TOWARD A MATHEMATICAL DEFINITION OF "LIFE"

In R. D. Levine and M. Tribus, The Maximum Entropy Formalism, MIT Press, 1979, pp. 477–498

Gregory J. Chaitin

Abstract

In discussions of the nature of life, the terms "complexity," "organism," and "information content," are sometimes used in ways remarkably analogous to the approach of algorithmic information theory, a mathematical discipline which studies the amount of information necessary for computations. We submit that this is not a coincidence and that it is useful in discussions of the nature of life to be able to refer to analogous precisely defined concepts whose properties can be rigorously studied. We propose and discuss a measure of degree of organization

and structure of geometrical patterns which is based on the algorithmic version of Shannon's concept of mutual information. This paper is intended as a contribution to von Neumann's program of formulating mathematically the fundamental concepts of biology in a very general setting, i.e. in highly simplified model universes.

1. Introduction

Here are two quotations from works dealing with the origins of life and exobiology:

> These vague remarks can be made more precise by introducing the idea of information. Roughly speaking, the information content of a structure is the minimum number of instructions needed to specify the structure. Once can see intuitively that many instructions are needed to specify a complex structure. On the other hand, a simple repeating structure can be specified in rather few instructions. [1]
>
> The traditional concept of life, therefore, may be too narrow for our purpose... We should try to break away from the four properties of growth, feeding, reaction, and reproduction... Perhaps there is a clue in the way we speak of living *organisms.* They are *highly organized,* and perhaps this is indeed their essence... What, then, is organization? What sets it apart from other similarly vague concepts? Organization is perhaps viewed best as "complex interrelatedness"... A book is complex; it only resembles an organism in that passages in one paragraph or chapter refer to others elsewhere. A dictionary or thesaurus shows more organization, for every entry refers to others. A telephone directory shows less, for although it is equally elaborate, there is little cross-reference between its entries... [2]

If one compares the first quotation with any introductory article on algorithmic information theory (e.g. [3–4]), and compares the second quotation with a preliminary version of this paper [5], one is struck by the similarities. As these quotations show, there has been a great

deal of thought about how to define "life," "complexity," "organism," and "information content of organism." The attempted contribution of this paper is that we propose a rigorous quantitative definition of these concepts and are able to prove theorems about them. We do not claim that our proposals are in any sense definitive, but, following von Neumann [6–7], we submit that a precise mathematical definition must be given.

Some preliminary considerations: We shall find it useful to distinguish between the notion of degree of interrelatedness, interdependence, structure, or organization, and that of information content. Two extreme examples are an ideal gas and a perfect crystal. The complete microstate at a given time of the first one is very difficult to describe fully, and for the second one this is trivial to do, but neither is organized. In other words, white noise is the most informative message possible, and a constant pitch tone is least informative, but neither is organized. Neither a gas nor a crystal should count as organized (see Theorems 1 and 2 in Section 5), nor should a whale or elephant be considered more organized than a person simply because it requires more information to specify the precise details of the current position of each molecule in its much larger bulk. Also note that following von Neumann [7] we deal with a discrete model universe, a cellular automata space, each of whose cells has only a finite number of states. Thus we impose a certain level of granularity in our idealized description of the real world.

We shall now propose a rigorous theoretical measure of degree of organization or structure. We use ideas from the new algorithmic formulation of information theory, in which one considers individual objects and the amount of information in bits needed to compute, construct, describe, generate or produce them, as opposed to the classical formulation of information theory in which one considers an ensemble of possibilities and the uncertainty as to which of them is actually the case. In that theory the uncertainty or "entropy" of a distribution is defined to be

$$-\sum_{i<k} p_i \log p_i,$$

and is a measure of one's ignorance of which of the k possibilities actually holds given that the *a priori* probability of the ith alternative is

p_i. (Throughout this paper "log" denotes the base-two logarithm.) In contrast, in the newer formulation of information theory one can speak of the information content of an individual book, organism, or picture, without having to imbed it in an ensemble of all possible such objects and postulate a probability distribution on them.

We believe that the concepts of algorithmic information theory are extremely basic and fundamental. Witness the light they have shed on the scientific method [8], the meaning of randomness and the Monte Carlo method [9], the limitations of the deductive method [3–4], and now, hopefully, on theoretical biology. An information-theoretic proof of Euclid's theorem that there are infinitely many prime numbers should also be mentioned (see Appendix 2).

The fundamental notion of algorithmic information theory is $H(X)$, the algorithmic information content (or, more briefly, "complexity") of the object X. $H(X)$ is defined to be the smallest possible number of bits in a program for a general-purpose computer to print out X. In other words, $H(X)$ is the amount of information necessary to describe X sufficiently precisely for it to be constructed. Two objects X and Y are said to be (algorithmically) independent if the best way to describe them both is simply to describe each of them separately. That is to say, X and Y are independent if $H(X,Y)$ is approximately equal to $H(X)+H(Y)$, i.e. if the joint information content of X and Y is just the sum of the individual information contents of X and Y. If, however, X and Y are related and have something in common, one can take advantage of this to describe X and Y together using much fewer bits than the total number that would be needed to describe them separately, and so $H(X,Y)$ is much less than $H(X)+H(Y)$. The quantity $H(X:Y)$ which is defined as follows

$$H(X:Y) = H(X) + H(Y) - H(X,Y)$$

is called the mutual information of X and Y and measures the degree of interdependence between X and Y. This concept was defined, in an ensemble rather than an algorithmic setting, in Shannon's original paper [10] on information theory, noisy channels, and coding.

We now explain our definition of the degree of organization or structure in a geometrical pattern. The d-diameter complexity $H_d(X)$ of an

object X is defined to be the minimum number of bits needed to describe X as the "sum" of separate parts each of diameter not greater than d. Let us be more precise. Given d and X, consider all possible ways of partitioning X into nonoverlapping pieces each of diameter $\leq d$. Then $H_d(X)$ is the sum of the number of bits needed to describe each of the pieces separately, plus the number of bits needed to specify how to reassemble them into X. Each piece must have a separate description which makes no cross-references to any of the others. And one is interested in those partitions of X and reassembly techniques α which minimize this sum. That is to say,

$$H_d(X) = \min[H(\alpha) + \sum_{i<k} H(X_i)],$$

the minimization being taken over all partitions of X into nonoverlapping pieces

$$X_0, X_1, X_2, \ldots, X_{k-1}$$

all of diameter $\leq d$.

Thus $H_d(X)$ is the minimum number of bits needed to describe X as if it were the sum of independent pieces of size $\leq d$. For d larger than the diameter of X, $H_d(X)$ will be the same as $H(X)$. If X is unstructured and unorganized, then as d decreases $H_d(X)$ will stay close to $H(X)$. However if X has structure, then $H_d(X)$ will rapidly increase as d decreases and one can no longer take advantage of patterns of size $> d$ in describing X. Hence $H_d(X)$ as a function of d is a kind of "spectrum" or "Fourier transform" of X. $H_d(X)$ will increase as d decreases past the diameter of significant patterns in X, and if X is organized hierarchically this will happen at each level in the hierarchy.

Thus the faster the difference increases between $H_d(X)$ and $H(X)$ as d decreases, the more interrelated, structured, and organized X is. Note however that X may be a "scene" containing many independent structures or organisms. In that case their degrees of organization are summed together in the measure

$$H_d(X) - H(X).$$

Thus the organisms can be defined as the minimal parts of the scene for which the amount of organization of the whole can be expressed as the

sum of the organization of the parts, i.e. pieces for which the organization decomposes additively. Alternatively, one can use the notion of the mutual information of two pieces to obtain a theoretical prescription of how to separate a scene into independent patterns and distinguish a pattern from an unstructured background in which it is imbedded (see Section 6).

Let us enumerate what we view as the main points in favor of this definition of organization: It is general, i.e. following von Neumann the details of the physics and chemistry of this universe are not involved; it measures organized structure rather than unstructured details; and it passes the spontaneous generation or "Pasteur" test, i.e. there is a very low probability of creating organization by chance without a long evolutionary process (this may be viewed as a way of restating Theorem 1 in Section 5). The second point is worth elaborating: The information content of an organism includes much irrelevant detail, and a bigger animal is necessarily more complex in this sense. *But if it were possible to calculate the mutual information of two arbitrary cells in a body at a given moment, we surmise that this would give a measure of the genetic information in a cell. This is because the irrelevant details in each of them, such as the exact position and velocity of each molecule, are uncorrelated and would cancel each other out.*

In addition to providing a definition of information content and of degree of organization, this approach also provides a definition of "organism" in the sense that a theoretical prescription is given for dissecting a scene into organisms and determining their boundaries, so that the measure of degree of organization can then be applied separately to each organism. However a strong note of caution is in order: We agree with [1] that a definition of "life" is valid as long as anything that satisfies the definition and is likely to appear in the universe under consideration, either is alive or is a by-product of living beings or their activities. There certainly are structures satisfying our definition that are not alive (see Theorems 3 to 6 in Section 5); however, we believe that they would only be likely to arise as by-products of the activities of living beings.

In the succeeding sections we shall do the following: give a more formal presentation of the basic concepts of algorithmic information theory; discuss the notions of the independence and mutual information

of groups of more than two objects; formally define H_d; evaluate $H_d(R)$ for some typical one-dimensional geometrical patterns R which we dub "gas," "crystal," "twins," "bilateral symmetry," and "hierarchy;" consider briefly the problem of decomposing scenes containing several independent patterns, and of determining the boundary of a pattern which is imbedded in an unstructured background; discuss briefly the two and higher dimension cases; and mention some alternative definitions of mutual information which have been proposed.

The next step in this program of research would be to proceed from static snapshots to time-varying situations, in other words, to set up a discrete universe with probabilistic state transitions and to show that there is a certain probability that a certain level of organization will be reached by a certain time. More generally, one would like to determine the probability distribution of the maximum degree of organization of any organism at time $t + \Delta$ as a function of it at time t. Let us propose an initial proof strategy for setting up a nontrivial example of the evolution of organisms: construct a series of intermediate evolutionary forms [11], argue that increased complexity gives organisms a selective advantage, and show that no primitive organism is so successful or lethal that it diverts or blocks this gradual evolutionary pathway. What would be the intellectual flavor of the theory we desire? It would be a quantitative formulation of Darwin's theory of evolution in a very general model universe setting. It would be the opposite of ergodic theory. Instead of showing that things mix and become uniform, it would show that variety and organization will probably increase.

Some final comments: Software is fast approaching biological levels of complexity, and hardware, thanks to very large scale integration, is not far behind. Because of this, we believe that the computer is now becoming a valid metaphor for the entire organism, not just for the brain [12]. Perhaps the most interesting example of this is the evolutionary phenomenon suffered by extremely large programs such as operating systems. It becomes very difficult to make changes in such programs, and the only alternative is to add new features rather than modify existing ones. The genetic program has been "patched up" much more and over a much longer period of time than even the largest operating systems, and Nature has accomplished this in much the same manner as systems programmers have, by carrying along all

the previous code as new code is added [11]. The experimental proof of this is that ontogeny recapitulates phylogeny, i.e. each embryo to a certain extent recapitulates in the course of its development the evolutionary sequence that led to it. In this connection we should also mention the thesis developed in [13] that the information contained in the human brain is now comparable with the amount of information in the genes, and that intelligence plus education may be characterized as a way of getting around the limited modifiability and channel capacity of heredity. In other words, Nature, like computer designers, has decided that it is much more flexible to build general-purpose computers than to use heredity to "hardwire" each behavior pattern instinctively into a special-purpose computer.

2. Algorithmic Information Theory

We first summarize some of the basic concepts of algorithmic information theory in its most recent formulation [14–16].

This new approach leads to a formalism that is very close to that of classical probability theory and information theory, and is based on the notion that the tape containing the Turing machine's program is infinite and entirely filled with 0's and 1's. This forces programs to be self-delimiting; i.e. they must contain within themselves information about their size, since the computer cannot rely on a blank at the end of the program to indicate where it ends.

Consider a universal Turing machine U whose programs are in binary and are self-delimiting. By "self-delimiting" we mean, as was just explained, that they do not have blanks appended as endmarkers. By "universal" we mean that for any other Turing machine M whose programs p are in binary and are self-delimiting, there is a prefix μ such that $U(\mu p)$ always carries out the same computation as $M(p)$.

$H(X)$, the algorithmic information content of the finite object X, is defined to be the size in bits of the smallest self-delimiting program for U to compute X. This includes the proviso that U halt after printing X. There is absolutely no restriction on the running time or storage space used by this program. For example, X can be a natural number or a bit string or a tuple of natural numbers or bit strings. Note that

variations in the definition of U give rise to at most $O(1)$ differences in the resulting H, by the definition of universality.

The self-delimiting requirement is adopted so that one gets the following basic subadditivity property of H:

$$H(\langle X, Y \rangle) \leq H(X) + H(Y) + O(1).$$

This inequality holds because one can concatenate programs. It expresses the notion of "adding information," or, in computer jargon, "using subroutines."

Another important consequence of this requirement is that a natural probability measure P, which we shall refer to as the algorithmic probability, can be associated with the result of any computation. $P(X)$ is the probability that X is obtained as output if the standard universal computer U is started running on a program tape filled with 0's and 1's by separate tosses of a fair coin. The algorithmic probability P and the algorithmic information content H are related as follows [14]:

$$H(X) = -\log P(X) + O(1). \tag{1}$$

Consider a binary string s. Define the function L as follows:

$$L(n) = \max\{H(s) : \text{length}(s) = n\}.$$

It can be shown [14] that $L(n) = n + H(n) + O(1)$, and that an overwhelming majority of the s of length n have $H(s)$ very close to $L(n)$. Such s have maximum information content and are highly random, patternless, incompressible, and typical. They are said to be "algorithmically random." The greater the difference between $H(s)$ and $L(\text{length}(s))$, the less random s is. It is convenient to say that "s is k-random". if $H(s) \geq L(n) - k$, where $n = \text{length}(s)$. There are at most

$$2^{n-k+O(1)}$$

n-bit strings which aren't k-random. As for natural numbers, most n have $H(n)$ very close to $L(\text{floor}(\log n))$ Here $\text{floor}(x)$ is the greatest integer $\leq x$. Strangely enough, though most strings are random it is impossible to prove that specific strings have this property. For an

explanation of this paradox and further references, see the section on metamathematics in [15], and also see [9].

We now make a few observations that will be needed later. First of all, $H(n)$ is a smooth function of n:

$$|H(n) - H(m)| = O(\log|n-m|). \tag{2}$$

(Note that this is not strictly true if $|n-m|$ is equal to 0 or 1, unless one considers the log of 0 and 1 to be 1; this convention is therefore adopted throughout this paper.) For a proof, see [16]. The following upper bound on $H(n)$ is an immediate corollary of this smoothness property: $H(n) = O(\log n)$. Hence if s is an n-bit string, then $H(s) \leq n + O(\log n)$. Finally, note that changes in the value of the argument of the function L produce nearly equal changes in the value of L. Thus, for any ϵ there is a δ such that $L(n) \geq L(m) + \epsilon$ if $n \geq m + \delta$. This is because of the fact that $L(n) = n + H(n) + O(1)$ and the smoothness property (2) of H.

An important concept of algorithmic information theory that hasn't been mentioned yet is the conditional probability $P(Y|X)$, which by definition is $P(\langle X, Y\rangle)/P(X)$. To the conditional probability there corresponds the relative information content $H(Y|X^*)$, which is defined to be the size in bits of the smallest programs for the standard universal computer U to output Y if it is given X^*, a canonical minimum-size program for calculating X. X^* is defined to be the first $H(X)$-bit program to compute X that one encounters in a fixed recursive enumeration of the graph of U (i.e. the set of all ordered pairs of the form $\langle p, U(p)\rangle$). Note that there are partial recursive functions which map X^* to $\langle X, H(X)\rangle$ and back again, and so X^* may be regarded as an abbreviation for the ordered pair whose first element is the string X and whose second element is the natural number that is the complexity of X. We should also note the immediate corollary of (1) that minimum-size or nearly minimum-size programs are essentially unique: For any ϵ there is a δ such that for all X the cardinality of {the set of all programs for U to calculate X that are within ϵ bits of the minimum size $H(X)$} is less than δ. It is possible to prove the following theorem relating the conditional probability and the relative information content [14]:

$$H(Y^*|X) = -\log P(Y|X) + O(1). \tag{3}$$

From (1) and (3) and the definition $P(\langle X, Y\rangle) = P(X)P(Y|X)$, one obtains this very basic decomposition:

$$H(\langle X, Y\rangle) = H(X) + H(Y|X^*) + O(1). \tag{4}$$

3. Independence and Mutual Information

It is an immediate corollary of (4) that the following four quantities are all within $O(1)$ of each other:

$$\begin{cases} H(X) - H(X|Y^*), \\ H(Y) - H(Y|X^*), \\ H(X) + H(Y) - H(\langle X, Y\rangle), \\ H(Y) + H(X) - H(\langle Y, X\rangle). \end{cases}$$

These four quantities are known as the mutual information $H(X : Y)$ of X and Y; they measure the extent to which X and Y are interdependent. For if $P(\langle X, Y\rangle) \approx P(X)P(Y)$, then $H(X : Y) = O(1)$; and if Y is a recursive function of X, then $H(Y|X^*) = O(1)$ and $H(X : Y) = H(Y) + O(1)$. In fact,

$$H(X : Y) = -\log\left[\frac{P(X)P(Y)}{P(\langle X, Y\rangle)}\right] + O(1),$$

which shows quite clearly that $H(X : Y)$ is a symmetric measure of the independence of X and Y. Note that in algorithmic information theory, what is of importance is an approximate notion of independence and a measure of its degree (mutual information), rather than the exact notion. This is because the algorithmic probability may vary within a certain percentage depending on the choice of universal computer U. Conversely, information measures in algorithmic information theory should not vary by more than $O(1)$ depending on the choice of U.

To motivate the definition of the d-diameter complexity, we now discuss how to generalize the notion of independence and mutual information from a pair to an n-tuple of objects. In what follows classical and algorithmic probabilities are distinguished by using curly brackets for the first one and parentheses for the second. In probability theory

the mutual independence of a set of n events $\{A_k : k < n\}$ is defined by the following 2^n equations:

$$\prod_{k\in S} P\{A_k\} = P\{\bigcap_{k\in S} A_k\}$$

for all $S \subset n$. Here the set-theoretic convention due to von Neumann is used that identifies the natural number n with the set $\{k : k < n\}$. In algorithmic probability theory the analogous condition would be to require that

$$\prod_{k\in S} P(A_k) \approx P(\bigsqcup_{k\in S} A_k) \tag{5}$$

for all $S \subset n$. Here $\bigsqcup A_k$ denotes the tuple forming operation for a variable length tuple, i.e.

$$\bigsqcup_{k<n} A_k = \langle A_0, A_1, A_2, \ldots, A_{n-1}\rangle.$$

It is a remarkable fact that these 2^n conditions (5) are equivalent to the single requirement that

$$\prod_{k<n} P(A_k) \approx P(\bigsqcup_{k<n} A_k). \tag{6}$$

To demonstrate this it is necessary to make use of special properties of algorithmic probability that are not shared by general probability measures. In the case of a general probability space,

$$P\{A \cap B\} \geq P\{A\} + P\{B\} - 1$$

is the best lower bound on $P\{A\cap B\}$ that can in general be formulated in terms of $P\{A\}$ and $P\{B\}$. For example, it is possible for $P\{A\}$ and $P\{B\}$ to both be $1/2$, while $P\{A\cap B\} = 0$. In algorithmic information theory the situation is quite different. In fact one has:

$$P(\langle A, B\rangle) \geq c_2 P(A)P(B),$$

and this generalizes to any fixed number of objects:

$$P(\bigsqcup_{k<n} A_k) \geq c_n \prod_{k<n} P(A_k).$$

Thus if the joint algorithmic probability of a subset of the n-tuple of objects were significantly greater than the product of their individual algorithmic probabilities, then this would also hold for the entire n-tuple of objects. More precisely, for any $S \subset n$ one has

$$P(\bigsqcup_{k<n} A_k) \geq c'_n P(\bigsqcup_{k\in S} A_k) P(\bigsqcup_{k\in n-S} A_k) \geq c''_n P(\bigsqcup_{k\in S} A_k) \prod_{k\in n-S} P(A_k).$$

Then if one assumes that

$$P(\bigsqcup_{k\in S} A_k) \gg \prod_{k\in S} P(A_k)$$

(here $\gg$ denotes "much greater than"), it follows that

$$P(\bigsqcup_{k<n} A_k) \gg \prod_{k<n} P(A_k)$$

We conclude that in algorithmic probability theory (5) and (6) are equivalent and thus (6) is a necessary and sufficient condition for an n-tuple to be mutually independent. Therefore the following measure of mutual information for n-tuples accurately characterizes the degree of interdependence of n objects:

$$[\sum_{k<n} H(A_k)] - H(\bigsqcup_{k<n} A_k).$$

This measure of mutual information subsumes all others in the following precise sense:

$$[\sum_{k<n} H(A_k)] - H(\bigsqcup_{k<n} A_k) = \max\{[\sum_{k\in S} H(A_k)] - H(\bigsqcup_{k\in S} A_k)\} + O(1),$$

where the maximum is taken over all $S \subset n$.

4. Formal Definition of Hd

We can now present the definition of the d-diameter complexity $H_d(R)$. We assume a geometry: graph paper of some finite number of dimensions that is divided into unit cubes. Each cube is black or white,

opaque or transparent, in other words, contains a 1 or a 0. Instead of requiring an output tape which is multidimensional, our universal Turing machine U outputs tuples giving the coordinates and the contents (0 or 1) of each unit cube in a geometrical object that it wishes to print. Of course geometrical objects are considered to be the same if they are translation equivalent. We choose for this geometry the city-block metric

$$D(X,Y) = \max |x_i - y_i|,$$

which is more convenient for our purposes than the usual metric. By a region we mean a set of unit cubes with the property that from any cube in it to any other one there is a path that only goes through other cubes in the region. To this we add the constraint which in the 3-dimensional case is that the connecting path must only pass through the interior and faces of cubes in the region, not through their edges or vertices. The diameter of an arbitrary region R is denoted by $|R|$, and is defined to be the minimum diameter $2r$ of a "sphere"

$$\{X : D(X, X_0) \leq r\}$$

which contains R. $H_d(R)$, the size in bits of the smallest programs which calculate R as the "sum" of independent regions of diameter $\leq d$, is defined as follows:

$$H_d(R) = \min[\alpha + \sum_{i<k} H(R_i)],$$

where

$$\alpha = H(R|\bigsqcup_{i<k} R_i) + H(k),$$

the minimization being taken over all k and partitions of R into k-tuples $\bigsqcup R_i$ of nonoverlapping regions with the property that $|R_i| < d$ for all $i < k$.

The discussion in Section 3 of independence and mutual information shows that $H_d(R)$ is a natural measure to consider. Excepting the α term, $H_d(R) - H(R)$ is simply the minimum attainable mutual information over any partition of R into nonoverlapping pieces all of size not greater than d. We shall see in Section 5 that in practice the

min is attained with a small number of pieces and the α term is not very significant.

A few words about α, the number of bits of information needed to know how to assemble the pieces: The $H(k)$ term is included in α, as illustrated in Lemma 1 below, because it is the number of bits needed to tell U how many descriptions of pieces are to be read. The $H(R|\bigsqcup R_i)$ term is included in α because it is the number of bits needed to tell U how to compute R given the k-tuple of its pieces. This is perhaps the most straight-forward formulation, and the one that is closest in spirit to Section 5 [5]. However, less information may suffice, e.g.

$$H(R|\langle k^*, \bigsqcup_{i<k}(R_i^*)\rangle) + H(k)$$

bits. In fact, one could define α to be the minimum number of bits in a string which yields a program to compute the entire region when it is concatenated with minimum-size programs for all the pieces of the region; i.e. one could take

$$\alpha = \min\{|p| : U(pR_0^*R_1^*R_2^* \ldots R_{k-1}^*) = R\}.$$

Here are two basic properties of H_d: If $d \geq |R|$, then $H_d(R) = H(R) + O(1)$; $H_d(R)$ increases monotonically as d decreases. $H_d(R) = H(R) + O(1)$ if $d \geq |R|$ because we have included the α term in the definition of $H_d(R)$. $H_d(R)$ increases as d decreases because one can no longer take advantage of patterns of diameter greater than d to describe R. The curve showing $H_d(R)$ as a function of d may be considered a kind of "Fourier spectrum" of R. Interesting things will happen to the curve at d which are the sizes of significant patterns in R.

Lemma 1. ("Subadditivity for n-tuples")

$$H(\bigsqcup_{k<n} A_k) \leq c_n + \sum_{k<n} H(A_k).$$

Proof.

$$\begin{aligned} H(\bigsqcup_{k<n} A_k) &= H(\langle n, \bigsqcup_{k<n} A_k\rangle) + O(1) \\ &= H(n) + H(\bigsqcup_{k<n} A_k|n^*) + O(1) \\ &\leq c' + H(n) + \sum_{k<n} H(A_k). \end{aligned}$$

Hence one can take

$$c_n = c' + H(n).$$

5. Evaluation of Hd for Typical One-Dimensional Geometrical Patterns

Before turning to the examples, we present a lemma needed for estimating $H_d(R)$. The idea is simply that sufficiently large pieces of a random string are also random. It is required that the pieces be sufficiently large for the following reason: It is not difficult to see that for any j, there is an n so large that random strings of size greater than n must contain all 2^j possible subsequences of length j. In fact, for n sufficiently large the relative frequency of occurrence of all 2^j possible subsequences must approach the limit 2^{-j}.

Lemma 2. ("Random parts of random strings")
Consider an n-bit string s to be a loop. For any natural numbers i and j between 1 and n, consider the sequence u of contiguous bits from s starting at the ith and continuing around the loop to the jth. Then if s is k-random, its subsequence u is $(k + O(\log n))$-random.

Proof. The number of bits in u is $j - i + 1$ if j is $\geq i$, and is $n + j - i + 1$ if j is $< i$. Let v be the remainder of the loop s after u has been excised. Then we have $H(u) + H(v) + H(i) + O(1) \geq H(s)$. Thus $H(u) + n - |u| + O(\log n) \geq H(s)$, or $H(u) \geq H(s) - n + |u| + O(\log n)$. Thus if s is k-random, i.e. $H(s) \geq L(n) - k = n + H(n) - k + O(1)$, then u is x-random, where x is determined as follows: $H(u) \geq n + H(n) - k - n + |u| + O(\log n) = |u| + H(|u|) - k + O(\log n)$. That is to say, if s is k-random, then its subsequence u is $(k + O(\log n))$-random.

Lemma 3. ("Random prefixes of random strings")
Consider an n-bit string s. For any natural number j between 1 and n, consider the sequence u consisting of the first j bits of s. Then if s is k-random, its j-bit prefix u is $(O(\log j) + k)$-random.

Proof. Let the $(n - j)$-bit string v be the remainder of s after u is excised. Then we have $H(u) + H(v) + O(1) \geq H(s)$, and therefore $H(u) \geq H(s) - L(n - j) + O(1) = L(n) - k - L(n - j) + O(1)$ since s is k-random. Note that $L(n) - L(n - j) = j + H(n) - H(n - j) + O(1)$

$= j + O(\log j)$, by the smoothness property (2) of H. Hence $H(u) \geq j + O(\log j) - k$. Thus if u is x-random (x as small as possible), we have $L(j) - x = j + O(\log j) - x \geq j + O(\log j) - k$. Hence $x \leq O(\log j) + k$.

Remark. Conversely, any random n-bit string can be extended by concatenating k bits to it in such a manner that the result is a random $(n + k)$-bit string. We shall not use this converse result, but it is included here for the sake of completeness.

Lemma 4. ("Random extensions of random strings")
Assume the string s is x-random. Consider a natural number k. Then there is a k-bit string e such that se is y-random, as long as k, x, and y satisfy a condition of the following form:

$$y \geq x + O(\log x) + O(\log k).$$

Proof. Assume on the contrary that the x-random string s has no y-random k-bit extension and $y \geq x + O(\log x) + O(\log k)$, i.e. $x < y + O(\log y) + O(\log k)$. From this assumption we shall derive a contradiction by using the fact that most strings of any particular size are y-random, i.e. the fraction of them that are y-random is at least

$$1 - 2^{-y+O(1)}.$$

It follows that the fraction of $|s|$-bit strings which have no y-random k-bit extension is less than

$$2^{-y+O(1)}.$$

Since by hypothesis no k-bit extension of s is y-random, we can uniquely determine s if we are given y and k and the ordinal number of the position of s in {the set of all $|s|$-bit strings which have no y-random k-bit extension} expressed as an $(|s| - y + O(1))$-bit string. Hence $H(s)$ is less than $L(|s| - y + O(1)) + H(y) + H(k) + O(1)$. In as much as $L(n) = n + H(n) + O(1)$ and $|H(n) - H(m)| = O(\log |n - m|)$, it follows that $H(s)$ is less than $L(|s|) - [y + O(\log y) + O(\log k)]$. Since s is by assumption x-random, i.e. $H(s) \geq L(|s|) - x$, we obtain a lower bound on x of the form $y + O(\log y) + O(\log k)$, which contradicts our original assumption that $x < y + O(\log y) + O(\log k)$.

Theorem 1. ("Gas")
Suppose that the region R is an $O(\log n)$-random n-bit string. Consider $d = n/k$, where n is large, and k is fixed and greater than zero. Then

$$H(R) = n + O(\log n), \text{ and } H_d(R) = H(R) + O(\log H(R)).$$

Proof that $H_d(R) \leq H(R) + O(\log H(R))$
Let β be concatenation of tuples of strings, i.e.

$$\beta(\bigsqcup_{i \leq k} R_i) = R_0 R_1 R_2 \ldots R_k.$$

Note that

$$H(\beta(\bigsqcup_{i \leq k} R_i) | \bigsqcup_{i \leq k} R_i) = O(1).$$

Divide R into k successive strings of size floor$(|R|/k)$, with one (possibly null) string of size less than k left over at the end. Taking this choice of partition $\bigsqcup R_i$ in the definition of $H_d(R)$, and using the fact that $H(s) \leq |s| + O(\log |s|)$, we see that

$$\begin{aligned} H_d(R) &\leq O(1) + H(k+1) + \sum_{i \leq k} \{|R_i| + O(\log |R_i|)\} \\ &\leq O(1) + n + (k+2)O(\log n) \\ &= n + O(\log n). \end{aligned}$$

Proof that $H_d(R) \geq H(R) + O(\log H(R))$
This follows immediately from the fact that $H_{|R|}(R) = H(R) + O(1)$ and $H_d(R)$ increases monotonically as d decreases.

Theorem 2. ("Crystal")
Suppose that the region R is an n-bit string consisting entirely of 1's, and that the base-two numeral for n is $O(\log \log n)$-random. Consider $d = n/k$, where n is large, and k is fixed and greater than zero. Then

$$H(R) = \log n + O(\log \log n), \text{ and } H_d(R) = H(R) + O(\log H(R)).$$

Proof that $H_d(R) \leq H(R) + O(\log H(R))$
If one considers using the concatenation function β for assembly as was done in the proof of Theorem 1, and notes that $H(1^n) = H(n) + O(1)$, one sees that it is sufficient to partition the natural number n into

$O(k)$ summands none of which is greater than n/k in such a manner that $H(n) + O(\log\log n)$ upper bounds the sum of the complexities of the summands. Division into equal size pieces will not do, because $H(\text{floor}(n/k)) = H(n) + O(1)$, and one only gets an upper bound of $kH(n) + O(1)$. It is necessary to proceed as follows: Let m be the greatest natural number such that $2^m \leq n/k$. And let p be the smallest natural number such that $2^p > n$. By converting n to base-two notation, one can express n as the sum of $\leq p$ distinct non-negative powers of two. Divide all these powers of two into two groups: those that are less than 2^m and those that are greater than or equal to 2^m. Let f be the sum of all the powers in the first group. f is $< 2^m \leq n/k$. Let s be the sum of all the powers in the second group. s is a multiple of 2^m; in fact, it is of the form $t2^m$ with $t = O(k)$. Thus $n = f + s = f + t2^m$, where $f \leq n/k$, $2^m \leq n/k$, and $t = O(k)$. The complexity of 2^m is $H(m) + O(1)$ $= O(\log m) = O(\log\log n)$. Thus the sum of the complexities of the t summands 2^m is also $O(\log\log n)$. Moreover, f when expressed in base-two notation has $\log k + O(1)$ fewer bit positions on the left than n does. Hence the complexity of f is $H(n) + O(1)$. In summary, we have $O(k)$ quantities n_i with the following properties:

$$n = \sum n_i, \;\; n_i \leq n/k, \;\; \sum H(n_i) \leq H(n) + O(\log\log n).$$

Thus $H_d(R) \leq H(R) + O(\log H(R))$.

Proof that $H_d(R) \geq H(R) + O(\log H(R))$

This follows immediately from the fact that $H_{|R|}(R) = H(R) + O(1)$ and $H_d(R)$ increases monotonically as d decreases.

Theorem 3. ("Twins")
For convenience assume n is even. Suppose that the region R consists of two repetitions of an $O(\log n)$-random $n/2$-bit string u. Consider $d = n/k$, where n is large, and k is fixed and greater than unity. Then

$$H(R) = n/2 + O(\log n), \text{ and } H_d(R) = 2H(R) + O(\log H(R)).$$

Proof that $H_d(R) \leq 2H(R) + O(\log H(R))$

The reasoning is the same as in the case of the "gas" (Theorem 1). Partition R into k successive strings of size floor$(|R|/k)$, with one (possibly null) string of size less than k left over at the end.

Proof that $H_d(R) \geq 2H(R) + O(\log H(R))$

By the definition of $H_d(R)$, there is a partition $\bigsqcup R_i$ of R into nonoverlapping regions which has the property that

$$H_d(R) = \alpha + \sum H(R_i), \;\; \alpha = H(R|\bigsqcup R_i) + H(k), \;\; |R_i| \le d.$$

Classify the non-null R_i into three mutually exclusive sets A, B, and C: A is the set of all non-null R_i which come from the left half of R ("the first twin"), B is the (empty or singleton) set of all non-null R_i which come from both halves of R ("straddles the twins"), and C is the set of all non-null R_i which come from the right half of R ("the second twin"). Let A', B', and C' be the sets of indices i of the regions R_i in A, B, and C, respectively. And let A'', B'', and C'' be the three portions of R which contained the pieces in A, B, and C, respectively. Using the idea of Lemma 1, one sees that

$$\begin{aligned} H(A'') &\le O(1) + H(\#(A)) + \sum_{i\in A'} H(R_i), \\ H(B'') &\le O(1) + H(\#(B)) + \sum_{i\in B'} H(R_i), \\ H(C'') &\le O(1) + H(\#(C)) + \sum_{i\in C'} H(R_i). \end{aligned}$$

Here # denotes the cardinality of a set. Now A'', B'', and C'' are each a substring of an $O(\log n)$-random $n/2$-bit string. This assertion holds for B'' for the following two reasons: the $n/2$-bit string is considered to be a loop, and $|B''| \le d = n/k \le n/2$ since k is assumed to be greater than 1. Hence, applying Lemma 2, one obtains the following inequalities:

$$\begin{aligned} |A''| + O(\log n) &\le H(A''), \\ |B''| + O(\log n) &\le H(B''), \\ |C''| + O(\log n) &\le H(C''). \end{aligned}$$

Adding both of the above sets of three inequalities and using the facts that

$$|A''| + |B''| + |C''| = |R| = n, \;\; \#(A) \le n/2, \;\; \#(B) \le 1, \;\; \#(C) \le n/2,$$

and that $H(m) = O(\log m)$, one sees that

$$\begin{aligned} n + O(\log n) &\leq H(A'') + H(B'') + H(C'') \\ &\leq O(1) + H(\#(A)) + H(\#(B)) + H(\#(C)) + \\ &\quad \sum\{H(R_i) : i \in A' \cup B' \cup C'\} \\ &\leq O(\log n) + \sum H(R_i). \end{aligned}$$

Hence

$$H_d(R) \geq \sum H(R_i) \geq n + O(\log n) = 2H(R) + O(\log H(R)).$$

Theorem 4. ("Bilateral Symmetry")
For convenience assume n is even. Suppose that the region R consists of an $O(\log n)$-random $n/2$-bit string u concatenated with its reversal. Consider $d = n/k$, where n is large, and k is fixed and greater than zero. Then

$$H(R) = n/2 + O(\log n), \text{ and } H_d(R) = (2 - k^{-1})H(R) + O(\log H(R)).$$

Proof. The proof is along the lines of that of Theorem 3, with one new idea. In the previous proof we considered B'' which is the region R_i in the partition of R that straddles R's midpoint. Before B'' was $O(\log |R|)$-random, but now it can be compressed into a program about half its size, i.e. about $|B''|/2$ bits long. Hence the maximum departure from randomness for B'' is for it to only be $O(\log |R|) + (|R|/2k)$-random, and this is attained by making B'' as large as possible and having its midpoint coincide with that of R.

Theorem 5. ("Hierarchy")
For convenience assume n is a power of two. Suppose that the region R is constructed in the following fashion. Consider an $O(1)$-random $\log n$-bit string s. Start with the one-bit string 1, and successively concatenate the string with itself or with its bit by bit complement, so that its size doubles at each stage. At the ith stage, the string or its complement is chosen depending on whether the ith bit of s is a 0 or a 1, respectively. Consider the resulting n-bit string R and $d = n/k$, where n is large, and k is fixed and greater than zero. Then

$$H(R) = \log n + O(\log \log n), \text{ and } H_d(R) = kH(R) + O(\log H(R)).$$

Proof that $H_d(R) \leq kH(R) + O(\log H(R))$

The reasoning is similar to the case of the upper bounds on $H_d(R)$ in Theorems 1 and 3. Partition R into k successive strings of size floor($|R|/k$), with one (possibly null) string of size less than k left over at the end.

Proof that $H_d(R) \geq kH(R) + O(\log H(R))$

Proceeding as in the proof of Theorem 3, one considers a partition $\bigsqcup R_i$ of R that realizes $H_d(R)$. Using Lemma 3, one can easily see that the following lower bound holds for any substring R_i of R:

$$H(R_i) \geq \max\{1, \log|R_i| - c\log\log|R_i|\}.$$

The max $\{1, \ldots\}$ is because H is always greater than or equal to unity; otherwise U would have only a single output. Hence the following expression is a lower bound on $H_d(R)$:

$$\sum \Phi(|R_i|), \tag{7}$$

where

$$\Phi(x) = \max\{1, \log x - c\log\log x\}, \;\; \sum|R_i| = |R| = n, \;\; |R_i| \leq d.$$

It follows that one obtains a lower bound on (7) and thus on $H_d(R)$ by solving the following minimization problem: Minimize

$$\sum \Phi(n_i) \tag{8}$$

subject to the following constraints:

$$\sum n_i = n, \;\; n_i \leq n/k, \;\; n \text{ large}, \;\; k \text{ fixed}.$$

Now to do the minimization. Note that as x goes to infinity, $\Phi(x)/x$ goes to the limit zero. Furthermore, the limit is never attained, i.e. $\Phi(x)/x$ is never equal to zero. Moreover, for x and y sufficiently large and x less than y, $\Phi(x)/x$ is greater than $\Phi(y)/y$. It follows that a sum of the form (8) with the n_i constrained as indicated is minimized by making the n_i as large as possible. Clearly this is achieved by taking

all but one of the n_i equal to floor(n/k), with the last n_i equal to remainder(n/k). For this choice of n_i the value of (8) is

$$\begin{aligned} & k[\log n + O(\log\log n)] + \Phi(\text{remainder}(n/k)) \\ = \;& k\log n + O(\log\log n) \\ = \;& kH(R) + O(\log H(R)). \end{aligned}$$

Theorem 6. For convenience assume n is a perfect square. Suppose that the region R is an n-bit string consisting of $\sqrt{n}$ repetitions of an $O(\log n)$-random $\sqrt{n}$ bit string u. Consider $d = n/k$, where n is large, and k is fixed and greater than zero. Then

$$H(R) = \sqrt{n} + O(\log n), \text{ and } H_d(R) = kH(R) + O(\log H(R)).$$

Proof that $H_d(R) \leq kH(R) + O(\log H(R))$

The reasoning is identical to the case of the upper bound on $H_d(R)$ in Theorem 5.

Proof that $H_d(R) \geq kH(R) + O(\log H(R))$

Proceeding as in the proof of Theorem 5, one considers a partition $\bigsqcup R_i$ of R that realizes $H_d(R)$. Using Lemma 2, one can easily see that the following lower bound holds for any substring R_i of R:

$$H(R_i) \geq \max\{1, -c\log n + \min\{\sqrt{n}, |R_i|\}\}.$$

Hence the following expression is a lower bound on $H_d(R)$:

$$\sum \Phi_n(|R_i|), \tag{9}$$

where

$$\Phi_n(x) = \max\{1, -c\log n + \min\{\sqrt{n}, x\}\}, \;\; \sum |R_i| = |R| = n, \;\; |R_i| \leq d.$$

It follows that one obtains a lower bound on (9) and thus on $H_d(R)$ by solving the following minimization problem: Minimize

$$\sum \Phi_n(n_i) \tag{10}$$

subject to the following constraints:

$$\sum n_i = n, \;\; n_i \leq n/k, \;\; n \text{ large}, \;\; k \text{ fixed}.$$

Now to do the minimization. Consider $\Phi_n(x)/x$ as x goes from 1 to n. It is easy to see that this ratio is much smaller, on the order of $1/\sqrt{n}$, for x near to n than it is for x anywhere else in the interval from 1 to n. Also, for x and y both greater than $\sqrt{n}$ and x less than y, $\Phi_n(x)/x$ is greater than $\Phi_n(y)/y$. It follows that a sum of the form (10) with the n_i constrained as indicated is minimized by making the n_i as large as possible. Clearly this is achieved by taking all but one of the n_i equal to floor(n/k), with the last n_i equal to remainder(n/k). For this choice of n_i the value of (10) is

$$\begin{aligned} & k[\sqrt{n} + O(\log n)] + \Phi_n(\text{remainder}(n/k)) \\ = \; & k\sqrt{n} + O(\log n) \\ = \; & kH(R) + O(\log H(R)). \end{aligned}$$

6. Determining Boundaries of Geometrical Patterns

What happens to the structures of Theorems 3 to 6 if they are imbedded in a gas or crystal, i.e. in a random or constant 0 background? And what about scenes with several independent structures imbedded in them—do their degrees of organization sum together? Is our definition sufficiently robust to work properly in these circumstances?

This raises the issue of determining the boundaries of structures. It is easy to pick out the hierarchy of Theorem 5 from an unstructured background. Any two "spheres" of diameter δ will have a high mutual information given δ^* if and only if they are both in the hierarchy instead of in the background. Here we are using the notion of the mutual information of X and Y given Z, which is denoted $H(X:Y|Z)$, and is defined to be $H(X|Z) + H(Y|Z) - H(\langle X,Y\rangle|Z)$. The special case of this concept that we are interested in, however, can be expressed more simply: for if X and Y are both strings of length n, then it can be shown that $H(X:Y|n^*) = H(X|Y) - H(n)$. This is done by using the decomposition (4) and the fact that since X and Y are both of length n, $H(\langle n,X\rangle) = H(X) + O(1)$, $H(\langle n,Y\rangle) = H(Y) + O(1)$, and

$H(\langle n, \langle X, Y\rangle\rangle) = H(\langle X, Y\rangle) + O(1)$, and thus

$$\begin{aligned} H(X|n^*) &= H(X) - H(n) + O(1), \\ H(Y|n^*) &= H(Y) - H(n) + O(1), \\ H(\langle X, Y\rangle|n^*) &= H(\langle X, Y\rangle) - H(n) + O(1). \end{aligned}$$

How can one dissect a structure from a comparatively unorganized background in the other cases, the structures of Theorems 3, 4, and 6? The following definition is an attempt to provide a tool for doing this: An ϵ, δ-pattern R is a maximal region ("maximal" means not extensible, not contained in a bigger region R' which is also an ϵ, δ-pattern) with the property that for any δ-diameter sphere R_1 in R there is a disjoint δ-diameter sphere R_2 in R such that

$$H(R_1 : R_2|\delta^*) \geq \epsilon.$$

The following questions immediately arise: What is the probability of having an ϵ, δ-pattern in an n-bit string, i.e. what proportion of the n-bit strings contain an ϵ, δ-pattern? This is similar to asking what is the probability that an n-bit string s satisfies

$$H_{n/k}(s) - H(s) > x.$$

A small upper bound on the latter probability can be derived from Theorem 1.

7. Two and Higher Dimension Geometrical Patterns

We make a few brief remarks.

In the general case, to say that a geometrical object O is "random" means $H(O|\text{shape}(O)^*) \approx \text{volume}(O)$, or $H(O) \approx \text{volume}(O) + H(\text{shape}(O))$. Here shape$(O)$ denotes the object O with all the 1's that it contains in its unit cubes changed to 0's. Here are some examples: A random n by n square has complexity

$$n^2 + H(n) + O(1).$$

A random n by m rectangle doesn't have complexity $nm + H(n) + H(m) + O(1)$, for if $m = n$ this states that a random n by n square has complexity

$$n^2 + 2H(n) + O(1),$$

which is false. Instead a random n by m rectangle has complexity $nm + H(\langle n, m\rangle) + O(1) = nm + H(n) + H(m|n^*) + O(1)$, which gives the right answer for $m = n$, since $H(n|n^*) = O(1)$. One can show that most n by m rectangles have complexity $nm + H(\langle n, m\rangle) + O(1)$, and less than two raised to the $nm - k + O(1)$ have complexity less than $nm + H(\langle n, m\rangle) - k$.

Here is a two-dimensional version of Lemma 2: Any large chunk of a random square which has a shape that is easy to describe, must itself be random.

8. Common Information

We should mention some new concepts that are closely related to the notion of mutual information. They are called measures of common information. Here are three different expressions defining the common information content of two strings X and Y. In them the parameter ϵ denotes a small tolerance, and as before $H(X : Y|Z)$ denotes $H(X|Z) + H(Y|Z) - H(\langle X, Y\rangle|Z)$.

$$\max\{H(Z) : H(Z|X^*) < \epsilon \,\&\, H(Z|Y^*) < \epsilon\}$$
$$\min\{H(\langle X, Y\rangle : Z) : H(X : Y|Z^*) < \epsilon\}$$
$$\min\{H(Z) : H(X : Y|Z^*) < \epsilon\}$$

Thus the first expression for the common information of two strings defines it to be the maximum information content of a string that can be extracted easily from both, the second defines it to be the minimum of the mutual information of the given strings and any string in the light of which the given strings look nearly independent, and the third defines it to be the minimum information content of a string in the light of which the given strings appear nearly independent. Essentially these definitions of common information are given in [17–19]. [17] considers an algorithmic formulation of its common information measure, while [18] and [19] deal exclusively with the classical ensemble setting.

Appendix 1: Errors in [5]

. . . The definition of the d-diameter complexity given in [5] has a basic flaw which invalidates the entries for $R = R_2, R_3$, and R_4 and $d = n/k$ in the table in [5]: It is insensitive to changes in the diameter d . . .

There is also another error in the table in [5], even if we forget the flaw in the definition of the d-diameter complexity. The entry for the crystal is wrong, and should read $\log n$ rather than $k \log n$ (see Theorem 2 in Section 5 of this paper).

Appendix 2: An Information-Theoretic Proof That There Are Infinitely Many Primes

It is of methodological interest to use widely differing techniques in elementary proofs of Euclid's theorem that there are infinitely many primes. For example, see Chapter II of Hardy and Wright [20], and also [21–23]. Recently Billingsley [24] has given an information-theoretic proof of Euclid's theorem. The purpose of this appendix is to point out that there is an information-theoretic proof of Euclid's theorem that utilizes ideas from algorithmic information theory instead of the classical measure-theoretic setting employed by Billingsley. We consider the algorithmic entropy $H(n)$, which applies to individual natural numbers n instead of to ensembles.

The proof is by *reductio ad absurdum.* Suppose on the contrary that there are only finitely many primes $p_1, \ldots, p_k$. Then one way to specify algorithmically an arbitrary natural number

$$n = \prod p_i^{e_i}$$

is by giving the k-tuple $\langle e_1, \ldots, e_k \rangle$ of exponents in any of its prime factorizations (we pretend not to know that the prime factorization is unique). Thus we have

$$H(n) \leq H(\langle e_1, \ldots, e_k \rangle) + O(1).$$

By the subadditivity of algorithmic entropy we have

$$H(n) \leq \sum H(e_i) + O(1).$$

Let us examine this inequality. Most n are algorithmically random and so the left-hand side is usually $\log n + O(\log \log n)$. As for the right-hand side, since

$$n \geq p_i^{e_i} \geq 2^{e_i},$$

each e_i is $\leq \log n$. Thus $H(e_i) \leq \log \log n + O(\log \log \log n)$. So for random n we have

$$\log n + O(\log \log n) \leq k[\log \log n + O(\log \log \log n)],$$

where k is the assumed finite number of primes. This last inequality is false for large n, as it assuredly is not the case that $\log n = O(\log \log n)$. Thus our initial assumption that there are only k primes is refuted, and there must in fact be infinitely many primes.

This proof is merely a formalization of the observation that if there were only finitely many primes, the prime factorization of a number would usually be a much more compact representation for it than its base-two numeral, which is absurd. This proof appears, formulated as a counting argument, in Section 2.6 of the 1938 edition of Hardy and Wright [20]; we believe that it is also quite natural to present it in an information-theoretic setting.

References

[1] L. E. Orgel, *The Origins of Life: Molecules and Natural Selection,* Wiley, New York, 1973, pp. 187–197.

[2] P. H. A. Sneath, *Planets and Life,* Funk and Wagnalls, New York, 1970, pp. 54–71.

[3] G. J. Chaitin, "Information-Theoretic Computational Complexity," *IEEE Trans. Info. Theor.* IT-20 (1974), pp. 10–15.

[4] G. J. Chaitin, "Randomness and Mathematical Proof," *Sci. Amer.* 232, No. 5 (May 1975), pp. 47–52.

[5] G. J. Chaitin, "To a Mathematical Definition of "Life"," *ACM SICACT News* 4 (Jan. 1970), pp. 12–18.

[6] J. von Neumann, "The General and Logical Theory of Automata," *John von Neumann—Collected Works, Volume V,* A. H. Taub (ed.), Macmillan, New York, 1963, pp. 288–328.

[7] J. von Neumann, *Theory of Self-Reproducing Automata,* Univ. Illinois Press, Urbana, 1966, pp. 74–87; edited and completed by A. W. Burks.

[8] R. J. Solomonoff, "A Formal Theory of Inductive Inference," *Info. & Contr.* 7 (1964), pp. 1–22, 224–254.

[9] G. J. Chaitin and J. T. Schwartz, "A Note on Monte Carlo Primality Tests and Algorithmic Information Theory," *Comm. Pure & Appl. Math.,* to appear.

[10] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication,* Univ. Illinois Press, Urbana, 1949.

[11] H. A. Simon, *The Sciences of the Artificial,* MIT Press, Cambridge, MA, 1969, pp. 90–97, 114–117.

[12] J. von Neumann, *The Computer and the Brain,* Silliman Lectures Series, Yale Univ. Press, New Haven, CT, 1958.

[13] C. Sagan, *The Dragons of Eden—Speculations on the Evolution of Human Intelligence,* Random House, New York, 1977, pp. 19–47.

[14] G. J. Chaitin, "A Theory of Program Size Formally Identical to Information Theory," *J. ACM* 22 (1975), pp. 329–340.

[15] G. J. Chaitin, "Algorithmic Information Theory," *IBM J. Res. Develop.* 21 (1977), pp. 350–359, 496.

[16] R. M. Solovay, "On Random R.E. Sets," *Non-Classical Logics, Model Theory, and Computability,* A. I. Arruda, N. C. A. da Costa, and R. Chuaqui (eds.), North-Holland, Amsterdam, 1977, pp. 283–307.

[17] P. Gács and J. Körner, "Common Information Is Far Less Than Mutual Information," *Prob. Contr. & Info. Theor.* 2, No. 2 (1973), pp. 149–162.

[18] A. D. Wyner, "The Common Information of Two Dependent Random Variables," *IEEE Trans. Info. Theor.* IT-21 (1975), pp. 163–179.

[19] H. S. Witsenhausen, "Values and Bounds for the Common Information of Two Discrete Random Variables," *SIAM J. Appl. Math.* 31 (1976), pp. 313–333.

[20] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers,* Clarendon Press, Oxford, 1962.

[21] G. H. Hardy, *A Mathematician's Apology,* Cambridge University Press, 1967.

[22] G. H. Hardy, *Ramanujan—Twelve Lectures on Subjects Suggested by His Life and Work,* Chelsea, New York, 1959.

[23] H. Rademacher and O. Toeplitz, *The Enjoyment of Mathematics,* Princeton University Press, 1957.

[24] P. Billingsley, "The Probability Theory of Additive Arithmetic Functions," *Ann. of Prob.* 2 (1974), pp. 749–791.

[25] A. W. Burks (ed.), *Essays on Cellular Automata,* Univ. Illinois Press, Urbana, 1970.

[26] M. Eigen, "The Origin of Biological Information," *The Physicist's Conception of Nature,* J. Mehra (ed.), D. Reidel Publishing Co., Dordrecht-Holland, 1973, pp. 594–632.

[27] R. Landauer, "Fundamental Limitations in the Computational Process," *Ber. Bunsenges. Physik. Chem.* 80 (1976), pp. 1048–1059.

[28] H. P. Yockey, "A Calculation of the Probability of Spontaneous Biogenesis by Information Theory," *J. Theor. Biol.* 67 (1977), pp. 377–398.