be formulated as

$$s_i(t+1) = \begin{matrix} 1 & \text{if} \quad s_{i-1}(t)+s_i(t)+s_{i+1}(t)=1 \\ 0 & \text{else} \end{matrix} \tag{30}$$

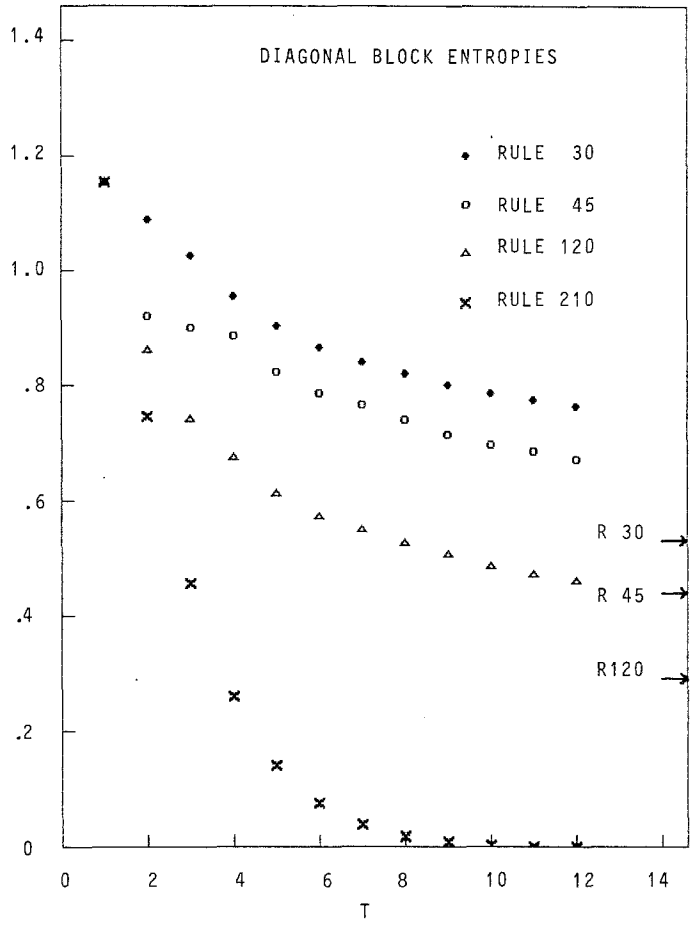

Fig. 14. Diagonal block entropies $h_{2,T}$ measuring the information increase in columns parallel to lines $i-t=\text{const}$ for the four rules shown in Fig. 12. The arrows indicate the estimates of $\lim_{N\to\infty} h_N$.

A pattern generated with this rule from a random start is shown in Fig. 15. Visual impression does not suggest any long-range effects in this and in similar patterns generated from other random configurations. Nevertheless, more detailed studies, which shall be published elsewhere (Grassberger, 1986), indicate that there are such long-range effects, reminiscent of critical phenomena. In the remainder of this section, we shall present alternative indications of such effects based on EMCs.

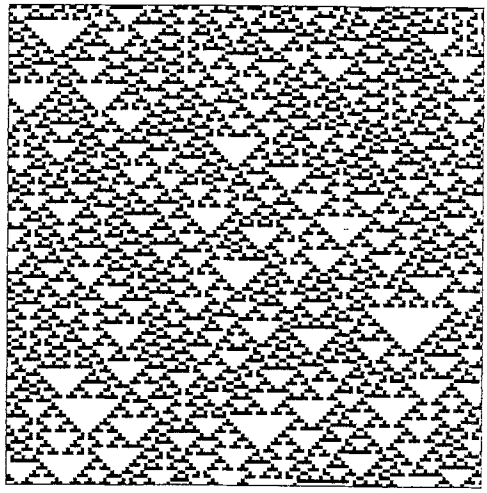

Fig. 15. Space-time pattern generated by rule 22 from random start.

Since we do not know exactly the invariant measure for rule 22, we cannot use exact enumeration as in the last subsection. Instead, we performed Monte Carlo estimates, based on very large lattices (up to 30,000 time steps and 36,000 lattice sites wide). Results for temporal block entropies $h_{N,T}$ are presented in Fig. 16 for $N = 1$–5. Results for $N = 2$, together

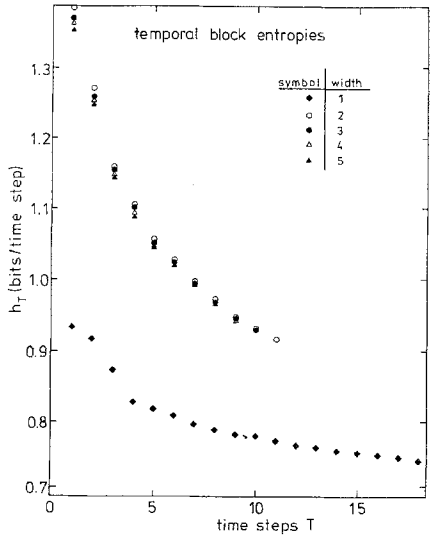

Fig. 16. Temporal block entropies, rule 22. The spatial width of the blocks is $N = 1$–5.

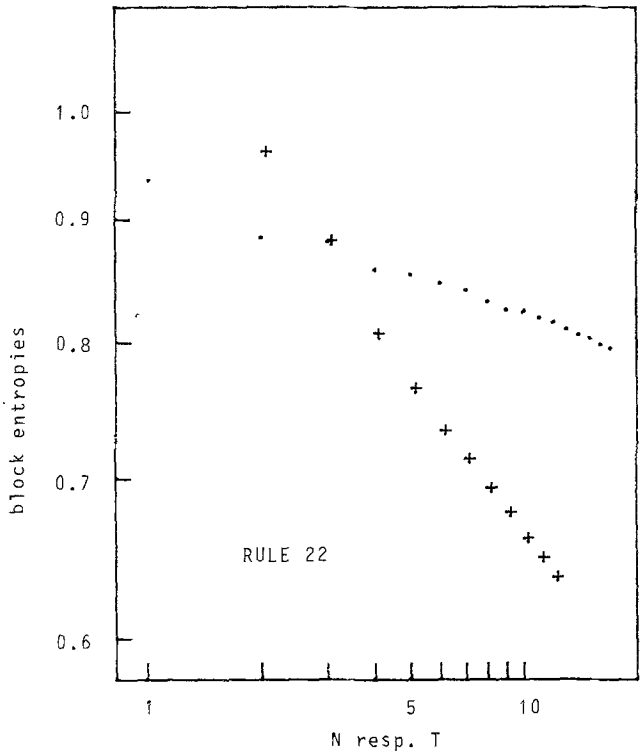

Fig. 17. (●) Spatial block entropies h_N and (+) temporal block entropies $h_{N,T}$ with width $N=2$ for rule 22, on a doubly logarithmic scale. Temporal entropies are in natural units (bits × log 2), in order to fit on the same scale.

with spatial block entropies, are presented on a doubly logarithmic scale in Fig. 17. Error bars in both figures are much smaller than the symbols.

The first thing to notice is that again it seems that the limit $N \to \infty$ in equation (28) is reached already for $N=2$. The second observation is that although the decrease of the block entropies with block length is very weak [so that it had been overlooked in Grassberger (1984)], it is very steady and does not show any tendency to vanish soon. Indeed, best fits were obtained with power laws (29) with $h=0$ in both cases, and with $\alpha=0.18$ for temporal entropies and $\alpha=0.06$ for spatial entropies (see Fig. 17). Deviations from such a power law are larger for spatial block entropies than for temporal ones.

If our interpretation of the data shown in Fig. 17 is correct, we have found a second example (in addition to Feigenbaum's map) with infinite EMC but zero randomness. In that case, it seems natural to call rule 22 a deterministic critical phenomenon. Notice that it would be a quite unusual

critical phenomenon, in the sense that it contains no continuous control parameter and no obvious order parameter.

6. DISCUSSION

In this paper, we have introduced several quantities which can serve as measures of complexity of ensembles of patterns. More precisely, we discussed only one-dimensional patterns in any detail. Such ensembles can then be considered as formal languages, endowed with probabilities which turn them into "styles."

In the simplest cases, a formal grammar is defined by a transition graph. If also the probability measure depends only on the graph in the sense that branching probabilities depend only on the actual node, then we have found

$$\mathrm{EMC} \leq \mathrm{TMC} = \mathrm{SC} < \mathrm{AC} \tag{31}$$

If the branching probabilities are not single-valued functions of the nodes in the transition graph, then the central equality in (31) is replaced by an inequality $\mathrm{TMC} > \mathrm{SC}$. Here , EMC and TMC stand for effective and true metric complexity, respectively, SC for set complexity, and AC for algorithmic complexity. All except the latter are related to Shannon entropies (and thus metric quantities), while the latter is a purely algorithmic concept and agrees with the complexity introduced by Wolfram (1984b). The EMC seems the only measure of complexity that is observable if the grammar is not known (except for the path-independent complexities mentioned briefly in Section 4). It is thus considered as the most relevant of the measures of complexity studied in the present paper. It is essentially a weighted sum over mutual information between distant letters.

The naturalness of our definition is indicated by the fact that the EMC was infinite in two cases that were also judged complex intuitively: the kneading sequence of the Feigenbaum map (Section 5.1) and some patterns created by cellular automata (Section 5.2). In both cases, we found scaling laws like those typical of critical phenomena.

Other most interesting examples to study would be natural languages and sequences of DNA. We conjecture that we should find similar scaling laws there, too. Unfortunately, existing numerical studies do not seem of sufficient detail to decide this question.

We mentioned in the introduction the concept of Kolmogorov complexity. In contrast to the quantities discussed in the present paper, this is not attached to an ensemble of strings, but to individual (infinite) strings: like Shannon entropy, it measures an amount of information per letter needed to specify a string; our complexities, in contrast, were informations per letter needed to guarantee that the string belongs to the ensemble, without specifying it further (except for the EMC, which was introduced as a lower estimate for such an information).

The difference between Kolmogorov complexity and Shannon entropy is that the latter is a measure-theoretic concept while the former is not. There are thus cases where the two do not match. But I claim that this happens only if the ensemble of strings one is considering is not stationary in the strong sense given in Section 2. Consider, e.g., the string of digits of pi, 3.141592.... The most efficient program to compute N digits on a general-purpose computer increases slower than linearly with N, and thus the Kolmogorov complexity of pi is zero. Nevertheless, by regarding sufficiently many digits, one gets the impression that they are more or less random [or "normal;" see Wagoner (1985)]. In order to test the latter, one has to do statistics over many short substrings, and verify that all different substrings of the same length occur with the same frequency. In this way, one discards the beginning of the string, which on the other hand is the crucial part in determining the Kolmogorov complexity: the shortest programs to generate other substrings of length N will in general not be shorter than $O(N)$. It is for this and similar examples (e.g., sequences of gaps between successive prime numbers or between energy levels of quantum systems) that we restricted ourselves to strictly stationary ensembles. There, a distinction between Shannon information and Kolmogorov complexity does not seem necessary. Notice that, although the concept of Kolmogorov complexity does not involve an *a priori* probability measure, it induces such a measure (Chaitin, 1979). Otherwise, it could not be equivalent to Shannon information, of course.

As already mentioned in the introduction, the idea of using mutual informations (like our EMC) to measure complexity of structures is not new. Within the framework of Shannon informations, we encountered it first in van Emden (1975). Using Kolmogorov complexity instead of Shannon information, it was proposed independently in Chaitin (1979). According to what we said above, we conjecture that the approaches of van Emden and of Chaitin are equivalent when applied to strictly stationary ensembles if the measure induced by the Kolmogorov complexity is equal to the true one.

ACKNOWLEDGMENTS

For stimulating discussions on the subjects of the present paper, I am most indebted to T. von der Twer, S. Wolfram, P. Szepfalusy, R. Dilao, J. Keymer, and H. Kantz. I also thank H. Kantz for a careful reading of the manuscript. I also want to acknowledge a very stimulating correspondence with J. Ford on the question of Kolmogorov complexity versus Shannon information. Finally, it is a pleasure to thank J. Dias de Deus for inviting me to an exciting meeting on cellular automata in Lisbon. The present paper is partly based on a talk given there.

NOTE ADDED IN PROOF

Unfortunately, I was unaware of the notion of "logical depth" of C. H. Bennett (in *Emerging Syntheses in Science*, D. Pines, ed., 1985) which measures essentially the time needed to run the shortest program producing the pattern.

REFERENCES

Alekseev, V. M., and Yakobson, M. V. (1981). *Physics Reports*, **75**, 287.

Allouche, J.-P., and Cosnard, M. (1984). Grenoble preprint.

Block, L., et al. (1980). Periodic points and topological entropy of 1-dimensional maps, in *Lecture Notes in Mathematics*, No. 819, Springer, Berlin, 1980, p. 18.

Chaitin, G. J. (1979). Toward a mathematical definition of 'life', in *The Maximum Entropy Principle*, R. D. Levine and M. Tribus, eds., MIT Press, Cambridge, Massachusetts.

Christol, G., Kamae, T., Mendes France, M., and Rauzy, G. (1980). *Bulletin Societé Mathematique France*, **108**, 401.

Collet, P., and Eckmann, J.-P. (1980). *Iterated Maps on the Interval as Dynamical Systems*, Birkhauser, Boston.

Crutchfield, J. P., and Packard, N. H. (1983). *Physica*, **7D**, 201.

Dias de Deus, J., Dilao, R., and Noronha da Costa, A. (1984). Lissabon preprint.

Eckmann, J. P., and Ruelle, D. (1985). *Review of Modern Physics*, **57**, 617.

Feigenbaum, M. (1978). *Journal of Statistical Physics*, **19**, 25.

Feigenbaum, M. (1979). *Journal of Statistical Physics*, **21**, 669.

Grassberger, P. (1984). *Physica*, **10D**, 52.

Grassberger, P., and Kantz, H. (1985). *Physics Letters*, **113A**, 235.

Grossmann, S., and Thomae, S. (1977). *Zeitschrift für Naturforschung*, **32a**, 1353.

Guckenheimer, J., and Holmes, P. (1983). *Non-linear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, Springer, New York.

Györgyi, G., and Szepfalusy, P. (1985). *Physical Review A*, **31**, 3477; and to be published.

Hofstadter, D. R. (1979). *Gödel, Escher, Bach*, Vintage Books, New York.

Hogg, T., and Huberman, B. A. (1985). Order, complexity, and disorder, Palo Alto preprint.

Hopcroft, J. E. and Ullman, J. D. (1979). *Introduction to Automata Theory, Lanaguages, and Computation*, Addison-Wesley.

Martin, O., Odlyzko, A., and Wolfram, S. (1984). *Communication in Mathematical Physics*, **93**, 219.

Packard, N. (1983). Complexity of growing patterns in cellular automata, Institute of Advanced Study preprint.

Schuster, H. G. (1984). *Deterministic Chaos*, Physik-Verlag, Weinheim, West Germany.

Shannon, C. E., and Weaver, W. (1949). *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, Illinois.

Sinai, Ya. (1985). *Commentarii Mathematici Helvetici*, **60**, 173.

Van Emden, M. H. (1975). *An Analysis of Complexity*, Mathematical Centre Tracts, Amsterdam.

Wagoner, S. (1985). Is pi normal;, *Mathematical Intelligencer*, **7**, 65.

Wolfram, S. (1983). *Review of Modern Physics*, **55**, 601 (1983).

Wolfram, S. (1984a). *Physica*, **10D**, 1.

Wolfram, S. (1984b). *Communications in Mathematical Physics*, **96**, 15.

Wolfram, S. (1985). Random sequence generation by cellular automata, Institute for Advanced Study preprint.