

A NOTE ON MONTE CARLO PRIMALITY TESTS AND ALGORITHMIC INFORMATION THEORY

Communications on Pure and Applied
Mathematics 31 (1978), pp. 521–527

Gregory J. Chaitin

IBM Thomas J. Watson Research Center

Jacob T. Schwartz¹

Courant Institute of Mathematical Sciences

Abstract

Solovay and Strassen, and Miller and Rabin have discovered fast algorithms for testing primality which use coin-flipping and whose con-

clusions are only probably correct. On the other hand, algorithmic information theory provides a precise mathematical definition of the notion of random or patternless sequence. In this paper we shall describe conditions under which if the sequence of coin tosses in the Solovay–Strassen and Miller–Rabin algorithms is replaced by a sequence of heads and tails that is of maximal algorithmic information content, i.e., has maximal algorithmic randomness, then one obtains an error-free test for primality. These results are only of theoretical interest, since it is a manifestation of the Gödel incompleteness phenomenon that it is impossible to “certify” a sequence to be random by means of a proof, even though most sequences have this property. Thus by using certified random sequences one can in principle, but not in practice, convert probabilistic tests for primality into deterministic ones.

1. Algorithmic Information Theory

To prepare for discussion of the Solovay–Strassen and Miller–Rabin algorithms, we first summarize some of the basic concepts of algorithmic information theory [1]–[4].²

Consider a universal Turing machine U whose programs are in binary. By “universal” we mean that for any other Turing machine M whose programs p are in binary there is a prefix μ such that $U(\mu p)$ always carries out the same computation as $M(p)$.

$I(X)$, the algorithmic information content of X , is defined to be the size in bits of the smallest programs for U to compute X . There is absolutely no restriction on the running time or storage space used by these programs. If X is a finite object such as a natural number or bit string, this includes the proviso that U halt after printing X . If X is an infinite object such as a set of natural numbers or of bit strings, then of course U does not halt. Sets, as opposed to sequences,

¹The second author has been supported by US DOE, Contract EY-76-C-02-3077*000. We wish to thank John Gill III and Charles Bennett for helpful discussions. Reproduction in whole or in part is permitted for any purpose of the United States Government.

²We could equally well have used in this paper the newer formalism of [7], in which programs are “self-delimiting.”

may have their members printed in arbitrary order. X can also be an r.e. function f ; in that case U prints the graph of f , i.e., the set of all ordered pairs $\langle x, f(x) \rangle$. Note that variations in the definition of U give rise to at most $O(1)$ differences in the resulting I , by the definition of universality.

It is easy to show (cf. [1]–[4]) that the maximum value of $I(s)$ taken over all n -bit strings s is equal to $n + O(1)$, and that an overwhelming majority of the s of length n have $I(s)$ very close to n . Such s have maximum information content or “entropy” and are highly random, patternless, incompressible, and typical. They are said to be “algorithmically random.” The greater the difference between $I(s)$ and the length of s , the less random s is, the more atypical it is, and the more pattern it has. It is convenient to say that “ s is c -random” if $I(s) \geq n - c$, where n is the length of s . Less than 2^{n-c} n -bit strings are not c -random. As for natural numbers, $I(n) \leq \log_2 n + O(1)$ and most n have $I(n)$ very close to $\log_2 n$. Strangely enough, though most strings are random, it is impossible to prove that specific strings have this property! For an explanation of this paradox see [1]–[6].

2. The Solovay–Strassen and Miller–Rabin Algorithms [8]–[10]

The general form of these algorithms is as follows: To test whether n is prime, take k natural numbers uniformly distributed between 1 and $n - 1$, inclusive, and for each one i check whether the predicate $W(i, n)$ holds. (Read “ i is a witness of n ’s compositeness.”) If so, n is composite. If not, n is prime with probability $1 - 2^{-k}$. This is because, as proved in [8]–[10], at least half the i ’s from 1 to $n - 1$ are in fact witnesses of n ’s compositeness, if n is indeed composite, and none of them are if n is prime. The definition of W is different in the Solovay–Strassen and the Miller–Rabin algorithms, but both algorithms are of this form, where $W(i, n)$ can be computed quickly, i.e., the running time of a program which computes $W(i, n)$ is bounded by a polynomial in the size of n , in other words, by a polynomial in $\log n$.

We shall now show that if sufficiently long random sequences are

supplied, the probabilistic reasoning of [8]–[10] can be converted into a rigorous proof of primality. To state our precise results, we need to make the following definition:

Definition 1. Let s be an m -bit sequence, and let J be an integer. Find the smallest integer k such that $(J - 1)^{k+1} > 2^m - 1$, and (by converting s to base $J - 1$ representation) find the unique sequence $d_k d_{k-1} \dots d_0$ of base $J - 1$ digits such that

$$\sum_{0 \leq i \leq k} d_i (J - 1)^i = s.$$

Calculate

$$Z(s, J) = \neg W(1 + d_0, J) \& \dots \& \neg W(1 + d_{k-1}, J),$$

where $W(i, n)$ is as above. Then we say that J passes the s -test for primality if and only if $Z(s, J)$ is true.

Lemma 1. Let m , J , k , and Z be as in Definition 1. Then the number of m -bit sequences s for which $Z(s, J)$ is true is 2^m if J is prime, but is not more than 2^{m+1-k} if J is not a prime.

Proof. If J is prime, then $W(i, J)$ is always false, so our assertion is trivial. Now suppose J is composite, so that $W(i, J)$ is true for at least $(J - 1)/2$ values of i . Since $d_k (J - 1)^k \leq 2^m$, i.e.,

$$0 \leq d_k \leq 2^m (J - 1)^{-k},$$

it follows that the number of s satisfying $Z(s, J)$ is at most

$$\left[2^m (J - 1)^{-k} + 1 \right] \left[(J - 1)/2 \right]^k = \left[2^m + (J - 1)^k \right] 2^{-k},$$

or 2^{m+1-k} since $(J - 1)^k \leq 2^m$.

It is now easy to prove our results:

Theorem 1. For all sufficiently large c , if s is any c -random $j(j + 2c)$ -bit sequence and J any integer whose binary representation is j bits long, then $Z(s, J)$ if and only if J is a prime.

Theorem 2. For all sufficiently large c , if s is any c -random $2j(i + c)$ -bit sequence and J any integer whose binary representation is j bits long and whose information content $I(J)$ is not more than i , then $Z(s, J)$ if and only if J is a prime.

Proof of Theorem 1. Denote the cardinality of a set e by writing $|e|$, and let $\sigma(m)$ be the set of all m -bit sequences. Let J be a non-prime integer j bits long. By Lemma 1,

$$|\{s \in \sigma(j(j+2c)) : Z(s, J)\}| \leq 2^{j(j+2c)+1-(j+2c)}.$$

Hence

$$\begin{aligned} |\{s \in \sigma(j(j+2c)) : \exists J \in \sigma(j)[J \text{ is composite} \ \& \ Z(s, J)]\}| \\ \leq 2^{j(j+2c)+1-2c}. \end{aligned} \quad (1)$$

Since any member s of the set S appearing in (1) can be calculated uniquely if we are given c and the ordinal number of the position of s in S expressed as a $j(j+2c)+1-2c$ bit string, it follows that

$$I(s) \leq j(j+2c)+1-2c+2I(c)+O(1) \leq j(j+2c)-2c+O(\log c).$$

(The coefficient 2 in the term $2I(c)$ is present because when two strings are encoded into a single one by concatenating them, it is necessary to add information indicating where to separate them. The most straightforward technique for providing punctuation doubles the length of the shorter string.) Hence if c is sufficiently large, no c -random $j(j+2c)$ bit string can belong to S .

Proof of Theorem 2. Arguing as in the proof of Theorem 1, let J be a non-prime integer j bits long such that $I(J) \leq i$. By Lemma 1,

$$|\{s \in \sigma(2j(i+c)) : Z(s, J)\}| \leq 2^{2j(i+c)+1-2(i+c)}. \quad (1')$$

Since any member s of the set S' appearing in (1') can be calculated uniquely if we are given J and the ordinal number of the position of s in S' expressed as a $2j(i+c)+1-2(i+c)$ bit string, it follows that

$$I(s) \leq 2j(i+c)+1-2(i+c)+2I(J)+O(1) \leq 2j(i+c)-2c+O(1).$$

(The coefficient 2 in the term $2I(J)$ is present for the same reason as in the proof of Theorem 1.) Hence if c is sufficiently large, no c -random $2j(i+c)$ bit sequence can belong to S' .

3. Applications of the Foregoing Results

Let s be a probabilistically determined sequence in which 0's and 1's appear independently with probabilities $\alpha, 1 - \alpha$, where $0 < \alpha < 1$. Group s into successive pairs of bits, and then drop all 00 and 11 pairs and convert each 01 (respectively 10) pair into a 0 (respectively, a 1). This gives a sequence s' in which 0's and 1's appear independently with exactly equal probabilities. If s' is n bits long, then the probability that $I(s') < n - c$ is less than 2^{-c} ; thus c -random sequences can be derived easily from probabilistic experiments. Theorem 2 gives the number of potential witnesses of compositeness which must be checked to ensure that primality for numbers of special form is determined correctly with high probability (or with certainty, if some oracle gave us a long bit string known to satisfy the randomness criterion of algorithmic information theory). Mersenne numbers $N = 2^n - 1$ only require checking $O(\log n) = O(\log \log N)$ potential witnesses. Fermat numbers

$$N = 2^{2^n} + 1$$

only require checking $O(\log n) = O(\log \log \log N)$ potential witnesses. Eisenstein–Bell³ numbers

$$N = 2^{2^{2^{\dots (n \text{ 2's altogether})}}} + 1$$

only require checking $O(\log n) = O(\log^k N)$ (for any k) potential witnesses. A number of the form $10^n + k$ only requires checking $O(\log n) + O(\log k)$ potential witnesses.

Concerning Theorem 1 it is worthwhile to remark the following: Using the extended Riemann Hypothesis, Miller was able to show that if n is composite, then the first natural number that is a witness of n 's compositeness (under the Miller–Rabin version of the predicate W)

³Quotation from Bell [11]: “F. M. G. Eisenstein (1823–1852), a first-rate arithmetician, stated (1844) as a problem that there are an infinity of primes in the sequence

$$2^2 + 1, 2^{2^2} + 1, 2^{2^{2^2}} + 1, \dots$$

Doubtless he had a proof. This looks like the sort of thing an ingenious amateur might settle. If anyone asks why I have not done it myself—I am neither an amateur nor ingenious.”

is always less than $O((\log n)^2)$. In contrast, we only need to check a “certified” random sample of $\log_2 n + O(1)$ potential witnesses.

4. Additional Remarks

The central idea of the Solovay–Strassen and Miller–Rabin algorithms and of the preceding discussion can be expressed as follows: Consider a specific propositional formula F in n variables for which we somehow know that the percentage of satisfiability is greater than 75% or less than 25%. We wish to decide which of these two possibilities is in fact the case. The obvious way of deciding is to evaluate F at all 2^n possible n -tuples of the variables. But only $O(I(F))$ data points are necessary to decide which case holds by sampling, if one possesses an algorithmically random sequence $O(nI(F))$ bits long. Thus one need only evaluate F for $O(I(F))$ n -tuples of its variables, if the random sample is “certified.”

These algorithms would be even more interesting if it were possible to show that they are faster than any deterministic algorithms which accomplish the same task. Gill [12], [13] in fact attacked the problem of showing that there are tasks which can be accomplished faster by a Monte Carlo algorithm than deterministically, before the current surge of interest in these matters caused by the discovery of several probabilistic algorithms which are much better than any known deterministic ones for the same task.

The discussion of extensible formal systems given in [14] raises the question of how to find systematic sources of new axioms, likely to be consistent with the existing axioms of logic and set theory, which can shorten the proofs of interesting theorems. From the metamathematical results of [1]–[3], we know that no statement of the form “ s is c -random” can be proved if s has a length significantly greater than c . This raises the question of whether statements of the form “ s is c -random” are generally useful new axioms. (Note that Ehrenfeucht and Mycielski [15] show that by adding any previously unprovable statement X to a formal system, one always shortens very greatly the lengths of infinitely many proofs. Their argument is roughly as follows: Consider a proposition of the form “either X or algorithm A halts,” where A in fact halts but takes a very long time to do so. Previously the proof of this assertion

was very long; one had to simulate A 's computation until it halted. Now the proof is immediate, for X is an axiom. See also Gödel [16].)

Hence it is reasonable to ask whether the addition of axioms “ s is c -random” is likely either to allow interesting new theorems to be proved, or to shorten the proof of interesting theorems which could have been proved anyhow (but perhaps by unreachably long proofs). The following discussion of this issue is very informal and is intended to be merely suggestive. On the one hand, it is easy to see that interesting new theorems are probably not obtained in this manner. The argument is as follows. If it were highly probable that a particular theorem T can be deduced from axioms of the form “ s is c -random,” then T could in fact be proved without extending the axiom system. For even without extending the axiom system one could show that “if s is random, then T ” holds for many s , and thus T would follow from the fact that most s are indeed random. In other words, we would have before us a proof by cases in which we do not know which case holds, but can show that most do. Hence it seems that interesting new theorems will probably not be obtained by extending a formal system in this way.

As to the possibility of interesting proof-shortenings, we can note that Ehrenfeucht–Mycielski theorems are not very interesting ones. Quick Monte Carlo algorithms for primality suggest another possibility. Perhaps adding axioms of the form “ s is random” makes it possible to obtain shorter proofs of primality? Pratt's work [17] suggests caution, but the following more general conjecture seems reasonable. If it is in fact the case that for some tasks Monte Carlo algorithms are much better than deterministic ones, then it may also be the case that some interesting theorems have much shorter proofs when a formal system is extended by adding axioms of the form “ s is random.”

References

- [1] Chaitin, G. J., *Information-theoretic computational complexity*, IEEE Trans. Info. Theor. IT-20, 1974, pp. 10–15.
- [2] Chaitin, G. J., *Information-theoretic limitations of formal systems*, J. ACM 21, 1974, pp. 403–424.

- [3] Chaitin, G. J., *Randomness and mathematical proof*, Sci. Amer. 232, 5, May 1975, pp. 47–52.
- [4] Schwartz, J. T., *Complexity of statement, computation and proof*, AMS Audio Recordings of Mathematical Lectures 67, 1972.
- [5] Levin, M., *Mathematical logic for computer scientists*, MIT Project MAC TR-131, June 1974, pp. 145–147, 153.
- [6] Davis, M., *What is a computation?* in *Mathematics Today — Twelve Informal Essays*, Springer-Verlag, New York, to appear in 1978.
- [7] Chaitin, G. J., *Algorithmic information theory*, IBM J. Res. Develop. 21, 1977, pp. 350–359, 496.
- [8] Solovay, R., and Strassen, V., *A fast Monte-Carlo test for primality*, SIAM J. Comput. 6, 1977, pp. 84–85.
- [9] Miller, G. L., *Riemann's hypothesis and tests for primality*, J. Comput. Syst. Sci. 13, 1976, pp. 300–317.
- [10] Rabin, M. O., *Probabilistic algorithms* in *Algorithms and Complexity — New Directions and Recent Results*, J. F. Traub (ed.), Academic Press, New York, 1976, pp. 21–39.
- [11] Bell, E. T., *Mathematics — Queen and Servant of Science*, McGraw-Hill, New York, 1951, pp. 225–226.
- [12] Gill, J. T. III, *Computational complexity of probabilistic Turing machines*, Proc. 6th Annual ACM Symp. Theory of Computing, Seattle, Washington, April 1974, pp. 91–95.
- [13] Gill, J. T. III, *Computational complexity of probabilistic Turing machines*, SIAM J. Comput. 6, 1977, pp. 675–695.
- [14] Davis, M., and Schwartz, J. T., *Correct-Program Technology/Extensibility of Verifiers—Two Papers on Program Verification*, Courant Computer Science Report #12, Courant Institute of Mathematical Sciences, New York University, September 1977.

- [15] Ehrenfeucht, A., and Mycielski, J., *Abbreviating proofs by adding new axioms*, AMS Bull. 77, 1971, pp. 366–367.
- [16] Gödel, K., *On the length of proofs* in *The Undecidable—Basic Papers on Undecidable Propositions, Unsolvability Problems and Computable Functions*, M. Davis (ed.), Raven Press, Hewlett, New York, 1965, pp. 82–83.
- [17] Pratt, V. R., *Every prime has a succinct certificate*, SIAM J. Comput. 4, 1975, pp. 214–220.

RECEIVED JANUARY, 1978.