PUC

Series: Monographs in Computers Science
and Computer Applications

Nº 3/70

COMPUTATIONAL COMPLEXITY AND GODEL'S INCOMPLETENESS THEOREM

and

TO A MATHEMATICAL DEFINITION OF 'LIFE'

by

G. J. Chaitin

Computer Science Department - Rio Datacenter

CENTRO TÉCNICO CIENTÍFICO
Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 209 — ZC-20
Rio de Janeiro — Brasil

COMPUTATIONAL COMPLEXITY AND GODEL'S INCOMPLETENESS THEOREM

and

TO A MATHEMATICAL DEFINITION OF 'LIFE'

by

G. J. Chaitin

Computer Science Department - Rio Datacenter

PREFACE:

There is currently much mathematical research concerned with the performance and hardware requirements of algorithms.

These two papers are intended as an introduction to this field.

The first shows in detail some typical ideas and applies them to metamathematics.

The second is a survey of one tendency in computational complexity studies and suggests a possible future application.

ACKNOWLEDGEMENTS:

The first paper was written during the author's stay at the Pontifical Catholic University under the sponsorship of IBM World Trade Co.

The second paper was prepared in the Computation Center of the National University of Technology, Argentina.

COMPUTATIONAL COMPLEXITY AND GODEL'S INCOMPLETENESS THEOREM

ABSTRACT: The state and time complexities are defined for computations of infinite sets of natural numbers by means of Turing machines. It's proved that there are arbitrarily complex infinite computable sets of natural numbers. In formal theories, however, it is not possible to demonstrate that specific examples of such sets are of arbitrarily high specific complexity.

KEY WORDS: Computational complexity, Godel's incompleteness theorem, Turing machines, recursive sets.

1. DEFINITIONS:

In this paper we are interested in infinite computations done by means of 4 - tape - symbol Turing machines. The machine will be used to generate in numerical order all the elements of an infinite set of natural numbers. The technical name for such sets of natural numbers is "infinite recursive set". We shall focus our attention on two measures of the complexity or difficulty of such computations:

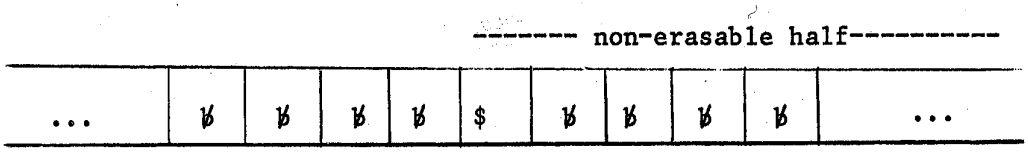
(i) the number of states of the Turing machine .

(ii) the time (number of state transitions) necessary for it to generate a natural number.

The Turing machines we use do their calculations on an infinite one dimensional tape divided into two halves. One half cannot be erased nor changed, and is used to write in binary notation, separated by $ signs, the successive elements of the infinite set of natural numbers being computed. It's considered that a natural number has

been output when a $ is written at its right. The four symbols used by the Turing machine on its tape are ∅ (blank) 0, 1, and $(delimiter).

The other half of the tape is used for intermediate results during the calculation. Initially the tape is all blank except for a $ on the first square of the non-eraseable half. The read-write head of the Turing machine is initially positioned on the $, and the machine is in state 1.



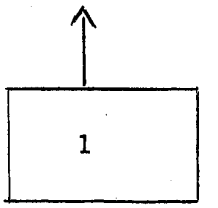


Fig. 1. Initial situation of the Turing machine.

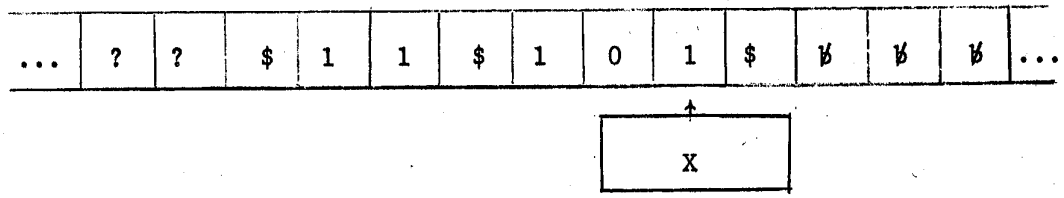


Fig. 2. The Turing machine generated the number 3 and has just generated the number 5.

The operation of any particular n - state Turing machine is defined by means of an n by 4 table.

Symbol on square on which read-write head is positioned.

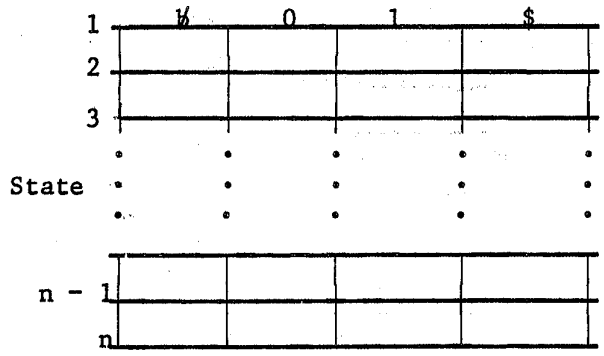


Fig. 3. A table for defining an n- state Turing machine.

Each entry in the table is an ordered pair consisting of an action to be performed (move the head one square to right (R) or to left (L) , write Ø (E), 0 (0), 1 (1), or $ ($)), and the number of the next state of the machine.

LEMMA: There are exactly $(6n)^{4n}$ different Turing machines with n states.

DEFINITION 1:

Let S be an i.r.s. (infinite recursive set), f be a t.r.f. (total recursive function from the natural numbers to the natural numbers), and n be a natural number. S is more difficult to compute in states than n iff every TM (Turing machine) that computes S has more than n states. We use the notation $S >n_s$. S is more difficult to compute in time than

f iff for every TM computes S there exists a natural number n_0 such that:

$\forall n\{((n \in S) \wedge (n > n_0)) \rightarrow$(the computation time for the TM to generate n is > f(n))$\}$

We use the notation $S >_t f$.

2. *ARBITRARILY HIGH COMPLEXITIES:*

In this section we shall show that although all i.r.s.'s, are computable some are easy to compute and others are difficult to compute. In fact, some are arbitrarily difficult to compute.

THEOREM 1:

For any natural number n there exists an i.r.s. S such that

$$S >_s n.$$

DEMONSTRATION:

There are infinitely many different i.r.s.'s, but by the Lemma there are only.

$$\sum_{k=1}^{n} (6k)^{4k}$$

TM's with not more than n states . Since each TM can compute at most one i.r.s., given any

$$1 + \sum_{k=1}^{n} (6k)^{4k}$$

different i.r.s.'s at least one of them cannot be computed by a TM with not more than n states.

q.e.d.

THEOREM 2:

For any t.r.f. f, there exists an i.r.s. S such that $S >_t f$.

DEMONSTRATION:

The set of all TM's can be put in an effective one-to-one onto mapping with the natural numbers; we will suppose that a subroutine is available for doing this. We will also need a subroutine for computing f, and another subroutine for simulating the operation of a given TM for a specific period of time ("universal" TM). From these subroutines we construct a TM that computes S.

The operation of this TM is divided into steps; during the nth step it's decided whether or not $n \in S$. Also there is a "garbage can", a set of TM's which is initially empty and may grow from step to step. In each step the following procedure is carried out.

(i) We take those TM's which correspond to natural numbers not greater than n/10.

(ii) We remove from this set of TM's those that are in the garbage can.

(iii) We simulate the operation of this set of TM's during time $f(n)$, and we select the subset consisting of those TM's which have generated the number n.

(iv) n is an element of S iff the selected subset of TM's is empty.

(v) We put the selected subset of TM's in the grabage can.

Evidently this construction defines a TM(Turing's thesis). S is in - finite because during the first n steps of the computation it is decided whether or not the first n natural numbers are in S, while only $\leq n/10$ TM's are simulated. Each TM can prevent at most one of these natural numbers from being in S, because when a TM prevents a natural number from being in S it is then put in the garbage can and will no longer be considered in future steps. Thus at most n/10 natural numbers $\leq n$ are not in S, that is to say, at least $n - (n/10) = 9n/10$ natural numbers $\leq n$ are in S.

It only remains to demonstrate that S is difficult to compute. By the definition we most prove that given any TM that computes S, there exists an n_0 such that each element n of S which is greater than n_0 generated by this TM a time greater than f(n). Let t be the natural numbers which corresponds to the given TM. We can take $n_0 = 10t - 1$.

The reason is as follows. The given TM will first make its appearance among the simulated TM's in step $n_0 + 1$. From this step on, the momment this TM generated a natural number n in time $\leq f(n)$, this n could not be an element of S, which is impossible.

q.e.d.

3. FORMAL THEORIES:

We use a stream-lined modern definition of formal theories which evolved from Hilbert's original conception of the meaning of "formal theory".

DEFINITION 2:

A formal theory is a recursively enumerable set of words over a finite alphabet. The words in the set are called theorems of the theory. (Recall that a recursively enumerable set is a set all of whose elements are generated by a TM).

For our present purposes, we shall suppose that all theorems are ordered pairs of two types. One type (TM, n) consists of the des - cription of a specific TM and a natural number written in decimal notation. The other type (TM_1, TM_2) consists of two descriptions of specific Turing machines.

The description of a TM is the ordered 4n - tuple consisting of the ordered pairs entered in the table defining the TM in the order that a person reads. Thus the TM.

	∅	0	1	$
1	(R, 2)	(L, 1)	(E, 2)	(L, 1)
2	(0, 2)	(1, 2)	($, 1)	(1, 1)

has the description ((R, 2), (L, 1), (E, 2), (L, 1), (0, 2), (1, 2), ($, 1), (1, 1)

We will show below that in a formal theory there exist upper bounds on the two types of complexities, contrasting with Theorems 1 and 2.

THEOREM 3: Hypothesis:

(TM, n) is a theorem of our formal theory only if the TM computes an i.r.s. S such that $S >_s n$,

Conclusion:

There exist a natural number N such that all ordered pairs (TM, n) which are theorems of the formal theory satisfy $n < N$.

DEMONSTRATION:

For any natural number n there is a TM with $\log_2 n + c$ states (where c is a constant depending only on our formal theory) which does the following. The first $\log_2 n$ states of the machine write the number n on the TM's tape in binary notation, and then they pass control to the remaining states. These states carry out the following procedure. They generate the theorems of our formal theory until a theorem of the form (TM, n) with $m > n$ is produced, (If no such theorem exist in our formal theory, the TM will loop forever). If such a theorem (TM, m) is found, the c last states cause the TM of the theorem to be simulated. By the hypothesis, an i.r.s. S satisfying $S >_s m$ will thus be computed.

In summary, for every n, if there is a theorem of the form (TM, m) $(m > n)$ in our theory, then there is a TM with $\log_2 n + c$ states which computes an i.r.s. S satisfying $S >_s m$. This implies.

$$\log_2 n + c > m > n$$

This inequality in n can only be satisfied for $n < N(c)$, where $N(c) \sim c + \log_2 n$ depends only on c and thus only on our formal theory. We conclude that there is an N such that no theorems of the form (TM,m) with $m > N$ are in our formal theory.

q.e.d.

THEOREM 4: Hypothesis:

(TM_1, TM_2) is a theorem of our formal theory only if TM_1 computes an i.r.s. S and TM_2 computes a t.r.f. f such that $S >_t f$.

Conclusion:

There exists a t.r.f. F such that all ordered pairs (TM_1, TM_2) which are theorems of the formal theory satisfy

$$\lim_{k \to \infty} (f(k)/F(k) = 0,$$

where f is the function computed by TM_2.

DEMONSTRATION:

The following is an algorithm for computing F(k) Generate the theorems of our formal theory until k theorems of the form (TM_1, TM_2) have been produced: (TM_1^i, TM_2^i) (i = 1, 2, ... k)

By the hypothesis, each TM_2^i computes a t.r.f. f_i.

Simulate each of the TM_2^i $(i = 1, 2, \ldots k)$ in order to determine the values of

$$f_i(k) \quad (i = 1, 2, \ldots k)$$

Compute

$$k \max_{i=1}^{k} f_i(k)$$

This is the value of F(k)

q.e.d.

Note that although the proof of Theorem 4 is a variant of the traditional diagonal-argument proof of Godel's theorem, the proof of Theorem 3 is quite different. It is related to the Berry paradox of "the smallest natural number not definable in 1000000 characters or less" (see pages 486/7 Beth - The foundations of Mathematics).

Theorems 1 and 3 carry over to the state complexity of computations of finite binary sequences by TM's; the resulting proof of Godel's theorem is even easier to digest.

REFERENCE:

Chaitin - On the difficulty of computations - IEEE
Transaction on Information Theory - January 1970

TO A MATHEMATICAL DEFINITION OF ' LIFE '

'Life' and its 'evolution' are fundamental concepts that have not yet been formulated in precise mathematical terms, although some efforts in this direction have been made. We suggest a possible point of departure for a mathematical definition of 'life'. This definition is based on the computer and is closely related to recent analyses of 'inductive inference' and 'randomness'. A living being is a unity; it is simpler to view a living organism as a whole than as the sum of its parts. If we want to compute a complete description of a region of space-time that is a living being, the program will be smaller in size if the calculation is done all together , than if it is done by independently calculating descriptions of parts of the region and then putting them together.

Section 1. The problem.

'Life' and its 'evolution' from the lifeless are fundamental concepts of science. According to Darwin and his followers, we can except living organisms to evolve under very general conditions. Yet this theory has never been formulated in precise mathematical terms. Supposing Darwin is right, it should be possible to formulate a general definition of 'life' and to prove that under certain conditions we can except it to 'evolve'. If mathematics can be made out of Darwin, then we will have added something basic to mathematics; while if it cannot, then Darwin must be wrong, and life remains a miracle which has not been explained by science.

The point is that the view that life has spontaneously evolved,and the very concept of life itself, are very general concepts, which it should be possible to study without getting involved in, for example, the details of quantum

chemistry. We can idealize the laws of physics and simplify them and make them complete, and then study the resulting universe. It is necessary to do two things in order to study the evolution of life within our model universe. First of all, we must define 'life', we must characterize a living organism in a precise fashion. At the same time it should become clear what the complexity of an organism is, and how to distinguish primitive forms of life from advanced forms. Then we must study our universe in the light of the definition. Will an evolutionary process occur? What is the expected time for a certain level of complexity to be reached? Or can we show that life will probably not evolve?

Section 2. Previous work.

Von Neumann devoted much attention to the analysis of fundamental biological questions from a mathematical point of view. (1) He considered a universe consisting of an infinite plane divided into squares. Time is quantized, and at any moment each square is in one of 29 states. The state of a square at any time depends only on its previous states of its four neighboring squares. The universe is homogeneous; the state transitions of all squares are governed by the same law. It is a deterministic universe. Von Neumann showed that a self-reproducing general-purpose computer can exist in his model universe.

A large amount of work on these questions has been done since von Neumann's initial investigations, and a complete bibliography would be quite lengthy. We may mention Moore (1962), Arbib (1966, 1967), and Codd (1968).

The point of departure of all this work has been the identification of 'life' with 'self-reproduction', and this identification has both helped and hindered. It has helped, because it has not allowed fundamental conceptual difficulties to be accomplished. But it has hindered because, in the end, these fundamental difficulties must be faced. At present the problem has evidenced itself as a

question of 'good taste'. As von Neumann remarks, (2) good taste is required in building one's universe. If its elementary parts are assumed to be very powerful, self-reproduction is immediate. Arbib (1966) is an intermediate case.

What is the relation between self-reproduction and life? A man may be sterile, but no one would doubt he is alive. Children are not identical to their parents. Self-reproduction is not exact; if it were, evolution would be impossible. What's more, a crystal reproduces itself, yet we would not consider it to have much life. As von Neumann comments, (3) the matter is the other way around. We can deduce self-reproduction as a property which must be possessed by many living beings, if we ask ourselves what kinds of living beings are likely to be around. Obviously, a species that did not reproduce would die out. Thus, if we ask what kinds of living organisms are likely to evolve, we can draw conclusions concerning self-reproduction.

Section 3 - Simplicity and complexity.

'Complexity' is a concept whose importance and vagueness von Neumann emphasized many times in his lectures. (4) Due to the work of Solomonoff, Kolmogorov, Chaitin, Martin-Lof, Willies, and Loveland, we now understand this concept a great deal better than it was understood while von Neumann worked. Obviously, to understand the evolution of the complexity of living beings from primitive, simple life to today's very complex organisms, we need to make precise a measure of complexity . But it also seems that perhaps a precise concept of complexity will enable us to define 'living organism' in an exact and general fashion. Before suggesting the manner in which this may perhaps be done, we shall review the recent developments which have converted 'simplicity' and 'complexity' into precise concepts.

We start by summarizing Solomonoff's work. (5) Solomonoff proposes the following model of the predicament of the scientis. A scientist is continually observing increasingly larger initial segments of an infinite sequence of 0's and 1's. This is his experimental data. He tries to find computer programs which compute infinite binary sequences which begin with the observed sequence. These are his theories. In order to predict his future observations, he could use any of the theories. But there will always be one theory that predicts that all succeeding observations will be 1's, as well as others that take more account of the previous observations. Which of the infinitely many theories should he use to make the prediction? According to Solomonoff, the principle that the simplest theory is the best should guide him.(6). What is the simplicity of a theory in the present context? It is the size of the computer program. Larger computer programs embody more complex theories, and smaller programs embody simpler theories.

Willis has further studied the above proposal, and also has introduced the idea of a hierarchy of finite approximations to it. To my knowledge, however , the success which predictions made on this basis will have has not been made completely clear.

We must discuss a more technical aspect os Solomonoff's work. He realized that the simplicity of theories, and thus also the predictions, will depend on the computer which one is using. Let us consider only computers whose programs are finite binary sequences, and measure the size of a binary sequence by its lenght. Let us denote by C(T) the complexity of a theory T. By definition, C(T) is the size of the smallest program which makes our computer compute T. Solomonoff showed that there are 'optimal' binary computers C that have the property that for any other binary computer C', $C(T) < C'(T)+d$, for all T. Here d is a constant that depends only on C and C', not on T. Thus, these are the most efficient binary computers, for their programs are shortest. Any two of these optimal binary computers C1 and C2 result in almost the same complexity measure for from $C1(T) < C2(T)+d12$ and $C2(T) < C1(T)+d21$, it follows that the difference

between C1(T) and C2(T) is bounded. The optimal binary computers are transparent theoretically, they are enormously convenient from the technical points of view. What's more, their optimality makes them a very natural choice. (7) Kolmogorov and Chaitin later independetly hit upon the same kind of computer in their search for a suitable computer upon which to base a definition of 'randomness'.

However, the naturalness and technical convenience of the Solomonoff approach should not blind us to the fact is it by no means the only possible one. Chaitin first based his definition of randomness on Turing machines, taking as the complexity measure the number of states in the machine, and he later used bounded-transfer Turing-machines. Although these computers are quite different, they lead to similar definitions of randomness. Later it became clear that using the usual 3-tape-symbol Turing machine and taking its size to be the number of states leads to a complexity measure CT(T) which is asymptotically just a Solomonoff measure CS(T) with its scale changed: CS(T) is asymptotic to 2 CT(T). It appears that people interested in computers may still study other complexity measures, but to apply these concepts of simplicity/complexity it is at present most convenient to use Solomonoff measures.

We now turn to Kolmogorov's and Chaitin's proposed definition of randomness or patternless. Let us consider once more the scientist confronted by experimental data, a long binary sequence. This time he is not interested in predicting future observations, but only in determining if there is a pattern in his observations, if there is a simple theory that explains them. If he found a way of compressing his observations into a short computer program which makes the computer calculate them, he would say that the sequence follows a law, that it has pattern. But if there is no short program, them the sequence has no pattern--it is random. That is to say, the complexity C(S) of a finite binary sequence S is the size of the smallest program which makes the computer calculate it. Those binary sequences S of a given lenght n for n, the random or patternless ones. This is a general formulation of the definition. If we use one of Solomonoff's optimal binary computers, this definition becomes even clearer. Most binary sequences of any given lenght n require programs of about lenght n. These are the patternless

or random sequences. Those binary sequences which can be compressed into programs appreciably shorter than themselves are the sequences which pattern. Chaitin and Martin-Lof have studied the statistical properties of these sequences, and Loveland has compared several variants of the definition.

This completes our summary of the new rigorous meaning which has been given to simplicity/complexity--the complexity of something is the size of the smallest program which computes it or a complete description of it. Simpler things require smaller programs. We have emphasized here the relation between these concepts and the philosophy of the scientific method. In the theory of computing the word 'complexity' is usually applied to the speed of programs or the amount of auxiliary storage they need for scratch-work. These are completely different meanings of complexity. When one speaks of a simple scientific theory, one refers to the fact that few arbitrary choices have been made in specifying the theoretical structure, not to the rapidity whith predictions can be made.

Section 4. What is life?--

Let us once again consider a scientist in a hypothetical situation. He wishes to understant a universe very different from his own which he has been observing. As he observes it, he comes eventually to distinguish certain objects. These are highly interdependent regions of the universe he is observing, so much so, that he comes to view them as wholes.Unlike a gas, which consists of independent particles that do not interact, these regions of the universe are unities, and for this reason he has distinguished them as single entities.

We believe that the most fundamental property of living organisms is the enormous interdependence between their components. A living being is a unity it is much simpler to view it as a whole than as the sum parts. That is to say, if we want to compute a complete description of a region of space-time that is a living being, the program will be smaller in size if the calculation is done all together, than if it is done by independently calculating descriptions of parts of the region and then putting them together. What is the complexity of a living-being, how can we

distinguish primitive life from complex forms? The interpendence in a primitive unicellular organism is far less than in a human being.

A living being is indeed a unity. All the atoms in it cooperate and work together. If Mr. Smith is afraid of missing the train to his office, all his incredibly many molecules, all his organs, all his cells, will be cooperating so that he finishes breakfast quickly and runs to the train station. If you cut the leg of an animal, all of it will cooperate to escape from you, or to attack you and scare you away, in order to protect its leg. Later the wound will heal. How different from what happens if you cut the leg of a table. The whole table will neither come to the defense of its leg, nor will it help it to heal. In the more intelligent living creatures, there is also a very great deal of interdependence between an animal's past experience and its present behavior, that is to say, it learns, its behavior changes with time depending on its experiences. Such enormous interdependence must be a monstrously rare occurrence in a universe, unless it has evolved gradually.

In summary, the case is the whole versus the sum of its parts, if both are equally complex, the parts are indepéndet (do not interact). If the whole is very much simples than the sum of its parts, we have the interdependence that characterizes a living being. (S) Note finally that we have introduced something new into the study of the size of programs (= complexity). Before we compared the sizes of programs that calculate different things. Now we are interested in comparing the sizes of programs that calculate the same thing in different ways. That is to say, the method by which a calculation is done is now of importance to us in the previous section it was not.

Section 5. Numerical examples.--

In this paper, unfortunately, we can only suggest a possible point of departure for a mathematical definition of life. A great amount of work must be done; it is not even clear what is the formal mathematical counterpart to the informal

definition of the previous section. A possibility is sketched here.

Consider a computer C1 which accepts programs P which are binary sequences consisting of a number of subsequences B, C, P(1),...P(K), A.

B, the leftmost subsequence, is a program for breaking the remainder of P into C, P(1),... P(K), and A. B is self-describing; it starts with a binary sequence which results from writing the length of B in base-two notation, doubling each of its bits, and then placing a pair of unequal bits at the right end. Also, B is not allowed to actually see whether any of the remaining bits of P are 0's or 1's, only to separate them into groups. (9).

C is the description of a computer C2. For example, C2 could be one of Solomonoff's optimal binary computers, or a computer which emits the program without processing it.

P(1),... P(K) are programs which are processed by K different copies of the computer C2. R(1) are the resulting outputs. These outputs would be regions of space-time, a space-time which, like von Neumann's, has been cut up into little cubes with a finite number of states.

A is a program for adding together R(1),... R(K) to produce R, a single region of space-time. A merely juxtaposes the intermediate results R(1), ... R(K) (perhaps with some overlapping); it is not allowed to change any of the intermediate results. In the examples below, we shall only compute regions R which are one-dimensional strings of 0's and 1's, so that A need only indicate that R is the concatenation of R(1) to R(K), in that order.

R is the output of the computer C1 produced by processing the program P.

We now define a family of complexity measures C(d, R), the complexity of a

region R of space-time when it is viewed as the sum of independent regions of diameter not greater than d.C(d, R) is the lenght of the shortest program P which makes the computer C1 output R, among all those P such that the intermediate results R(1) to R(k) are all less than or equal to d in diameter. C(d, R) where d equals the diameter of R is to within a bounded difference just the usual Solomonoff complexity measure. But as d decreases, we may be forced to forget any patterns in R that are more than d in diameter, and the complexity C(d, R) increases.

We present below a table with four examples. In each of the four cases, R is a 1-dimensional, region, a binary sequence of lenght n. R1 is a random binary sequence of lenght n ('gas'), R2 consists of n repetitions of 1 ('crystal') . The left half of R3 is a random binary sequence of lenght n/2. The right half of R3 is produced by rotating the left half about R3's midpoint ('bilateral symmetry'). R4 consists of two identical copies of a random binary sequence of lenght n/2 ('twins').

C(d, R) = approx.	R = R1 'gaz'	R = R2 'crystal'	R = 3 'bilateral symmetry'	R = R4 'twins'
d = n	n	log2 n (note 1)	n/2	n/2
d = n/k (k > 1 fixed, n large)	n	k log2 n (notes 1,2)	n-(n/2k) (note 2)	n (note 2)
d = 1	n	n	n	n

Note 1. This supposes that n is represented in base-two notation by a random binary sequence. These values are too high in those rare cases where this is not true.

Note 2. These are conjectured values. We can only show that C(d, R) is approximately less than or equal to these values.

Footnotes

1. See in particular his fifth lecture delivered at the University of Illinois in December of 1949, 'Re-evaluation of the problem of complicated automata-Problems of hierarchy and evolution', and his unfinished The Theory of Automata: Construction, Reproduction, Homogeneity. Both are published posthumously in von Neumann (1966).

2. See pages 76-77 of von Neumann (1966).

3. See page 78 of von Neumann (1966)

4. See especially pages 78-80 of von Neumann (1966)

5. The earliest generally available appearance in print of Solomonoff's ideas of which we are aware is Minsky's summary of them on pages 41-43 of Minsky (1962). A more recent reference is Solomonoff (1964).

6. Solomonoff actually proposes weighing together all the theories into the prediction, giving the simplest theories the largest weight.

7. Solomonoff's approach to the size of programs has been extended in Chaitin (1969b) to the speed of programs.

8. The whole cannot be more complex than the sum of its parts, because one of the ways of looking at it is as the sum of its parts, and this bounds its complexity.

9. The awkwardness of this part of the definition is apparently its chief defect.

Bibliography

Arbib, M. A. (1966). 'Simple self-reproducing universal automata,'Information and Control.

Arbib, M. A. (1967). 'Automata theory and development: Part 1, 'Journal of Theoretical Biology'.

Arbib, M. A. 'Self-reproducing automata--some implications for theoretical biology'.

Biological Sciences Curriculum Study. (1968). Biological Science: Molecules to Man, Moughton Mifflin Co.

Chaitin, G. J. (1966). 'On the lenght of programs for computing finite binary sequences, 'Journal of the Association for Computing Machinery'.

Chaitin, G. J. (1966). 'On the lenght of programs for computing finite binary sequences: Statistical considerations' ibid.

Chaitin, G. J. (1969). 'On the Simplicity and speed of programs for computing infinite sets of natural numbers'

Chaitin, G. J. (1970). 'On the difficulty of computations,' IEEE Transactions on Information Theory.

Codd, E. F. (1968). Cellular Automata. Academic Press.

Kolmogorov, A.N. (1965). 'Three approaches to the definition of the concept' 'amount of information' ',Problemy Peredachi Informatsil.

Kolmogorov, A.N. (1968). 'Logical basis for information theory and probability theory,' IEEE Transactions on Information Theory.

Loveland, D. W. 'A variant of the Kolmogorov concept of complexity,' report 69-4, Math. Dept., Carnegie-Mellon University.

Loveland, D. W. (1969). 'On minimal program complexity measures,' Conference Record of the ACM Symposium on Theory of Computing, May 1969.

Martin-Lof, P. (1966). 'The definition of random sequences,' Information and Control.

Minsky, M. L. (1962). 'Problems of formulation of artificial inteligence', Mathematical Problems in the Biological Sciences, American Math. Society.

Moore, E. F. (1962). 'Machine models of self-reproduction,' ibid.

Von Neumann, J. (1966). Theory of Self-Reproduction Automata, (Edited by A.W. Burks) University of Illinois Press.

Solomonoff, R. J. (1964) 'A Formal theory of inductive inference,' Information and Control.

Willis, D. G. (1969) 'Computational complexity and probability constructions,' Stanford University.