# A COMPLEXITY THEORETIC APPROACH TO RANDOMNESS

Michael Sipser*

M.I.T.
Mathematics Department
Cambridge, MA 02139

Abstract: We study a time bounded variant of Kolmogorov complexity. This notion, together with universal hashing, can be used to show that problems solvable probabilistically in polynomial time are all within the second level of the polynomial time hierarchy. We also discuss applications to the theory of probabilistic constructions.

## I. Introduction

Complexity theory is a natural setting in which to study randomness. In part, complexity theory, recursive function theory, and set theory are theories of definability. It is no coincidence that notions of randomness play a role in such theories because a random object is in a certain sense the opposite of a definable one.

A number of previous results establish connections between randomness and complexity. There are a variety of probabilistic algorithms [M,R1,R2,SS,K]. A few results, such as Adleman's [Ad1], directly concern complexity classes. Paul and Seiferas [PS] have used Kolmogorov complexity to simplify counting arguments used in lower bound arguments. Recently, pursuing a line suggested by Shamir [Sh], Blum and Micali [BM] and Yao [Y] give definitions of the notion of a random set of strings based upon polynomial circuit predictors and polynomial tests. This exciting direction is further explored in Blum, Blum and Schub [BBS] and Goldwasser, Micali, and Tong [GMT].

In this paper we present a complexity theoretic definition of randomness derived from Kolmogorov complexity. The Komogorov complexity of a finite binary string is the length of its shortest describing program. Intuitively, it is a measure of the amount of information the string contains. A string is random if it cannot be compressed, i.e., its shortest program is as long as the string itself. This idea originally appeared in [Ko] and was subsequently independently discovered by Chaitin [Ch1] and Solomonoff [Sol]. Significant other work in this area includes [Ch2,L,S, KS]. Gewirtz's thesis [Ge] contains an excellent introduction.

We modify the Kolmogorov complexity by considering only programs that run for a limited amount of time. This is similar to a notion proposed some time ago by Levin [L] and subsequently by Adleman [Ad2] where the log of the running time is added to the length. Daley [D] and Keri-Ko [Ke] give definitions much like ours and prove several results of a recursion-theoretic nature; Peterson [P] also considers related notions.

As an interesting application we show that the class BPP [G] is contained within the polynomial time hierarchy [MSt,St]. The intuition is fairly simple: since BPP = P relative to a

*Supported by NSF grant MCS-81-05555 and Air Force grant AFOSR-82-0326.

random oracle with probability one [BG], it follows that BPP = P relative to a Kolmogorov-Chaitin-random oracle. A combinatorial argument involving Carter-Wegman universal hash functions [CW] shows that it is sufficient to use oracles that are random with respect to the time bounded complexity. Such oracles exist within the polynomial time hierarchy. Peter Gacs subsequently distilled the combinatorial essence of this argument, obviating the need for random strings, and proving a stronger theorem. (See Section V.)

II. The Complexity Measure

Let $A_1, A_2, \ldots$ be a standard enumeration of all Turing machines and let $A_i$ also denote the partial function computed by $A_i$. Let $\Sigma = \{0,1\}$ and $N = \{1,2,\ldots\}$. For all $s \in \Sigma^*$ let $U(0^i|s) = A_i(s)$. The partial function $U$ is computed by the standard universal Turing machine.

Definition: Let $t: N \to N$, $s \in \Sigma^*$ and $A \in \Sigma^*$ or $A \subseteq \Sigma^*$

1) $K^t(s) = \min\{|p|: U(p) = s$, halting within $t(|s|)$ steps$\}$

2) $K^t(s|A) = \min\{|p|: U^A(p) = s$, halting within $t(|s|)$ steps$\}$ Here $A$ is given as an oracle to $U$.

Remarks: There are no restrictions on the function $t$. For sufficiently fast growing (non-recursive) $t$ the time bound is inoperative and $K^t(\ )$ becomes equal to $K(\ )$.

Definition: For $t: N \to N$, $s \in \Sigma^*$ or $A \subseteq \Sigma^*$, let $KD^t(s|A) = \min\{|p|: \forall v \in \Sigma^*[U^{v,A}(p)=1$ iff $v=s]$ and $U^{s,A}(p)$ halts within $t(|p|)$ steps The notation $U^{v,A}$ means that both $v$ and $A$ are given as oracles.

The intuition is that while $K(s)$ is the length of the shortest program generating $s$, $KD(s)$ is the length of the shortest program which accepts only $s$. In pure Kolmogorov complexity these two measures differ by only an additive constant. In the time restricted complexity they appear to be quite different, though we can prove the following relationship.

Theorem 1: Let $p$, $q$ denote polynomials and NP an oracle for a complete set.

1) $\forall p\, \exists q \quad KD^q(s) \leq K^p(s) + O(1)$

2) $\forall p\, \exists q \quad K^q(s|NP) \leq KD^p(s) + O(1)$

Proof: 1) Immediate

2) Use the NP-oracle repeatedly to determine successive bits of $s$.

III. Coding Lemma

The usefulness of Kolmogorov complexity in simplifying counting arguments stems from the following relationship:

Theorem: If $A \subseteq \Sigma^n$ then $\forall s \in A$:

$$K(s|A) \leq \log |A| + O(1)$$

The short program for $s$ is simply its index in a lexicographic ordering of $A$.

Thus if an easily definable set $A$ is sparse, i.e., has only a small fraction of $\Sigma^n$, then its members are all nonrandom, given some side information depending only upon $A$. The principal theorem of this paper is an analogous theorem about $KD^c$. If a sparse set $A$ is definable by a polynomial size circuit, then all of its members have short, fast descriptions given some side information depending only upon $A$.

Theorem 2: $\forall c\, \exists d$ if $A \subseteq \Sigma^n$ and is accepted by a circuit of size $n^c$, then $\forall s \in A$:

$$KD^d(s|A,i_A) \leq \log |A| + \log\log |A| + O(1)$$

where $i_A$ depends only upon $A$.

The proof can be viewed as an application of Carter-Wegman universal hashing [CW]. A randomly chosen hash function from a certain distribution is likely to isolate $s$ from the remainder of $A$. Its description is its image under the hash function.

Let $k = |A|$, $m = 1 + \lceil \log k \rceil$, and for $x \in \Sigma^n$ let $x_i$ denote the $i$-th bit. Let $h: \Sigma^n \to \Sigma^m$ be a linear transformation given by a randomly chosen $m \times n$ binary matrix $R = \{r_{ij}\}$, i.e. for $x \in \Sigma^r$, $Mx$ is a string $y \in \Sigma^m$ where $y_i = (\sum_j r_{ij} \wedge x_j) \bmod 2$.

Lemma: For each $i \leq n$ and distinct $x,y \in \Sigma^n$, $Pr[(h(x))_i = (h(y))_i] = 1/2$

Lemma: For distinct $x,y \in \Sigma^n$, $Pr[h(x) = h(y)] = 2^{-m}$

Lemma: For any $x \in \Sigma^n$, $Pr[\exists y \in A\ (y \neq x \text{ and } h(x) = h(y))] \leq k \cdot 2^{-m} \leq 1/2$

Lemma: Let $H$ be a collection $m$ randomly selected functions $h$ as above.

$$Pr[\forall h \in H\ \exists y \in A\ (y \neq x \text{ and } h(x) = h(y))] \leq 2^{-m}$$

Lemma: $Pr[\exists x \in A\ \forall h \in H\ \exists y \in A\ (y \neq x \text{ and } h(x) = h(y))] \leq 1/2$

Definition: Let $h: \Sigma^n \to \Sigma^m$ and $H$ be a collection of such functions. Let $A,B \subseteq \Sigma^n$ and $x \in \Sigma^n$. Say $h$ <u>separates</u> <u>$x$</u> <u>within</u> <u>$A$</u> if for every $y$ in $A$, different from $x$, $h(x) \neq h(y)$. Say $h$ separates $B$ within $A$ if it separates each $x \in B$ within $A$. Say $h$ separates $B$ within $A$ if for each $x \in B$ some $h \in H$ separates $x$ within $A$. Recapitulating the preceeding lemmas we have:

Coding Lemma: $A \subseteq^n$, where $k = A$, $m = 1 + [\log k]$. If $H$ is a collection of $m$ randomly chosen linear transformations $\Sigma^n \to \Sigma^m$ then:

$$Pr[H \text{ separates } A \text{ within } A] \geq 1/2.$$

Proof of Theorem 2: Let $H$ be a collection of functions as given by the coding lemma. Let $i_A$ be an encoding of $H$. For each $s \in A$ there is some $h_i \in H$ that separates it within $A$. The description of $s$ is then $b_i h_i(s)$ where $b_i$ is the binary representation of $i$ padded out to length $\lceil \log m \rceil$. Thus $KD^p(s|A,i_A) \leq \log m + m + 0(1) = \log|A| + \log\log|A| + 0(1)$. The polynomial $p$ is large enough to permit the algorithm to check that $s \in A$ and $h_i(s)$ is correct.

On information theoretic grounds we speculate that it should be possible to remove the additive $\log\log|A|$.

Theorem 3: $\forall c\ \exists d$ if $A \subseteq \Sigma^n$ is accepted by a circuit of size $n^c$ then there is a string $i_A$ such that then for each $s \in A$:

1) $K^d(s|A,i_A,NP) \leq \log|A| + \log\log|A| + 0(1)$
2) $K^d(s|A,\Sigma_2) \leq \log|A| + \log\log|A| + 0(1)$

Proof: 1) Immediate from Theorems 1 and 2.

2) The algorithm computes the lexicographically first $H$ separating $A$ within $A$ instead of obtaining one for free. This may be done with a $\Sigma_2$ oracle since testing that $H$ separates $A$ within $A$ is a $\Pi_1$ predicate.

## IV. Random Strings and the Class BPP

Definition: String $s$ is said to be <u>$(p,c)$-random</u> if $K^p(s) \geq |s| - c$, and <u>$(p,c)$-$\Sigma_i$-random</u> if $K^p(s|\Sigma_i) \geq |s| - c$. Often we will write only $\Sigma_i$-random when the time bound $p$ is implicit and $c$ is 0.

Theorem 4: For a fixed polynomial $p$, $\{(s,c);\ s \text{ is } (p,c)\text{-random}\} \in CO\text{-}NP$.

This can be generalized to show that the $(p,c)$-$\Sigma_i$-random strings form a language in $\Pi_{i+1}$. Note that $(p,0)$-random strings exist.

The classes R [Adl] and its symmetric version BPP [G] consist of problems which can be solved with a high probability.

Definition: A language $A$ is in BPP iff there is a polynomial time predicate $S_A$, a polynomial $p_A$, and an error bound $\varepsilon$, $0 < \varepsilon < 1/2$ such that $A = \{x:\mu\{y: |y| = p_A(|x|) \text{ and } S_A(x,y)\} > \varepsilon\} = \{x: \mu\{y; |y| = p_A(|x|) \text{ and } S_A(x,y)\} > 1-\varepsilon\}$ where for $Y \subseteq \Sigma^n$, $\mu Y = |Y|/2^n$.

Theorem 5: $BPP \subseteq \Sigma_4 \cap \Pi_4$ .

Proof: Let $B \in BPP$ . There is a probabilistic algorithm $M$ for $B$ which, on input $x$ of length $n$ , chooses $n^2$ $y$'s at random and accepts if the majority of them satisfy $S_B(x,y)$. $M$ will fail to give the correct answer with probability less than $2^{-2n}$ . Here, as elsewhere in the coming arguments, we assume wlog that $n$ is large. Thus, calling such a sequence of $n^2$ $y$'s an experiment $e$ of length $m=n^2 p_B(n)$ the set $F_x = \{e \in \Sigma^m$: $M$ fails on input $x$ when using random experiment $e\}$ has cardinality at most $2^m/2^{2n} = 2^{m-2n}$ . Therefore, by theorem 3, if $e \in F_x$ , for some polynomial $b$ ,

$$K^b(e|F_x,\Sigma_2) \leq m - 2n + \log(m-2n) + 0(1)$$

Since membership in $F_x$ may be quickly computed knowing $x$ and whether $x \in A$ , by adding $x$ to the description of $e$ we have:

$$K^b(e|\Sigma_2) \leq n + m-2n + \log(m-2n) + 0(1)$$
$$< m \quad \text{for large } n .$$

Hence, the experiments in $F_x$ are all $\Sigma_2$-nonrandom. The language consisting of those strings accepted by $M$ when using a $\Sigma_2$-random experiment is equal to $B$ except perhaps at finitely many points. This language is in $\Sigma_4$ and since $BPP$ is closed under complement, $B \subseteq \Sigma_4 \cap \Pi_4$ .

A slightly more involved argument shows $BPP \subseteq \Sigma_3 \cap \Pi_3$ by appealing to theorem 3 part 2. The next section gives a further improvement.

V. BPP According to Gacs

Peter Gacs improved the author's original theorem by showing that $BPP \subseteq \Sigma_2 \cap \Pi_2$ with a somewhat simpler proof that relies only on the coding lemma and not on the notion of a random string.

Theorem 6: $BPP \subseteq \Sigma_2 \cap \Pi_2$

Proof: Let $B \in BPP$ be accepted by a probabilistic algorithm with error probability $< 2^{-n}$ on inputs of length $n$ , which uses $m = n^k$ random bits.

Let $E_x \subset \Sigma^m$ be the collection of random inputs on which $M$ rejects $x$ . For $x \in B$ , $|E_x| \leq 2^{m-n}$ . Letting $\ell = 1+m-n$ , the coding lemma states that there is a collection $H$ of $\ell$ linear transformations $\Sigma^m \to \Sigma^\ell$ separating $E_x$ within $E_x$ . If $x \notin B$ , $|E_x| > 2^{m-1}$ and by the pidgeon hole principle no such collection exists. Hence $x \in B$ iff such an $H$ exists. Expanding this, $x \in B$ iff

$$(\exists H)(\forall a \in E_x)(\exists h \in H)(\forall e' \in E_x)[e \neq e' \wedge h(e) \neq h(e')]$$

The second existential quantifier has polynomial range and is eliminable. Hence $A \in \Sigma_2 \cap \Pi_2$ .

Definition: Let $R_2 = R^{NP}$ be the collection of languages accepted in random polynomial time [Ad1] with an oracle for an NP-complete set.

Corollary: $BPP \subseteq R_2 \cap Co\text{-}R_2$

Proof: The coding lemma states that most $H$ work.

Steve Mahaney also later independently discovered the proof of theorem 6.

VI Probabilistic Constructions

The probabilistic method is a powerful one for proving the existence of certain objects. In the presence of a "certified random" string, a probabilistic construction can be made deterministic. We formalize this notion and measure the degree of randomness required to carry out various constructions.

Definition: Let $A \subseteq \Sigma^* \times \Sigma^*$ . A probabilistic construction for members in $A$ is a polynomial time computable function $f: \Sigma^* \times \Sigma^* \to \Sigma^*$ and a polynomial $p$ such that for each $s \in \Sigma^*$ of length $n$ $\mu\{r \in \Sigma^{p(n)}$: $(s,f(s,r)) \in A\} \geq 1/2$ . Here, $r$ may be thought of as the random input to the probabilistic algorithm. The construction is deterministic if for each $s$ and $r$ , $(s,f(s,r)) \in A$ . Typically, $s$ might represent the length of a desired object expressed in unary.

A $\Sigma_i$- or $\Pi_i$-probabilistic construction is one where the set $\{(s,f(s,r)): s,r \in \Sigma^*, |r| = p(|s|)\} \cap A$ is in the corresponding level of the polynomial time hierarchy.

Examples: 1) There is a $\Pi_1$-probabilistic construction for universal sequences. [AKLLR]

2) Pippenger [Pi] gives a $\Pi_1$-probabilistic construction for superconcentrations of size $29n + O(1)$. Gabber and Galil [GG] give a deterministic construction for $504n + O(1)$ size superconcentrators. Note that the superconcentrator property is $\Pi_2$ even though the construction, by building up from the more simply described expanding graphs, is only $\Pi_1$.

3) There are $\Pi_0$-probabilistic constructions for the restrictions employed in the lower bound on circuits given in [FSS] and [S2]. Pippenger [personal communication] points out that $\Pi_0$-constructions can be excecuted with certainty by an algorithm that runs in expected polynomial time.

4) Pippenger and Yao [PY] give a $\Pi_2$-probabilistic construction for rearrangable networks of size $O(n(\log n)^{1/k})$ for depth $k$.

Theorem: If there is a $\Sigma_i$-probabilistic construction for members of a set $A$ then there is a deterministic construction for members of $A$ relative to an oracle for $\Sigma_{i+2}$-random strings.

Acknowledgements

I am very grateful to a number of people who helped to clarify these ideas. I am especially indebted to Peter Gacs for his graciously offered, penetrating insights. Benny Chor and Michael Ben-Or made a number of important suggestions to the proof of the coding lemma. Larry Carter, Mark Wegman, and Ron Rivest pointed out the connections with hashing functions. The idea of looking at probabilisitc constructions stemmed from a conversation with Dick Lipton. I also wish to thank my collegues at the IBM Yorktown Research Center who provided a stimulating research environment last summer.

References

[Ad1] L. Adleman, "Two theorems on random polynomial time," 19th FOCS, 1978, 75-83.

[Ad2] L. Adleman, "Time, space and randomness," MIT/LCS/TM-131 , 1979.

[AKLLR] R. Aleliunas, R.M. Karp, R. Lipton, L. Lovasz, and C. Rackoff, "Random walks, universal traversal sequences, and complexity of mazeproblems," 20th FOCS, 218-223, 1979.

[BBS] L. Blum, M. Blum, M. Shub, "A simple, secure pseudo-random number generator," CRYPTO 1982.

[BG] C.H. Bennett and J. Gill, "Relative to a random oracle $A$, $P^A \neq NP^A \neq co\text{-}NP^A$ with probability one," SKOMP, to appear.

[Ch1] G.J. Chaitin, "On the length of programs for computing finite binary sequences," JACM 13 , 1966, 547-569 and JACM 16, 1969, 145-159.

[Ch2] G.J. Chaitin, "A theory of program size formally identical to information theory," JACM 22 , 329-340.

[CW] J.L. Carter and M.N. Wegman, "Universal classes of hash function," JCSS 18, no 2, 143-154, 1979.

[D] R.P. Daley, "On the inference of optimal descriptions," TCS 4, 301-319, 1977.

[G] J. Gill, "Complexity of probabilistic Turing machines," SIAM J. of Computing 6, 675-695.

[GG] O. Gabber and Z. Galil, "Explicit constructions of linear size superconcentrators," 20th FOCS, 364-370, 1979.

[GMT] S. Goldwasser, S. Micali, and P. Tong, "Why and how to establish a private code on a public network," 23rd FOCS, 134-144, 1982.

[FSS] M. Furst, J.B. Saxe, M. Sipser, "Parity, circuits, and the polynomial time hierarchy," 22nd FOCS, 260-270, 1981.

[K] R. Karp, "Probabilistic analysis of partitioning algorithms for the travelling-salesman problem in the plan," Math. Op. Res. 2 , 209-224.

[KS] H. Katseff and M. Sipser, "Several results in program size and complexity," 18th FOCS, 1977, 82-89.

[Ke] K. Ko "Resource-bounded program-size complexity and pseudo-random sequences," to appear.

[Ko] A.N. Kolmogorov, "Three approaches for defining the concept of information quantity," Prob. Inform. Trans. 1, 1-7, 1965.

[L] L. Levin, "Universal sequential search problems," Prob. Info. Trans. 9, 1973, 265-266.

[MM] A.R. Meyer and E.M. McCreight, "Computationally complex and pseudo-random zero-one valued functions," in Theory of Machines and Computations, Z. Kohavi and A. Paz, eds. Academic Press, 19-42, 1971.

[M] G. Miller, "Rieman's hypothesis and tests for primality," 7th SIGACT, 1975, 234-239.

[P] G.L. Peterson, "Succinct representations, random strings, and complexity classes," 21st FOCS, 86-95, 1980.

[Pi] N. Pippenger, "Superconcentrators," SIAM J. Comput. 6, 298-304, 1972.

[PY] N. Pippenger and A. Yao, "Rearrangeable networks with limited depth," SIAM J. Alg. and Disc. Meth. 3, 411-417, 1982.

[PSS] W. Paul, J. Seiferas, and J. Simon, "An information-theoretic approach to time bounds for on-line-computations," 12th STOC, 357-367, 1980.

[R1] M. Rabin, "Probabilistic algorithms in finite fields," MIT/CCS/TR-213.

[R2] M. Rabin, "N-process synchronization by 4 log N-valued shared variable," 21st FOCS, 1980, 4-7-410.

[S] C. Schnorr, "the network complexity and Turing Machine complexity of finite functions," Acts Informa. 7, 1976, 95-107.

[Sh] A. Shamir, "On the generation of cryptographically strong pseudo-random sequences," 8th ICALP, 545-550, 1981.

[Si] M. Sipser, "Borel sets and circuit complexith," 15th STOC, 1983.

[Si2] M. Sipser, "Three approaches to a definition of finite state randomness," unpublished manuscript.

[SS] R. Solovay and V. Strassen, "A fast Monte-Carol test for primality," SIAM J. on Comput. 6, 1977, 84-85.

[St] L. Stockmeyer, "The polynomial time hierarchy," TCS 3, no.1, 1-22, 1976.