RELATION BETWEEN THE COMPLEXITY AND THE PROBABILITY OF LARGE NUMBERS

by

Peter Gacs

STAN-CS-79-765
September 1979

DEPARTMENT OF COMPUTER SCIENCE
School of Humanities and Sciences
STANFORD UNIVERSITY

Relation Between the Complexity and the Probability of Large Numbers

Peter Gacs

Computer Science Department
Stanford University
Stanford, California 94305

September, 1979

Abstract.

$H(x)$, the negative logarithm of the apriori probability $M(x)$, is Levin's variant of Kolmogorov's complexity of a natural number x . Let a(n) be the minimum complexity of a number larger than n , $s(n)$ the logarithm of the apriori probability of obtaining a number larger than n . It was known that

$$s(n) \leq \alpha(n) \leq s(n) + H(\lfloor s(n) \rfloor) \ .$$

We show that the second estimate is in some sense sharp.

This research was supported in part by National Science Foundation grant MCS-77-23738 and by Office of Naval Research contract N00014-76-C-0330.

Relation Between the Complexity and the Probability of Large Numbers

Peter Gacs

Let $T(p)$ be a partial recursive function defined over binary sequences with values among the natural numbers which is <u>prefixless</u>:

(a) If p_1 is a beginning segment of p_2 and $T(p_1)$ is defined then $T(p_2) = T(p_1)$

and <u>optimal</u>:

(b) for any other prefixless p.r. function T', there is a sequence p such that $T(pq) = T'(q)$ for all q.

Let $\ell(p)$ denote the length of the sequence p, Levin introduced the complexity

$$H(x) = \min\{\ell(p): T(p) = x\}$$

as a useful variant of Kolmogorov's complexity. See e.g. [1], also Chaitin [2], Gacs [3].

We denote by $T(p;t)$ a computable "approximation" of $T(p)$: on some Turing machine computing $T(p)$, $T(p;t)$ is $T(p)$ if $T(p)$ is computed within time t, undefined otherwise. We write

$$H(x;t) = \min\{\ell(p): T(p;t) = x\}$$

$$M(x) = 2^{-H(x)}, \quad M(x;t) = 2^{-H(x;t)},$$

$$s(n) = -\log\left(\sum_{i=n}^{\infty} M(i)\right)$$

$$a(n) = \min_{i>n} H(i).$$

$\alpha(n)$ and $s(n)$, two extremely slowly (slower than any unbounded, recursive function) growing functions, are closely related. It is known that

(1) $$s(n) \le \alpha(n) \lesssim s(n) + H(\lfloor s(n) \rfloor),$$

where $\overset{+}{<}$ and $\overset{+}{\asymp}$ denote inequality and equality to within an additive, $\lesssim$ and $\approx$ to within a multiplicative constant.

The first inequality is trivial, the second one is well-known (see e.g. [4]). A hint to the proof: to find a number $\ge n$, we have only to know $2^{-s(0)}$ to within an error term $2^{-s(n)}$.

We will show that the second estimate in (1) is sharp.

Theorem. Let g(n) be any positive, monotone recursive function such that

(2) $$\sum_n 2^{-g(n)} = \infty .$$

Then $a(n) > s(n) + g(s(n))$ infinitely often.

Proof. It is well-known (see e.g. [3]) that, if $\mu(n;t)$ is a computable nonnegative rational function over pairs of natural numbers, monotone in t and $\sum_n \mu(n;t) \le 1$, i.e., for each t, $\mu(n;t)$ is a semimeasure, then

$$\mu(n;t) \lesssim M(n) .$$

Put

$$s(n;t) = \sum_{i \ge n} M(i;t)$$

$$s\mu(n;t) = \sum_{i \ge n} \mu(i;t)$$

$$m(k;t) = \max\{n: s(n;t) < k\}$$

$$m_\mu(b;t) = \max\{n; s_\mu(n;t) < k\} .$$

The construction depends on n_k, a fast-growing recursive sequence. We will see at the end of the proof, how we should choose it in dependence of g.

Let $\mu(n;0) = 0$.

Suppose that $\mu(n;t)$ is already constructed. Put

$$k(t) = \max\{k \geq -\log(1 - s_\mu(0;t)): \exists i \in [n_{k-2}+1, n_{k-1}]$$

(3) $$\alpha(m_\mu(i - g(i);t);t) > i\} .$$

Put $n(t) = n_{k(t)}$. Let $j(t) = \max\{j: \mu(j;t) > 0\}$. Put

$$\mu(j(t)+1;t) = 2^{-n(t)}$$

$$\mu(j;t+1) = \mu(j;t) \quad \text{for } j \neq j(t) .$$

We will show that there are infinitely many i's such that for almost all t, (3) holds.

This implies, of course, that

$$\alpha(m_\mu(i - g(i)) > i .$$

That is, for some n, with

$$i-g(i) > s_\mu(n)$$

$$a(n) > i > s_\mu(n) + g(i) \geq s(n) + g(i) \geq s(n) + g(s(n))$$

and the theorem will be proved.

Suppose that, on the contrary, there is a largest i_0 among the i such that (3) holds for almost all t and a least t_0 such that (3) holds for i_0 and all $t \geq t_0$.

Under the above assumptions,

$$s_\mu(0;t) \to 1 \quad .$$

Therefore

$$\sum_t 2^{-n(t)} = 1 \quad .$$

<u>Notation</u>. $A(t_1,t_2) = \sum_{t=t_1}^{t_2} 2^{-n(t)}$;

$$B(t_1,t_2,k_0) = \sum \{2^{-n(t)} : t \in [t_1,t_2] \text{ , } k(t) = k_0\} \text{ .}$$

<u>Lemma</u>. There exists a triple (k_0,t_1,t_2) with $k_0 \geq k(t_0)$, $t_2 \geq t_1 \geq t_0$ such that

(a) $k(t) \geq k_0$ for $t \in [t_1,t_2]$;

(b) $2^{-n_{k_0}-1} \leq A(t_1,t_2) \leq 3\ B(t_1,t_2,k_0)$.

<u>Proof</u>. For some t^0 , $(k(t_0), t_0, t^0)$ will satisfy (a) and the first inequality of (b).

Let us say that $(k_0,t_1,t_2) < (k_0',t_1',t_2')$ if $k_0' \leq k_0$, $t_1' \leq t_1 \leq t_2 \leq t_2'$.

Let (k_0,t_1,t_2) be a minimal triple $\leq (k(t_0), t_0, t^0)$, among the triples satisfying (a) and the first part of (b).

(A) For $t_3 \in [t_1,t_2]$ we have $k(t) = k_0$, otherwise the triple is not minimal.

For similar reasons we have

(B) If $t_1 \leq t_1' \leq t_2' \leq t_2$ and $k(t) > k_0$ in $[t_1',t_2']$ then then $B(t_1',t_2') < 2^{-n_{k_0}}$.

Therefore we have

$$A(t_1,t_2) \le B(t_1,t_2,k_0) + (1+\#\{t \in [t_1,t_2]: k(t) = k_0\}\cdot 2^{-n_{k_0}}$$

$$\le 2B(t_1,t_2,k_0) + 2^{-n_{k_0}} . \qquad \square$$

We concentrate now on a triple $(k,t_1,t_2) \le (k(t_0),t_0,t^0)$ satisfying (a) and (b).

<u>Notation</u>. For $i \in [n_{k-1},n_k]$ put

$$E_i = \{t \in [t_1,t_2]: \exists n\ H(n;t) < i , H(n;t) < H(n;t-1)\} .$$

We now estimate $s_i = \# E_i$ from below (see (5)). Let us write $E_i = \{t_{i1},t_{i2},\ldots,t_{is_i}\}$, where $t_{ij} < t_{ij+1}$. Put $t_{i0} = t_1-1$, $t_{is_i+1} = t_2$. Let u_{ij} = the last t in $[t_{ij}+1,t_{ij+1}]$ (if any) with $k(t) = k$. If there is no one, $u_{ij} = t_{ij}$.

Let $\sigma_{ij} = \sum_{t = t_{ij}+1}^{u_{ij}-1} 2^{-n(t)}$, $\lambda_{ij} = -\log \sigma_{ij}$. Then by our algorithm we have

$$\alpha(m_\mu(i-g(i))\ ; u_{ij}-1) \le i .$$

On the other hand, by the definition of u_{ij},

$$\alpha(j(t_{ij}+1)\ ; u_{ij}-1) > i .$$

Therefore we have

$$\lambda_{ij} = s(j(t_{ij}+1)\ ; u_{ij}-1 \ge i-g(i) ,$$

(4) $$\sigma_{ij} \le 2^{-i+g(i)} .$$

On the other hand,

$$2^{-n_{k-1}} < \sum_{t=t_0}^{t_2} 2^{-n(t)} = \sum_{t \in E_i} 2^{-n(t)} + \sum_j \sigma_{ij} + B(t_1,t_2,k)$$

$$< s_i \cdot 2^{-n_k} + (s_i+1)2^{-i+g(i)} + B(t_1,t_2,k) .$$

Using (b) of the Lemma,

$$\frac{2}{3} \cdot 2^{-n_{k-1}} \leq (s_i+1)(2^{-n_k} + 2^{-i+g(i)}) \leq 2(s_i+1)(2^{-i+g(i)}) ,$$

Hence

$$s_i \geq \frac{1}{3} \cdot 2^{-n_{k-1}+i-g(i)} - 1 ,$$

that is, for $i-g(i) > n_{k-1}+2$:

(5) $$s_i \geq \frac{1}{4} \cdot 2^{-n_{k-1}+i-g(i)} .$$

Put $m_k = \min\{i: i-g(i) > n_{k-1}+2\}$.

We have

$$1 \geq s(0;t_2) - s(0;t_1) \geq \sum_{i=m_k+1}^{n_k} \cdot 2^{-i} \cdot (s_i - s_{i-1}) + 2^{-m_k} \cdot s_{m_k}$$

$$= \sum_{i=m_k}^{n_k} \cdot 2^{-i} s_i - \sum_{i=m_k}^{n_k-1} 2^{-i-1} \cdot s_i$$

$$> \sum_{i=m_k}^{n_k-1} 2^{-i-1} \cdot s_i \geq \frac{1}{8} \cdot 2^{-n_{k-1}} \cdot \sum_{i=m_k}^{n_k} 2^{-g(i)} .$$

If n_k is chosen far enough from n_{k-1} , this will obviously lead to a contradiction. □

References

[1] L. A. Levin, "Laws of information conservation," Problems of Information Transmission 10, 3 (1974), 206-210.

[2] G. Chaitin, "A theory of program size formally identical to information theory," Journal ACM 22 (1975), 329-340.

[3] P. Gacs, "On the symmetry of algorithmic information," Soviet Math. Doklady 15 (1974), 1477-1480; Corrections, ibid, 6, v.

[4] R. Solovay, unpublished manuscript.