Notes on Self-Representing, and Other, Information Structures

by

Peter Schorer

(Hewlett-Packard Laboratories, Palo Alto, CA (ret.))
2538 Milvia St.
Berkeley, CA 94704-2611
Email: peteschorer@gmail.com
Phone: (510) 548-3827

Feb. 21, 2023

Key words: self-representation, self-similarity

Motivation for These Notes

These Notes have their foundations in three sources: first, and probably most important, Chaitin's papers an algorithmic information theory (see Bibliography); second, Mandelbrot's *Fractals: Form, Chance, and Dimension* (Mandelbrot 1977); and third, Hofstadter's *Gödel, Escher, Bach: An Eternal Golden Braid* (Hofstadter 1979).

But my interest in the subject goes back to the early 1960s, when, as a technical writer, I wondered why manuals were always written after the equipment they described had been built, especially as the equipment sometimes turned out to be difficult if not impossible to explain entirely on paper. I wondered why engineers didn't first decide on the maximum size and difficulty they wanted the manual to have, then build the machine to match?

In the early 1970's I became a computer programmer. Around 1975 1 happened upon Gregory Chaitin's article on algorithmic information theory in *Scientific American* (Chaitin 1975a). This article, and several papers by Chaitin which I soon read (see Bibliography), introduced me to the idea of measuring the complexity (degree of randomness) of a string of binary digits by the size of the smallest program needed to generate the string.

The parallel between program-to-generate-a-string and manual-to-describe-a-program seemed obvious. I began thinking about what I called "self-representing" programs, i.e., programs which contained all the instructions for their use. I wondered (not surprisingly, considering how difficult most programs were to use) if there might be programs so complicated that they couldn't be self-representing, relative to a given class of users defined by a minimum vocabulary and set of abilities.

In 1976 1 completed a Master's thesis on the denotational semantics of programming languages (Schorer 1976), a subject which introduced me to the idea of a computable function being the limit of a sequence of approximating functions. That same year, Dijkstra's *A Discipline of Programming* (Dijkstra 1976) appeared; this book developed further Dijkstra's idea of deriving a correct program through a sequence of successive approximations. I began wondering how one might define approximation informationally.

In Mandelbrot's book I confronted the term "self similarity" for the first time, a term he applied to certain mathematical curves which look the same in the large as in the small. The concept of "self-reference" as set forth in Hofstadter's *Gödel, Escher, Bach* (Hofstadter 1979) seemed closely related.

Throughout these years, my mind was also on the successes of Paul MacCready, Jr., and his associates in first achieving extended man-powered flight (Grosser 1981). The beautiful aircraft which MacCready designed also seemed related to my preoccupations with self-representing information structures, though I wasn't sure in what way. Later on, the underlying similarity became clearer with the question, "Are there vehicles, in particular ships and aircraft, which are so complicated they could not carry all the drawings needed to describe their construction?" Here are two extreme examples:

(1) Suppose it is found that the solution to a certain math problem requires that π be expanded so that the number of digits is equal to the number of atoms in the visible universe, namely, about 10^{82} digits. But then there would be no room in the universe for the text of the solution to the problem.

(2) Suppose it is found that a complete description of the entire universe will require say, a third of the atoms in the universe. But then that will only be a complete description of two-thirds

of the universe. A complete description of the entire universe will be impossible.

Introduction

It is often said that we live in an age of information, and indeed, for those of us who spend large amounts of time with computers, information, as represented in bits, is becoming the very coinage of our thought. (A bit is the amount of information we obtain from the reception of one of two equally probably symbols.)

In these Notes, I would like to set forth certain ideas, questions, and speculations which arose from considering the idea of a "self-representing information structure", i.e., an information structure (e.g., a network of roads, a government bureaucracy) in which a description of the entire structure can be contained within the structure itself, e.g., a map of the entire network can be put on a sign at each crossroads, an organization chart of the entire bureaucracy can fit in each office. However, there is much in these Notes which is not obviously related to this idea, but which I have included because my intuition is that it might be related.

Underlying everything is the basic fact of information theory and of its offspring, algorithmic information theory, that information is a measure of predictability, of regularity of occurrence, of pattern, hence of the opposite of these, complexity. Thus, e.g., not all 200-page books contain the same amount of information.

The reader should keep in mind that throughout these Notes, I mean by "information" anything that can be represented in bits. This includes books and all other printed or written material, reproductions of paintings, music, movies, video programs, and, of course, computer programs.

Representation

Classical Definition of Information

We begin with the question of representation because we will usually have in mind that the information with which we will be concerned, represents something. Does information *have to* represent something? The classical definition of information is merely concerned with the probability of occurrence of a symbol emitted by an information source. That is, if an information source emits symbols from a finite source alphabet $S = \{s_1, s_2, ..., s_q\}$, then the information $I(s_i)$, $1 \leq i \leq q$, obtained from the emission of the symbol s_i is given by:

$$(1) \quad I(s_i) = \log (1/(P(s_i))) \text{ bits},$$

where $P(s_i)$ is the probability of s_i being emitted by the source (Abramson 1963, 14).

"The fundamental problem of communication is that of reproducing at one point exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is, they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem." (Shannon 1948, "Introduction")

One of the principle aims of information theory is to discover rules for designing efficient codes. This aim arises from the fact that, if the symbols from the source are transmitted through a noisy information channel, it is possible to substitute for each symbol s_i a longer string of symbols a_i which will reduce the chance of the receiving apparatus failing to detect that s_i was in fact the symbol sent. Thus a code is a mapping C from symbols in S to strings over some finite alpha-

bet A. And *now* we may say that each code string a_i represents a string s_i. Representation is in the picture even though we still do not know what the original symbols s_1, s_2, ..., s_q represent, if anything.

It follows from the definition (1) that the more probable the occurrence of symbol s_i, the less the information obtained from its occurrence. (If every time your grandmother writes she says the same thing, you get very little information from her.)

As a result, it can be shown that the maximum length of the code symbol a_i representing s_i is shorter for high probability s_i than for low, over any information channel. In brief: high probability = low information = short code symbol.

Algorithmic Information Theory Definition of Information

Algorithmic information theory arose from the idea that the randomness of a finite string of symbols, e.g., a binary string, s, could be measured by the length of the shortest computer program needed to generate the string, all programs being written in some language, e.g., some Turing Machine formalism. The string s may or may not mean anything, but the program can be said to represent the string in the same way that the code symbol a_i can be said to represent the original source symbol s_i.

The binary string s generated by the program, "Print '1 0' 1000 times'", is less random than the the binary string typically generated by, say, flipping a fair coin 1000 times and counting a head as 1 and a tail as 0, because in the latter case the shortest representation of s is close to the length of s itself.

Most strings are random, as can be seen from the following argument: there are 2^m strings of length m. As r decreases below m, a smaller and smaller fraction of the 2^m strings can be matched one-to-one with the strings of length r (these are possible programs for generating strings of length 2^m).

A Digression

The following question, which may be nonsensical, occurred to me while I was considering the illusion of cause and effect in the sub-section,"The Two Planets Problem" on page 29.

What does it mean, physically, *to represent*, e.g., to say that this particular set of magnetic elements, in this particular state, on this particular disc or tape, *represents* that particular state of physical affairs over there? It is clear that part of the universe has been used — disturbed — to represent another part.

Are there "accidental" representations? Suppose that one night a computer breaks down and, in the process, writes random 1's and 0's into its disc memory. Later, when the computer is repaired, it is discovered that the contents of the memory are a description of certain physical events that purportedly took place on the night of the breakdown on a star several light years away. Several years later, astronomical observations seem to confirm the description. Should the contents of the computer memory be considered a description?

Information Structures

The Three Types of Information Structure

So far as I know, neither classical information theory nor algorithmic information theory has

investigated deeply the nature of "too little", "enough", or "too much" information. True, Shannon, in his original paper, discusses approximations to English text. Also, an inadequate code means ambiguity, at the receiver, as to which symbols were transmitted. But what about cases where we want "too little" information, e.g., in the case of abstraction, where, e.g., we may want to know only the structure of a computer program, and not the actual instructions in the various procedures.

On the other hand, is there such a thing as "too much" information? One of the ideas we will consider is the possibility that certain paradoxes arise from too much information.

In between, of course, lies the case of "enough" information, one example of which will be a self-representing structure.

Finally, what about the evolution of information systems, e.g., computer networks, government and industrial bureaucracies, in which initially it is possible for all the information "about" the system to get to where it is needed by the time it is needed, but, as the system evolves, the lines of communication become overburdened or clogged, so that it is no longer possible for the left hand to know what the right is doing; the system can no longer represent itself, 'know' itself. So one of our goals will be to develop mathematical models that will describe, *informationally*, the evolution of bureaucracies, bodies of scientific knowledge, telephone, road, and railway systems, etc.

We shall consider each of the cases — "too little", "enough", "too much" information — separately, but first we give an informal definition of "information structure". The reader should understand that information structures, as I will define them, are only examples — although, I believe, important examples — of the three cases of information adequacy.

Definition of an "Information Structure"

The following definition is too formal, considering the discussion in the remainder of these Notes. But it will serve to focus some of the ideas that follow.

By an "information structure" I mean a finite, connected, bidirectional graph $G(V, E, NC, t(i))$ where:

V is the set of vertices. A vertex is hereafter called a "node". A node may represent, e.g., an office in a bureaucracy, a town in a network of roads. Each node $v(i)$ in V is capable of containing some maximum finite amount of information $\text{MaxInf}(v(i))$.

E is the set of edges. An edge is hereafter called, usually, a "connection" or a "connection line" or a "line". A connection may represent, e.g., a means for sending mail in a bureaucracy, a road in a network of roads. There is associated with each edge $e(i)$ a channel capacity $C(e(i))$. Informally: there is a maximum rate at which we can send information through each connection line such that the frequency of errors is as small as we desire.NC is the (possibly null) set of "node coordinates" which specify, under some coordinate system, the location of some or all nodes in V. NC may, e.g., represent a checkerboard of cells, each of which can communicate, via a "negligible length" connection, with its neighboring cell, as in Conway's Game of Life (see "A Computational Model: Conway's Game of "Life"" on page 28).

$t(i)$ is an integer ≥ 0, representing a moment in time. The information structure G contains a time parameter because we will want to model the *evolution* of information structures.

On the Information Content of the Nodes

The basic idea is that each node contains information describing the structure as a whole at

any time $t(i)$. The node may contain too little, enough, or too much such information. Each case will be considered as an example in the following sections.

"Too Little" Information

Intuitive Meaning of "Too Little" Information

If we are given only the first m bits of an n-bit string s, where $1 \leq m < n$, then we are given too little information to uniquely describe s.

There are now two possibilities: (a) there is a way to find out the remaining $(n - m)$ bits, or, (b) there is not.

Under (a), suppose we are allowed to guess the remaining $(n - m)$ bits, receiving a truthful "yes" or "no" reply as to the correctness of each guess. Then, because the reply to each guess will determine the value of the bit, we will need exactly $n - m$ guesses to determine what s is.

Examples of "Too Little" Information

Approximation

At present I will consider all examples of too little information under the heading of "approximation." I think of an approximation of something as too little information about the thing. Successive approximation I think of as the gradual acquiring of more information about the thing. There appear to be two types of successive approximation: (a) one in which the gradual acquiring of more information can be "run" by a program, i.e., by a finite mechanism, e.g., a computer program that generates successively the digits of π, and, (b) one for which there is no such program or mechanism: the means of acquiring each successive unit of information about the thing must be "figured out from scratch" each time.

This section may be thought of as a proposal for an information-theory-based Theory of Approximation.

Approximation in Infinite Domains

In this type of approximation, there is no limit to the "amount" of approximating material, i.e., the material, e.g., bits in a computer memory, with which we can represent each approximation: we do not "run out of it"; furthermore, the material we use to approximate the thing does not change the thing, as in the case of finite domains discussed below.

Approximations to Mathematical Proofs

"In mathematics, there are more truths than proofs." That is, an axiomatic system has only a countable infinity of strings; but there is an uncountable infinity of sets of integers, therefore it is impossible to find a one-to-one match between strings (some of which are proofs) and sets of integers (these being the "truths" we want to have proofs of).

But suppose we think of the countable infinity of strings as "bits" which we can spread over the uncountable infinity of truths. In other words, suppose we are willing to settle for mere approximations to truths in return for being able to "cover" more of them. I believe it was the mathematician M. O. Rabin who in the late seventies described an algorithm which computed the probability that a number n was prime. Probabilities close to 1 were achieved for very large n in a matter of 12 minutes or so (if I recall correctly), whereas the best known algorithms for determin-

ing the primality of the same numbers "with certainty" would have taken many years.

Approximation in Finite Domains

We now consider approximation in finite domains, in particular, approximation in domains where the "material" representing the approximation must be taken from the thing to be approximated, as in the following examples.

Approximate Descriptions of Information Structures

The first example is that of an Information Structure in which the information in some node, possibly in all, is insufficient to describe the Structure as a whole. This may be because the amount of information required for an adequate description is larger than MaxInf($v(i)$) for one or more i, or because the information that is present in one or more nodes, is simply inadequate, for whatever reason.

Structured Programming as Approximation

The method of computer programming known as "structured programming" can be thought of as one in which the program is obtained through successive approximation: at each level of refinement, we add more information about the program until, at the lowest level, we have the entire program specified.

We now turn to some other examples. I am not sure whether they should be considered examples of approximation in finite or infinite domains.

Other Types of Approximation

Approximation as a 'Wrapping'

In Mandelbrot's *Fractals: Form, Chance, and Dimension* (Mandelbrot 1977) we read:

"In other words, at certain scales and for certain methods of investigation, many phenomena may be represented by regular continuous functions, somewhat in the same way that a sheet of tinfoil may be wrapped around a sponge without following accurately the latter's complicated contour." (Mandelbrot 1977, 7)

A school of sculpture (e.g., Cristo) in the seventies wrapped objects, including buildings and bridges, in cloth. Suppose we wrap an object, e.g., Rodin's sculpture, "The Thinker", in a rubber-sheet, making the enclosure airtight. Call this a "first approximation" to the sculpture. Then suppose we slowly remove the air inside the rubber, so that the rubber gradually conforms more closely to the outline of the actual sculpture. (A "method of exhaustion".) Some industrial products are wrapped in transparent material by a similar process. Consider a computer program as a "skin" which, in the process of computation, slowly collapses around the object which is the value computed.

Standing on a Rubber Sheet

Imagine a large empty swimming pool covered with a rubber sheet. The sheet supports the weight of one person without touching the bottom of the pool. Under the person's feet, the sheet will conform closely to the actual position and shape of each foot, but farther away the sheet will not. Putting it another way, if the sheet were suddenly "frozen" and the person then removed, we could tell more about the shape of his feet from the rubber near or under his feet, than we could from the rubber farther away.

The Death of Hal

In Stanley Kubrick's film *2001*, the astronaut Dave (Keir Dullea) is forced to partially disable the rebellious computer Hal in order to gain control of it. He leaves just enough of Hal's circuitry intact so that Hal will be able to maintain the basic functions of the ship.

The scene suggests the idea of approximation "in reverse", and also raises the question of how Hal's circuits were organized so that it was possible to easily remove his faculties of higher intelligence while maintaining the routine control faculties.

A related question is the following.

The Aging Mathematician

It is well-known that mathematicians lose their ability with age. Are there better and worse ways for the aging mathematician to deal with this problem, assuming he wishes to continue to do mathematics? For example, if this loss is due to the cumulative loss of brain cells, are there better and worse brain cells to lose? If some transfer of capabilities from one group of brain cells to another is possible, e.g., through certain mental exercises, then, which capabilities should be transferred first?

The Turing IQ Test

Suppose the aging mathematician (or any student) wishes to make the most of the mental ability he has. Suppose he develops, on paper or on a small computing system, an information Environment for the particular branches of mathematics he works in (see Curtis, William, *How to Improve Your Math Grades*, accessible on the web site www.occampress.com). Such an Environment is not an artificial intelligence system in the sense of a system that is able to solve any problem automatically, but rather a system of aids to solving problems, tailored to the mathematician's own needs and preferences, and characterized above all by making possible *rapid look-up* of desired information.

Suppose now, one mathematician, call him "Inferior", is placed in a closed room, with his Environment. Another mathematician, call him "Superior", is placed in another closed room with only standard textbooks and papers on his subject. Both are given ample supplies of paper and pencils.

An Interrogator in another room asks questions of the two via computer terminal and attempts to guess which is the Inferior, which the Superior mathematician. If he were unable to do so (under some reasonable statistical criterion), in what sense would the Inferior mathematician be inferior?

Approximation in a Cellular Physical Universe

This topic is covered in the sub-section "How Much Can We Know About The Universe?" on page 31.

What Good Is Structure?

Suppose I ask for a mathematical answer to the question, "What good is structure?" Assume we have n oysters, only one of which contains a pearl. Without further information, we can always find which oyster contains the pearl by opening each one in succession. At most we must open n oysters.

We can represent the task of finding the oyster containing the pearl as that of searching a binary tree of depth $m = [\log n]$, $n \geq 2$, where "$[x]$" denotes the smallest integer larger than x. Thus

the root of the tree is at level 0, the next two nodes at level 1, etc.

For convenience from here on, but without loss of generality as far as the purpose of our discussion is concerned, we limit n to a power of 2.

Definition of a "Node-Visit" First, we define the searching, or traversing, of a binary tree in "preorder". The (recursive) definition of preorder is:

a) Visit the root of the first tree;
b) Traverse the subtrees of the first tree (in preorder);
c) Traverse the remaining trees (in preorder).

(Knuth 1969, 334)

We define a "node-visit" as the entering, and leaving, of a node in a binary tree. Thus, it is easy to see that, in visiting all nodes of a finite binary tree in preorder, each node except a leaf node receives 3 node-visits, while a leaf node receives only 1, because, for a non-leaf node, we:

(1) enter the node *q* on the branch that enters *q* from above and leave on the left branch below *q*;

(2) enter the node *q* from below *q* on the left branch and leave on the right branch below *q*;

(3) enter the node *q* from below *q* on the right branch and leave on the branch that enters *q* from above,

whereas, for a leaf-node, we simply:

(1) enter the node *q* on the branch that enters *q* from above and leave *q* via the same branch.

Therefore, the number of node-visits that must be made in the search of all nodes of a binary tree of depth $m \geq 1$ is:

[1] $3(1 + 2 + 4 + \ldots + 2^{(m-1)})$ (for the non-leaf nodes) + $1(2^m)$ (for the leaf nodes) =

$3(2^m - 1) + 2^m$

Assume now that each node is identified by an address or "map" for reaching it from the root, e.g., a binary string, read from left to right, in which:

0 = "Go down the left branch."
1 = "Go down the right branch."

Then clearly each leaf-node has an *m*-bit address. Thus if each oyster occupies a separate leaf node, we can specify its location with an *m*-bit address.

From [1] it is clear that the specification of the first *k* bits of an *m*-bit address, $1 \leq k \leq m$, has the effect of reducing the maximum number of node-visits that must be made to find the oyster containing the pearl. If we correlate "structure" with "number of address bits given of the oyster

containing the pearl", then my reply to the question, "What good is structure?", is, "It saves time (as measured here by number of node-visits) in finding things."

On the Size of the Search Program

Observe that the one program can be used to visit all nodes of any size binary tree. The inputs to the program are (a) a description of the pearl, and (b) the binary tree to be searched.

A Naive View of Structure

With the above example in mind, I think, naively: We have structure when things "overlap". All n-bit addresses whose first k bits are the same overlap in the first k bits.

Are There Better And Worse Structures? Approximation in Learning

Suppose you need to learn the fundamentals of a number of different technical subjects. The "proper" way to do this, i.e., the way that experts in each field would suggest you do it, would be to take at least one university course in each subject, and do all the assigned exercises. But this is an extremely inefficient approach. You might find it far better to attempt to get a global view of the subject, however imprecise, then proceed "top-down" on a conceptual basis, i.e., attempting to understand the concepts at each level, and working the exercises only as this seems to lead to that goal.

This approach is an answer to the question, "Given too little time to learn properly what you want to know, what is a good way to proceed, particularly if your main interest is in learning 'where' the concepts are, for future reference — learning the 'addresses' of the concepts, rather than learning to be adept at applying them?"

This might be called an "approximative" approach to understanding.

If one begins building Environments for each subject, one soon begins wondering about different organizations of a subject — different views of the subject obtained through different settings of the lens of abstraction. For example, suppose we view all, or a part, of mathematics through the concept of "function" in its full generality; or through the concept of "periodicity", or of "limits". Some parts will be rendered invisible under each of these settings of the lens. We ask, What constitutes better and worse settings of the lens for a given purpose? (Consider statements like, "If you view x as simply a set of y's, then everything follows!")

Series in Mathematics as Approximations

Consider a convergent series in mathematics, e.g., one for π or for e, the base of the natural logarithms. A computer program can be written to compute successive digits of the series. The program takes an input n and computes the first n digits of the series. The program consists of a finite number x of bits. Program plus n can be represented in $x + \log n$ bits. From the curve of $y = \log n$, it is clear that, as n grows, the rate of growth of $x + \log n$ becomes less. We think: The cost in bits of obtaining the next digit in the series grows progressively less as n increases. (The cost of obtaining all the digits is not infinite, but merely x, since we can just let the program run forever. Is this fact — that it costs more (bits) to get some information than it does to get all — significant?)

Informationally, how does a divergent series differ from a convergent series?

Does classical and/or algorithmic information theory provide a more general view of convergence and divergence in the same sense that, say, topology provides a more general view of geometry?

Problem Solutions as "Shapes"

In general, and not only in reference to technical problems, we can think of a solution to a problem as a "shape" that fits the problem to a better or worse degree. The goodness of a given shape is proportional to how few bits we need to describe it.

"Fuzzy" Images

There is a kind of photographic image composed of small squares, each square of a uniform shade of gray, different squares typically being of different shades. Such images appear clearer the farther away (up to a point) one views them. Informationally, how would we describe this phenomenon?

"Enough" Information: Self-Representing Information Structures

Obviously, the subject of "enough" information can, and does, fill volumes. In the following, I will only give examples that to me seem related to the concept of self-representation. Self-representation is similar to, if not the same as, self-similarity as described in (Mandelbrot 1977) and to self-referencing as described in (Hofstadter 1979). These two books may be said to have popularized the concepts in the intellectual community.

Informal Definition of "Self-Representing Information Structure"

A "self-representing information structure" (SRIS) is an Information Structure as defined in the section, "Definition of an "Information Structure"" on page 5, such that a description of the entire structure can be contained in each node. Thus, the Structure may be self-representing at time $t(i)$, but not self-representing at some earlier or later time $t(j)$.

A certain class of infinite sequences of self-representing Information Structures is important, namely, the class consisting of those sequences such that, for each sequence, there exists a single program p that generates all the Structures in the sequence, i.e., the Structure $G\ (V, E, NC, t(i)) = p(i)$. Each node of each Structure, in this case, simply contains a representation of p and a representation of the argument i. Since the length of $i = \log(i)$, it is clear that the maximum number of bits in each node of successive Structures increases as $\log(i)$, or, in short, that the incremental increase in the size of nodes approaches zero as i approaches infinity.

The "Weight" Metaphor

I think of SRIS's as being "light", and of non-SRIS's as being "heavy" because I think of SRIS's as being able to "support themselves" informationally, whereas non-SRIS's cannot. If an SRIS is thought of as a vehicle, e.g., a ship or a plane, then an SRIS can carry all the plans for its construction, whereas a non-SRIS cannot.

Original Inspiration for the Weight Metaphor

In 1977 Bryan Allen, in a ship called *The Gossamer Condor,* which had been designed by Paul MacCready Jr. and his team, made the first man-powered flight over a 1-mile figure-of eight course. In 1979, Allen pedalled another MacCready ship, the *Gossamer Albatross,* across the English Channel. These aircraft were extremely light and simple of construction — indeed, they may be considered to be simply large-scale versions of some of the indoor model planes MacCready built in his youth.

When I first read about these beautiful aircraft, and later saw TV documentaries about them, I began wondering if they might not provide a useful metaphor for a certain kind of information structure. For example, would it be possible to fly these ships with all the drawings describing their construction aboard? The illustrations in (Grosser 1981) suggest the answer is yes. But of course we must qualify this with, ". . . assuming a certain level of ability in the users of the drawings". Perhaps we should put the question this way: "Assume there is an airfield somewhere in the Englishspeaking world, with a dozen or so competent aeronautical engineers in residence. Assume the airfield is reasonably close to sources of Mylar, carbon-fiber-reinforced-plastic, and other necessary materials. Would it be possible to load the Gossamer Albatross with a set of drawings such that, when it landed at that airfield, the resident engineers would be able to build another Albatross which would fly?"

One reply to the question that evades the whole issue of the weight of the drawings is: "Yes, because if all the necessary materials were present at the airfield as stipulated, the engineers could simply build another Albatross by copying the first one." I do not wish to dismiss this reply, since it may contain an insight of importance in what follows, but for the time being let us say that the second ship must be built from the plans aboard the first, and not by copying it directly.

"The boat [belonging to the RUR Company, where "RUR" stands for "Rossum's Universal Robots"] moves slowly out to sea and out of sight. A month later, somewhere in the Indian Ocean, two boats appear where one was before. The original boat carried a miniature factory with all the necessary equipment, plus a computer program which enables it to construct a complete replica of itself. The replica contains everything that was in the original boat, including the factory and a copy of the computer program. — Dyson, Freeman, *Disturbing the Universe*, Basic Books, 1979, p. 197.

The man-powered aircraft, and the question of their being able to fly with all their drawings aboard, leads us to think of self-representing structures as being "light", whereas aircraft whose plans are too heavy to allow them to fly with all their drawings aboard, i.e., non self-representing structures, we may think of as "heavy".

Examples of (Possibly) Self-Representing Information Structures

The following examples are in many cases merely suggestive; I realize it is not obvious how they fit our informal definition of an SRIS.

The Model House in Albee's Play, "Tiny Alice" "

The central image of the play is the mysterious model of the great mansion, in which the action takes place, that occupies the centre of the stage. Inside this model every room corresponds to one in the real house, and tiny figures can be observed repeating the movements of the people who occupy it. Everything that happens in the macrocosm is exactly repeated in the microcosm of the model. And no doubt inside the model there is another smaller model, which duplicates everything that happens on an even tinier scale, and so on *ad infinitum,* upwards and downwards on the scale of being." (Esslin 1980, 314.)

Esslin's description raises a question which we must dispose of now, namely, is an SRIS really possible? That is, if each node *n* contains a representation of the entire structure, then it must be that, within *n* there are (representations of) nodes that in turn contain (representations of) nodes that in turn contain . . . My answer at present is that this in no way implies that we need an infinite amount of information in each node, as any reader of (Mandelbrot 1977) will agree, or indeed, as anyone familiar with recursive functions will agree. To take one of the simplest examples, one can

define an infinite binary tree via the following (finite amount of) information: (1) the root of the tree is a node; (2) each node has two branches, each of which end in a node.

DNA

Probably the most familiar example of an SRIS, or rather, of something that is "SRIS-like", is DNA (deoxyribonucleic acid). See, e.g., the description of DNA in (Hofstadter 1979).

Leibniz's Monads

There seems to be some similarity between self-representing structures and Leibniz's monads, specifically, in the monad's property of "mirroring" the rest of the universe.

" . . . they [the monads] have been so arranged by God that the spontaneous activity of each in fact mirrors the whole of the universe, more or less clearly or confusedly, from its particular point of view." (Hampshire 1956, 164)

Tools That Can Carry Their Own Instructions

In our culture, the familiar carpenter's hammer is "light" because almost no instructions are required for any adult to use it. We could describe its use — e.g., by a sequence of pictures, drawings — on the tool itself, say on the handle.

A telephone would be "light" if not only the instructions for using it but also the entire phone directory could be contained on each phone.

Anyone who is working on the design and implementation of computing systems, in particular so-called "interactive systems", which attempt to be nearly or totally self-explanatory to anyone in the intended set of users, knows how he feels constrained to keep the computer programs in the system simple to use and change, because the instructions for these programs must be part of the system itself.

Low- vs. High-Information Prose

Consider technical prose that we feel is too long-winded, that contains relatively little information (i.e., the presentation is longer than necessary, relative to some audience). Using our weight metaphor, we are inclined to say that such prose is self-representing because it could easily support its own description, where, here, "its own description" refers to the idea that the prose is attempting to express. We could write down this idea "in a few words", make reduced images of these words and place these images at many places throughout the prose passage. There would be lots of room for them.

Self-Representation in Mathematics

Group Theory

A set of elements G, with an operation defined between every pair of elements x, y in G, is a *group* if all of the following properties hold:

• For all x, y in G, xy and yx are in G (closure property);

• There exists a unique element e in G (called the "identity element") such that, for each x in G, $xe = ex = x$;

• For all x, y, z in G, $x(yz) = (xy)z$ (associative property);

• For each x in G, there exists a unique element x^{-1}(called the "inverse of x") such that $xx^{-1} = x^{-1}x = e$.

An automorphism is a one-one mapping of a set onto itself. The set of all automorphisms of a group G, denoted $Aut(G)$, is itself a group, in which composition of automorphism is the group operation. Therefore, for each group G there exists an infinite hierarchy of groups.

We may think of this hierarchy of groups as self-representing in the sense that whatever we discover about groups by studying the groups at some arbitrary level i, we automatically discover about the groups at any other level.

Regular Functions in Analysis

In a well-known text on the theory of functions (Knopp 1945, 86), we read:

"Suppose we restrict our attention to the class of entire rational functions (polynomials) of the third degree (i.e., to curves of the third degree):

$y = a_0 + a_1x + a_2x^2 + a_3x^3$ (a_n, x, y, real)

Such a function is already completely determined by very few conditions (requirements). If we know, for example, that the curve passes through four specific distinct points (i.e., if we know the values of the function for four distinct values of x), the function is fully defined, no matter how close to one another the four points may lie. The behavior of the curve, with all its regular and singular properties, in the whole xy-plane can thus be inferred from the behavior of the function in an arbitrarily small interval.

". . . a function belonging to [the class of regular functions, i.e., the class of differentiable functions] possesses such a strong inner bond, that from its behavior in a region, however small, of the z-plane one can deduce its behavior in the entire remaining part of the plane."

Knopp then gives the following theorem:

"The Identity Theorem for Analytic Functions. If two functions are regular in a region R, and if they coincide in a neighborhood, however small, of a point z_o of R, or only along a path segment, however small, terminating in z_o, or also only for an infinite number of distinct points with the limit z_o, then the two functions are equal everywhere in R." (Knopp 1945, 87).

The self-representing idea here for me is the fact that a small amount of information about the functions reveals "everything" about the functions. The functions are represented in small parts of themselves.

Holograms

"Holograms and the images they produce have other fascinating properties besides dramatic 3-dimensional realism. For example, any fragment of the hologram can regenerate the entire image even if the fragment is extremely small. Since no image-forming device, such as a lens, is used, *each point on the subject sends its reflected light waves to each portion of the plate.* The reconstruction process regenerates these recorded waves, each portion of the hologram contributing to the entire image.

"This property, as well as others, can best be understood by regarding the hologram as a win-

dow, although it is indeed a strange window, that generates its own scene. A fragment of the hologram is like a small window, or perhaps a keyhole, if the hologram is small. *The observer can see just as much through the small window,* but of course he must come quite close to the window in order to see as much." (Italics mine.) (*Encyclopedia Americana* 1979, 303ff)

Esthetics

In a paper titled "Mathematics of Aesthetics" (Birkhoff 1956), G.D. Birkhoff sets forth an argument that the aesthetic measure *M* of an "aesthetic object" is given by

$$M = \frac{O}{C}$$

where *O* is the "order" (defined below) which is present in the object, and C is the "complexity" (defined below) in the object.

Birkhoff discusses the contemplation of a visual or auditory aesthetic object, e.g., a painting or a piece of music. His discussion seems related to the algorithmic information theoretic definition of pattern.

"From the physiological-psychological point of view, the act of perception of an aesthetic object begins with the stimulation of the auditory or visual organs of sense, and continues until this stimulation and the resultant cerebral excitation terminate. In order that the act of perception be successfully performed, there is also required the appropriate field of attention in consciousness. The attentive attitude has of course its physiological correlative, which in particular ensures that the motor adjustments requisite to the act of perception are effected when required." (Birkhoff 1956, 2186)

". . . Now although these automatic adjustments are made without the intervention of motor ideas, nevertheless there is a well-known feeling of effort or varying tension while the successive adjustments are called for and performed." (Birkhoff 1956, 2187)

He then sets forth his definition of psychological "complexity" *C*:

"Suppose that *A*, *B, C*. . . . are the various automatic adjustments required, with respective indices of tension *a*, *b*, *c*,. . . , and that these adjustments *A*, *B*, *C*, . . . , take place *r*, *s*, *t*, . . . times respectively. Now it is the feeling of effort or tension which is the psychological counterpart of what has been referred to as the complexity *C* of the aesthetic object. In this manner we are led to regard the sum of the various indices as the measure of complexity, and thus to write

$$C = ra + sb + tc + \ldots$$

(Birkhoff 1956, 2187)

He then discusses the two types of "associations" which occur during the contemplation of an aesthetic object: the first type he calls "formal", of which an instance is symmetry; the second type he calls "connotative", of which an instance is the associations "which are stirred by the meaning of a beautiful poem." (Birkhoff 1956, p. 2190)

He later on sets forth a definition of the psychological meaning of "order" *O*:

"Let us suppose that associations of various types *L*, *M*, *N*,. . . . take place with respective indices of tone of feeling *l*, *m*, *n*,. . . . In this case the indices may be positive, indifferent, or, negative, according as the corresponding tones of feeling are positive, indifferent, or negative. If the associations *L*, *M*, *N*, . . . occur *u*, *v*, *w*, . . . times respectively, then we may regard the total tone of feel-

ing as a summational effect represented by the sum $ul + vm + wn + \ldots$

"This effect is the psychological counterpart of what we have called the order O of the aesthetic object, inasmuch as $L, M, N, \ldots$ correspond to what have been termed the elements of order in the aesthetic object. Thus we are led to write

$O = ul + vm + wn + \ldots$"

(Birkhoff 1956, 2191)

Birkhoff then argues that the aesthetic measure M of the aesthetic object is given by:

$$M = \frac{O}{C}$$

Style

Consider a person's handwriting, or a cartoonist's drawing style. We find ourselves saying things like, "He makes his *r*'s this way", "He draws feet like that," etc. Should we think of this as though he has a program for making "*r*"'s, a program for drawing feet, which take parameters of size (how big to make the "*r*" or the foot) and location (where to make it on the page)? A program is a finite amount of information. But what is an "overall" style? What makes it easy for experienced classical music listeners to recognize within a few seconds that, say, Vivaldi is the composer of a piece they have never heard before? It is as though some composers have placed a stamp every second or two in their compositions: "Made by Vivaldi"... "Made by Vivaldi". . . Like a self-representing structure.

Is the domain of styles in a given art continuous or discrete? Should we think of styles as "flowing" into each other, or as each style being separate from the next? In the first case, then, changing a style is "simply" a matter of changing the parameters of a program, whereas in the second case, we have to change the program itself.

Book Publication

I think of using photolithography to publish a handwritten manuscript that contains mathematical drawings: rectangles, circles, other curves. Here too there seems to be a weight question: if the drawings were done by a professional illustrator, they would seem out of place; the handwritten "form" wants the crudity of uneven "straight" lines, lopsided circles, hastily drawn arrows, scribbled notes. I want to say that the amount of information present in the handwritten form is too much for the amount of information in the "standard" drawings.

Suppose I publish the manuscript in a very expensive format (paper, binding). Will that too seem out of place? (Consider editions of Leonardo's drawings.)

Jazz

In jazz, a "simple" improvisation, i.e., one requiring little technique or range, can be a "better" improvisation than a difficult one, i.e., one requiring considerable technique or range.

Thus a performer with little technical skill may outdo one with greater skill. But his performance immediately becomes worse if he attempts improvisations that are beyond his ability. Is this a question of "fit", i.e., weight?

Why does great music, e.g., a Bach violin partita, sound bad when it is poorly played? Why doesn't the greatness of the music "overcome" the ineptitude of the performer? (We must keep in mind, here, the difference between quality of performance and quality of reproduction: consider

good/bad readings of poetry, vs. good/bad recordings of these readings.)

The Wilderness Explorer

There is a Wilderness, which is actually an infinite binary tree. An Explorer wants to explore this Wilderness. He has a small Truck (a "Touring" machine) which can carry only a limited amount of information, i.e., it has a capacity of only *c* bits. The rule is that he must always have a complete map of his trip in his Truck before he starts (compare "What Good Is Structure?" on page 8). Maps consist of a finite string of binary digits such that 0 means "Turn right at the next node", 1 means "Turn left at the next node." A map only describes how to *reach* a destination, since a return map can always be obtained by the rule, "If you are on a 0 branch, turn left at the next node, if you are on a 1 branch, turn right at the next node."

At first, the Explorer wants to be able to go anywhere. He conjures up interesting trips by flipping a coin to decide which way to turn at each node. Unfortunately, he soon finds that interesting trips are only about *c* nodes long. No matter how hard he tries, he can't seem to find a way of describing interesting trips by shorter descriptions. He does find, however, that if he is willing to settle for a boring trip, he can go very far indeed: for example, a trip like, "Repeat a thousand times: 'turn left, turn right'". He also realizes that there is a class of boring trips from which no Explorer ever returns, namely, trips like, "Repeat forever: 'turn left', 'turn right'."

So he concludes, "Short, interesting trips are possible. Long boring trips are possible. But long interesting trips are not possible."

The above is a rather obvious parable for anyone acquainted with the rudiments of algorithmic information theory (see the Chaitin references in the Bibliography).

Environments

The way of presenting a mathematical subject described in William Curtis' *How to Improve Your Math Grades* (www.occampress.com) seems related to the idea of self-representing structures, but I am not sure exactly how.

"Too Much" Information: Non Self-Representing Information Structures

Informal Definition of a "Non Self-Representing Information Structure"

Informally, a "heavy" or "non self-representing" information structure is one that cannot be represented in one of its nodes. However, I must immediately point out that this definition does not obviously take into account *conflicting* information, which, as we shall see, may also be considered a case of too much information.

One way a heavy structure may arise is that, as *i* in *t(i)* (the time parameter in the sequence of structures G (V, E, NC, $t(i)$)) increases, a given light structure *evolves* into a heavy structure.

Consider the evolution of structures such as bureaucracies, bodies of knowledge, telephone and road systems. Initially, the amount of communication relative to the capacity of the connecting lines is small. You can accomplish what you want any time at the maximum transmission speed of the lines. You seldom have to wait for someone else to finish using the lines. Thus, in the case of bodies of knowledge, it is easy to find out what has already been done: it would be foolish *not* to try to find out, since the cost of the search is so much less than the cost of duplicating the experiment.

As the system evolves, it grows in size and number of users. You must plan for the use of the

system; maps are required. The cost of finding out about any part increases. The advantage the system originally provided slowly decreases — for we must keep in mind that *updating* these descriptions of the system is an essential function that the system must perform for itself: the left hand must be able to tell what the right is doing, and has done. If you can bicycle across Manhattan faster than you can drive across, you might as well not own a car; if the line is almost always busy, you might as well not have a phone. In the movie *The Ladykillers,* Prof. Markus explains to his landlady, Mrs. Wilberforce, that it would actually cost the insurance company *more* if the professor and his gang were to return the money they have stolen than if they kept it, because in the former case a great deal of bureaucratic labor would be involved, whereas in the latter case, the company can simply raise the premium for all its customers a farthing or two.

The basic idea here is that of background (the "wires" that connect the nodes of the system) becoming foreground (becoming the very subject, the problem, with which the system must spend more and more of its resources on). (See e.g, (Hofstadter 1979).)

Before giving some examples of heavy information structures, I would like briefly to describe two types of information domain which seem relevant to this discussion.

"Balloon" Domains and "Modular" Domains

A "balloon" domain is one in which, loosely speaking, if one part changes, another part must change accordingly (as when we squeeze a balloon), i.e., in which there *must be* trade-offs. One collection of techniques for dealing with a certain type of balloon domain is known as "linear programming".

The "opposite" of a balloon domain is a "modular" domain, where a part can be changed without affecting other parts.

Examples of balloon domains are:

- ecological systems;
- scientific civilizations (scientific discoveries change the very environment in which science is done);
- computing systems plus their users (users change the systems as they use them);
- economic systems in which predictions about the system affect its future.

Examples of "Too Much" Information

The Problem of Recording "All" of History

Suppose a group of intellectuals got together and decided to make the task of future historians easier by recording every event that occurred in these intellectuals' own time. Video cameras would record the sounds and activities surrounding every building, street, vehicle, inside and outside at all times. Every reel of video tape would then be saved for the future.

We immediately sense a flaw in this idea. For one thing, how would the enormous increase in demand for video tape, cameras, and camera operators affect the history of the time? Would it always be possible to film the cameras that are doing the filming, since this filming is now also a part of the history of the time? And the filming of these secondary cameras, and of the tertiary cameras, etc.?

The problem, we may say, is that there probably isn't enough "room" informationally, to represent this new "history obsessed" system within itself.

Structuring Mathematical Proofs and Computer Programs

Suppose, in structuring mathematical proofs or computer programs, we make a rule: "Each level of each proof, or each procedure, must be complete on one page!", attempting, in this way, to make the "visual whole" which a page is, also to be a logical whole. But if a proof or a program is complex, I may have so many levels that the benefit of structuring is lost or "swamped" by the need to turn pages: I no longer remember where I am at any given level, or how I got there.

Question: How shall we think about the informational cost of structuring?

Can Programs Always Contain All Their Error Messages?

Portions of many computer programs are devoted to checking for errors: errors in the data which is input to the program, e.g., numerical data not being in the proper range, and for errors which occur during the course of computation, e.g., attempted division by zero. We ask: Is it possible for every computer program to contain "all" its error messages? We must remember that errors may arise in the outputting of the error message portions of the program itself. Does the number of potential errors in every program grow more rapidly than the size of the program?

Of course, some error messages can never be issued by the program itself, e.g., the message, "Operator error: hardware not turned on."

Aircraft Controllers

Consider the problem of finding the most efficient number of air traffic controllers for a given airport. If there are only a few controllers, then each controller is responsible for a large section of the air space, and there is a greater chance he will overlook a potential air collision; on the other hand, the frequency with which he will have to pass control of a given aircraft to another controller will be will be relatively low. If there are many controllers, each assigned a small section of the air space, there is less chance of a controller overlooking a potential accident but the frequency at which he passes control from one section to another becomes much greater — conceivably, the time lost in doing this may increase the chance of his not detecting a potential collision.

Controlling Complexity

We ask if there is an informational analogue to Parkinson's well-known Law, "Work expands so as to fill the time available for its completion." (Parkinson 1957, 1)

It seems that the more persons involved in setting up the rules for doing something, the more complicated the thing will be, not because of the incompetence of the persons involved, but because (1) there are more bits available (namely, in the minds of the persons involved), and (2) because it takes more and more information to encode the procedures simply to tell the right hand what the left is doing, i.e., to keep the appropriate persons involved in the work, informed as to the state of the work. (For more on this latter point, see, e.g., (Brooks 1975).) As far as (1) is concerned, our analogue to Parkinson's Law thus might run:

"Complexity expands so as to consume the amount of information storage available."

Thus, one way to simplify government regulations, e.g., those governing civil aeronautics, might be to require that only one person, of suitable intelligence and experience in civil aviation, draw them up and maintain them.

Languages, Meta-Languages, Meta-MetaLanguages, . . .

Suppose we have n objects, say $n = 3$. Call the 3 objects A, B, and C. Let us assume that all we ever want to say about n objects is which of the n objects we are currently considering. Thus we

might want to say, "I am considering *A* and *C*", or, "I am considering none of *A*, *B*, or *C*", or, "I am considering *B* alone". Such statements can easily be encoded in *n*-bit strings. Thus, under the column headings "A", "B", and "C", we list all 3-bit strings (a total of $2^3 = 8$). Each string indicates which of *A*, *B*, and *C* we are considering. "0 1 1", e.g., means we are not considering A, but we are considering B and C. We can now repeat the process for the 3-bit strings. There will be a total of $2^8 = 256$ 8-bit strings. Each string indicates which of the 3-bit strings we are considering. "0 0 0 0 0 1 1 0" means that we are only considering the 3-bit strings "0 0 1" and "0 1 0". We can repeat the process over the 8-bit strings, yielding a total of 2^{256} 256-bit strings.

As every student of mathematics and computer science knows, there are precisely 2^n subsets of *n* objects, or, for us, 2^n statements we can make about *n* objects. But these 2^n statements themselves constitute a set of objects, and so we might wish to repeat the process, now using the same mechanism to make statements about our original 2^n statements, i.e., creating a first meta-language.Moral of the story: if we want to say "everything" about *n* objects, and everything about what we have said about the *n* objects, and everything about what we have said about everything about what we have said about the *n* objects, etc., we need more and more space (bits) to do it in.

Wittgenstein's *Tractatus*

Wittgenstein's *Tractatus Logico-Philosophicus* (Wittgenstein 1961), was written in 1918, some 12 years before Gödel's Incompleteness Theorem and some 30 years before the invention of information theory.

The *Tractatus* attempts to describe what cannot be expressed in a logical language, e.g.:

"Propositions cannot express logical form: it is mirrored in them." (Section 4.121) "What expresses *itself* in language, *we* cannot express by means of language." (Section 4.121) "What *can* be shown, *cannot* be said." (Section 4.1212) "What we cannot speak about we must consign to silence." (Section 7)

By Wittgenstein's own testimony, the *Tractatus* was misunderstood by most of its first readers, including Wittgenstein's former teacher, Bertrand Russell. Was Wittgenstein attempting to get at the kind of cardinality argument described in the previous sub-section?

The Computer Mail System Problem

Why can't I always use the computer mail system to inform the system operator that the computer is not running?

The Broken Bike Problem

Why can't I always ride my bike to the bike shop to have it fixed?

The Garbage Can Problem

If the garbage man always takes only what is in the garbage can, how do you throw out the garbage can? (This question is asked by comedian George Carlin on one of his record albums.)

"The other day I bought a wastebasket and took it home in a paper bag. Then I threw the bag into the basket." — Lily Tomlin

The Halting Problem

It is well known in computation theory that there does not exist a Turing Machine *M* which can predict, for each Turing Machine *M'* and each input *i* to *M'*, whether *M'* will eventually halt. Yet the reason is not that the instructions in *M* or *M'* are vaguely defined. Rather, it is as though

there isn't enough room in the space of Turing Machines for Machines such as *M*.

Thinking about the matter fancifully, we might ask, "Well, if there isn't enough room in one domain to hold all the machines we would like, then why not put the 'extras' in another domain, and then, when we need an answer from one of these machines, simply go into that domain, get it, and bring it back?" (See also "Piet Hein's Boxes" on page 26.) We envision a trap door through which we climb from one domain to the other, and through which we bring back these answers.

The problem may not be so much whether we can "put" the extra machines in another domain, or whether we can construct a trap door to get them, but whether we can bring back the answer through the trap door. Suppose, for each answer I bring back, an answer that already existed downstairs would have to be moved upstairs?

Perhaps this is related to the problem of doing computations on a machine that has too-little memory. Perhaps we could bring some answers back if we settled for partial, incomplete answers, e.g., assertions about the probability that theorems are valid.

Finite-State Machines

A finite-state Machine can be regarded as a type of Turing Machine all of whose memory is "in" the program, (A program by definition consists of only a finite number of bits.) Thus, a finite-state Machine can only remember up to a certain number, the number of the input tape cell it is currently reading. Thereafter, it "loses track"; its counter is reset to an earlier number. We might say that a countable infinity of tape cells all "have the same number" as far as the Machine is concerned.

The Coffee Shop Problem

A man is looking for female companionship. He believes that, on a particular Saturday morning, the right woman might be at one of three coffee shops. How should he go about finding her? If he decides to go to Coffee Shop *A*, she may be in Coffee Shop *B* or *C*, and be gone by the time he gets there.

It occurs to him that he could get two of his friends to help him, giving each a description of the type of woman he is looking for, so that all three coffee shops could be covered at once. But by the time he got the friends together, she might be gone. Furthermore, three searchers might not be enough, especially if the Coffee Shops are crowded, and especially if the woman's personality is important to him, so that it will be necessary to engage every likely prospect in conversation. If the woman is shy, the approach by a strange male may frighten her off. Or, she may begin to like the friend, and the two may go off together.

"Si elle avait recontré seulement
A, B, C, ou D, au lieu de Moi,
Elle les eût amés uniquement!"

(If she had only met
A, B, C, or D, instead of Me,
She would have loved them exclusively!)

— "About a Defunct Lady", *Poems of Jules Laforgue*,
tr. Patricia Terry, University of California Press,
Berkeley, CA, 1958, p. 177.

This scenario is another example of our universe seeming not to have enough "room" informationally. Why can't a thousand searchers descend on each coffee shop and not disturb a thing? Why can't there be simultaneous multiple copies of every coffee shop, so that in this one, the woman quietly reads a book while simultaneously in another she politely answers the questions of the searcher and in another she leaves because she doesn't want to be bothered? Thinking of our Turing Machine example above, we are inclined to reply that, even if such simultaneous universes were possible, the problem might be bringing back what you learned in one, to the other.

Find the Answer

Consider those mathematical problems in which we are asked to find a certain integer. But we already know all the integers. We simply need to hunt through all the integers in some systematic fashion until we find the one that supplies a correct answer to the problem. We can use a computer to carry out the search. The only reason this doesn't provide us with the answer to all integer-valued problems is that the number of integers that might have to be searched through may be so large that we can't wait for the computer to complete its search.

The Consumer's Problem

Conservative economic thinkers are fond of extolling the virtues of the marketplace, but here too there is an important informational aspect, for, suppose the marketplace is so large, or the wares so complex, that no prospective buyer can possibly examine all the wares offered for sale before they are sold out (to buyers who do not care to examine all the wares) or before they are no longer manufactured. Thus, the virtue of the marketplace rests on the assumption that there will be time to examine at least a "representative sample" of all the wares before making a choice.

One might turn to professional product examiners. But suppose the number and variety of these grows to such an extent that one cannot even search the marketplace of product examiners in time.

Phone Lines

"Clinton Hotlines Overwhelmed — It May Be Quicker to Write" — headline in *San Francisco Chronicle*, Mar. 6, 1993, p. A1.

Paperwork in a Business

Some businesses will not accept orders for less than a minimum dollar amount because the cost of handling the paperwork is greater than the profit they can make on a sale below that amount. We can imagine a company becoming more and more inefficient in its paperwork handling, so that its minimum order amount continues to rise until the cost of handling the order is greater than the cost of making and shipping the product.

Entropy

"The appearance of temporal asymmetry in thermodynamic phenomena is extraordinarily varied. For example, suppose that we remove the stopper from a flask filled with coloured gas, and watch it diffuse away; or place an ice cube in a jar of warm water and observe it melting; or witness the radiation of heat by an electric wire. In all these processes there is asymmetry in the following sense: the reverse sequence of events is never seen to occur. The spontaneous accumulation of gas atoms in an open flask is not encountered, nor the spontaneous freezing of a

small part of warm water in a jug, nor the generation of electricity by heating up an electric wire." (Davies 1977, 29)

Suppose we had a fault-free computer with unlimited memory on which we wanted to simulate a gas diffusing from a flask by simulating the movement of each molecule. Suppose, further, that we wanted to attach to each molecule a record of its travels, so that we could run the diffusion backwards from the molecular level, i.e., without making a single record of the entire simulation. What difficulties would we run into? Might it be that, "in the limit", such a simulation would no longer proceed due to the enormous time required to update the record of each molecule's travels?

Note the similarity between this example and our "canonical" model, namely, the Information Structure which becomes so complex that it is no longer possible to represent the whole in each of the parts (nodes)

Chemicals and Their Names

Some chemical names consist of long strings of letters, numbers, and hyphens. Suppose each molecule had to be able to carry its own name. Would some molecules then be legislated out of existence? Suppose each molecule had to contain a map of the entire universe, with a "You are here" indicator showing the location of the molecule. Would some, or all, molecules then be legislated out of existence? Suppose the map only had to cover part of the universe. How small would that part have to be in order for the molecule not to be legislated out of existence?

Why Is There Learning At All?

"Heredity is nothing but stored environment." — Luther Burbank.

Why doesn't every living thing carry all the information it will need to govern its behavior throughout its life? E.g., "On day 25, fly down from nest to a point 2 ft. NNW from base of tree and eat 17 pine seeds there." We are inclined to respond: because no reasonably-sized animal or plant (perhaps no possible animal or plant!) could carry that much information. It is as though Nature decided, *for plants and animals beyond a certain complexity,* to leave most of the information in the environment, and instead instructed each member of the species, "Don't worry about the species losing what you've learned in your life. The species will pick it up next generation." "The environment is information", and each generation recaptures the information it needs from the environment by the process called "learning". For example, instead of each bird being born with a program describing exactly how to move its wings each moment of each flight (a "static" universe), it is born instead with a program which says, in effect, "Keep trying this kind of thing until that results, then remember what you did."

Certainty in Mathematical Proofs

Consider the question of certainty in mathematics. We feel that a proof that runs several hundred pages is more likely to contain an error than a proof that runs, say, half a page. We are inclined to say that certainty is a local, not a global, property of proofs. The shorter the proof, or the more proofs there are for a given theorem, and the more mathematicians who review these proofs without finding an error, the more likely the proofs are correct.

We may say, "'Very long' proofs begin to defeat the whole purpose of a proof", namely, that of demonstrating logical validity.

Informational Cost of Program Proving

In the 1970s, I attempted to prove the correctness of a program I was writing for my job, in short, a "real-life" program. As I recall now, the program, written in an Algol-like language, consisted of perhaps a dozen one- or two-page procedures, plus several pages of declarations.

I attempted to do inductive proofs of every iteration in every procedure. Typically, the proofs were at least as long as the procedures themselves. Furthermore, the proofs themselves contained errors, as I found out when a procedure supposedly proved correct was revealed to have errors in actual practice.

This experience was one of the motivations for my beginning to think about information systems which are too "heavy" to support their own weight. Here, the cost (in errors) of writing and checking proofs was greater than the cost of the normal write-check-and-debug process by which most programs are written.

Scientific Research

An acquaintance of mine once told me that, during a conversation with a scientist at a leading university, the scientist remarked that it is sometimes easier to redo an experiment than to look it up in the literature. And yet certainly one of the reasons why scientists publish the results of their work is to save others from having to duplicate that work.

We are inclined to say that there are information systems so complicated that they begin to "fold over on themselves", meaning not that they become more regular, more patterned and orderly, but that, for their users, they begin to look the same as they were at an earlier stage of their history. (E.g., scientific knowledge at a stage when it was extremely unlikely to find the answer to your question in the literature.)

Medical Care

We are reverting to an earlier time when the non-existent treatment then has become the unaffordable treatment now.

Bureaucracies

Consider a bureaucracy in which each department has a staff of couriers to carry messages, memoranda, etc. to other departments. There are two possibilities: each office can maintain a supply of maps of the entire bureaucracy which the couriers then carry with them on their rounds. Or the maps can be posted at every intersection of corridors, so that the couriers don't need to carry maps themselves. Are there bureaucracies so complex that it is not possible to store a map in each office, or in each corridor?

Highway Systems and Subways

There are many ways of finding your way around a country that is new to you: one is to carry a map in your car; another is by reading signs. Suppose, at each crossroads there is a sign containing a map of the entire country, with a "You are here" marker. Are there highway systems so complicated that you would need most of the room in your car for maps, or such that the signs would be inordinately large?

In some subway systems, e.g., in London, there are maps of the system in each car and on each waiting platform, so that passengers can figure out how to get to their destination. Are there subway systems so complicated that such maps are impossible? Le., such that the maps themselves would have to be as large as the subway system?

The Problem of Too Many Masterpieces

How many artistic masterpieces have been created in this century? In all past centuries? Is it possible for one person to view, or hear, or read all of them? If not, then those masterpieces which the person never gets to know, might just as well never have been created as far as he is concerned. (But that statement overlooks the influence of works of art on later artists.)

Probability

Why is it uncertain what number will turn up when I throw a fair die? Is it because there isn't enough "room" in the information domain which is our physical universe for the information regarding the outcome of each case? ("All we can tell you is that there is an equal chance that some number between 1 and 6 will turn up.")

"Things have a shape, and that shape is the probability of their occurrence."

Why can't there be probabilities greater than 1 or less than 0? An event having probability < 1 is not certain to occur; an event having probability 1 is certain to occur; why couldn't it be that an event having probability > 1 is certain to occur simultaneously with other events? Does probability "wrap around" at some point, so that some probability > 1 and some probability < 0 are equal?

Describing a Waterfall

Suppose that the only way we could represent a waterfall was by another waterfall. (Like a random binary string which can only be described by a copy of itself.) What good would such a representation be?

Paradoxes

Some paradoxes seem to involve an endless cycle, or what programmers call an "infinite loop", which arises out of some form of self-reference. Are there assertions which are "almost" paradoxical but not quite, i.e., which involve a very large, but always finite, number of repetitions of something?

The underlying idea in the following is that paradoxes occur when two information spaces have been "superimposed" on one another, or, to put it another way, when two information spaces are being viewed, via a kind of optical illusion, as though they were one.

Would there be any advantage in describing paradoxes informationally?

The classical definition of information contains no obvious susceptibility to paradox. Given any assignment of probabilities to the occurrences of source symbols, we can compute the average amount of information per source symbol, and, as a result, know what the minimum size of any code strings must be with which we propose to represent source symbols.

A code, in information theory, is a mapping from source symbols to strings over some code alphabet. Do I have a paradox when the same code string is mapped to by two or more different source symbols?

Examples of Paradoxes

We begin with a few short examples of a kind the reader is probably familiar with.

A asked "Am I able to speak?" But questions of the form, "Is *x* able to speak?" are not paradoxical as long as *x* is not the speaker.

B wrote the message, "Teach me how to write." Again, commands of the form, "Teach *x* how to write" are not paradoxical as long as *x* is not the writer of the command.

Consider this widely known paradox: A coin has written on each side, "The statement on the

other side of this coin is false." Is either statement true, or false? But statements of the form, "The statement on the other side of coin *x* is false", are not paradoxical as long as they do not occur on both sides of coin *x*.

Irresistible Cannonball, Immoveable Post

Another well-known paradox first postulates an irresistible cannonball which can knock over anything in its way, and an immovable post which cannot be knocked over by anything. The question is then asked, what happens when the cannonball hits the post?

The answer is that "it is logically impossible that there can exist *both* an irresistible cannonball *and* an immovable post" (Smullyan 1978, 15). Either the cannonball or the post can exist separately without logical contradiction, i.e., each can exist in a separate domain. The paradox only arises when we merge the two domains.

Russell's Paradox

Russell's Paradox arises from the question, "Does the set of all sets which do not contain themselves, contain itself?" (If it does, it doesn't; if it doesn't, it does.) Russell's solution, namely, the Theory of Types, can also be regarded as an expanding of an information space: each Type can only contain elements of a lower Type, or, looking at it from our informational point of view, we may say that the originally "flat" information domain that produced the paradox, has been pulled out, like a telescope with an infinite number of sections (Types), such that each section contains enough information space to represent all the lower sections (Types).

The Barber Paradox is an expression of Russell's Paradox in more familiar, everyday terms. It postulates a town in which the barber shaves all and only those men who do not shave themselves, and then asks if the barber shaves himself. (If he does, he doesn't; if he doesn't he does.) The paradox disappears if we expand the information space by placing the barber in a separate town, say Town A, and his customers in Town B. Then the barber can shave all and only the men in Town B who do not shave themselves. In Town A, whether or not the barber shaves himself, there is no paradox.

For more on Russell's Paradox",sets that contain themselves, and that do not, see "Thoughts on Russell's Paradox" in the section, "Set Theory" in the second file of "A Few Off-the-Beaten-Track Observations..." in occampress.com.

Piet Hein's Boxes

The Dutch mathematician Piet Hein has published a number of riddle poems which he calls "grooks". One of them is:

"A bit beyond perception's reach
I sometimes believe I see
That Life is two locked boxes, each
Containing the other's key."

(Gardner 1973, 104)

I suggest that the paradox disappears if we create more information space, in this case, a "mirror", or "complementary", universe. Let there be:

Two universes, Universe 1 and Universe 2, separated by a vertical Interface.

We have the following rules:

Rule 1: The complement of Universe 1/Universe 2 is Universe 2/Universe 1.

Rule 2: The complement of object A/B is object B/A.

Rule 3: Complementary objects can be made to appear at any point on the Interface, or to vanish by bringing them together at opposite sides of the Interface at the same point.

Rule 4: Whatever is done to an object in Universe 1/Universe 2, is done to its complement in Universe 2/ Universe 1.

To explain Hein's riddle, proceed as follows:

1. Make Box A/Box B appear simultaneously in Universe I/Universe 2 from some point of the Interface.

2. Make Key B/Key A appear simultaneously in Universe 1/Universe 2 from some point of the Interface.

3. Put Key B/Key A in Box A/Box B.

4. Make a new Key A/Key B appear simultaneously in Universe 1/Universe 2 from some point of the Interface.

5. Lock Box A/Box B with this second Key A/Key B.

6. Make this second Key A/Key B vanish by bringing them together at some point of the Interface.

"Paradoxes" in Quantum Mechanics

A famous "paradox" in quantum mechanics is that of Schrödinger's Cat, which, being unobserved inside a closed box, must be regarded as being both dead and alive. In the 1950s, Hugh Everett, a graduate student at Princeton University, proposed, in his PhD thesis, the so-called "many worlds", or "parallel universes" theory, which states that, whenever a choice between two or more different outcomes exists in the universe, each outcome branches into a separate universe. So, in one universe, Schrödinger's Cat is alive, and in another it is dead.

Surely this theory is an extreme case of solving a paradox by "making more space".

Information-Based Speculations on the Physical Universe

Computational Models Of The Physical Universe

I would now like to set forth some speculations on the physical universe which are based on computational models. My aim here is to discuss informally the possibility that the universe is a

self-representing information structure, i.e., that it entirely contains the program that "makes it go". In the background of most of these speculations is John H. Conway's well-known Game of "Life", which is described briefly in the following section.

A Computational Model: Conway's Game of "Life"

I quote Martin Gardner's original description of the game. There have been numerous computer simulations of the game.

"To play life you must have a fairly large checkerboard and a plentiful supply of flat counters of two colors. (Small checkers or poker chips do nicely.) An Oriental 'go' board can be used if you can find flat counters that are small enough to fit within its cells. (Go stones are unusable because they are not flat.) It is possible to work with pencil and graph paper but it is much easier, particularly for beginners, to use counters and a board.

"The basic idea is to start with a simple configuration of counters (organisms), one to a cell, then observe how it changes as you apply Conway's 'genetic laws' for births, deaths and survivals. Conway chose his rules carefully, after a long period of experimentation, to meet three desiderata:

"1. There should be no initial pattern for which there is a simple proof that the population can grow without limit.

"2. There should be initial patterns that *apparently* grow without limit.

"3. There should be simple initial patterns that grow and change for a considerable period of time before coming to an end in three possible ways: fading away completely (from overcrowding or becoming too sparse), settling into a stable configuration that remains unchanged thereafter, or entering into an oscillating phase in which they repeat an endless cycle of two or more periods.

"In brief, the rules should be such as to make the behavior of the population unpredictable.

"Conway's genetic laws are delightfully simple. First note that each cell of the checkerboard (assumed to be an infinite plane) has eight neighboring cells, four adjacent orthogonally, four adjacent diagonally. The rules are:

"1. Survivals. Every counter with two or three neighboring counters survives for the next generation.

"2. Deaths. Each counter with four or more neighbors dies (is removed) from overpopulation. Every counter with one neighbor or none dies from isolation.

"3. Births. Each empty cell adjacent to exactly three neighbors — no more, no fewer — is a birth cell. A counter is placed on it at the next move.

"It is important to understand that all births and deaths occur simultaneously. Together they constitute a single generation, or, as we shall call it, a 'move' in the complete 'life history' of the initial configuration." (Gardner 1970)

The Cellular Hypothesis

The Game of Life takes place on a checkerboard. In the following speculations I assume that

space-time is "cellular", i.e., that space-time consists of "cells" such that each cell can contain only a finite amount of information (finite number of bits).

Why Is There Order In The Universe?

Consider the events in the physical universe as being governed by a program. One possibility is that the program is entirely contained within the universe itself (the universe is describable within the universe itself). One way of expressing this is to say that each cell must contain a program to describe how that cell is to behave, based on the behavior of adjacent cells. I now assert that the universe must have order (i.e., regularity), because after the universe has persisted for a certain time the program in each cell, consisting of a finite number of bits, must begin to repeat itself, although miracles are not impossible (a program can always contain a test for a condition, e.g., the solution to an unsolved mathematics problem, which may or may not ever occur, but whose occurrence in any case has nothing to do with the size of the program).

If each cell could contain an infinite number of bits, or if all or part of the program could lie outside the universe, then miracles could occur any time the Creator wished, e.g., after billions of years, the Law of Gravity could be suspended in any space-time locale.

How Large Is The Program That Runs The Universe?

If space-time is cellular, what is the maximum number of bits that could occupy a cell? Could the physical universe be simpler or more complicated than it is? (I must recommend to the reader the article, "The Anthropic Principle" (Gale 1981), in which such questions are rephrased as, "Would there be intelligent beings to ask questions like this if the universe were different?").

What Would It Be Like To Read The Book Of All Knowledge?

The Book of All Knowledge necessarily contains knowledge of the reader's reading it. If the reader read those parts, what would he read? Would the text run something like, "Now you are reading the part of the Book that describes your reading of part of the Book . . . Your eyes are now seeing these words, and these words, and these words, . . ."

("The sentence I am now writing is the sentence you are now reading" (Hofstadter 1979, p. 495). Hofstadter's column, "Virus-like sentences and self-replicating structures" (Hofstadter 1983), seems related to this idea.)

When you are not reading the Book, there must be a part devoted to describing your behavior while you are not reading it — a part you could never read, of course. Other persons, however, would be able to read this part, which might run, "At (space-time coordinates), *x* stopped reading the Book. He busied himself with . . . Then, at (space-time coordinates) he reopened the book and turned to page *n."*

We sense that the Book of All Knowledge would have to contain *too much* knowledge, that there wouldn't be enough room, no matter how big it was, to describe Everything, or that the description would overflow into that which was being described.

How Does The Effect Know What To Do When The Cause Causes It?

The Two Planets Problem

A planet of a distant star appears, to observers on earth, to be in communication with a planet of another star, the means of communication being long and short flashes of light. Following the sending of each sequence of flashes, and a time period equal to that required for the light to traverse the distance between the two planets, the receiving planet repeats the sequence, always with

a prefix and a suffix of, say, three short flashes; then, after a period in which the receiving planet apparently ponders its reply, a period ranging from a few hours to several days, the receiving planet proceeds to send another message in reply.

Earth astronomers attempt to decipher the messages. Then, one day, the speed of response by the then-receiving planet seems to increase: soon the response is made within seconds after the message is received, then, it is made before the message could have been received, even though the repetition of the message is always correct. Earth scientists conclude that other intelligent beings have simply played a joke on observers by preprogramming separate light transmitters on each planet to give the illusion of communication.

Is The Universe Deterministic?

Some people spend many hours in their youth wondering if the universe is deterministic. Some of these people end their pondering with the thought that, if we believe the universe is deterministic, then we behave differently than if we believe it isn't, or than if we had never thought about the question in the first place.

It is well known in computation theory that there does not exist a Turing Machine M that can predict, for every Turing Machine M' and every input i to M, whether M' will eventually halt. Yet the reason is not that the instructions in M or M' are vaguely defined. In keeping with one of the themes of these notes, we may say, it is as though there isn't enough room in the space of Turing Machines for Machines such as M.

Thus, even if God does not play dice, there may be non-determinism in the universe, simply because there are not enough bits for the left hand always to represent what the right is doing.

Does Nature Calculate Its Dimensions?

Does Nature calculate? Does it "figure out" how far apart things are to be, for example? If not, why do they wind up just that far apart? These questions derive from a passage in a biography of Buckminster Fuller:

"Imagine him then, standing, age twenty-two, at the stem of a running ship, gazing at its white wake on the dark water. Millions of little bubbles compose this whiteness, each one a little sphere or partial sphere, each exemplifying the mathematics of spheres, including presumably the famous π which enters the mathematics of anything circular.

"'I'd learned at school that in order to make a sphere, which is what a bubble is, you employ π, and I'd also learned that π is an irrational number.' So 'when', he recalls asking, 'does nature have to fudge it and pretend it comes out even and then make some kind of compromise bubble?' And millions of them per second. 'I think it's too many decisions for nature to make.'" (Kenner 1973, 132)

Does calculation only come into being with intelligence?

How does the electron know what to do next? Where is the program that governs its behavior? (Which may be a question of the same type as "Have you stopped beating your wife?".)

"Here was I, sitting at my desk for weeks on end, doing the most elaborate and sophisticated calculations to figure out how an electron should behave. And here was the electron on my little oil drop, knowing quite well how to behave without waiting for the result of my calculation. How could one seriously believe that the electron really cared about my calculation, one way or the other? And yet the experiments at Columbia showed that it did care. Somehow or other, all this complicated mathematics that I was scribbling established rules that the electron on the oil drop

was bound to follow. We know that this is so. Why it is so, why the electron pays attention to our mathematics, is a mystery that even Einstein couldn't fathom." — Dyson, Freeman, Disturbing the Universe, Basic Books, N.Y., 1979, p. 50.

Digression On The Laws Of Nature

What kinds of behavior distinguish those who have discovered certain laws of Nature from those who have not?

How Much Can We Know About The Universe?

The Heisenberg Uncertainty Principle

"This is the way Heisenberg stated the uncertainty principle originally [in 1926-27]: If you make the measurement of any object, and you can determine the *x*-component of its momentum with an uncertainty delta (Δp), you cannot, at the same time know its *x*-position more accurately than $\Delta x = h/\Delta p$). The uncertainties in the position and momentum at any instant must have their product greater than Planck's constant [*h*]." (Feynman 1963, 37-11)

Does the Principle arise because of the cellular nature of the universe, i.e., because there is a limit as to how much information one cell, or one group of cells, can contain about another cell or group of cells?

Or is it that, when dimensions become small enough, we see Nature in the act of never completing its calculations, so that a "blur" surrounds "what happens next". Suppose, at the quantum level, we think of measurement as a kind of robbing Peter to pay Paul. That is, suppose that, at this level, the only way of describing, of making a picture of, say, a mountain, is by taking some of the stone that constitutes the mountain, and making a smaller model of the mountain. Suppose the mountain is made of boulders of a certain uniform size. Then the model will have a basic "roughness", "crudeness", which is established by the size of the boulders. Suppose we make the model bigger to overcome this. Then we need to take more boulders from the original mountain, thus changing its shape. Perhaps our model-building effort must stop when the model is the same size as the mountain.

Data Banks and Libraries

Consider man's attempts to know more about the universe. We model the universe now as *a finite checkerboard,* in which each square can contain one bit of information. Clearly, some bits of the original universe must be sacrificed to the representation of the rest. If we think of the knowledge which humans on earth have, and which can be represented symbolically, then all information storage media correspond to this part of the checkerboard. At first, we are inclined to think that the quantity of matter in these media is trivial: what proportion of the earth's atoms were in all magnetic tapes and disks in 1983, for example? (How could we find out?) But magnetic tapes, like magnetic disks, movies, books, video disks, video tape, microfilm, etc., do not come into being out of nothing, but instead are the product of much greater "disturbances" in the pre-knowledge universe, e.g, the disturbances produced by mining and processing and transporting the ores that contain the special metals that are used to make these media. So the amount of "turf" that had to be disturbed to accumulate the knowledge which is represented in the above media, is perhaps not negligible.

Now, suppose that we want to know not only about the original, primordial universe, but also about our knowledge of that universe, i.e., suppose we want to study "the nature of human knowl-

edge". Now we will need a representation of our first representation. As I believe Chaitin has stated somewhere: in general, the patterns constituting computer programs are themselves near-random, so the representation of the the representation of ... of knowledge is relatively expensive in terms of bits.

Given a finite universe, then, eventually we will be unable to represent, in that universe, everything that we want to represent.

Entropy

Irreversible processes are well-known in nature, e.g., when we drop a drop of ink in water, we are familiar with the ink spreading through the water until the water is of uniform color, but we never see the reverse process, i.e., a jar or pitcher of water with ink dissolved in it coalescing into a drop.

Is this because there aren't enough bits in the universal fabric to record "everything"? Suppose each molecule were like a fortune cookie, where the fortune described the molecule's future, what it was to do next, as well as its entire past. Would there be enough bits?

Additional Thoughts

One way of regarding this chapter is as a posing of the question, "Does anything of interest lie at the intersection of topology, information theory, and computation theory?"

"Suppose All There Is, Is Structure"

Suppose all there is, is structure: what would this mean? Among other things, it would mean that "what" = "where".

The Lens of Abstraction

The "lens of abstraction" is a suggestive term for the act of restricting our attention to certain aspects of a subject. For example, in mathematics it sometimes seems desirable to be able to take "horizontal" slices through many different subjects at the same time (viewing the normal studying of one subject at a time starting at the beginning as studying "vertically"). These horizontal slices are centered on concepts which appear in different subjects, e.g., the concept of function, or the concept of "quotient" as represented by the familiar quotient in arithmetic, quotient sets, quotient groups and rings, quotient topologies, etc., or the concept of periodicity ("going-around-ness") as represented by loops and recursion in programming, the congruence relation, finite groups, periodic functions, certain operations on complex numbers, winding numbers, etc.

Of course, all branches of mathematics look the same under the lens setting, "axiomatic system".

Topology, Information Theory And Computation Theory

Does anything interesting result if we merge topology and information theory? Can we speak of information "surfaces" (..."Here there is almost no change in information, there we have a very steep information gradient"...)? Valleys (low information) then correspond to patterns, regularity; peaks correspond to randomness. Is there a meaningful sense in which we can assert that informa-

tion *x* is "near" to information *y*? Perhaps in the sense that the two pieces of information are measured by nearly the same number of bits?

What does a rough surface represent? A smooth surface?

When we learn something, at first almost everything is new to us, hence unpredictable (high information content); then, gradually, through repetition, less and less is new, there are fewer and fewer surprises each time we return to the task or subject. How might we model this as an information surface?

Can we impose a topology on information structures? (The question should really be: Is there a mathematical model which captures what we have been loosely calling an "Information Structure", upon which we can impose a useful topology?) For example, can we speak of a given sequence of bits, "random" in the sense of algorithmic information theory, as being "near to" or "far from" some other random sequence?

Can we impose a topology on the theorems themselves in a branch of mathematics? Can we say that theorem *y* is "near" to theorem *x* in any meaningful sense other than that theorem *y* can be deduced from theorem *x* in a relatively small number of steps? Informally, theorem *y* might be "near" to theorem *x* if, e.g., the two theorems have similar logical forms, or if they consist of roughly the same number of characters, or if they are near to each other in an alphabetical listing of theorem statements, or if they concern the same objects (e.g., prime numbers, Cauchy sequences, Hausdorff spaces), or if their proofs utilize the same kind of argument, etc.

How much information is there in a given theorem? In a given axiomatic system? (See Chaitin references in Bibliography.)

Suppose we think of each branch of mathematics as an "information surface". We may then describe some branches as "lumpier" than others, meaning that each new problem, each new concept in such a branch, seems to require its own intellectual machinery: the terrain is jagged, hilly, unpredictable, whereas in other branches, things seem "smoother": one concept or technique seems to go a long way, things seem to lead into each other in a nice way.

For me, unquestionably, classical number theory is "jumpy" in this sense; theory of functions of a complex variable is not.

Is lumpiness a sign that we have not yet gotten to the fundamentals of the subject? Consider the topological approach which Scott and Strachey took in developing their mathematical semantics of programming languages (see Bibliography). The functions dealt with there are, of course, the computable functions, which include most, if not all, of the functions of classical number theory. Can it be that this type of approach will take us closer to the fundamentals of number theory than do the approaches one confronts in the typical number theory textbook?

Is it possible to distinguish one branch of mathematics from another "blindfolded", i.e., by structure ("feel") alone?

How Much Mathematics Is There?

It has been remarked that "ninety percent of all the mathematics we know has been discovered (or invented), in the last hundred years" (Temple 1981, xv).

Is this because more mathematicians have lived in the past 100 years than in all previous centuries combined? Is progress in mathematics limited by the size of the mathematical community?

Is there an optimum number of mathematicians for a given stage of technological development, such that fewer mathematicians could not make use of the available technology, and more mathematicians would overload it, causing, in effect, a regressing, a "folding back", to the equiv-

alent of a more primitive stage? ("It is often easier to prove the theorem yourself than to find it in the literature.")

"In mathematics, many areas show signs of internal exhaustion — for example, the elementary geometry of the circle and the triangle, or the classical theory of functions of a complex variable. While one can call on the former to provide five-finger exercises for beginners and the latter for applications to other areas, it seems unlikely that either will ever again produce anything that is both new and startling within its bounded confines." (Davis and Hersh 1981, 24-25)

Even though I am inclined to agree with this opinion, I can't help wondering if it might be wrong. Suppose I write a program to generate all the theorems of some axiomatic system. Such a system can be regarded as a grammar, and each theorem as a sentence produced by that grammar. If the system is "interesting", then there will be rules in the grammar which produce a shorter theorem from a longer one. Thus, in general it will not be possible to know when all the theorems of a given length have been produced.

Now since there will be a vast number of trivial theorems (every string produced by the grammar which represents the axiomatic system is, formally speaking, a theorem), I will not want to look at every theorem. Can I add to my program a criterion for deciding which theorems to show me, i.e., for deciding which are the potentially "interesting" theorems? Should I do this on a length-of-theorem criterion alone (length in number of symbols)? How do I decide what the maximum length of an "interesting" theorem should be?

"Interesting conjectures, like interesting numbers, tend to be concisely describable. It is hard to imagine a mathematically interesting, finitely refutable conjecture so verbose that it could not be encoded in the halting problem for a small program, one a few thousand or tens of thousands of bits long." (Bennett 1979, 28, remainder of reference no longer known)

Suppose I set some c as the maximum length of a theorem I am willing to call "interesting": do I really want my program not to show me any theorem of length, say, c + I? What about length $c + 2$, $c + 3$?

Why shouldn't there be interesting theorems which resist Bennett's criterion, i.e., which are irreducibly very long, say, hundreds of thousands of bits?

Probably most mathematicians would agree with Davis and Hersh's observation concerning the internal exhaustion of geometry and theory of functions of a complex variable. Yet I can't help wondering what the computer would "eventually" reveal if it began systematically to generate the theorems within some pre-established length.

We are again confronted with the idea of a topology on theorems themselves. We would like to be able to say things like, "Within such-and-such distance of theorem x it is highly unlikely there are other important theorems."

On Limited Space For Working Mathematics Problems

Until now, all mathematics has been done as though the space to do it in were unlimited! The tacit assumption has been that there is enough scratch paper to copy down each problem and to use in attempting to find an answer to each problem, and enough paper to publish the results. The business of the mathematician is to advance the subject — solve old problems, discover new ones.

I have suggested in these Notes several possible reasons for questioning this assumption: e.g., what happens when mathematics becomes so large a discipline that it frequently is easier for a mathematician to (re)prove theorems than to search for them in the literature? Is it correct to continue to call mathematics "a body of knowledge" under those circumstances?

Suppose you were only permitted a limited amount of paper to work on a mathematical problem, or only a limited number of bits in a computer memory — suppose, e.g., (borrowing an idea from (Spencer-Brown 1979, 79)) these bits were evenly spaced on the surface of a sphere? In this limited space you would have to do all your calculations, scribbling, drawing. How would this limitation affect the way you went about solving the problem? If the limited space were demonstrably not enough to enable you to solve the problem, would there be better and worse ways to make use of the inadequate space?

To express this another way: scholars, including scientists and mathematicians, are still living in a "flat-earth" information world, an information world of "absolute" space and time, where there is always enough room for more knowledge, and where adding to the existing body of knowledge does not change the "information space" surrounding that body of knowledge: a researcher can always, or almost always, find if a given item of knowledge exists, and if so, where it exists, more rapidly than he can recreate the item of knowledge.

On The Importance Of Knowing "What Lies Near To What"

When we study computer science or mathematics, we often find ourselves wanting to know *where* things in a given subject or problem are, relative to other things — "what lies near to what" — because this seems an obvious way to simplify things and understand them. For me, an isolated theorem or definition "floats", I feel I have to work like hell whenever I think about it because I have to imagine all its possible homes. As soon as I know where it goes in one or more structures, I feel I have begun to understand it. So for me, "where" is as important, even more important, than "what".

Suppose we are asked to multiply several pairs of large numbers, e.g.,

80791	80791	80791	etc.
<u>32654</u>	<u>32655</u>	<u>32656</u>	

We may simply plunge right in and do each calculation (by hand, or with a calculator or computer) or we may look at all of the pairs first, and notice that each successive multiplier is one more than the previous, and that all the multiplicands are the same. Thus, we may realize that we need only perform the initial multiplication and then add the multiplicand to obtain each successive product. Here, the second method is more inexpensive computationally. This may be taken as a crude illustration of what I mean by the importance of knowing "what lies near to what" in a problem or subject domain.

Or suppose we are trying to determine what function a given program computes. The normal way to go about this is to analyze the program, perhaps construct a proof to show that it computes a certain function. But if the program is known to occupy a space of programs in which one can talk about how "near" any two programs are to each other, then perhaps another way to determine the function is by "moving the program slightly", i.e., by changing it slightly and observing some of the changes in its behavior. Again, the idea is the same: "Don't ask *what* it is, ask *where* it is."

Theory Of Searching Vs. Theory Of What Is Searched

Suppose one were to agree that the sciences, and mathematics, were rapidly on the way to becoming non-self-representing structures. Suppose a new discipline called, say, "information

science", were established to deal with this problem — at least to improve our ability to find what we need within some reasonable time after we need it. Suppose this discipline gradually became the dominant discipline, since everyone recognized that a branch of knowledge so complex that it is often more efficient to perform the experiment, or prove the theorem, yourself than to attempt to find it in the literature, is no branch of knowledge at all. And now suppose that this new science began to grow at such a rate that soon it became difficult to find out if a given experiment had been carried out, or a given theorem proved, so that eventually a need arose for a new science . . .

Suppose the theorems in the first (and possibly the subsequent) of these new sciences began more and more to resemble the theorems in the original sciences and branches of mathematics?

Curves and Straight Lines

Why are there curves, waves, wrinkles in physical space? Why is it that, if you tie the ends of a 10-foot rope to two stakes 5 feet apart, the rope must curve? In elementary integral calculus, why are rectangles the primitive elements by which we compute the areas of figures with curved boundaries? Why, and in what respect, are straight lines and rectangles "simpler" than curved lines and curved areas? Are there informational answers to these questions?

Bibliography

Abramson, Norman. 1963. *Information Theory and Coding*. New York:McGraw-Hill.

Birkhoff, George David. 1956. Mathematics of aesthetics. In *The World of Mathematics*. *See* Newman 1956.

Brooks, Frederick P., Jr. 1975. *The Mythical Man-Month: Essays on Software Engineering*. Menlo Park: Addison-Wesley.

Chaitin, Gregory. 1966. On the length of programs for computing finite binary sequences. *JACM* 13 (October): 547-569.
——— . 1969. On the length of programs for computing finite binary sequences: statistical consideration. *JACM* 16 (January): 145-159.
——— . 1970. On the difficulty of computations. *IEEE Trans. on Information Theory* IT-16 (January): 5-9.
——— . 1974. Information theoretic computational complexity. *IEE Trans. on Information Theory* IT-20 (January): 10-15.
——— . 1974. Information-theoretic limitations of formal systems. *JACM* 21 (July): 403-424.
——— . 1975. Randomness and mathematical proof. *Scientific American* (May): 47-52.
——— , 1975. A theory of program size formally identical to information theory. *JACM* 22 (July): 329-340.
——— . 1977. Algorithmic information theory. *IBM J. Res. Dev.* (July): 350-359.

Davies, P. C. W. 1977. *The Physics of Time Asymmetry.* Los Angeles:University of California Press.

Davis, Philip. J., and Hersh, Reuben. 1981. *The Mathematical Experience*. Boston:Houghton Mifflin Co.

Dijsktra, E. W. 1976. *A Discipline of Programming*. Englewood Cliffs, N.J.: Prentice-Hall.

Encyclopedia Americana. 1979. s. v. “holography”.

Esslin, Martin. 1982. *The Theater of the Absurd*. New York: Penguin Books.

Feynman, Richard P., Leighton, Robert B., and Sands, Matthew. 1963. *The Feynman Lectures on Physics*. Vol. 1, *Mainly Mechanics, Radiation, and Heat*. Menlo Park, Calif.:Addison-Wesley.

Gale, George. 1981. The anthropic principle. *Scientific American* (December): 154-171.

Gardner, Martin. 1970. The fantastic combinations of John Conway’s new solitaire game “Life”. *Scientific American* (October): 120-123.
———. 1973. Free will revisited, with a mind-bending prediction paradox by William Newcomb. *Scientific American* (July): 104-109.

Grosser, Morton. 1981. *Gossamer Odyssey: The Triumph of Human-Powered Flight*. Boston,

Mass.: Houghton Mifflin Co.

Hampshire Stuart, ed. 1961. *The Age of Reason: The 17th Century Philosophers*. New York:The New American Library.

Hofstadter, Douglas R. 1979. *Gödel, Escher, Bach: An Eternal Golden Braid*. New York: Basic Books.

——. 1983. Virus-like sentences and self-replicating structures. Metamagical Themas, *Scientific American* (January): 14-22.

Kenner, Hugh. 1973. *Bucky: A Guided Tour of Buckminster Fuller*. New York:William Morrow and Co.

Knopp, Konrad. 1945. *Theory of Functions. Part 1*. Trans. Frederick Bagemihl. New York: Dover Publications.

Knuth, Donald E. 1969. *The Art of Computer Programming. Vol. 1, Fundamental Algorithms*. Menlo Park, Calif.:Addison-Wesley.

Mandelbrot, Benoit B. 1977. *Fractals: Form, Chance, and Dimension*. San Francisco: W. H. Freeman and Co.

Newman, James R. 1956. *The World of Mathematics*. 4 vols. New York:Simon and Schuster.

Parkinson, C. Northcote. 1957. *Parkinson's Law and Other Studies in Administration*. New York:Bantam Books.

Schorer, Peter. 1976. The mathematical semantics of the Lambda Calculus. Master's thesis. San Jose (Calif.) State University.

Scott, Dana. 1970. *Outline of a Mathematical Theory of Computation*. Oxford University Computing Laboratory, Programming Research Group Technical Monograph PRG-2. Oxford.

——, and Strachey, Christopher. 1971. *Toward a Mathematical Semantics of Computer Languages*. Oxford University Computing Laboratory, Programming Research Group Technical Monograph PRG-6. Oxford.

Shannon, Claude E. 1948. A mathematical theory of communication. *Bell Syst. Tech. J.* 27 (July): 379-423.

Smullyan, Raymond. 1978. *What Is the Name of This Book?* Englewood Cliffs, N.J.:Prentice-Hall.

Spencer-Brown, G. 1979. *Laws of Form*. New York:E. P. Dutton.

Temple, George. 1981. *100 Years of Mathematics*. New York:Springer-Verlag.

Wittgenstein, Ludwig. 1961. *Tractatus Logico-Philosophicus*. Trans. D. F. Pears and B. F. McGuinness. London: Routledge and Kegan Paul.