ON MINIMAL-PROGRAM COMPLEXITY MEASURES

by

D. W. Loveland

Report 69-18

March, 1969

This paper is to be presented at the ACM Symposium on Theory of Computing, May 5-6, 1969.

Introduction

Kolmogorov[2,5] introduced two measures of complexity for finite sequences based on the length of minimal descriptive programs, which he employed in a new concept of tf randomness tf . Chaitin,[1] independently, introduced a similar measure for the same purpose. The author has subsequently introduced a third measure of the same type.[4] In this section these measures are defined and comments are made regarding some properties of the measures. The following section contains some results concerning a natural hierarchy for infinite sequences constructed using the third of the above mentioned measures. The structure of hierarchies using the other measures is not considered explicitly although the corresponding questions may receive as interesting, or even more interesting, answers within these other hierarchies.

We shall be concerned with only binary strings (i.e., finite sequences) and (infinite) binary sequences. Sequences are represented by x or y and strings by u^n, v^n, x^n, y^n or z^n where the superscript specifies the length of the string. The initial segment of length n of a sequence x is called the n-prefix of x and notated, in context, by x^n. Also p and q denote strings whose lengths are then denoted by $l(p)$ and $l(q)$. X denotes the set of all strings and N denotes the set of positive integers. Finally, the capital letters A and B denote partial effectively computable functions, or algorithms, from X to X (or from XxN to X in proper context) and should be regarded dually as partial recursive functions and also as "computers" which (perhaps) yield a certain output string when given a certain input string, or program, and optionally a "stopping rule".

The <u>Kolmogorov complexity</u> (or <u>K-complexity</u>) of x^n with respect to algorithm A is given by

$$K_A(x^n) = \min_{A(p)=x^n} l(p)$$

if there exists a string p such that $A(p) = x^n$, otherwise

$$K_A(x^n) = \infty .$$

The (restricted) <u>Kolmogorov conditional complexity</u> (or <u>conditional complexity</u>) of x^n with respect to algorithm A is given by

$$K_A(x^n|n) = \min_{A(p,n)=x^n} l(p)$$

if there exists a string p such that $A(p,n) = x^n$, otherwise

$$K_A(x^n|n) = \infty .$$

The adjective "restricted" is applicable because in the original definition x^n may be conditioned by objects other than the length of x^n.

The interest in the (restricted) conditional complexity is due to work of Martin-Löf.[5] He develops an analytical tool by which he shows that strings of sufficiently high conditional complexity asymptotically have all the standard properties expected of a random sequence.

Let us consider a natural interpretation of each measure. The K-complexity represents the length of the shortest program which when run on "machine" A produces x^n. The conditional complexity represents the length of the shortest program which when run on "machine" A which is told when to halt produces string x^n. More precisely, in the conditional complexity measure algorithm A receives as input a program p and a positive integer n the latter of which may be used by A or p to help define x^n, most obviously by determining the length of x^n. Thus this information of up to Log n(= $\log_2 n$) bits need not be supplied by the program p as is required in the K-complexity measure. The above interpretation, however, does not forbid the use of the information contained in n also for the determination of the pattern of 0's and 1's in the output sequence. The behavior of the conditional complexity supports this interpretation on this point. It is easily seen[4] that for any n an arbitrary string x^n of highest conditional complexity (approximately equal to n) can be "lengthened" by adding 0's to form a string y^r of length r so that the conditional complexity of y^r is a predetermined constant (independent of n). Here r is the integer whose binary expansion is x^n.

The author[4] introduced a third measure of complexity which would reflect the interpretation where the positive integer is simply a halt command and would also have the related property that the complexity of the entire string would never be less than that of an initial part. For convenience in giving the definition, we write $u^X \prec v^3$ if $i \leq 1\ j$ and u^1 is the first i bits of v^{-3}. The <u>uniform complexity</u> of x^n with respect to the algorithm A is given by

$$K_A(x^n;n) = \min_{D(A,x^n)} l(p)$$

where

$$D(A,x^n) = \{p | A(p,i) = x \backslash x^X \prec x^n{}_5 \quad \text{all} \quad i \text{ £ } n\}$$

if $D(A_5 x^n)$ is non-empty, otherwise

$$K_A(x^n;n) = \infty.$$

2

There are several basic properties noted by Kolmogorov which apply to all three measures. We state them in terms of the uniform measure.

1. There exists a universal algorithm B such that for an arbitrary algorithm A and for all x^n

$$J_{\_} \backslash x^n, *TL) \; J\wedge \; JX_a \backslash x^n, Tl) + c$$

where c depends only on A and B.

2. If B_1 and B_2 are two universal algorithms then there exists a constant c such that

$$|K_{g_1}(x^n;n) - K_{g_i}(x^n;n)| \leq c$$

holds for all x^{11}.

3. There exists a constant c such that

$$K(x^n;n) \text{ £ } n + c$$

for all x^n.

4. Less than 2^r strings of length n satisfy

$$K(x^n;n) < r.$$

Alternately, property 4 states that for any constant c $K(x^n;n)$ ^ $n - c$ holds for at least $2^n(l-2"^c)$ strings of length n. Thus almost all strings of length n have their complexity close to n.

Property 1 states that any two universal algorithms differ only by an additive constant. Thus we can suppress our dependence on a particular underlying algorithm by " reading" our results only up to the additive constant. Agreeing to this, we may choose an arbitrary universal algorithm as our underlying algorithm. We write $K(x^n;n)$ for K^$(x^n;n)$ if B is our chosen standard (and likewise for the other measures). By choosing the underlying algorithms for the conditional and uniform complexity measures the same we observe $K(x^n|n)$ <£ $K(x^n;n)$. Also, it is readily seen that $K(x^n;n)$ <^ $K(x^n) + c$ for an appropriate constant c. It can be shown[4] that for infinitely many n

$$K(x^n;n) - K(x^n|n) \sim \text{Log } n$$

holds for some x^n, yet for each constant c there is a constant d such that for all n

$$(x^n|K(x^n;n) \wedge n - c\} \underline{c} (x^n|K(x^n|n) \wedge n - d\}.$$

We now consider some complexity properties of sequences; here x^n represents the n-prefix of sequence x. Clearly, x is a recursive sequence if there exists a constant c such that for infinitely many n, $K(x^n;n) \le c$. For the conditional complexity we have that x is recursive if for some constant c and for all n, $K(x^n|n) \le c$ as shown by Meyer.[4] Conversely, x is recursive implies a constant upper bound on the complexity of x^n for either of the above measures.

We define x to be a random sequence if and only if there exists a constant c such that $K(x^n;n) > n - c$ for infinitely many n. A present weakness of this definition is that it has not been established (to the author's knowledge) that all sequences included by this definition possess the standard properties associated with randomness, such as the law of large numbers or the law of the iterated logarithm. A perhaps more reasonable definition might seem to be achieved by replacing "for infinitely many n" by "for all but finitely many n". Martin-Löf[6] has shown this condition is satisfied by no sequence, whereas it is easily seen that the given definition is satisfied by "almost all" sequences (see the next section). The author conjectures that the standard properties of randomness are satisfied by this set of sequences. There is heuristic support that the definition is a suitable one. Unfortunately, although the relationship between these measures of complexity and the notion of randomness[1,2,5]* is the most exciting aspect of this notion of complexity, it is beyond the scope of this paper. It will be seen in the next section that the definition is conveniently expressed in the notation of the hierarchy considered. To better understand the invariance of the set defined, it would

---

* See the Note on page 14.

be desirable to know if the K-complexity can replace the uniform complexity in the definition without altering the defined set of sequences. Using a theorem stated earlier, we observe we can replace the uniform complexity by the conditional complexity in the definition and obtain the same set of sequences.

There is a further important comparison to be noted between the conditional complexity and the uniform complexity. We shall notate by $\overline{x^n}$ the string of length n obtained by reversing the order of bits of x^n. Thus $\overline{1100} = 0011$. Clearly for an appropriate constant *c*, $|K(x^n|n) - K(\overline{x^n}|n)| < c$ holds for all x^n. (This property and its importance was pointed out to the author by Martin-Löf in a private communication.) This relationship above holds if $\overline{x^n}$ denotes any recursive permutation of x^n. Thus the conditional complexity of x^n may be regarded as bounded by the conditional complexity of its least complex permutation plus the cost of the permutation. This is certainly a natural property for a measure of "information content" to possess when evaluating complete units of patterns.

It is easily observed that the uniform complexity lacks this property. For let x^n be a string with the last bit a 1 such that $K(x^n;n) \sim n$. Let $y^m = 011011100...\overline{x^{11}}$ denote the string of concatenated successive binary integers through $\overline{x^{11}}$. Then $K(y^m;m) = c$, a constant independent of m, yet $K(\overline{y^m};m) \geq K(x^n;n) \sim n \gtrsim \text{Log } m - \text{Log Log } m$ as $x^n < \overline{y^m}$ and the uniform complexity is a monotonic non-decreasing function over the second argument. We conclude it is impossible for a constant to exist which bounds the difference in uniform complexity of x^n and $\overline{x^n}$ for all x^n.

Thus the conditional and uniform complexity measures are incompatible in their characteristics and must serve different purposes. When a string is considered as an isolated unit, it seems desirable to judge the complexity, conditioned by a given length n, by the lowest complexity of any permutation plus the cost of the permutation, and with full use of $^{ft}n^n$ as a partial codification of the distribution of 0's and 1's. This is accomplished by the conditional complexity. If the string is viewed as a part of a nested collection of strings (i.e., of any two members, one is a prefix of the other) and a reflection of the information content of the generating process is sought, the uniform complexity seems more satisfactory. In the latter situation the conditioning value $^{tf}n^n$ is often viewed as imposed upon the process from outside as a stopping rule only. In summary, the uniform complexity reflects the complexity of generating an <u>ordered</u> string <u>uniformly</u>, meaning that any program describing a string outputs an initial portion of the string if terminated earlier. The consideration of the complexity of infinite sequences using a conditioned complexity of their n-prefixes as intermediate readings is one situation where the uniform complexity seems quite suitable.

A hierarchy for infinite sequences

A natural hierarchy of complexity for infinite binary sequences is suggested by each of the previously mentioned complexity measures. We shall consider exclusively the uniform measure. The hierarchy is defined as follows. For every non-decreasing function $f: N \to N$ there is a class C_f of infinite sequences x defined by

$$x \in C_f \Leftrightarrow K(x^n;n) \leq f(n), \quad \forall n \geq n_0(x).$$

That is, C_f consists of those x with all but a finite number of their n-prefixes bounded by f.

We restate in this notation several statements of the preceding section concerning the uniform measure. Any constant function c defines a class C_c which contains only those recursive sequences defined by a program of length less than or equal to c. Also, there exists a constant c such that C_{n+c} contains all infinite sequences. It follows directly from the counting property that if g is an unbounded function and $f(n) = n-g(n)$ is non-decreasing then C_f has measure 0 under the standard product measure with $\mu(0) = \mu(1) = 1/2$ as coordinate measure.

A sequence x is seen to be random as defined previously if and only if there exists a constant c such that $x \notin C_{n-c}$. It follows from the result in the previous paragraph that the set of random sequences has measure 1. An open problem is whether or not there exists a (non-decreasing) function f defining a class consisting of precisely all non-random sequences. Such an f would be of form $f(n) = n-g(n)$ for a very slow growing unbounded function g.

We will note later the existence of an analogous function defining the class of all recursive sequences.

It is informative to locate familiar non-recursive sequences in the hierarchy to serve as landmarks. Perhaps the most obvious question is whether there is a non-trivial class G_f containing the characteristic sequences of all r.e. sets. The classes C^ with f(n) = (l+€)Log n for any € > 0 qualify by the following fact. <u>JEf</u> x <u>i£</u> <u>an</u> <u>r.e.</u> <u>sequence</u> <u>then</u> $K(x^n;n)$ <£ Log n + c <u>for</u> <u>some</u> <u>constant</u> c (dependent on x). This is seen as follows. If x contains an infinite number of l's (otherwise result is immediate), let recursive function h(i) enumerate without repetition the l's of the sequence. We specify an algorithm A and a program p such that $A(p,i) = x^i$, for x^i-<x^n. The algorithm obtains integer j from p, calculates h(l), h(2), ... , h(j), forms the binary string of length i with l's precisely at the values of h(k), k <; *j,* and prints the sequence. The number *j,* chosen such that h(j) <^ n but k > j implies h(k) > n, can be reported by p in length Log j. Thus

$$K_A(x^n;n) \text{ £ Log } n$$

or

$$K(x^n;n) \text{ <£ Log } n + c.$$

for a suitable c.

Certain subsets of r.e. sequences whose distinguishing property is a denseness of l's (or 0's), such as sequences characterizing simple or hypersimple sets, perhaps are contained in smaller (" lower") classes. Rather than report the last value below n of the enumerating function, the program might report the number of 0's (or 1's) less than n. This information in itself is insufficient in general to meet the uniformity condition (but would suffice as a device to lower the bound on the conditional complexity). If one could establish that the spacing between 0's is increasing with n, however, it would suffice to code into the program the length of the run of 1's which includes the n^{th} component, together with the number of 0's preceding this run. This approach can yield a shorter description of a sequence.

Sequences which are not r.e. but have few 0's in them can often be shown to fall in a small class, with $f(n) \sim O(\text{Log } n)$, by having the program explicitly report the location of the occurrences of the 0's. In general, tricks which reduce bounds on complexity are hard to find for sets of sequences specified in usual terms. Bounding the complexity seems difficult even for individual sequences. The science (art?) of seeking economical descriptions (programs) of finite approximations, by initial segments, of non-recursive sequences promises to be quite challenging.

As the r.e. sequences all appear within a rather small class in the sense that $\text{Log } n < n$, we might inquire about slightly "less constructive" sequences, sequences characterizing sets recursive in any r.e. set. We call such sequences ZL_2-sequences. Our results are inconclusive here but we can show the following.

If g J^ <a (total) recursive function such that $\sum_{k=1}^{\infty} 2^{-g(k)} < \infty$, then there exists .a ^-sequence x such that $K(x^n;n) > n - g(n)$ for all $n > n_0$, for some positive integer n_0. For example, (l-fe)Log n is a suitable g for any $e > 0$. Thus $f(n) \sim n$ and we see the complexity of /^-sequences can be quite high. Further information is given by the following theorem.

For any partial recursive function g defined on an infinite domain D and unbounded on D there exists a. A^-sequence x such that $K(x^n;n) > n - g(n)$ infinitely often on D.

The proofs of the two theorems are similar and reasonably direct. Use is made of the fact that for a given k and r the set $\{x^k | K(x^k;k) \leq r\}$ is r.e.

It should be noticed that no /^-sequence asserted to exist, by the above statements is definitely a random sequence although this is, of course, not precluded. The statements do assert that if the Ao-sequences are not random they have complexity characteristics "very close" to that of a random sequence. We conjecture that there is a A2""sec5uence which is a random sequence and, indeed, it would be very interesting should there not be such a /^-sequence. To date the various definitions of randomness based on the explicit concern for recursive (or r.e.) strategies have yielded A>-sequences which are random. (Examples are the random sequences defined by the sequential test concept of Martin-Löf [5] or by attempts to root the von Mises notion of randomness in recursive function theory.)

Finally, we consider several properties of the hierarchy which concerns the smallest classes C_f containing non-recursive sequences.

Every non-decreasing unbounded recursive function f *defines a complexity class* C_f *with a continuum of members*.

This is clearly not true if f is a bounded function as such classes include only recursive sequences. The proof uses the existence of an algorithm which "places" the i^{th} bit of a string of length n at that integer where the i^{th} jump of f occurs, and fills in with zeros.

There exists a non-recursive sequence y *such that* $y \in C_f$ *for every non-decreasing unbounded recursive* f.

This asserts the existence of a sequence whose uniform complexity grows slower than any unbounded monotonic recursive function. The proof uses a Rado-type ("Busy Beaver") function.[7]

There exists a non-decreasing unbounded function E *such that* $x \in C_E \Leftrightarrow x$ *is recursive*.

Recall that any non-decreasing unbounded function defines a class which contains all the recursive functions and that no constant function can define such a class. This theorem asserts that the set of recursive sequences itself forms a class. Further, the theorem states that no non-recursive sequence can have its uniform complexity measure grow strictly slower than a certain minimum growth rate with respect to n. This is also a "gap" theorem stating that there exist non-decreasing functions f and g such that, say, $(f(n))^2 < g(n)$, all n, yet $C_f = C_g$. The definition of E uses the fact that for each n there is an integer s such that if any two programs of length at most n calculate the same sequence of length s, they calculate the same infinite sequence.

An open problem is to characterize E. It seems reasonable to expect that E can be expressed in terms of the Rado "Busy Beaver" function. If this is true, then E is a tight bound in that sequences exist with complexity growth rate about that of E. Otherwise one might ask if there is a (non-trivially) faster growing function than E defining the same class as E. Mentioned before is the question of the existence of a function f which separates the random sequences as E separates the recursive sequences. Finally, it should be recalled that the definition of random sequence adopted here itself requires formal justification.

Note: Martin-Löf has pointed out to the author that sequences random in the sense given here are random in the sense of his paper.[5] Thus it follows that all the standard properties of randomness hold for the random sequences defined above. This definition (expressed in terms of the conditional complexity) apparently was first proposed informally by Martin-Löf in 1965 and recently used by Kolmogorov in a recent set of lectures given in Moscow.

References

1. Chaitin, G. J., ff On the length of programs for computing finite binary sequences'1 , J. ACM 13, 4 (Oct. 1966), 547-569.

2. Kolmogorov, A. N., u Three approaches to the definition of the concept fquantity of information1ff , Problemy Peredachi Informatsii 1 (1965), 3-11, (in Russian).

3. Loveland, D. W., n The Kleene hierarchy classification of recursively random sequences11 , Trans. AMS 125, 3 (Dec. 1966), 497-510.

4. ________________9 M A variant of the Kolmogorov concept of complexity11 , (forthcoming).

5. Martin-Löf, P., M The definition of random sequences,u Infor. and Control 9, (1965), 602-619.

6. Martin-Löf, P., M O kolebanii složnosti beskonečnykh dvoicnykh posledovatel fnostejf! , (unpublished).

7. Rado, T., M On a simple source for non-computable functionsf! Proc. Symposium of Math. Theory of Automata, MRI Symposia Series, XII, Polytechnic Press, Brooklyn, (1962), 75-81.