Solovay Contents

Kerko Goy

I. Randomness properties of $\Omega = \sum M(n)$.

Relation of various definitions of randomness.

**Definition**

A weak test $\delta$ is defined as

$$\lim_{m \to \infty} \mu \{ x : \delta(x|\mu) > m \} = 0.$$

**Theorem**

Each r.e. real fails under some weak test.

II. $H(H(x)|x)$.

**Conjecture**

$\forall n \exists x \in \mathbb{Z}$, $H(x) \approx n$, $H(H(x)|x) \approx \log n$.

(In our examples we may have $H(x) < n / \log \log n$)

Why not $n / \log n$?

III. Various formulas relating $H$ and $K$.

Can be approximately summarized in

$$H(x) \approx K(x \mid H(x)) + H^2(x).$$

where $H^2(x) = H(H(x)).$
IV. The relation of the tests

\[ m_H(x) = |x| - H(x) \text{ and } m_K(x) = |x| - K(x). \]

It is shown that

\[ m_H \geq m_K + O(\log m_K) \]

\[ m_K(x) \geq m_H(x) + O(\log_2 |x|), \]

and this is sharp.

V. Upper bounds on \( H \) that are sharp

infinitely often. (See my dissertation for a detailed consideration.)

More: \( \exists h \)

\[ \lim_{n \to \infty} \frac{1}{n} \sum_{m = n \leq m \leq n} h(m) = H(m) = \frac{1}{3} > 0. \]

\[ \forall h, \lim_{n \to \infty} h(n) = 0, \]

Moreover: no rec. reg. \( D_n \) with \( m \in D_n \rightarrow m > n \& \exists x \in D_n, h(x) \leq H(x) + C. \)
On $s(n)$, $\alpha(n)$, $D_n = \{ x : l(x) \leq n \ \& \ \U(x) \ is \ defined \},$ and $\Omega_n$, and the sets $\{ x : H(x) = m \}$. Here $s(n) = -\log(\sum_{n>m} H(n))$, $\alpha(m) = \min_{n \geq m} H(n)$.

I think I can prove that $\alpha(m) \geq s(m) - H(s(m))$ cannot be substantially improved. 

$\log \# D_n \approx m - H(n)$, as well as the cardinality of some related sets.

$H(D_n) \leq n - H(H(D_n) \cup \Omega_n) \leq H(-\Omega_n \cup D_n + H(n))$

On $H(\Omega_n)$.

Let $g_n$ be recursive. $\sum 2^{-g_n(i)} = \infty.$

$g_n$ be recursive such that $g_n \rightarrow \infty$.

Let $x$ be random. For $n$

1) $H(n) \leq g_n(n)$ and

either 2a) $H(x^n) \geq n + g_n(n)$

or 2b) $H(x^n) \leq n + g_n(n)$ (analogous to Martin-Löf's oscillation result)

$H(x) - n \rightarrow$ for recursive $x$, because $\sum 2^{-H(x^n)}$ converges.
To the proof of 2b: let $\Lambda_n \subset 2^\mathbb{Z}$, $\Sigma \mu(A_n) = \infty$.

\[ g_2(n) = -\log \mu(\Lambda_n), \quad \Lambda_n \subseteq \text{the whole space.} \]

Then, \[ M(n), \frac{\mu(A_n)}{\mu(\Lambda_n)} \leq M(x). \]

For $x \in \Lambda_n$, \[ H(x) \leq -\log \mu(x) + H(\Lambda) + g_2(n). \]

Now for $g_1$ let $g_2$ be such that \[ \Sigma \Sigma q^2 = \infty, \Sigma q^{2+\epsilon} < \]

Then \[ H(x) \leq -\log \mu(x) + g_1(n), \quad \text{for all } x, \]

\[ H(x) - H(x) + \log \mu(x) + g_1(n) + g_2(n) \geq g_2(n) \quad \text{for random } x. \]

**Def.** $\Omega$-like r.e. real: \[ \exists F, \forall q \in \Omega \quad |\Omega - F(q)| \leq C(n-q) \]

Note: $\alpha$ is $\Omega$-like iff $H(\alpha^+) = H(\Omega_n)$.

**VII A. $\Omega$ and the within random reals.**

\[ \exists \omega \in \Omega \]

1. If $x$ is within random then $\exists C \exists m$

\[ H(x^n) \geq n + H(\Lambda)_n - C \]

2. $H(\Omega_n) \leq n + H(\Lambda)_n - g_2(m) + O(\log \alpha_2(n))$

Similarly, $KCL_n \leq n - \alpha_2(n) + O(\log \alpha_2(n))$

3. If $x$ is within random then

\[ H(x^n) \geq n + \alpha_2(m) + O(\log \alpha_2(m)) \]

4. $\exists m \quad H(\Omega_n) \leq n + \alpha_1(n) + O(\log \alpha_1(n))$. 
For 1) the function \( \sup_{m} \inf_{n} m + H(n) - l H(x^n) \)

is an arithm. test.

2) Knowing \( m \) we know a little segment of \( \Omega \).

3) \( M(x^n) \geq M(\ell_5(n)), M(x^n), 2^5(n) \),

\( n = H(x^n) \) is an arithm. test.

4) Using a segment of \( \Omega \) to compute a length.

\[ \text{VIII. Non-recursive sets with } H(x^n) \neq H(n). \]
Notes to Solovay's Chapter VIII (other case).

The only thing we want to replace is Lemma VIII.3.

Let \( \Phi_e(x) \) always measure the storage requirement of the computation of \( \{e\}^*(x) \).

We assert that the priority construction will work

with \( t_i = 6^i \). To see this, we prove a lemma replacing Lemma VIII.3.

**Lemma VIII.3'**

Let

\[
H(x; t) = \min \{ \mathbb{1}_p : H(1p) = x & \Phi(p) \leq t \} \quad \text{and} \quad M(x; t) = \exp(-H(x; t))
\]

\[
s(n; t) = -\log \left( \sum_{i \geq n} M(x; t) \right)
\]

\[
m(k; t) = \min \{ n : s(n; t) > k \}
\]

Note that \( m(k; t) \neq m(k; \infty) < \infty \).

**Lemma VIII.3'** We have for all \( k \) and \( n \geq m(k, t) \)

\[
H(n; t) - H(n; 6t) = \kappa - H(k; t; 6t)
\]
Proof of Lemma VIII.3

Let \( n \geq m(k,t) \).

We give a program of length

\[ H(k,t; \xi t) + H(m,t) - k \]

computing \( n \) within space \( 6 \xi t \).

Let \( p \) be a program computing \((k,t)\) within space \( t \).

We act off, with the help of \( p \), on the \( 6 \xi \) areas of length \( S_0, \ldots, S_6 \). We store \( t \), \( k \), and a piece of program \( q \) of length \( H(k,t) - k \) to be specified later in areas \( S_0 \) and \( S_4 \).

Let \( \xi \) be an enumeration of the words of length \( \leq t \).

\( S_2 \) will keep track of \( \xi \).

\( S_3 \) will keep track of the numbers \( \xi \) with \( \log_2 \leq t \).

\( S_4 \) will keep track of \( \sum 2^{-\xi x} : \xi < i, \xi(x) \leq t, m(x) \geq t \).

\( S_5 \) is reserved for the actual computations.

After computing and storing \( t \) and \( k \), we proceed to compute \( m(k,t) \). For each \( \xi \) with \( \log_2 \leq t \) we compute \( \xi(\xi,t) \) by accumulating the sum mentioned above in \( S_4 \).
We stop when \( \frac{m}{m} \leq k \). Then we arrived at \( r = m(k, t) \).

Now we again begin accumulating, in \( S_k \), the sum

\[
q_i = \sum \left\{ 2^{-1} x_i^1 \cdot x_i^2 \mid \beta_j < i, \ \Phi(x_j) \leq t, \ U(x_j) > m(k, t) \right\}.
\]

Let \( U(x_i) = m, \ \Phi(x_i) \leq t, \ \beta_i = H(n; t) \). We have

\[
\alpha_i = M(m; t), \ \text{hence if } \ q \text{ localizes the interval}
\]

\((a_i, \beta_i)\), we will be able to recover \( m \). O.E.D.

**Application of Lemma VII. 3'**

We must show that with this \( e_2 \), and the \( \Phi(m) \) defined in Solovay VII, for any recursive function \( e_3(x) \) we have infinitely often \( \Phi_e(\Phi(m)) \leq m \).

**Proof** Let \( j \) be the least \( i \) with the following property: for all \( j \leq m(k, t^5) \), \( \{ e_3(j) \beta \} < i \).
Clearly $i_k$ is computable from $k$ within space $5 \cdot 6^k$, since, as we have seen in the proof of lemma $\overline{III}_3$, $m(k, 6^k)$ is. Thus $H(k; 6^k; 6^k) \leq 2 \log k$,

and for $n \geq m(k, 6^{i_k})$,

\[ H(n; 6^k) - H(n; 6^{i_k+1}) \geq k - 2 \log k > 1. \]

Hence

$\sigma(i_k) \leq m(k, 6^{i_k}),$

i.e.

$\Phi_e(\sigma(i_k)) \leq i_k$. \quad Q.E.D.
I. Randomness properties of $\Omega$

Recall that the real $\Omega$ is defined as the

$$\sum \{ 2^{-|p|} : U(p) \text{ is defined} \},$$

(Chaitin uses lower case $w$, but we use $\Omega$ as our notation for this.

Here $U$ is the universal Chaitin computer and the $|p|$ is non-negative.

$|p|$ is the length of the program $p$.

It is clear that $\Omega$ is an r.e. real in the sense of the following definition.

**Definition.** A real $x$ is r.e. if there is a recursive sequence of rationals $\langle x_n, n \rangle$ which is weakly increasing (i.e., $n \leq m \rightarrow x_n \leq x_m$) and has the limit $x$.

Equivalently, $x$ is r.e. if $\{ q \in \mathbb{Q} : q \leq x \}$ is r.e.
(These two definitions are, in fact, effectively equivalent, so that we can effectively go from a Gödel number for \( (\forall n \in \mathbb{N}) \) to one for \( 1 \leq q \leq \mathbb{Q} : q < x \), and vice versa.) Note also, that we can choose the sequence \( x \) in Definition 1 to consist of dyadic rationals.

A randomness property of reals is a property that holds of all reals except for a set of Lebesgue measure zero. (For example, let \( M \) be a transitive model of some suitable fragment of axiomatic set theory. We say that \( x \) is random over \( M \), if \( x \) lies in no Borel set of measure zero which can be
encoded by a real in \( M \). (Cf. [Sol. 1, Chapter II].)

...where this notion is discussed and shown to be a randomness property.

Now obviously almost all reals are not r.e., and thus sufficiently strong randomness properties are not possessed by \( B \). We are going to investigate in this chapter the following two questions.

1) What randomness properties are properties of some r.e. real?

2) Which of these are properties of \( B \)?

Of course these questions are somewhat vague...

Before stating our results precisely, we need some
preliminary definitions. We pick an effective enumeration of all open intervals with rational end-points, \( \langle I_0, I_1, \ldots \rangle \). Then we say a recursive function \( h : \omega \to \omega \) gives a recursive enumeration of the open set \( U \subseteq \mathbb{R} \) if

\[
U = \bigcup_{n \in \omega} I(h(n))
\]

By definition, \( \langle U_n \rangle \) is r.e. if \( h \) such an \( n \) exists.

A sequence of open sets \( \langle U_n \rangle \) is simultaneously r.e. if there is a recursive map \( h : \omega^2 \to \omega \) such that

\[
U_n = \bigcup_{j \leq n} I(h(n,j)).
\]

We now proceed to list certain properties of a real \( x \in [0,1] \).
\( P_1(x) \): Let \( \langle U_n; n \in \mathbb{N} \rangle \) be a simultaneously
er sequence of open sets. Suppose that the sequence \( \langle U_n \rangle \)
is decreasing (i.e., \( m < n \Rightarrow U_n \subseteq U_m \)) and
that \( \mu(U_n) \to 0 \). Then \( x \notin \bigcap_n U_n \).

\( P_2(x) \): Let \( \langle U_n; n \in \mathbb{N} \rangle \) be a simultaneously
er sequence of open sets. Suppose further that

\[ \sum_{i=0}^{\infty} \mu(U_i) < \infty. \]

Then \( x \in U_i \) for only finitely many \( i \).

\( P_3(x) \): Similar to \( P_2 \) except we require the
\( U_i \)'s to satisfy the following additional requirement:

There is a recursive function \( h: \omega \to \omega \) such
that
\[ \sum_{i=0}^{\omega} \mu(U_i) \leq 2^{-h(i)} \]
(I.e. we should have an effective bound on the rate of convergence in (1).)

\( P_3 \): Similar to \( P_4 \) except we require that

\[ p(U_n) < 2^{-n}. \]

\( P_4(x) \) says first (to avoid some trivial cases) and lies in the unit ball that \( x \) is not a dyadic rational. Thus the dyadic expansion of \( x \) is well-defined. Let \( x(n) \) be the first \( n \)-digits of \( x \). Then for some \( C \),

\[ H(x(n)) > n - C \]

infinitely often in \( n \).

We shall see in a moment that \( P_4(x) \) holds. For almost all reals \( x \), and that the implications
\[ P_4(x) \rightarrow P_2(x) \rightarrow P_3(x) \iff P_3'(x) \rightarrow P_4(x) \]

are trivial.

\[ P_3(x) \] is the notion of Martin-Löf randomness.

(4. [7]) \( P_4(x) \) is the notion of randomness proposed by Chaitin in [7]. Schnorr has announced the theorem: \( P_3(x) \iff P_4(x) \).

Chaitin shows that \( P_4(52) \) in [7]. Here are the main results of this chapter.

1) \( P_4(x) \) holds for no \( x \) in \( x \).

2) \( P_4(52) \) is true. (The results of Chaitin and Schnorr that \( P_4(52) \) and \( P_3(52) \)
of course follow.)
3) \( P_2(x) \rightarrow P_3(x) \).

Thus \( P_2(x) \) is just another way of expressing Martin-Löf randomness. Our result is a bit surprising since the obvious Borel-Cantelli variant of M.L. randomness is \( P_3 \).

I. 1. Let us begin with the trivial facts recalled above. In the first place for any sequence \( \langle U_i, v_i \rangle \) as in a \( P_1 \), \( \cap U_i \) is a Borel set of measure zero.

But there are only countably many \( v_i \) simultaneously, i.e. sequences of open sets, and the union over all such as in \( P_2 \), \( \langle U_i, v_i \rangle \), of the corresponding sets \( \cap U_i \) has measure zero. Thus almost all reals avoid all such
sets and so satisfy $P_1$.

Next let $x$ satisfy $P_1$. Let $(U_i)_{i \in \mathbb{N}}$ be as in $P_2$. Put $W_0 = \bigcup_{m=0}^{\infty} U_m$. Then

$<W_n |_{new}>$ is simultaneously r.e. ad

$$\lim_{n \to \infty} \mu(W_n) \leq \lim_{n \to \infty} \sum_{m=1}^{\infty} \mu(U_m) = 0.$$ 

Thus $x \notin W_n$, some $n$, by $P_1$. Thus $x \in U_1$, only

Thus for $1 < n$, so $x$ satisfies $P_2$.

It is utterly trivial that $P_2(x) \rightarrow P_3(x)$, and the proof that $P_3(x) \rightarrow P_3(x)$ can be gotten along the lines of the proof that $P_1(x) \rightarrow P_2(x)$ just given.

Suppose conversely that $x$ satisfies $P_3(x)$. 
Let \( \langle U_n \rangle \) be as in \( P_3 \), and let
\( h : \omega \to \omega \) be also as in \( P_3 \). Define \( \langle W_n \rangle \) by
\[
W_n = \bigcup_{m = h(n)}^{\infty} U_m.
\]
Then \( \mu(W_n) < \frac{\gamma}{2^n} \) & the \( W_n \)'s are simultaneously r.e.
By \( P_3'(x) \), \( x \notin W_n \), some \( n \).
But then \( x \notin U_i \) for \( i > h(n) \), as will be shown.

The proof that \( P_3(x) \to P_4(x) \) is essentially in Chaitin. Recall that for some constant \( C \) independent of \( m, n \),
\[
\nu(\{ x : x \notin \mathbb{E}_n, H(x) \leq H(n) + n - k \}) \leq C \cdot 2^{-k}.
\]
Thus, \( \mu(\{x \in \mathbb{Z}^n : H(x) \leq n - k \}) \leq C \cdot 2^{-H(x)} \).

On the other hand, \( \sum_{n=0}^{\infty} H(n) < 1 \). So

\[
\mu(\{x \in \mathbb{Z}^n : H(x) \leq \sum_{n=0}^{\infty} H(n) \leq n - k \}) 
\leq C \cdot 2^{-k}.
\]

Let \( W_k := \{ x \in \mathbb{Z}^n : H(x(n)) \leq n - k \} \).

Then, \( \mu(W_k) \leq C \cdot 2^{-k} \left( \sum_{n=0}^{\infty} 2^{-H(n)} \right) \).

Also, the \( \langle W_k \rangle_{k \in \mathbb{N}} \) are simultaneously r.e. (For the purpose of this proof, view each dyadic rational as given its non-terminating dyadic expansion. It is necessary to check that each dyadic rational lies in \( W_k \), all \( k \). But \( n_0 \) is clear since \( H(x) \).)
is evident: \( H(x(n)) \leq H(n) + o(1) < n. \)

By a theorem of Chaitin, \( \sum_{n=1}^{\infty} 2^{-H(n)} < 1. \)

Thus \( \mu(W_0) \leq C \cdot 2^{-n} \) and we can apply

\( P_3 \) to conclude \( x \not\in W_k \) for some \( k \). Hence

\[ H(x(n)) \geq n - k, \text{ all } n \]

and \( x \) satisfies \( P_4. \)

1.2 We turn to the proofs of our main results of this chapter. The first is

**Theorem 1.1.** Let \( x \) be an r.e. real. Then \( x \) does not satisfy \( P_4. \)

The proof we present is due to D.A. Martin and was arrived at at about the same time as the
authors' proof. We present Martin's proof since it is conceptually simpler.

Let \( x \) be an r.e. real. We may assume \( 0 \leq x \leq 1 \), and let \( \{x_n\} \) be a monotone increasing recursive sequence of reals with limit \( x \). We define \( U_n \). Note that if \( x \) is rational, the sequence

\[
W_n = (x - 2^{-n}, x + 2^{-n})
\]

even shows that \( x \) does not satisfy \( P_3 \). So we assume \( x \) irrational.

Now define \( U_n = (x_n, x_n + 2(x - x_n)) \).

It is clear that the \( U_n \)'s are decreasing.
sequence of open sets containing x. Moreover

\[ m(U_n) = 2 |x - x_n| \to 0. \] To complete

the proof that x does not satisfy P_2, we show

that \( \{U_n \mid n \in \mathbb{N}\} \) are simultaneously

But this is clear since

\[ U_n = U_m \cap (x_n, x_n + 2(x_m - x_n)). \]

**Theorem I.3.** Let \( \Sigma \) be as

in Cluett's paper. Then \( \Sigma \) satisfies \( P_2 \).

**Proof.** (Again, I would like to thank

D. A. Martin for making a remark which led to

the present proof.)

\( P_2 \Rightarrow P_4 \& \)

Cluett proves \( P_4(\Sigma) \)
Let the \( \{U_n \} \) be a simultaneously \( \text{r.e.} \) sequence of open sets such that

\[
\sum_{n=0}^{\infty} \mu(U_n) \leq C < \infty.
\]

By deleting a finite no. of the \( U_n \)'s, we may assume \( C = 1 \).

For the moment, let \( N \) be any positive integer. \( N \) will be fixed presently.

Let

\[
W_N = \frac{1}{N} \times 1 \quad x \in U_n \quad \text{for} \quad \text{N distinct values of} \ n.
\]

First, what is the measure of \( W_N \). If \( A \) is any set, let \( \chi_A \) be its characteristic function: \( \chi_A(x) = 1 \) for \( x \in A \), and equals 0 otherwise. Let

\[
g = \sum_{n=0}^{\infty} \chi_{U_n}.
\]

Then \( \int g = \sum_{n=0}^{\infty} \mu(U_n) = \Theta_n \).

On the other hand, by the definition of \( W_N \),
\[ 0 \leq N \leq 9 \]

Integrating this equation we get \( N \mu(W_N) \leq \frac{\varepsilon}{\varepsilon N} \frac{1}{\varepsilon N} \).

\[ \mu(W_N) \leq \frac{\varepsilon}{\varepsilon N} \frac{1}{\varepsilon N} \]

Next we claim that \( \langle W_N / N = 0 \rangle \) is simultaneously \( \varepsilon \) i.e. \( \varepsilon \)-closed.

\[ W_N = \bigcup \{ (a, b): a, b \in \mathbb{Q} \text{ and } c, d, e \in \mathbb{N} \} \]

Thus it suffices to see that

\[ \varepsilon_{53}: \varepsilon < b \leq \varepsilon \]

Let \( x \in W_N \). Then for some \( n_0, \ldots, n_N \) \( x \in U_{n_i}, 1 \leq i \leq N \). Since the \( U_i \)'s are open, there are rectangles \( a < x < b: [a, b] \subset U_{n_i}, 1 \leq i \leq N \) (So \( [a, b] \subset W_N \)). It follows
\[ W_N = \bigcup \{ (e, s) : a, b \in \mathbb{Q}, a < b \leq 2e, s \in \omega \} \]

\text{for at least } N \geq 3.

Where it suffices to see that

\[ \{ (e, b, a) : s, e \in \mathbb{Q}, e, s \leq 1/3 \} \]

\text{is r.e. Let } \varphi : \omega^2 \rightarrow \omega \text{ be as in the definition of simultaneously r.e. Then, by compactness of } [e, s]

\[ \exists n \quad ( [e, s] \subseteq \bigcup_{j < n} \varphi(x, j)) \]

That the expression in parentheses is recursive is a trivial piece of combinatorics that we leave to the reader.
We now indicate our plan to show \( R \subseteq U_n \) for only finitely many \( n \). \( R \) is defined in terms of the universal Chaitin computer, \( U \). \( U \) must in turn simulate all other Chaitin computers. We shall construct a Chaitin computer \( C \) whose sole purpose in running its programs converge is to issue \( R \subseteq W_N \), so for some value of \( N \), \( C \) will compute. We turn to the details.

First we need an analog of the recursion theorem. Let \( h : \Sigma^* \times \Sigma^* \rightarrow \Sigma^* \) be partial recursive. Suppose also that for any \( y \), \( h(\cdot, y) \) is a Chaitin computer in its first argument.
Then for some Chaitin computer $C,$

$$C(x) = h(x, \Pi e)$$

deducing

We may thus from the usual recursion theorem

follows. Let $s : \Sigma^* \rightarrow \Sigma^*$ be such that

$s(e) = \Pi e$, for $e$ the Gödel number of the computer $C$. By the recursion theorem, we may find $C$ such that Let $s_2 : \Sigma^* \rightarrow \Sigma^*$ be such that if $e$ is the number of a Chaitin computer as a partial recursive function, $s_2(e)$ is its Gödel number of Chaitin computer. By the usual recursion theorem, find $C : C(x) \equiv h(x, s_2(s_2(e)))$. 
where e is a Gödel number for x.

We now describe c:

We next need a result of Chaitin which we express as follows. Suppose we are given a series of rationals \( x_n \) in \([0,1]\). We imagine receiving the sequence in \( x_n \) in real time. Then we can output a prefix-free code \( \langle s,1w \rangle \) which will be generated from the \( x_n \)'s in an effective fashion so that \( \sum 2^{-|s|} = x \).

This will follow easily from Chaitin's description of the effective construction of an instantaneous code.
We easily construct a sequence of dyadic rationals $x'_n$ such that 
\[ x_0 - 2^{-n} \leq x'_n \leq x_0. \]

If $y_0 = x'_0$, $y_{n+1} = x'_n - x'_n$. And 
\[ y_n = \frac{m}{2^n}, \text{ at stage } n, \text{ we put } \]

in requests and Chvátal's proof on the code words of length $n$. Thus the sum over the constructed code $S$ of $2^{-18}$ will be 
\[ \sum_{n=0}^{\infty} y_n = \lim_{n \to \infty} x'_n = x. \]

We turn to the construction of $C$. $C$ will first compute the length of its prefix code, and pick $N > 2^k$.

Let $V = W_N$. Note $p(W) = 2^{-\epsilon}$. Let $W = W_N$. Since $D_{\epsilon}$,

$C$ will proceed in stages. At any stage $n$,
\( E \) is the limit of a sequence

C will add a finite set of strings. Also at any time \( n \), we will have some finite approximation \( V_n \) to \( V \).

Suppose we come to stage \( n \) of the definition.

C. We select a time \( t_n \) so large that \( n \to t_n \).

For any \( s \), if \( n > 0 \), 2) Any string added to the domain of \( C \) at previous stages, the corresponding string \( T_c^s \) has been computed by \( U \) before time \( t_n \).

and so is reflected in the value of \( S(t_\infty) \).

Let \( S' = S_{t_\infty} \), \( V' = V_{t_\infty} \). If \( S' \notin V' \), we do nothing. If \( S' \in V' \), let
let be the least real such that \( r \notin V' \) or \( r = 1 \). Now if \( r \) is rational, we see since \( V' \) is a finite union of open intervals with rational endpoints.

We send to our string generating program the request \( y_n = (r - \mathcal{S}_{n-1}) \cdot 2^k \). We associate to this request the subset \( (\mathcal{S}_n', r) \) of \( V \).

Suppose by the inductive hypothesis that
\[
\sum_{i=0}^{n} y_i \leq 2^k \mu(V \mid \mathcal{C}_0, 2^k I) \text{.}
\]

Since
\[
2^k \mu(V) \leq 1
\]

\[
1 - \sum_{i=0}^{n} y_i > 2^k \mu(V \mid \mathcal{C}_n', 1 I)
\]

\[
> y_{n+1}.
\]
So $\sum_{i=0}^{n} y_i < 1$. Thus we can process the request and issuers that by the time on is completed

$$\sum_{i} 2^{-i^3} : s.e. S_{on} = \sum_{i} 2^{-i} y_i.$$ 

This after $t_{n+1}$, $S_{on} \geq \Delta + 2^{-n} \cdot y_n.$

So $S_{on}$ will be done $\geq \tau$. This shows that different requests have different sub-intervals of $V$ associated to them, and shows that ($\tau$)

continues to hold.

Now suppose that $S \in V$, for infinitely many $i$. Then $S \in V$. Thus $S$ for a

some $\alpha \in Sc, \epsilon, \delta \in \Sigma, \epsilon, \delta \in V$.

Pick $n$ so large that $S_{on} \geq \epsilon, \delta \in V.$
Possibly give the proof of $E_3$ now. E. 1st

Then our construction of $C$ will produce $R_x$

$S_x \geq b$, a contradiction.

**Proof**

**1.4 Finally, we give the $a_i$ of Theorem 1.3:**

If $x$ satisfies $P_2$, then $x$ satisfies $P_2$.

Let $\{U_i, n_i\}$ be as in the statement of $P_2$.

That $x \in U_0$ for infinitely many $n_i$. We shall show $x$ does not satisfy $P_3$. We may assume, by deleting finitely many $U_i$'s that

$$\sum_{i=0}^{\infty} \mu(U_i) < 1.$$  Let

$$V_3 = \{x \mid x \in U_0 \text{ for } 2^{-i} \text{ different values of } i\}.$$

Then the $V_3$'s form a decreasing sequence.
sequence of open sets. In the proof of the
previous theorem, we established that the
\[ \mu(V_n) \leq 2^{-n}. \]
But \( x \in V_n \) for all \( n \). Where \( x \) fails to satisfy

P3.

This last part of I is for my own
information and will not appear in Marginalia.

I send proof of preliminary notes not intended
for the final run by a # after the
number.
Let the \((U_n)_{n=0}^{\infty}\) be a simultaneously

\[ U_n \subseteq 2^n \]

equence of open sets, with \(\mu(U_n) < 2^{-n}\). We can effectively associate to \(U_n\) a prefix free code

\[ S_n : x \in U_n \Rightarrow \exists s \in S_n \quad s \subseteq x. \text{ To do this, let} \]

let \((S_n^{\leq n})_{n=0}^{\infty}\) be a recursive enumeration of strings

\[ U_n = \bigcup_{s \subseteq x} S_n^{\leq n}(x). \text{ We can continue} \]

a prefix code \(S_0, S_1, \ldots, S_n, \ldots\) by adding

\[ y \supset S_0 \quad \text{if} \quad y \supset S_0 \quad \text{such that} \quad y \supset S_0 \quad (\text{Ref. 1, Example}) \]

Now apply the Christia simulation lemma to get a new code for the \(S_i\):

\[ C(y, t) = S_i \quad \epsilon \subseteq U(y) = n \]

for some \(t : 1 + t \leq |S_i|, 1 - n \)
\[ \exists C_0. \]

Thus \( x \in U_0 \Rightarrow \exists m: \]

\[ H(x(m)) \leq m - n + H(n) + C_0. \]

(We use here that the \( U_n \)'s are small, i.e.)

Now given \( k \), pick \( n: n - H(n) > k \)

(Possible since \( H(n) \leq 2 \log_2 n \).)

Then \( x(n) \) does not satisfy

\[ x(n) \geq m - k \quad \text{all } m, \quad \text{since } x \in U_n. \]

This completes our reconstruction of Subramaniam's proof that \( P_2(x) \leftrightarrow P_4(x). \)
The main result of this section is the following theorem.

There is a $C > 0$ such that for every string $s$ of length $n$ with $\tilde{H}(s) = \log n - \log \log n - C$, (All log's to base 2)

Here $\tilde{H}(x|y)$ is the length of the shortest program $p$ $\geq U(p, y) = x$.

Note that it suffices to prove the theorem for large $n$, since we afterwards can adjust $C$ so as to make it true for the finite set of exceptional values.

We shall see that any $C > 1$ will work for all sufficiently large $n$. In what follows, $n$ is always "sufficiently large."
Lemma 1. Let \( \varepsilon \) be sufficiently small.

Let \( k = \log n - \log \log n - C \)

\( k = 2^k + 1 \)

\[ M = \sum_{0 \leq i \leq k} n^{c_i} \]

Then \( M < 2^n \), for \( n \) sufficiently large.

**Proof**

\[ M \leq \frac{n^{c_\varepsilon c^{\varepsilon + 1}}}{n^{c_\varepsilon} - 1} \]

\[ \sim n^{c_\varepsilon (c^{\varepsilon + 1})} \]

\[ \log M \leq c_\varepsilon k \log n \]

\[ \log \log M \leq \log \log n + k + \log c_\varepsilon \]

\[ \leq \log n + [ \log c_\varepsilon - C ] \]

If \( C > \log c_\varepsilon \)

\[ M < 2^n \]
Our bound in the theorem can be traced back to this lemma.

2. Corollary Suppose that $2^n$ is divided into $l$ disjoint pieces $A_1, \ldots, A_l$. Then for some $i$

$$\#(A_i) > n^{c_i}.$$  

(Else $2^n = \sum \#(A_i) \leq M^*$.)

Keeping the situation of the corollary, let $i$

minimal $\exists \#(A_i) > n^{c_i}.$

$$\sum_{1 \leq j < i} \#(A_j) \leq \sum_{j \leq i} n^{c_i} = n^{c_i} (n - 1) \leq n^{c_i} (n - 1) + o(n^{c_i})$$
Thus for an \( \varepsilon' < \varepsilon \) we have:

\[
(\exists \nu) \left( \#(A_{\nu}) > n^{2+\varepsilon'} \sum_{1 \leq j < \nu} \#(A_j) \right)
\]

3. We now indicate the particular partition of strings of length \( \nu \) to which we will apply the theorem.

Let \( |y| < k \). We say that \( y \) is active for \( x \) if

1) \( U(y, x) \) is defined (and equals \( z \) say)

2) \( U(z) = x \)

We assume for the remainder of the proof

1) \( \nu \) is sufficiently large.
2) \( \tilde{H}(x^{x \nu} x) < k \),

* \( k \) is in Lemma 1
It follows that for some p active in x, \( U(p, x) = x \).

There is no analogous notion of \( p \) being active in \( x \) at time \( t \).

Clearly the number of programs active for \( x \) is \( 2^k - 1 \). We let

\[ A_i = \{ x \mid x \text{ has } 2^k - i \text{ active programs} \} \]

(There is an analogous notion of \( A_i \):

\[ A^t_i = \{ x \mid x \text{ has } 2^k - i \text{ active programs at time } t \} \]

Clearly, if \( x \in A^t_i \), then \( x \in A^t_j \) for some \( j \leq i \).

We apply our corollary and get an \( i \) such that \( \bigcup_{j \leq i} A_j = S \), then \( \#(S) \leq \frac{k}{i} n^{2k - i} \). Since \( \#A > n^{2k} \),

\[ \# A > n^{2k} \]
Cost of chanting:

1. \( n \cdot \log n + O(\log \log n) \) (5.4)

2. \( i \cdot 2^k \cdot k + O(\log k) \) (5.4)

3. \( \log n - \log \log n + O(\log \log n) \) (5.4)

4. \( \#(S_i) \leq n^{c(i-1)} + O(n^{c(i-2)}) \)

5. \( \log \#S_i \leq c'(i-1) \cdot \log n \)

6. \( \log \log \#S_i \leq \log j \)

7. \( \#S_i + \#S_{i+1} \leq C'(i-1) + 1 \cdot \log n \) (5.4)

The upshot is that we can describe \( \gamma, \delta \), and \( \#S_n \) as:

\( (C'(i-1) + 3) \log n + O(\log \log n) \) (5.4)
Now proceed as follows.

1) Wait for all elements of $U_{-1}$ to arrive. (This uses $S_1$.)

2) At this point, members of $A_i$ can be identified recognizably, & when $x$ appears, we can compute $x^*.$

Look for the least $x.$ $|x| \geq C'(i)(\log n)$

Such an $x$ must exist.

The description of $x$ we have given takes $O(1) + C'(i-1) + 3\log n + O(\log \log n)$

Thus if $C' > 3,$ and $n$ is sufficiently large, we have a contradiction.

[So in write-up, take $C' = 4,$ $C = 3.$]
Note that in the current version, the "countable Gödel numbering" is eliminated.

There is a variant of this result, suggested to the author by Christin, which seems true and which I am unable to obtain by the method above.

**Conjecture**: There is an infinite sequence \( \{x_n\} \) of strings so that

1. \(|x_n| = n\)

2. \(H(x_n) \sim n\)

3. \(\hat{H}(x_n | x_n) \sim \log n\)

The point is that the \(x_n\)'s we construct might well satisfy \(H(x_n) \leq n / \log \log n\), for example.
III. Relationships between $H$ and $K$

Our goal in the following section is to prove the following formula which relates $H$ and $K$.

1) $H(x) = K(x) + K(K(x)) + O(K^5(x))$

2) $K(x) = H(x) - H(H(x)) + O(H^3(x))$

These formulas are really equivalent since we shall also prove:

3) $K^2(x) - H^2(x) = O(H^3(x))$

4) $H^3(x) \sim K^3(x)$

(where $\sim$ stands for "are asymptotically equal").

Granted 3) and 4), 1) and 2) are clearly equivalent and give a formula for the number of bits needed to make the optimal Kolmogorov program self-delimiting or the cost of the optimal chain program.
Our proof will proceed in three stages. We first get the bound for $H(x)$ in terms of $K(x)$:

5) $H(x) \leq K(x) + H(K(x)) + O(1)$

This is quite easy. Slightly harder is the inequality

6) $K(x) \leq H(x) - H^2(x) + H^3(x) + O(1)$

is more difficult. (Here, e.g., $H^2(x) = H(H(x))$.

We shall use a similar notation for iterated logarithms.)

5) and 6) are close to giving 1), the difficulty being that 5) contains $HK$ rather than $H^2$. However, we can bootstrap our estimate on $K - H$ to get one on $H^2 - HK$ proving 2). At that point 5) and 6) are easy (though somewhat tedious) to obtain, and with them 1).
Lem 1. \( H(x) \leq K(x) + H(K(x)) + O(1) \).

Proof. Let \( U, V \) be respectively the universal Chaitin computer and the universal Kolmogorov computer.

Define a Chaitin style computer, \( C \), as follows:

\( C \) on input \( x \), first simulates \( U \). Thus, \( x = x, x_2 \) with \( x_2 \in \text{dom} U \). \( C \) will compute \( U(x) \). It then reads exactly \( U(x) \) further bits of the input, if possible, getting a word \( x_3 \), computes \( V(x_3) \) and gives it as its output.

Let \( y_3 \) be a minimal Kolmogorov program for \( x_3 \) and \( y_1 \), a minimal Chaitin program for \( ly_3 l \). Then
\[ U(\pi_c^n y_1, y_3) = C(y_1, y_3) = V(y_3) = x \]

Where \( H(x) \leq K(x) + H(K(x)) + 1 \pi_c l \)

**Lemma 2.** Let \( S_n \) be the number of \( x \) such that \( x \in \text{dom } U \) and \( |x| = n \).

Then \( S_n \leq C 2^{-n - H(n)} \)

**Proof.** We shall use a Chaitin's lemma for constructing an instantaneous code for requests. Each time that we first recognize that \( |S_n| \geq 2^k \), we put in a request for a code for \( n \) of length \( n-k \).

Let \( n_k \) be the largest \( k \) such that \( |S_{n_k}| \geq 2^k \).

To be able to cite Chaitin's theorem, we need...
\[ \sum_{n=0}^{\infty} \sum_{0 \leq j \leq n_k} 2^{-(n-j+2)} \leq 1. \]

We now verify this. In the first place,

\[ \sum_{0 \leq j \leq n_k} 2^{-\left(n-j+2\right)} \leq \sum_{n-n_k+2 \leq j < \infty} 2^{-j} \leq 2^{\frac{n_k-n_k}{n_k-n_k}}. \]

On the other hand,

\[ \sum \left\{ 2^{-|x|} : x \in \text{dom}(U) \text{ and } |x| = n \right\} = \]

\[ 2^{-n} \cdot \# S_n \geq 2^{-n} \cdot 2^{n_k+n}. \]

Thus the sum in \( 7 \) is less than

\[ \sum \left\{ 2^{-|x|} : x \in \text{dom}(U) \right\} \leq 1. \]

Let \( C \) be the C-language compiler that implements this code. Let \( \theta \in m = 1 \llbracket 1 \rrbracket. \quad \therefore \]

\( H(n) \leq m + n-n_k+2, \)
since $C(y) = \frac{1}{y}$ for some $UL_{\pi_c}^c - y) = C(y)$.

for some $y$ of length $n - \eta + 2$. From (2),

$s \leq n \left( H(n) + (n + 3) \right)

from which the lemma follows.

Lemma 3. $K(x) \leq H(x) - H^2(x) + H^3(x) + O(1)$

Proof. We begin by describing a certain $M$ Kolmogorov-style computer $C, M$ will depend on a fixed constant $C$ to be

$M$, on input $x$, first simulates the universal Chaitin computer $U$. If $U$ halts on input $x$, we must have $x = x_x \in \text{domain}(U)$. 
1 M computes $U(x_i) = d$, say, and lets $N = \lfloor x_i \rfloor + d - C$. M next interprets $x_2 = x_1$ in the obvious way. Next M proceeds to list the elements of domain $(U)$ of length $n$ in some definite order.

(E.g. by the order in which they are computed by $U$, using lexicographical ordering to resolve ties.)

If there are $g$ elements of this set, let $y$ be the $j^{th}$ element. (If there are $< g$ elements, $M$ will be undefined at $x_i$.) M then outputs $U(y)$.

We claim that one can choose $C$ sufficiently large that for every $x_i$, there is a $Z$ of length
\[ H(x) = H^2(x) + H^3(x) + C \quad \text{such that} \]

\[ M(Z) = 0. \quad \text{From this the lemma clearly follows.} \]

Let \( x \) be given, let \( x_1 \) be the minimal Chaitin program for \( x \). Let \( x_2 \) be a minimal Chaitin program for \( |x_1| = H(x) \). Let \( x_3 \) be a minimal Chaitin program for \( |x_2| = H^2(x) \). The \( x_2 \) we will construct will have the form \( x_3 \triangleright W \).

\[ |W| = H(x) - H^2(x) + C. \quad \text{(So the length of} \]

\( x_2 \) will be \( |W| + |x_3| = |W| + H^3(x) \), as claimed.) To construct such a \( x_2 \), we need to know, with no digits is in \( \text{Line}^2 \).

9) \( S_{H(x)} \leq 2 \)

This is clear from Lemma 2, if we take \( C \geq \log_2 C_2 \).
So we can choose now \( M \) on input \( 2^n \) will compute
\[ \text{the value of } y \text{ to be } H(x). \] Since \( q \) holds, we can choose \( w \) so that \( y \) is the \( w \)th member of \[
\{ y \in \text{dom } U : |y| = H(x) \}.
\]
But then \( M \) will evaluate \( y \) to be \( x \) and will finally output \( x \) as its answer. Thus \( M(\tau) = x \), and the lemma is proved.

**Lemma 4**: \[ |H(x_1) - H(x_2)| \leq H(\lfloor x_1 - x_2 \rfloor) + O(1) \]

**Proof**. Given \( x \) a constant easily computed computer that on input \( y \), tries to parse \( y \) as \( y_i \) \( y_j \) with \( y_i, y_j \in \text{dom } (U) \) and the output \( U(y_i) + U(y_j) \). It follows that
\[ H(a+b) \leq H(a) + H(b) + O(1) \]

and similarly for \( a \geq b \), \( H(a-b) \leq H(a) + H(b) + O(1) \)

From this the lemma follows easily.

**Lemma 5** \( H(x) = K(x) + H^2(x) + O(H^3(x)) \)

\textit{Proof} Let \( D(x) = H(x) - K(x) + H^2(x) \). By

Lemm 1, \( D(x) \leq H(K(x)) - H^2(x) + O(1) \).

By Lemm 2, \( -H^3(x) \leq D(x) \).

By Lemm 4, \( H(K(x)) - H^2(x) \leq \)

\[ H(|K(x) - H(x)|) + O(1) \]

\[ H(|D(x)| + H^2(x)) + O(1) \leq \]

\[ H(D(x)) + H^3(x) + O(1) \]

Putting the results of the first two paragraphs together
we get

10. \[ |D(x)| \leq H^3(x) + H(|D(x)|) + O(1). \]

I say that from 10) it follows that

1) \[ |D(x)| \leq 2H^3(x) \]

for all but finitely many \( x \), whence \( D = O(H^3) \),

as claimed. (Note that \( H(x) \geq 1 \) for all \( x \) else \( \phi \in \text{cln}(U) \) & \( \phi \notin C(U) \).

\[ \exists y \mid |D(x_n)| \rightarrow 2H^3(x) \text{ when } x_n \rightarrow y \text{ for } \]

\( y \rightarrow \infty \). Thus \( |D(x_n)| \rightarrow \infty \text{ when } \]

\( H(|D(x_n)|) \leq \log (|D(x_n)|) = o(D(1/n)) \).

Dividing 10) by \( |D(x_n)| \), we get

\[ 1 \leq \frac{H^3(x)}{|D(x_n)|} + \frac{H(|D(x_n)|)}{|D(x_n)|} + O\left(\frac{1}{|D(x_n)|}\right). \]
The last two terms are $o(1)$ and the first is

less than $\frac{1}{2}$. Thus it is small, so we must have

$$D(x) \leq 2H^3(x) \quad \text{for all but finitely many} \ x.$$

In the following we will need estimates

for $Hf(x)$ when $f = O(g)$. We shall

use the estimate $Hf(x) = O(\log g(x))$, which

clearly

$$Hf(x) \leq \log f(x) \leq \log g(x) + O(1)$$

($f(x) = O(g(x))$).

We have now established 2). Note first that by

applying $H$ to both sides of 2), we get

12) $HK(x) = H^2(x) + O(\log H^3(x))$.

Next substitute $Kx$ for $x$ in 2). We get
\( K^2(x) = HK(x) - H^2K(x) + O(H^3K(x)) \)

Now using Lemma 4 in 12, we get

\[ H^2K(x) = H^3(x) + O(\log H^3(x)). \]

So \( H^2K(x) \) is \( O(H^3(x)) \). It follows that the two right hand terms in 13 are \( O(H^3(x)) \).

Thus \( K^2(x) = HK(x) + O(H^3(x)) \) and from this and 12, 3) follows. Finally we know \( K \sim H^* \) when, since \( H^2 \to \infty \),

\( K^3 \sim HK^2 \). But applying \( H \) to 3),

\[ HK^2 - H^3 = O(\log H^3). \]

Where \( HK^2 \sim H^3 \), establishing 4).

* we suppress mention of \( x \).
IV. Relations between H-randomness and K-randomness of finite sequences.

In this section we study two related questions. First, Kolmogorov and Chaitin have proposed two related notions of randomness for finite sequences of 0's and 1's, which we will define precisely in a few moments. We prove that every Chaitin random finite string is Kolmogorov random but that the converse is false.

The refutation of the converse involves a procedure for constructing counterexamples. These also shed light on the question discussed in the preceding section of formulas that compute
H(x) from K(x) or conversely. The formula of the last section have an error term $O(H^3(x))$.

We show in this section that this error term is, in a certain sense, best possible.

Let us recall the notion of Kolmogoroff randomness for finite strings. It is based on two facts:

1) $K(x) \leq |x| + C_0$, in all $x \in \Sigma^*$.

2) $\# \{ x : |x| = n \ & \ K(x) < n + C_0 - \log 3 \} = O(2^{-n-3})$.

We define $M_K(x) = |x| + C_0 - K(x)$.

Then $0 \leq M_K(x) \leq |x| + C_0$. Roughly speaking, $M_K(x)$ measures the degree of non-randomness of $x$.

Kolmogoroff random strings are those for which $M_K(x)$
is small.

The analogous facts in the Chaitin context are as follows.

1'): \( H(x) \leq 1x1 + H(1x1) + C_4 \)

2'): \( \# \{ x : |x| = n \land H(x) \leq n + H(1x1) + C_4 - 1 \} \)

\( = O(2^{n-1}) \).

We put \( m_H(x) = 1x1 + H(1x1) + C_4 - H(x) \).

The intuitive interpretation of \( m_H \) is similar to that of \( m_K \).

We shall prove

(i) \( m_H(x) \geq m_K(x) + O(\log_2 m_K(x) + 1) \).

It follows from (i) that if \( m_H \) is small

\( m_K \) must be small, and this is the sense in which
every Chaitin random real is Kolmogorov random.

It is easy to prove, using the formulae relating
\[ 1 \text{ and } K \text{ of the last section that} \]
\[ m_K(x) \approx m_H(x) + O(\log^2 1x1). \]

Thus, if \( m_H(x) \gg \log^2 1x1 \), \( m_K(x) > 1 \). However, we shall construct an infinite series of strings \( w_n \)
with the following properties:

a) \( |w_n| \to \infty \text{ as } n \to \infty \).

b) \( K(w_n) = |w_n| + O(1) \).

(Thus the \( w_n \)'s are K-random, for \( n \) large.)

c) \( \lim_{n \to \infty} \frac{m_H(w_n)}{\log^2 |w_n|} = 1. \)

It is in this sense that we show that K-randomness
does not entail H-randomness.

Suppose that \( A \) is a finite set of integers. The


t hickness of \( A = \max(A) - \min(A) \). If \( A \) is empty,


t hickness \( (A) = 0 \).

We remark that it would be easy to modify the
proof of the preceding section to show

\[
\lim_{n \to \infty} \frac{H(n) - K(n) - H(H(n))}{H^3(n)} \leq 1.
\]

It follows from the example promised in the
preceding paragraph that this lim sup is \( = 1 \).

The same method provides a counterexample
to various improvements in the error term of the

formulas relating \( H \) and \( K \).
Also rules out

\[ H(v) = K(v) + H(k(x)) + o(1) \]
We shall prove the following: there are

infinite seq. of strings \( y_n, z_n, w_n \): 

\[
\lim_{n \to \infty} \frac{H(y_n)}{\log \log H(y_n)} = 1.
\]

1) \( K(y_n) = K(z_n) + O(1) \)

2) \( H(y_n) = H(w_n) + O(1) \)

3) \( \lim_{n \to \infty} H(y_n) = \infty \).

To see how this rules out improvements
In the results of the preceding section, we show that the relation

\[ H(x) = K(x) + K^2(x) + K^3(x) + O(K^4(x)) \]

is false. Indeed, if this were true, since \( K(y_n) = K(\varepsilon) + O(\varepsilon) \), we would have

\[ H(y_n) = H(\varepsilon) + O(K^*(y_n)) \]

A fortiori,

\[ H(y_n) = H(\varepsilon) + O(\log^3 H(y_n)) \]

Hence \( \lim_{\log \log H(y_n)} \frac{H(y_n) - H(\varepsilon)}{\log H(y_n)} = 0 \),

contrary to our promised example.

We complete these introductory remarks by indicating how the example of the K-random walks...
that are not $H$-random are considered. To start with,

let $k_n$ be an increasing sequence of integers such that

$$H(k_n) = O\left( \log^{3^k} k_n \right).$$

(For a given $k$ such a sequence is easy to construct.) Pick $x_n$ so that

1) $|x_n| = k_n$

2) $H(x_n | x_n) = \log k_n + O(\log \log k_n).$ (This is the main result of Chapter II guarantees such $x_n$'s.)

Let $y_n$ be the minimal Kolmogorov program for $x_n$ given $x_n$. Let $z_n$ be a suitably random string of length $x_n$. Then $x_n = y_n z_n$ will turn out to be $K$-random, but not $H$-random.
IV.1 Let us now show that

\[ m_H(x) \geq m_K(x) + O(\log m_K(x) + 1) \]

Indeed, let \( L = m_K(x) \),

\[ K(x) \leq 1x1 + C_4 - L \]

Thus \( H(K(x)) = H(1x1) + O(\log L + 1) \)

So \( H(x) \leq \lg K(x) + H(K(x)) + O(1) \)

\[ \leq 1x1 + H(1x1) + O(L) + O(\log L + 1) - L \]

So \( m_H(x) = L + O(\log L + 1) \)

as was to be proved.

IV.2. The following technical lemma will play a key role

in our construction of counterexamples:

**Lemma** \((\forall j)(\exists L)(\forall n)(\forall y): I \)
\[ K(y|n) \leq 2^{n-j} \]

\begin{align*}
\{ z \in \{0,1\}^n : K(y^z) \leq 1 \} \\
\text{less than} \\
\text{has at most} 2^{n-j} \text{ elements.}
\end{align*}

So, roughly speaking, if \( y \) is \( K \)-random over \( z \), then for many \( z \)'s of length \( n \), \( y^z \) is \( K \)-random.

Our proof will proceed by denying the conclusion and then showing that \( y \) has a short \( K \)-program.

Consider the following special purpose \( K \)-minimizer computer \( M \).

On inputs \( x \) \& \( n \), \( M \) first tries to prove \( x \) is
\[ x_1, x_2, x_3, x_4 \]

where \( x_1, x_2, x_3 \) are in \( \text{dom} \ U \). If so, it sets
\[ j = U(x_1) \]
\[ e = U(x_2) \]
\[ m = U(x_3) + 1 \times 4! \]

M now proceeds to enumerate those \( y \)'s such that
\[ 1) \quad |y| = m \]
\[ 2) \quad \text{For at least } 2^{n-3} \times 3^3, \]
\[ K(y^{x_2}) \leq m + n - L \]

this means \( x_2 \) is a number \( \leq 2^{x_2} \).

M also considers \( x_4 \) as a number \( \leq 2^{x_4} \).

and outputs the \( m \) \( x_n \)'s such \( y \) if it exists.

let \( C_0 = \text{sim} \ M \).
Now there are at most \(2^{n+m-b+1}\) pairs \((y, z)\) such that \(K(y, z) \leq n+m-b\). Thus there are at most \(2^{n+m-b} y's\) which occur in this way with \(2^{n+m-b} z\) 's. Thus, if we let \(x_1, x_2, x_3, x_4\) denote respectively \(j, k, l-j, l+j\), and the position of the same particular \(y\) which is \(2^{n+m} z\) 's associated in the list of all such \(y's\), we get a \(K\)-program for \(y\) of length

\[
S \leq m + l + j + H(l) + H(j) + H(l-j) \leq m - l + O(j) + O(\log l + 1).
\]

where the constants are independent of \(n, m\). We now select \(l\) so that \(l \geq j + O(j) + O(\log l + 1) + 1\).
Then we get a $K$-program by the same technique.

So let $K(y) \leq |y| - 1$, and the particular $L$ just chosen we must have

$$K(y^{\oplus x}) \leq |y| + n - L.$$ 

IV.4. We now begin our construction of the $w_i$'s which are $K$-random but not $H$-random. To start off,

let $x = 2^x$. Then clearly $H(x) = H(i) + O(1) = O(\log^3 x)$. By the main result of chapter II, select $n_0$ so that

1) $|n_i| = li$

2) $H(n_i \mid n_0) \geq \log x + O(\log^2 x)$
Now by Chaitin's weak

\[ \tilde{H}(n, \nu) = \tilde{H}(H(n, \nu) \mid \nu) \leq \log \frac{\tilde{H}(n, \nu)}{\nu} + O \left( \frac{1}{\nu} \right) \leq \log \frac{1}{\nu} + O \left( \frac{1}{\nu^2} \right) \]

\[ \log \frac{\tilde{H}(n, \nu)}{\nu} + O \left( \frac{1}{\nu^2} \right) = \log \tilde{H}(\nu) \leq \log \nu + O \left( \log \nu \right) \]. So 2) can be sharpened to

3) \[ \tilde{H}(n, \nu, \nu) = \log \nu + O \left( \log \nu \right) \]

Let \( y_\nu \) be the minimal \( K \)-program for computing \( n, \nu \) from \( \nu \). Then \( |y_\nu| = K(n, \nu, \nu) \]

\[ \tilde{H}(n, \nu, \nu) + O \left( \log \tilde{H}(n, \nu, \nu) \right) \] by arguments using the relative version of the identity

\[ K \leq H + o(1) \leq K + HK + O(1) \]
So \( |y_1| = \log k + \Theta(\log \log k) \).

Moreover, \( K(y_1, \eta) = |y_1| + O(1) \)

since \( \eta \) is the \( K \)-minimal program that computes \( y_1 \) from \( \eta \).

We now quote the lemma of IV.3 and get that we can choose \( z_0 \) so that

1) \( |z_0| = \eta_0 \)

2) \( K(y, z_0) = |y| + \eta_0 + O(\log \log k) \)

3) \( H(z_0) = \eta_0 + H(\eta_0) + O(1) \).

We put \( w_0 = y \cap z_0 \).

Then 1) of the preceding paragraph says that \( W_0 \) is \( K \)-random, and it is evident that \( |w_0| \to \infty \) with \( 1 \).
We next compute \( H(\|w\|) \). Indeed

\[
H(\|w\|) = H(\|y_1 + n + O(1)\|) = H(n) + O(H(\|y\|)) + O(1)
\]

But \( H(\|y_1\|) = H(\log l + O(\log^2 l)) = H(\log l) + O(\log l) = \log l + O(\log l) \).

Thus \( H(\|w\|) = H(n) + O(\log l) \).

Next, we are going to establish an upper bound on \( H(\|w\|) \):

\[
H(\|w\|) \leq H(\|w_1 + z\| + H(z) + O(1)) \leq H(\|y_1 + z\|) + n + O(1)
\]

Now we have \( n = 121 \), so \( H(n, 121) = O(1) \).
Also, $H(n \cdot 1n_e) = O(1)$, so

$$H(n \cdot 1z_e) \leq H(n \cdot 1n_e) + H(1z_e) + O(1)$$

$$= O(1).$$

Now by an easy relativization of a result of Chaitin's, there are $O(1)$ programs of length $|y|_e$ for $n$ from $n_e$

where $H(y, 1z_e) \leq H(y, 1n_e, n_e) + O(1) \leq H(1y, 1n_e, n_e) + O(1) \leq O(\log |y|_e + O(\log \log |y|_e)) = O(\log^2 |y|_e + O(\log^3 |y|_e))$

The upshot is that
\[ H(w_c) \leq H(n_c) + \log l_c + n_c + H(n_c) + O(\varepsilon), \log 3n_c) \]

So, \[ 1w_c + H(1w_c) - H(w_c) \geq \]

\[ \log l_c + n_c + O(\log^2 l_c) + H(n_c) - [O(\log^2 l_c) + n_c + H(n_c)] = \]

\[ \log l_c + O(\log^2 l_c) \]

Thus, \[ m_H(w_c) \geq \log l_c + O(\log^2 l_c) \]

\[ \geq \log(\log n_c) + O(\log^3 n_c) \]

\[ \geq \log(\log n_c) + 0(\log^3 n_c) \]

Which proves the claim that

\[ \lim_{1w_c} \frac{m_H(w_c)}{\log^2 1w_c} \geq 1 \]
As we remarked earlier, it is easy to get a bound on $\mathcal{H}$ in the other direction, using the methods of proof of \textbf{III}. (In revised version of this, should be explicitly stated. Also in revised version of \textbf{Z}, should state upper bound on $\tilde{H}(\mathcal{H}, l_w)$ that follows from Chernoff’s work.)

The upshot is:

$$H(w_c) = l_w l_1 + H(l_w l_1) + \log Q_w Q_1$$

$$Q_w \sim \log l_w l_1$$

Now if $z_c$ is $H$-random of length $l_w l_1$, then & since $z_c$ will also be $K$-random,

$$K(z_c) = l_w l_1 + O(1), \quad H(z_c) = l_w l_1 + H(l_w l_1) + O(1)$$
This is with \( \mathbf{H}(z) = H(z) + \mathbf{H}(z) \) and

\[
\frac{H(2z) - H(1z)}{\log H(1z)} \to 1 \quad (\text{as} \quad 1 \to \infty).
\]

We need a lemma to get this. Say:

\[
H(u_c) = H(u_c) + O(1).
\]

\( \exists C: \)

Lemma. Let \( n \geq 0 \). Then \( \exists m: \)

\[
| m + H(m) - n | \leq C.
\]

Proof. We know

\[
| H(x + D) - H(x) | \leq H(D) + C_0.
\]

Thus if \( D \) is large enough, \( D > 2 \log D + C_0 \geq \)

\[
H(D) + C_0. \text{ Fix } D. \text{ Let now } f(x) = x + H(x).
\]

By what we have said, \( f(D) + f(x + D) > f(x). \)
Also, \( f(x+D) \leq f(x) + 2D \).

Now given \( m \), pick \( n \): \( |f(n) - m| \) is minimal.

Then clearly \( |f(n) - m| \leq 2D \) else one of \( f(n+D), f(n-D) \) would be closer to \( m \).

Now pick \( u_n \): \( \theta u_n \) is H-random. 

1) \( H(u_n) = |u_n| + H(1|u_n|) + O(2) \)

2) \( |u_n| + H(1|u_n|) = H(u_n) + O(2) \)

So of course \( H(u_n) = H(u_n) + O(2) \).

Now \( K(u_n) = |u_n| + O(2) \).

It remains to compute \( |u_n| \).

\[
|u_n| + H(|u_n|) = |u_n| + H(|u_n|) + \frac{|u_n|}{|u_n|}. 
\]
Work \( \|w_1\| = \|w_1\| + R_0 \)

\[
H(\|w_1\|) = H(\|w_1\|) + O(\log R_0)
\]

Thus \( R_0 + O(\log R_0) \sim \log^2 \|w_1\| \)

It follows that \( R_0 \sim \log^2 \|w_1\| \)

Thus, \( K(\|w_1\|) \)

\[
K(\|w_1\|) - K(\|w_1\|) = R_0 \sim \log^2 \|w_1\|
\]

So all the results claimed in the introduction to this section are now proved.
The result claiming correctness on a set of upper density is not proved, only that it is within \( O(1) \) on a set of upper density 1. I should try to let us begin by recalling the following result of Chaitin. Let \( f : \omega \to \omega \) be recursive. Then

1) If \( \sum_{j=0}^{\infty} 2^{-f(j)} < \infty \),

then

\[ H(n) \leq f(n) + O(1), \quad \text{all } n. \]

2) If \( \sum_{j=0}^{\infty} 2^{-f(j)} = \infty \), then

\[ H(n) > f(n) \quad \text{for infinitely many } n. \]

In the second case, it is not hard to construct \( g : \omega \to \omega \), recursive, such that \( g(n) - f(n) \to \infty \) as \( n \to \infty \) and \( \sum_{j=0}^{\infty} 2^{-g(j)} = \infty \). One can even show that \( g \) can be chosen weakly monotone increasing if it is...
weekly monotone. Thus the conclusion is that strengthening to

\[ \lim_{n \to \infty} (H(n) - f(n)) = \infty. \]

This remark also shows that any recursive r.o. lower bound can be infinitely often improved (by replacing \( f \) by \( f + g \) as above.)

We show that the situation in the upper bound case is quite different. We shall construct a recursive function \( h: \omega \to \omega \) such that

1) \( h(n) > H(n) \) for all \( n \)

2) \( h(n) = H(n) \) for infinitely many \( n \).

In fact, by an easy argument, it can be strengthened to
\[ \lim \frac{1}{n} \# \{ m : m \leq n \land h(m) = H(m) \} \geq 0. \]

Our construction of \( h \) and our proof of (1) will be quite non-constructive, and in some sense, \( \{ n : h(n) \leq H(n) \} \) is quite sparse. We shall prove, in this direction that there is no effective procedure which given \( n \) guesses a finite set \( D_n \) such that:

1. \( m \in D_n \Rightarrow m \geq n \)
2. \( (\exists x \in D_n) \ h(x) \leq H(x) + C \)

(Our proof will hold for any recursive upper bound on \( h \).) It will follow easily that

\[ \lim \frac{1}{n} \# \{ m : m \leq n \land h(m) \leq H(m) + C \} = 0. \]

The basic idea behind our construction is to choose \( f \) so that \( \sum 2^{-f(n)} \) converges "as slowly" as possible.
One byproduct is the construction of a recursive convergent sequence of positive reals \( (a_n) \rightarrow q \) if \( (b_n) \rightarrow q \) is any recursive sequence of positive reals, then

\[
\lim_{n \to \infty} \frac{a_n}{b_n} > 0.
\]

We remark that with a little extra work we can arrange that \( f \) be monotone increasing. We sketch his modification without giving complete details.

V. 1. We begin by giving the construction of the slowest growing convergent sequence, \( a_n \). (Since there is no sort of uniqueness proved, we should really say "maximally slow growing" but we permit one to talk ourselves to be sloppy.)
Let us recursively enumerate all pairs consisting of

1) a partial recursive function mapping \( \omega \to \omega \) to the positive rationals, \( h \).

2) A positive integer \( \alpha \).

\[ \exists \alpha \beta \gamma \geq \sum_{i \in \omega \cap \omega} h(i) \]

Such an enumeration is easy to construct by standard techniques.

The first standard enumeration of partial recursive functions
by partial recursive functions \( \uparrow \) and closing off the partial recursive function \( \uparrow \) it is about to violate \( \alpha \) 1)

We define a recursive function \( a : \omega \to \omega^+ \) as follows: Case 1 \( n = 2^3 3^5 \), \( h(i) \text{ has } a_n \).
is first computed in exactly $k$ steps. Then

$$a_n = 2^{-(n+2)} \cdot h_k(j).$$

Case 2. Suppose Case 2 is occurring at stage $n$ for the $m$th time. Put

$$a_n = 2^{-n-1}.$$

We claim first $\sum a_n < 1$.

Indeed Case 2 contributes at most $\frac{1}{2}$ to the sum. Numbers of the form $2^i 3^j 5^k$ for even $i$ contribute at most

$$2^{-i-2} \cdot \sum_{j \in \mathbb{N}, k \in \mathbb{N}} h_k(j) < 2^{-i-2}.$$

Less than

Thus Case 1 contributes at most $\frac{1}{2} + \frac{1}{2^2} + \cdots = \frac{1}{2}$ to the sum. So $\sum a_n < 1$. 

}\vspace{10pt}
Suppose \( \{b_i\} \) is recursive and \( \sum b_i < \infty \).

Pick \( \lambda \in \mathbb{Q} : \sum a_i < 1 \).

Now \( \lim \frac{a_i}{b_i} = \lambda \Rightarrow \lim b_i = \infty \).

Thus, without loss of generality, we may assume \( \sum a_i < 1 \).

So \( b_i = h_n(i) \) for some fixed \( n \).

If \( \lim \frac{a_i}{b_i} = 0 \),

then \( a_i < 2^{-(n+3)} b_i \) for \( i \) large enough.

We now define an infinite series \( c_n \):

\[
c_n = 2^{-n} 3^{-n} 5^{-n}
\]

where \( h_n(i) \) is computed in exactly \( n \) steps.
By construction, \( a_n = 2^{-n-2} b_n \).

\[
 b_n > 2^{k+3} a_n \geq 2^{-k} b_n.
\]

But then \( \sum b_n \) diverges exponentially contradicting our assumption that \( \sum b_i \) is convergent.

\[ \sum_{i=0}^{\infty} b_i \]

5.2. Our construction of \( f \) will be similar to that of the \( a_i \)'s. Again \( f \) at various stages will be imitating \( H \) at earlier stages, and if \( H \) too closely imitates \( f \), we would get \( \sum 2^{-H(i)} = \infty \), contrary to fact.

\[ \text{Let } \langle l_n \rangle \text{ be an enumeration with } l_n \leq n \text{ so that } l_n < n \]

repetitions of all pairs \( \langle l, j \rangle : j \geq H(l) \).

\[ \text{Let } \langle l_n \rangle \text{ be an enumeration with } l_n \leq n \text{ so that } l_n < n \]

repetitions of all pairs \( \langle l, j \rangle : j \geq H(l) \).
\[ g(n) = \sum_{j=0}^{\infty} 2^{-g(n)} = \sum_{j=0}^{\infty} \sum_{i=0}^{j \geq H(i)} 2^{-i} = \sum_{i=0}^{\infty} 2^{-H(i)+1} \leq \frac{1}{2} < 1. \]

It follows that \( g(n) \geq H(n) - C \) for some positive \( C \), by the result of Chaitin recalled at the start of this section.

I say that \( g(n) < H(n) + 3 \) for infinitely many \( n \).

Suppose to the contrary that \( g(n) \geq H(n) + 3 \) for \( n \geq k \). Define an infinite sequence of indices \( b_k, j_n \) as follows

\[ \sum_{j=0}^{\infty} 2^{-g(n)} = \sum_{j=0}^{\infty} \sum_{i=0}^{j \geq H(i)} 2^{-i} = \sum_{i=0}^{\infty} 2^{-H(i)+1} \leq \frac{1}{2} < 1. \]
$f_n = H(c_n)$;

$c_{n+1}$ is the stage at which $<c_{n+1}, c_n>$ is listed.

Then $c_n < c_{n+1}$ &

$g(c_{n+1}) = H(c_n) + 2$.

Thus $H(c_{n+1}) \leq g(c_{n+1}) - 3 \leq H(c_n) - 3$.

So $H(c_n)$ is an infinite decreasing sequence of non-negative integers, and we have reached a contradiction.

Thus $H(n) \leq g(n) + C$ for all $n$.

Let $H(n) > g(n) - 3$ for infinitely many $n \in \mathbb{Z}$.

Pick $C'$ minimal such that $H(n) \leq g(n) + C'$ for infinitely many $n \in \mathbb{Z}$.

Thus $-2 \leq C'$. But $C > E$.

Evidently, $H(n) = g(n) + C'$. For infinitely many $n$. (Etc.)
C' could be replaced by C' - \ell \). Now put \( b(x) = g(x) + C' \). Then \( h \) now satisfies 1) and 2) of the introduction.

\[ \text{If we desired to get } h \text{ monotone we would construct a monotone } g \text{ and arrange to assign to each pair } <i,j> \text{ a block } B_{i,j} \text{ so that} \]

\[ \sum_{i \in B_{i,j}} 2^{-g(i)} = 2^{-g(j) - 2} \]

2) The \( B_{i,j} \)'s are pairwise disjoint \&

\[ c < \min(B_{i,j}) \]

We would then construct assuming

\[ H(\ell) \leq g(\ell) - 3, \text{ an infinite series of blocks. So:} \]
\[ \sum_{s \in S_{5,5}} 2^{-H(s)} \geq 2 \left( \sum_{s \in S_{5,5}} 2^{-H(s)} \right) \]

which would contradict \( \sum_{j \in \mathbb{G}} 2^{-H(j)} < 1 \).

**V. 3.** Recall from clustering work that \( h_{\mathcal{C}} \in \mathcal{C} \):

1. \( h(n) \leq H(\log n) + C_0 \),

2. \( h(n) \geq H(\log n) - C_0 \),

Thus, if we let \( h_{2}(n) = h(\log n) + C_0 \),

then \( h_{2}(n) \geq H(n) \).

Also, if \( h(\log n) = H(n) \), then

\( h_{2}(n) \leq H_{\mathcal{C}} \cdot H(n) + 2C_0 \), in which case

Thus, for some \( n \leq d \)

\( 2^{m} \) of the \( n \) is \( (2^{m}, 2^{-m}) \), and thus for \( h_{2}(n) \)

\( \leq 2^{m} \cdot (\log 2)^{m+1} \).
Thus for some $j$ and $m_l, l_j = m_l,$

\[ \text{if } n < 2^{-m_l} : \ h_2(n) = H(m_l) + j \frac{n}{2} \geq \frac{1}{8}c_0.2^{\frac{n}{3}} \]

This establishes $2^*$ with $h$ replaced by $h_2.$

(We may obviously change $2^*$ of both as well by replacing $h^* \text{ if necessary by } h^* - \epsilon.$)

V.4. We now let $h: \omega \mapsto \omega$ be an arbitrary recursive upper bound for $H.$ We shall assume that we have an effective procedure that assigns to each $n \mapsto \bar{h}(n)$ finite set $D_n$ so that

1) $\min D_n > n$

2) $\exists x \in D_n: h(x) + c \leq H(x).$
We shall proceed to derive a contradiction.

First, we define an infinite series \( E_0 \) of finite sets by putting \( E_0 = D_0 \).

\[
E_{n+1} = D_{m_n} \text{ when } m_n = \max(E_n).
\]

Thus, in addition to 2), the \( E_n \)'s satisfy

3) \( i < j \rightarrow E_i \cap E_j = \emptyset \).

Let \( Y_n = \sum_{j \in E_n} 2^{-g(j)} \).

Now \( \sum_{j} 2^{-g(j)} \leq \sum_{j} 2^{-h(j)} < 2 \).

So 3) entails \( \lim_{n \to \infty} Y_n = 0 \).

We now define a Chaitin machine \( M \) that runs as follows:
1) M searches until it finds an $n$ such that
   
   \[-\log_2 y_n \geq \sin M + A \cdot C + 1.\]

   (Note that by a standard use of the recursion theorem, M may be allowed to know its simulation cost. Since $f, E$ are recursive so is $y_n$. Since $y_n \to 0$ as $n \to \infty$, M's search will succeed.)

2) M now constructs an instance $S$ instantaneous code for $E_n$ so that the code word for $x$ has length \( A(x) - \sin M - C - 1 \).

   (This is possible by $(\star)$ & Church's Theorem.)

3) M now examines $S$ and $y_n$. If $y_n \geq A(x)$
code words of $S$, then $M$ outputs the corresponding

element of $E_r$.

It follows that for $x \in E_r$,

$$H(x) \leq h(x) - A - 1,$$

contrary to our assumption 2.

Now if

$$\lim \frac{1}{n} \# \{ m < n : h(m) - A \leq H(m) \} > c,$$

then we could take $D_n = [E_r, \lambda n \lambda]$ where

$\lambda$ is chosen so that $\frac{n-1}{\lambda n} < c$, contrary to the theorem just proved. This all claims made in the

introduction to this section have been established.
Enclosed is a revised version of Chapter II.
I should note that precisely this result was published in a Russian journal (by Gac, I believe). Oddly, I don't know the reference; Greg Chaitin (IBM Watson Labs), probably would have it.

Some background to put the result in perspective (See also the remark at the end of this letter.) Let \( \phi(n) = \sup \{ \tilde{H}(x^x | x) : |x| = n \} \). Chaitin had previously shown that \( \phi(n) \to \infty \), as \( n \to \infty \).

His proof was non-constructive, and did not get a recursive lower bound tending to \( \infty \).

It is not hard to show \( \tilde{H}(x^x | x) \leq \log n + \log \log n + O(\log \log \log n) \) (if \( |x| = n \)). (This lower bound is fairly sharp.) In fact let \( w = \)
but finitely many $n$.

Let $C_1$ be chosen with $2 < C_1$; $\log C_1 < C$

Let $k = \log n \log \log n - C_1$. Let $l = 2^k$.

Let $M = \sum_{1 \leq i \leq k} n^{c_i}$. Then $M \leq 2^n - 1$, for all sufficiently large $n$.

Hence, if the set of strings of length $n$ is partitioned into $l$ pieces, then for some $i \leq l$, \(#(A_i)\) >

If $i$ is least such, then

\[ \#(U_{j < i} A_j) \leq \sum_{1 \leq j \leq i} n^{c_j} \]

\[ \leq 2 \cdot n^{c_i (i-1)} \] (for $n$ sufficiently large.)

Hence, if we pick $\varepsilon > 0$ sufficiently small,

then

\[ n^{2 + \varepsilon} \cdot \#(U_{j < i} A_j) < \#(A_i). \]
Next I wish to describe the particular partition of strings of length \( n \) that I will apply these remarks to. Let \( x \) be a string of length \( n \). By our assumption that \( \widetilde{A}(x^*|x) < k \), for some \( p \) with \( 1 \leq p \leq k \), we have \( U(p|x) = x^* \); \( U(x^*) = x \).

**Definition.** Let \( |x| = n \), \( |y| \leq k - 1 \). Then \( y \) is active for \( x \) if \( U(y|x) \) is defined \((d = z)\) and \( U(z) = x \).

Let \( 1 \leq j \leq 2^k \). Then \( A_j = \{ x : \text{There are precisely } 2^k - j \text{ programs active for } x \} \).

At a given stage \( t \), we can define an analogous notion of active for \( x \) at stage \( t \), and get a corresponding notion partition \( A_j^t \). Note that \( x \in A \) implies that for all \( s \geq t \), \( x \in A_i^s \) for some \( i \).
Also \( A_{2^k} = \emptyset \) since \( x^* \) gives rise (as remarked earlier) to at least one program active for \( x \).

Let \( i \) be least such that \( \#(A_i) > n^{C_i} \).

We code finite sequences of integers into integers in some standard way. \( \langle x_0, \ldots, x_{n-1} \rangle \) is the seq. no. of the sequence \( \langle x_0, \ldots, x_{n-1} \rangle \). Let \( S_i = U_{j<i} A_j \)

**Lemma.** \( H(\langle n, i, \#(S_i) \rangle) \leq \left\lceil C(i-1) + 2 \right\rceil \cdot \log n + O(\log \log n) \).

**Proof.**
1. \( H(n) \leq \log n + O(\log \log n) \)

2) \( H(\langle n, i \rangle) \leq H(n) + H(i) + H(n) \leq H(n) + H(i \ln n) + \Theta(1) \).

(Recall: \( H(x|y) = \min z : U(z | y^*) = x \); \( \tilde{H}(x|y) = \min z : U(z | y) = x \) )
\[ \tilde{H}(\langle i \parallel n \rangle) \leq \tilde{H}(\langle i \parallel k \rangle) + \tilde{H}(k \parallel n) + O(1) \]

Since clearly \( \tilde{H}(k \parallel n) = O(1) \), \( \tilde{H}(i \parallel n) \leq \tilde{H}(i \parallel k) \)

3) Since \( i \leq 2^k \), clearly \( \tilde{H}(i \parallel k) \leq k + O(1) \)

So \( H(\langle \langle n, i \rangle \rangle) \leq 2 \log n + O(\log \log n) \)

4) By previous remarks,

\[ \#S_i \leq 2 \cdot n^{c(i-1)} \]

Thus \( H(\langle \langle n, i \rangle, \#S_i \rangle \rangle) \leq \)

\[ \tilde{H}(\#S_i \parallel \langle \langle n, i \rangle \rangle) + H(\langle \langle n, i \rangle \rangle) + O(1) \]

\[ \leq \tilde{H}(\#S_i \parallel 2 \cdot n^{c(i-1)}) + \tilde{H}(2 \cdot n^{c(i-1)} \parallel \langle \langle n, i \rangle \rangle) \]

\[ + H(\langle \langle n, i \rangle \rangle) \]

\[ \leq \left( i - 1 \right) \cdot \log n + 1 + O(1) + \]

\[ 2 \log n + O(\log(\log n)) + O(1) \text{ q.e.d. Lemma} \]
The remainder of the argument can be summarized as follows. We will describe a uniform recursive procedure $\Psi$ with the following property. Suppose 1) $n$ is sufficiently large, 2) $\phi(n) < k$. Then $\Psi(\langle\langle n, i, \#S_i \rangle\rangle)$ will be a string of length $n, x$, such that $H(x) \geq LC \cdot i \cdot \log n$. 

On the other hand, $H(x) \leq H(\langle\langle n, i, \#S_i \rangle\rangle) + O(1) \leq [C \cdot (i-1) + 2] \log n + O(\log \log n)$. This gives a contradiction for $n$ sufficiently large.

Here is our description of $\Psi$:

1) Wait for a time to so large that a) $A_{2^k} = \emptyset$;
2) $\#S_i^t = \#S_i$. (Recall that $\Psi$ is given as input the parameters $n, i, \#S_i$.)

At any time $t > t_0$, $S_i^t \supseteq S_i$. (Recall that numbers tend to migrate downward in our protocol.) Since $S_i^t \subseteq S_i^* \subseteq S_i$ & $\#S_i^t = \#S_i$, we have
$S_{i^0} = S_i^+ = S_i$.

2) Hence if $x \in A_i^+$ for some $t \geq t_0$, then $x \in A_i$. (For otherwise, $x \in S_i \setminus S_i^+$, contradicting 1)

3) Suppose $x \in A_i^+$, then we can compute $x^*$ by the following procedure. Let $y_1, \ldots, y_m$ be all those strings active for $x$ at time $t$. Since $x \in A_i$, these are the only strings ever active for $x$. Let $z_1 = U(y$.

Then since $\phi(x) < k$, some $z_1 = x^*$. Since $U(x^* z_1) = x$, all $i$, $x^*$ is the least $z_1$. Hence for $x \in A_i^+$, $t \geq t_0$, we can effectively compute $H(x)$ (from the data $y_1, n, i, \# S_i$).

4) Since $\# A_i \geq n^c \cdot i$, some $x \in A_i$ has $H(x) > L C \cdot i \cdot \log n \cdot i$. Let $<x, t_1>$ be the least pair $\exists$ $i \colon t \geq t_0$

2) $x_1 \in A_{i_1}^{t_1}$

3) $H(x_1) > L C \cdot i \cdot \log n \cdot i$

Then $x_1$ is the output of the procedure $\psi$. 
As indicated earlier, this completes the proof.

Note that \( \tilde{H}(x^y | x) \) is of interest in the Chaitin theory as the difference between the more conceptual \( \tilde{H}(x^y) \) and the correct \( H(y | x) \) (\( = \tilde{H}(y | x^y) \)). I.e. the identity

\[
H(\langle\langle x,y\rangle\rangle) = H(x) + H(y | x) + O(1)
\]

would not be valid with \( H(y | x) \) replaced by \( \tilde{H}(y | x) \) (and in the case \( y = x^y \) would be off by the term \( \tilde{H}(x^y | x) \)).

As ever,

Bob.