On analogues of Miller-Yu theorem in Resource-Bounded Measures

Himanshu Shukla
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur
Email-hshukla@cse.iitk.ac.in

Mentor: Dr. Satyadev Nandakumar
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

July 9, 2015

SURGE-2015 IIT Kanpur

Abstract

We study the analogues of Miller-Yu theorem in Resource-bounded measures. Miller-Yu [7] gave the first characterization of 1-randomness in terms of Plain Kolmogorov complexity. Hence closing a long standing open problem. Its analogues in Resource Bounded measures were still unknown. We explore them and discover the behaviour of existing theorems in resource-bounded measures. We present the resource-bounded version of Chaitin's inequality[2] and present the proof for the same. Further we define randomness in resource-bounded measures and prove that the set of strings random with respect to computational paradigm in resource bounded measures is a subset of set of random strings in terms of measure-theoretic paradigm. Also prove one side of implication of Miller-Yu theorem in resource-bounded measures. We stop with a conjecture that set of computationally random strings is a *"proper subset"* of set of strings which are measure-theoretically random.

Contents

Introduction

Infinite random sequences have been interesting mathematicians and computer-scientists since long. Using measure theoretic approaches Von Misses in 1931 tried to define infinite random sequences as follows

> *A sequence is random if any of its sub-sequence created using an admissible place selection function which is independent of value of the nth term chosen, follows law of large numbers.*

. This definition did not work and there were many counter-examples showing such sequences can be non-random in Geneva conference (1937). In 1966 Martin Löf[6] gave the first criterion of randomness using measure theory. In 1970 Chaitin and Levin gave the criterion for randomness in terms of Computational aspect of random sequence which means that *"All initial segments of random sequences have longer descriptions"*.

In 1972-73 Schnorr showed that the definition of random sequences using computational aspect and measure theory were equivalent. But still there was no characterization in terms of Plain Kolmogorov complexity. In 1971 Martin Löf[5] went very close in deriving such a criterion but just missed it. In 2008 Joshep S. Miller and Liang Yu solved this long standing open problem by giving a criterion for 1-random infinite binary sequences in terms of Plain Kolmogorov complexity. Resource bounded complexity is informally defined as the Kolmogorov complexity when time and space are bounded. The characterization of Miller and Yu does not assume resource-boundedness. The analogues of this theorem in resource-bounded measures were still unknown. We studied them and have proved the behaviour of various inequalities and theorems in resource-bounded measures. First of all I will be stating some definitions and previous results of this field without proof. A detailed discussion on them and proofs can be found in standard texts like[3] and [4].

1 Definitions and previous results

1. **Kolmogorov's complexity:** The Kolmogorov complexity of any finite binary string x with respect to a Turing machine T is defined as shortest program p such that on input of y which is also a binary string and the program p, T outputs x. Mathematically

   $$C_T(x|y) = \min\{l(p) : T(p,y) = x\}$$

   $l(.)$ represents the length of any string. Note that this definition is conditional to y. For obtaining the unconditional definition put $y = \epsilon$. Also note that same definition works taking x and y as natural numbers because we can inter-convert finite binary sequences and natural numbers and infinite binary sequences into reals.
   There is a similar definition of Kolmogorov complexity if the programs are prefix-free i.e. if a program p produces x and q produces x' then p can not be prefix of q or vice-versa. For prefix-free (instantaneous) Kolmogorov complexity we have the following definition:

   $$K_T(x|y) = \min\{l(p) : T(p,y) = x\}$$

   Here the Turing machine T is prefix-free.

2. **Theorem 1.0.1** *: There exists a universal turing machine U such that for every turing machine T there exists a constant c_T for all x and y*

   $$C_U(x|y) \leq C_T(x|y) + c_T$$

   A similar invariance theorem can be obtained for prefix-complexity by replace C by K and taking T and U as prefix-free machines.

3. **Theorem 1.0.2** *:The number of strings of length n such that $C(x) \leq n-k$, $l(x) = n$ are at most 2^{n-k}*

4. **Theorem 1.0.3** ***(Kraft's inequality)***: $\sum\limits_{p\ is\ instantaneous\ code} 2^{-l(p)} \leq 1$

5. **Paradigms of Randomness:** There are three paradigms of randomness namely *computational*, *measure-theoretic* and *unpredictability* paradigm. We give the definition of all of the three.

   (a) *Computational Paradigm:* Intuitively Random sequences should be hard to describe i.e. they should not have shorter description. This notion was formalised by Chaitin and Levin and the sequences following this criterion are called 1-random sequence. A sequence ω is said to 1-random iff

   $$K(\omega \upharpoonright n) > n - O(1)\ \forall n$$

   (b) *Measure-theoretic Paradigm:* Martin Löf in 1966 [6] gave the first criterion for randomness in terms of measure theory. It can be simply stated as a sequence is random according to a property iff it does not lie in an effectively null set. Formally it can be stated as follows:[3]

      i. A *Martin Löf* test is a sequence $\{U_n\}_{n \epsilon \omega}$ of uniformly $\sum_0^1$ classes such that $\mu(U_n) \leq 2^{-n}$.

      ii. A class $C \subset 2^\omega$ is *Martin-Löf null* if there is a Martin- Löf test $\{U_n\}_{n \epsilon \omega}$ such that $C \subseteq \cap_n U_n$.

      iii. A set $A \ \epsilon\ 2^\omega$ is *Martin Löf random* if $\{A\}$ is not Martin Löf null.

   (c) *Unpredictability Paradigm:* Intuitively for a random sequence we should not be able to guess the n+1 the bit if we are given n bits of the sequence. Formally it is defined as follows:
   Let d be a computably enumerable (*c.e.*). martingale then an infinite sequence ω is said to unpredictably non-random iff $\omega \epsilon S[d]$ where $S[d]$ represents the success set of d.

   Note: In 1972-1973 Schnorr showed that these three paradigms are equivalent.

6. **Theorem 1.0.4** *Let d be a c.e. martingale.*

   (a) For any string σ and any prefix-free set S of extensions of σ, we have $\sum_{\tau \epsilon S} 2^{-|\tau|} d(\tau) \leq 2^{-|\sigma|} d(\sigma)$.

   (b) Let $R_k = \{\sigma : d(\sigma) \leq k\}$. Then $\mu([\![R_k]\!]) \leq \frac{d(\lambda)}{k}$.

7. **Theorem 1.0.5** ***(Chaitin's Inequality):***

   $$\forall k \forall n |\{\sigma \epsilon 2^n : K(\sigma) < n + K(n) - k\}| < 2^{n-k+O(1)}$$

   . This theorem has got its importance because it gives a counting inequality for prefix complexity.

8. **Universal Probability:** Universal Probability is of any finite string x conditioned to y is defined as follows:

   $$P(x|y) = \sum_{p(y)=x} 2^{-l(p)}$$

   I present the following theorem which compiles the results of prefix free complexity. The proof of all these theorems are present in [2].

9. **Theorem 1.0.6** *Here T is a prefix-free turing machine.*

   (a) $K(x) \leq K(x, y) + O(1)$

   (b) $K(x, y) \leq K(x) + K(y|x) + O(1)$

   (c) $K(x, y) \leq K(x) + K(y) + O(1)$

   (d) The set of all true propositions of the form "$P_T(x) > 2^{-n}$" is recursively enumerable and given y one can recursively enumerate over all true propositions of the form "$P_T(x|y) > 2^{-n}$"*

(e) *For every prefix-free turing machine T there is a constant c such that*

   i. $K(x) \leq -log_2(P_T(x)) + c$

   ii. $K(x|y) \leq -log_2(P_t(x|y)) + c$

(f) *There is a computer $T^{'}$ for each turing machine T such that the following to hold:*

   i. $K_{T'}(x) = [-log_2(P_T(x))] + 1, P_{T'}(x) = 2^{-[-log_2(P_T(x))]}$

   ii. $K_{T'}(x|y) = [-log_2(P_T(x|y))] + 1, P_{T'}(x|y) = 2^{-[-log_2(P_T(x|y))]}$

(g) *For each computer T there is a constant c such that*

   i. $P(x) \geq 2^{-c}P_T(x)$ *and* $P(x|y) \geq 2^{-c}P_T(x|y)$

   ii. $K(x) = -log_2(P(x)) + O(1)$ *and* $K(x|y) = -log_2(P(x|y)) + O(1)$

(h) $P(x) \approx \sum\limits_{y} P(x, y)$

(i) $\exists$ *a turing machine T such that* $K_T(y|x) = K(x, y) - K(x) + c$

(j) $K(s, t) = K(s) + K(t|s) + O(1)$

10. **Theorem 1.0.7** ***(Schnorr's theorem):*** *An infinite sequence ω is Martin Löf random iff $\forall n$ $K(\omega \upharpoonright n) > n - O(1)$*

11. **Resource-bounded Kolmogorov Complexity:** I am only concerned with the time hence will give only time-bounded definition of Kolmogorov complexity. Let T be a turing machine and $t(n)$ be a time bound when n is length of string in interest. The time-bounded Plain Kolmogorov complexity of the x conditioned to y is defined as length of shortest program producing from input y in at most $t(n)$ steps.

$$C_T^{t(n)}(x|y) = min\{l(p) : T(p, y) = x, \text{in at most t(n) steps}\}$$

We have a similar definition for prefix-free complexity obtained by taking T as prefix-free machine and C to be K.

$$K_T^{t(n)}(x|y) = min\{l(p) : T(p, y) = x, \text{in at most t(n) steps}\}$$

12. **Theorem 1.0.8** ***(Invariace theorem for resource-bounded complexity):*** *There exists a universal partial recursive turing machine U such that for every other partial recursive turing machine T there is a constant c such that $\forall x$ $\forall y$*

$$C_U^{ct(n)logt(n)}(x|y) \leq K_T^{t(n)}(x|y) + c.$$

A similar theorem holds for Resource-bounded prefix-free complexity which is as follows

$$K_U^{ct(n)logt(n)}(x|y) \leq K_T^{t(n)}(x|y) + c.$$

2 Miller-Yu Theorem

2.1 Original Proof

I present the original proof of Miller-Yu theorem as given in [7] with some explanations. Before going on I will state the following theorem which I will be proving later in this section.

Theorem 2.1.1 *Let G(n) be defined as follows:*

$$G(n) = \begin{cases} K_{s+1}(t), & \text{if } n = 2^{\langle s,t \rangle} \text{ and } K_{s+1}(t) \neq K_s(t) \\ n, & \text{otherwise.} \end{cases}$$

Here K_s means sth stage approximation of Kolmogorov complexity. That is we run all the programs of length i for j steps such that $i + j = s$. Hence if ω is an infinite sequence and $\forall n$ $C(\omega \upharpoonright n) \geq n - G(n) - c$ then is Martin-Löf random.

Theorem 2.1.2 ***(Miller-Yu)[7]:*** *For ω which is an infinite binary sequence, the following statements are equivalent:*

1. *ω is 1-random.*
2. *$\forall n$ $C(\omega \upharpoonright n) \leq n - K(n) - O(1)$.*
3. *$\forall n$ $C(\omega \upharpoonright n) \geq n - g(n) - O(1)$ for every computable $g : \mathbb{N} \Rightarrow \mathbb{N}$ such that $\sum_{n \epsilon \mathbb{N}} 2^{-g(n)}$ is finite.*
4. *$\forall n$ $C(\omega \upharpoonright n) \geq n - G(n) - O(1)$. $G(n)$ is same as used in theorem 2.1.*

I prove the following lemma before starting with the proof of the theorem:

Lemma 2.1.3 $\sum_{n \epsilon \mathbb{N}} 2^{-G(n)} < \infty$

Proof:

$$\textstyle\sum_{n\epsilon\mathbb{N}} 2^{-G(n)} \leq \sum_{n\epsilon\mathbb{N}} 2^{-n} + \sum\limits_{t\epsilon\mathbb{N}} \sum_{m\epsilon\mathbb{N}} 2^{-K_s(t)}$$
$$\textstyle\sum_{n\epsilon\mathbb{N}} 2^{-G(n)} \leq \sum_{n\epsilon\mathbb{N}} 2^{-n} + \sum\limits_{t\epsilon\mathbb{N}} \sum_{m \geq K(t)} 2^{-m}$$
$$\textstyle\sum_{n\epsilon\mathbb{N}} 2^{-G(n)} \leq \sum_{n\epsilon\mathbb{N}} 2^{-n} + 2 \sum\limits_{t\epsilon\mathbb{N}} 2^{-K(t)}$$

using Kraft's inequality the sum converges. Hence $\sum\limits_{n\epsilon\mathbb{N}} 2^{-G(n)} < \infty$.

Now we prove the theorem: $1 \Rightarrow 2$: Define

$$I_k = \{\omega \epsilon 2^\omega : (\exists n) C(\omega \upharpoonright N) < n - K(n) - k\}.$$

Now as K_s and C_s represent the sth stage approximation of K and C. Then $\exists n \exists s$ such that $C(\omega \upharpoonright n) + K(\omega \upharpoonright n) < n - k$ iff $\omega \epsilon I_k$. This makes I_k to be a $\sum_0^1$ class. Fewer than $2^n - K(n) - k$ programs have length less than $n - K(n) - k$, so $|\{\sigma \epsilon^n : C(\sigma) < n - K(n) - k\}| \leq 2^{n-K(n)-k}$. Hence

$$\mu I_k \leq \sum_{n\epsilon\mathbb{N}} \mu\{\omega \epsilon 2^{\mathbb{N}} : C(\omega \upharpoonright n) < n - K(n) - k\}$$
$$\leq \sum_{n\epsilon\mathbb{N}} 2^{-n+n-K(n)-k} = 2^{-k} \sum_{n\epsilon\mathbb{N}} 2^{-K(n)} \leq 2^{-k}$$

Hence $\{I_k\}_{k\epsilon\mathbb{N}}$ is a Martin Löf test. Now if ω is 1-random hence $\omega \notin I_k$ for some k. Hence $\exists k$ such that $(\forall n)$ $C(\omega \upharpoonright n) \geq n - K(n) - k$

$2 \Rightarrow 3$: Let $g : \mathbb{N} \Rightarrow \mathbb{N}$ be a computable function such that $\sum\limits_{n\epsilon\mathbb{N}} 2^{-g(n)} < \infty$. By minimality of K as an information content measure, $\forall n$ $K(n) \leq g(n) + O(1)$. Therefore, if $\forall n$ $C(\omega \upharpoonright n) \geq n - K(n) - O(1)$ hence $\forall n$ $C(\omega \upharpoonright n) \geq n - g(n) - O(1)$.

3 ⇒ 4 : This follows because G is computable functions and $\sum\limits_{n\epsilon\mathbb{N}} 2^{-G(n)}$ is finite.

4 ⇒ 1 : This part is implied directly by theorem 2.1.1. Hence we now prove theorem 2.1.1 using contrapositive argument. We would prove the following:

If ω is not 1-random then $\forall c\ \exists n$ such that $C(\omega\restriction n) \leq n - G(n) - c$

Proof: Suppose ω is not 1-random then $\forall k\epsilon\mathbb{N}\ \exists t$ such that $K(\omega\restriction t) \leq t - k$. This is by Schnorr's theorem (theorem 1.0.7) and t is large enough so that

$$K(t) \leq 2^t - k - 1$$

as if not then we know that $t > k$, also suppose for contradiction $K(t) \geq 2^t - k - 1$ for some pair (t, k) then $K(t)$ has a max value as $log(t) \Rightarrow k + 1 \geq 2^t - log(t)$ since $k < t$ hence this will not hold. Hence a contradiction. Therefore $K(t) \leq 2^t - k - 1$ is a valid assumption.
Now take the least s such that $K_{s+1}(t) = K(t)$ then put $n = 2^{\langle s,t\rangle}$ (from now on we will assume n to be of this form and to be the least s such that $K_{s+1}(t) = K(t)$) and we will show that for this n ∃ a program of most length $n - G(n) - k + O(1)$ such that $C(\omega\restriction n)$ can be described.

Define a partial computable no prefix-free function $M : 2^{\mathbb{N}} \Rightarrow 2^{\mathbb{N}}$. $\forall t\epsilon\mathbb{N}$ let $n = 2^{\langle s,t\rangle}$ (this s is minimum s such that $K_{s+1}(t) = K(t)$). To $\langle s, t\rangle$ we devote all the programs from length $n/2 + c + 1$ to $n + c$. Note that no to pairs $s_1, t_1\rangle$ and $\langle s_2, t_2\rangle$ will have conflict in the program sets associated with them i.e. if $\{\langle s_i, t_i\rangle\}$ are the programs associated with $\langle s_i, t_i\rangle$ then $\{\langle s_i, t_i\rangle\} \cap \{\langle s_j, t_j\rangle\} = \phi$. This is because the range of length associated with both of these sets are different.

Now for $k\epsilon\mathbb{N}$, let $m = n - K_{s+1}(t) - k + c$. Now m is obviously $< n + c$ and if $m > n/2 + c + 1$ then $\forall y\sigma\epsilon 2^n$ such that $K(\sigma\restriction t) \leq t - k$, we try to give each σ a M-program of length m. We required this m to be $> n/2 + c + 1$ because k will be at least 1 and the number of strings with one bit compression can be at most $n/2$. Note that different m do not compete for the programs. Also note that since $K_{s+1}(t) = K(t)$ so total number of strings in the set $|\{\sigma\epsilon 2^n : K(\sigma\restriction t) \leq t - k\}| \leq 2^{t-K(t)-k+c}.2^{t-k} = 2^m$. Hence we have enough M-programs of length m for such $\sigma's$.
Now because ω is 1-random hence we have

$$\begin{aligned} m = n - K(t) - k + c \geq n - K(t) - k + c &\geq n - 2^t + k + 1 - k + c \\ &\geq n/2 + c + 1 \end{aligned}$$

. This happens because $2^t < n/2$ as :

$$n = 2^{\frac{(s+t)(s+t+1)}{2}+t}.$$

Now as s and $t > 0$ so

$$\begin{aligned} \tfrac{(s+t)(s+t+1)}{2} &\geq 1 \\ \Rightarrow n/2 = 2^{\frac{(s+t)(s+t+1)}{2}+t-1} &\geq 2^t. \end{aligned}$$

Hence $m > n/2 + c + 1$ which was also one of the required condition for giving M-programs.
Therefore ∃ a program of length m for $\omega\restriction n$ as for ω a non 1-random $\omega\restriction n\ \epsilon 2^n$ and $\omega\restriction t = \sigma\restriction t$ for some $\sigma\epsilon 2^n$ and for that σ there is a description of length m.
Therefore

$$\begin{aligned} C(\omega\restriction n) \geq C_M(\omega\restriction n) + O(1) &\leq n - K(t) - k + c + O(1) \\ &\leq n - G(n) - k + O(1) \end{aligned}$$

as k is an arbitrary value hence $\forall c\ \exists n\ C(\omega\restriction n) \leq n - G(n) - c$.

2.2 Simpler proof of Miller-Yu theorem

Bienvenu, Merkel and Shen[1] gave a simpler proof of Miller-Yu theorem by showing the equivalence of statement 1 and 3 in theorem 2.1.2. $1 \Rightarrow 3$ is evident using $g(n)$ instead of$K(n)$ in the proof of side $1 \Rightarrow 2$. We look at the simpler proof for the equivalence $3 \Rightarrow 1$
$3 \Rightarrow 1$*:* By the universal randomness test we generate for every $c = 1, 2, 3\ldots$ a sequence of strings

$$x(c,0), x(c,1), x(c,2), \ldots$$

such that total measure of all the intervals is less than 2^{-c}, and for every non random sequence ω and every c, one of the strings $x(c,i)$ is a prefix of ω.

We without loss of generality assume that $x(c,i)$ is a total function and the enumeration is done in the increasing order of length i.e. $l(x(c,0)) \leq l(x(c,1)) \leq l(x(c,2))\ldots$ for any c, as dummy intervals can be added without altering the total measure.

Now for each c there are finitely many strings of length n, and let m(c,n) represent the total measure of such strings. Hence we have $\sum_n m(n,c) \leq 2^{-c}$ for every c.
Now consider the function g defined by the equation

$$2^{-f(n)} = \sum_c 2^{c/2} m(n,c).$$

Since the quantity $\sum_n m(n,c) < 2^{-c}$, hence

$$\sum_n 2^{-f(n)} = \sum_{n,c} 2^{c/2} m(n,c) \leq \sum_c 2^{-c/2} \leq 1$$

. Now any string of length n is the sequence $x(c,.)$ is uniquely determined by c and then we number these strings of length n from 1 to $2^n m(n,c)$. Hence its Kolmogorov's Complexity does not exceed

$$2log(c) + log(2^n m(n,c)) + O(1) \leq 2log(c) + n - f(n) - c + O(1)$$

As $-2log(c) + c$ can be arbitrarily large hence putting this equal to c', we prove the implication using contra-positive argument.

3 Our Work

We took a two way approach for studying Miller-Yu theorem in the resource-bounds. One was using original proof of Miller-Yu and another was using the simpler proof. We do all our study for polynomial time bounds i.e. polynomial of length of string.

- In course of studying it through the original proof we derived a resource bounded-version of Chaitin's inequality (theorem 1.0.5). For which we derived all the parts of theorem 1.0.6 in resource-bounded measures.
- We converted the proof in section 2.2 and theorem 1.0.7 into unpredictability paradigm. Then we defined the definition of randomness in resource bounded measures analogous to paradigms of randomness in unbounded case.
- We further studied the relation between these newly defined paradigms of resource bounded randomness.
- We prove the $3 \Rightarrow 1$ of theorem 2.1.2 for resource bounded complexity.
- Finally we end have ended with the conjecture that the other implication of theorem 2.1.2 does not hold.

3.1 Resource-bounded version of Chaitin's Inequality

I reproduce whole of theorem 1.0.6 before proving the resource bound version. Let $x*$ represent the shortest program in lexicographic order which outputs x with respect to a turing machine.

Theorem 3.1.1 *If p and q be polynomial time bounds then*

$$\forall p \exists q \textit{ such that } \forall k \forall n |\{\sigma \epsilon 2^n : K^p(\sigma) < n + K^q(n) - k\}| < 2^{n-k+O(1)}$$

Before proving, we give the following lemmas:

Lemma 3.1.2 *Let p and q be the polynomial time bounds then*

$$\exists q \ \forall p (K^q(x) \leq K^p(x,y) + O(1))$$

Proof: The proof for this lemma goes as follows, let p be a fixed polynomial time bound, let a represent a program and as there is a Turing machine T such that $T^p(a) = x$ iff $U^p(a) = (x,y) \Rightarrow K^p_T(x) = K^p(x,y)$ hence using the invariance theorem for the prefix-free resource bound version of Kolmogorov Complexity there is a polynomial time bound q such that $K^q(x) \leq K^p(x,y) + O(1)$

Lemma 3.1.3 *let p, q and q' be polynomial time bounds then*

$$\forall p \forall q \exists q' (K^{q'}(x,y) \leq K^p(x) + K^q(y|x) + O(1))$$

Proof: We claim that there is computer T with the following property. If $U^q(a, x*) = y$ and $|a| = K^q(y|x)$, then $T^{q+p+O(1)}(x * a, \epsilon) = \langle x, y \rangle$. Hence by invariance theorem $K_T^{q+p+O(1)}(x,y) \leq |x * a| = |x * | + |a| = K^p(x) + K^q(y|x)$ and $\exists q'$ which is polynomial time bound such that $K^{q'}(x,y) \leq K_T^{p+q+O(1)}(x,y) + O(1)$
I now verify the claim as follows: T does the following when given $x * a$ on its program tape and ϵ on its work tape. Now first it simulates the computation of U and reads $x*$ and performs the computation of x and then it performs the simulation of U as $x*$ given on work tape and a given on program tape and computes y. Finally it takes $O(1)$ time to calculate$\langle x, y \rangle$. One can see that the amount of steps that T takes to compute $\langle x, y \rangle$ is $p + q + O(1)$.

Lemma 3.1.4 *Let p and q be polynomial time bounds*

$$\forall p \exists q (K^q(x,y) \leq K^p(x) + K^p(y) + O(1))$$

Proof: Fix some polynomial time bound p then $\exists$ a computer T such that given $x*$ and $y*$ (with respect to polynomial time bound p and $U(x*, \epsilon) = x$ and $U(y*, \epsilon) = y$) on its program tape and ϵ on its program tape it will compute x and then y and finally $\langle x, y \rangle$. Hence $K_T^{2p+O(1)}(x,y) \leq K^p(x) + K^p(y)$ hence using invariance theorem we have $K^q \leq K^p(x) + K^p(y) + O(1)$ where q is a polynomial time bound.

Lemma 3.1.5 *Let p be a fixed polynomial time bound then define*

$$P_T^p(x) = \sum_{a: T^p(a,\epsilon) = x} 2^{-|a|}$$

then $\forall p \exists q$ such that all true propositions of the form "$P_T^p(x) > 2^{-n}$" are recursively enumerable and this statement is polynomial time decidale in q steps.

Proof: We just present a rough draft of the proof by chosing the *canonical Kraft's tree*, i.e. for a given turing machine T there is exactly one program of each length such that it can halt and produce some string. Now note that we run all the programs in this canonical Kraft's Tree of length $\leq n$ for exactly p steps using devotailing on this Kraft's tree. Note that there are exactly n programs so total steps are $n * p$. Now if there is any program that produces x then clearly $P_T^p(x) \geq 2^{-n}$ and we can see that if there is no program that gives this then all the programs that can produce x in p steps are of length $\geq n + 1$ hence P_T^p can at most be 2^{-n}. Hence there exists a q such that one can check the truth value of the expression "$P_T^p(x) > 2^{-n}$" in at most q steps.

Lemma 3.1.6 *Let p and q be polynomial time bounds and let T be a prefix-free turing machine then*

1. $\forall T \forall p \exists q$ *such that* $K^q(x) \leq -log_2(P^p_T(x)) + c$

2. $\forall T \forall p \exists q$ *such that* $K^q(x|y) \leq -log_2(P^p_T(x|y)) + c$

Proof: Define a turing machine T' which functions as follows first of all it checks whether it has been given ϵ or $y*$ on its work tape. If it has been given ϵ then it enumerates the true propositions of the form "$P^p_T(x) > 2^{-n}$" and simulates the turing machine defined by the requirements of the form $\langle x, n+1\rangle$ ("$P^p_T(x) > 2^{-n}$") i.e $\langle x, n$ is said to be a requirement iff $n \geq [-log_2(P^p_T(x))] + 1$ ([.] represents Greatest Integer Fucntion). Now this check can be made in polynomial time bound as for a particular x start from $n = 1$ and then move forward in polynomial time we will get the least n such that $P^p_T(x) > 2^{-n}$ and note that all the n's after that will automatically satistfy this. Hence the programs a of length n such that $T^{'}(a, \epsilon) = x$ is 1 if $n \geq [-log_2(P^p_T(x))] + 1$ and 0 otherwise. This said because note that if n is choses then $n+1, n+2, \ldots$ all be chosen hence in the machine T' we will be assigning the first program available of each length to x hence we can conclude that $\exists q'$ which is a polynomial time bound such that $K^{q^{'}}_{T'}(x) = [-log_2(P^p_T(x))] + 1, P^{q^{'}}_{T'}(x) = 2^{-[-log_2(P^p_T(x))]}$ now use invariance theorem to get the theorem part a. The other part also follows in a similar fashion.

Lemma 3.1.7 *Let p and q be polynomial time bounds and let T be a prefix-free turing machine then*

1. *there exists a constant c such that*

$$P^q(x) \geq 2^{-c}P^p_T(x), P^q(x|y) \geq 2^{-c}P^p_T(x|y)$$

2. $K^q(x) = -log_2(P^p(x)) + O(1), K^q(x|y) = -log_2(P^p(x|y)) + O(1)$

Proof: The proof of part a follows from lemma 3.1.6 using the fact that $P^q(s) \geq 2^{-K^q(s)}$ and proof of second part is obtained by taking $C = U$

Lemma 3.1.8 *Let p, q, q′ be a polynomial time bounds then*

1. $\forall p \exists q$ *and a constant c such that* $P^q(x) \geq 2^{-c} \sum\limits_y P^p(x, y)$.

2. $\forall p \exists q'$ *and a constant* c' *such that* $P^{q^{'}}(x) \leq 2^{c^{'}} \sum\limits_y P^p(x, y)$.

Proof: This is a two way proof, first of all there is a turing machine T such that $\exists l$ which is a polynomial time bound $T^l(a, \epsilon) = x$ if $U^p(a, \epsilon) = \langle x, t\rangle$. Thus $P^l_T(x) \geq \sum\limits_y P^p(x, y)$. This statement simply means that set of programs producing x in machine T in at most p time bound is a superset of set of programs producing $\langle x, y\rangle$ using the machine U in at most p time bound. Hence $P^l_T(x) \geq \sum\limits_y P^p(x, y)$. Using the lemma 3.1.8.1 part a we have a q which is a polynomial time bound and a c such that $P^q(x) \geq 2^{-c} \sum\limits_y P^p(x, y)$.

Secondly there is a turing machine T' such that l' which is polynomial time bound $T^{'^{l'}}(a, \epsilon) = \langle x, x\rangle$ if $U^p(a, \epsilon) = x$ Thus using the same subset argument we have $\sum\limits_y P^{l'}_{T'}(x, y) \geq P^{l'}_{T'}(s, s) \geq P^p(s)$. Now again using part a of lemma 3.1.8 $\exists q'$ which is a polynomial time bound and a c' such that $\sum\limits_y P^{q^{'}}(x, y) \geq 2^{-c^{'}} P^p(x)$. This proves the theorem.

Lemma 3.1.9 $\forall p$ *which is a polynomial time bound there is a turing machine T , a constant c and polynomial time bounds q and l such that*

$$K^l_T(y|x) = K^p(x, y) - K^q(s) + c$$

Proof: The set of programs such that $U(a, \epsilon)$ is defined is recursively enumeratble and each program in this set can be checked in polynomial time that $U^p(a, \epsilon)$ is defined or not. Let a_k be the kth program in some recursive enumeration of this set and $x_k, y_k\rangle = U^p(a_k, \epsilon)$. Now by 3.1.8 we have a c' and a polynomial time bound q' such that $P^{q'}(x) \geq 2^{-c'} \sum\limits_y P^p(x, y)$. Hence $\frac{2^{-c'} \sum\limits_y P^p(x,y)}{P^{q'}(x)} \leq 1$. Now by lemma 3.1.7.2 we have a polynomial time bound q and a constant c'' such that $K^q(x) = -log_2(P^{q'}(x)) + c$ and hence writing the combination of c' and c'' as c we have $2^{K^q(x)-c} \sum\limits_y P^p(x, y) \leq 1 \forall x$. Given $x*$ on its work tape T simulates T_s defined by the requirements $\langle t_k, |a_k| - |x*| + c$ and $x = U^q(x*, \epsilon)$. This simulation of T will also be in polynomial time as every other thing is in polynomial time hence for some l which is a polynomial time bound $T(a', x*) = T_x(a', \epsilon) = y$. and $a' = |a_k| - K^q(x) + c$ i.e. $K^l_T(y|x) = K^p(x, y) - K^q(x) + c$. Hence we have constructed such a T such that the condition of the theorem is satisfied.

Lemma 3.1.10 *Let p, q and q' be the polynomial time bounds then $\forall p \exists q$ and q' and c' such that*

$$K^p(x, y) \geq K^{q'}(y|x) + K^q(x) - c'$$

Proof: By lemma 3.1.9 $\forall p$ which is polynomial time bound we have a turing machine T and polynomial time bounds q and l and a constant c such that $K^l_T(y|x) = K^p(x, y) - K^q(x) + c$, now by invariance theorem we have a polynomial time bound q' and c' such that $K^{q'}(y|x) \leq K^p(x, y) - K^q(x) + c'$. Hence $K^p(x, y) \geq K^{q'}(y|x) + K^q(x) - c'$.

Proof of theorem 3.1: Let n be the length of string σ and p be polynomial time bound then $\exists q$, q' and c such that $K^p(\sigma) = K^p(n, \sigma) + O(1) \geq K^q(n) + K^{q'}(x|n) - c + O(1)$. Now by counting argument there are less than 2^{n-k} strings such that $K^{q'}(\sigma|n) < n - k$. Hence fewer than 2^{n-k} strings satisfy $K^p(\sigma) < K^q(n) + n - k - c + O(1)$. This proves the inequality for resource bounded version.

3.2 Proof of Miller-Yu theorem and Schnorr's Theorem using Unpredictability paradigm

3.2.1 Miller-Yu theorem in Unpredictability paradigm

Unpredictability formulation of the simpler proof will be as follows:

Theorem 3.2.1 *The following statements hold:*

1. Let $f : \mathbb{N} \Rightarrow \mathbb{N}$ be such that $\sum 2^{-f(n)} < \infty$. Then if $\forall$ c.e. martingales d, $\omega \notin S[d]$, $\exists$ a constant c such that $\forall n$.

2. There exists a total computable function $f : \mathbb{N} \Rightarrow \mathbb{N}$ such that $\sum 2^{-f(n)} < \infty$ and $\omega \ \epsilon \ S[d]$ every non-random sequence ω and $\forall$ c $\exists$ n such that $C(\omega \upharpoonright n) < n - f(n) - c$

I present the proof of second part first.

Proof: Let d be c.e. martingale. Let $\chi_c = \{\sigma : d(\sigma) \geq 2^c\}$. Now let ω be such that $\omega \ \epsilon \ S[d]$. Also without loss of generality assume these strings enter the set χ_c in lexicographic order. Hence for every c $\exists$ n_c such that $d(\omega \upharpoonright n_c) > 2^c$, hence this initial segment belongs to χ_c. Note that the measure of all the segments of length n_c= $m(n_c, c)$ in χ_c

$$= \frac{\#ofsegmentsoflengthn_c}{2^{-n_c}} = \sum_{\substack{\sigma \epsilon \chi_c \\ |\sigma|=n_c}} 2^{-n_c} \frac{d(\sigma)}{d(\sigma)}$$

$$\leq 2^{-c} \sum_{\substack{\sigma \epsilon \chi_c \\ |\sigma|=n_c}} 2^{-n_c} d(\sigma)$$

$$\leq \tfrac{d(\epsilon)}{2^c} \text{ using theorem 1.4.}$$

Also $\sum\limits_n m(n,c) \leq d(\epsilon)2^{-c}$. Now I consider the function f defined by the equation

$$2^{-f(n)} = \sum_c 2^{\frac{c}{2}} m(n,c).$$

Since each $m(n,c)$ and even the sum $\sum_n m(n,c)$ does not exceed 2^{-c}, the right hand side is a computably convergent series and f is computable. Hence,

$$\sum 2^{-f(n)} = \sum_{n,c} 2^{c/2} m(n,c) \leq \sum_c 2^{-c/2} < \infty.$$

Number of strings of length n in χ_c can be uniquely and computably determined by c and the ordinal number of this string among $2^n mn, c$ of them. Hence Kolmogorov complexity of the initial segment till n_c.

$$C(\omega \restriction n_c|n) \leq 2log\ c + log(2^n m(c, n_c)) + O(1) \leq 2logc + n - f(n) - c + O(1)$$

Keep the constant $2logc - c$ as another constant say $c^{'}$.

The proof of first part is as follows:
let f be a total computable function such that $\forall\ c\ \exists\ n$ such that $C(\omega \restriction n) < n - f(n) - c$. Let $R_0, R_1 \ldots$ be prefix free generators such that $[\![R_c]\!] = \{\omega : \exists n$ such that $C(\omega \restriction n) < n - f(n) - c\}$. Now we construct a martingale d_c such that whenever a string σ is added to R_c $\forall\ \tau$ such that $\tau \succeq \sigma$ we add 1 and we add $2^{k-|\sigma|}$ to all $d_c(\sigma \restriction k), k < |\sigma|$. As d_c are uniformly c.e. maringales and $d_c(\epsilon) \leq 2^{-c}$. Thus $d = \sum_n d_n$ is a c.e. martingale. note that if for a sequence ω the condition holds then $\forall\ c$, it will lie in $[\![R_c]\!]$, hence for different $c's$, $d_c(\omega)$ would have been incremented by 1 hence. $\sum_c 1$ diverges and hence if ω satisfies the condition then there exists a martingale such that $\omega \epsilon S[d]$.

3.2.2 Schnorr's Theorem in unpredictability paradigm

Unpredictability formulation of Schnorr's theorem is as follows:

Theorem 3.2.2 *The following two statements hold:*

1. *If d is c.e. martingale such that $\omega\epsilon\ S[d]$ then it is not 1-random.*
2. *If $\forall$ d which are c.e martingales such that $\omega \notin S[d]$ then ω is 1-random.*

Proof: First statement has the following proof, construct prefix free sets $R_0, R_1, \ldots$ such that

$$[\![R_n]\!] = [\![\{\sigma : d(\sigma) > 2^k\}]\!].$$
$$R_n \subseteq \{\sigma : d(\sigma) > 2^n\},$$

R_n is constructed such that it is minimal (if $\sigma \epsilon R_n$ then $\sigma \restriction 1\ \forall\ l < |\sigma| \notin$ in R_n).

Hence

$$\sum_{n\geq 2} \sum_{\sigma \epsilon R_{n^2}} 2^{-|\sigma|+n} = \sum_{n\geq 2} 2^n \frac{d(\sigma)}{d(\sigma)}$$
$$\leq \sum_{n\geq 2} d(\epsilon) 2^{n-n^2}$$

without loss of generality lets assume $d(\epsilon) = 1$, hence

$$\sum_{n\geq 2} d(\epsilon) 2^{n-n^2} \leq \sum_{m\geq 2} 2^{-m} < 1.$$

So by minimality of K among information content measures $\exists$ a constant c such that if $\sigma\epsilon R_{n^2}$ for some $n \geq 2$ then $K(\sigma) \leq |\sigma| - n + c$. Since $\omega\epsilon S[d]$, hence $\omega\epsilon[\![R_{n^2}]\!]$, for each n $\exists$ a k such that $K(\omega \restriction k) \leq k - n + c$, hence ω is not 1- random.
Second statement can be proved as follows. Let R_k be a prefix-free set such that $[\![R_k]\!] = \{\omega : \exists n K(\omega \restriction n) \leq n - k\}$. Construct d_k in the sense of proof of first part of simpler proof[1]. Hence if $\nexists\ d$ such that $\omega\epsilon\ S[d]$. Hence $\exists\ k\ K(\omega \restriction n) > n - k\ \forall n$. Thus ω is 1-random.

3.3 Polynomial time randomness

If we look at the proof of all theorems that are concerned randomness of strings. They rely on the following idea:

All the three paradigms defining randomness of infinite binary sequences are equivalent.

This statement simply means that if a string is random with respect of computational paradigm then it is random with respect to measure theoretic paradigm and also with respect to unpredictability paradigm. Hence I give the following definitions in resource-bounded measures as analogues of the three randomness paradigms.

- **1-k-randomness:** An infinite binary sequence is said to be *1-k random* iff the following holds.

  $$K^{n^k}(\omega \upharpoonright n) \geq n - O(1)\ \forall n.$$

- **p-Unpredictably random:** An infinite binary sequence ω is said to be *p-unpredictably random* iff the following holds:

  $$\forall d_p \text{ such that } d_p \text{ is a polynomial time computable martingale } \omega \notin S[d_p].$$

- **Polynomially Martin-Löf random:** A Martin-Löf test U_n is said to be *polynomial Martin Löf test* iff $\exists$ a d_p which is a polynomial time computable martingale such that it is possible to construct sets R_n which is a prefix-free set such that

  $$[\![R_n]\!] = [\![\sigma : d_p(\sigma) > 2^n]\!].$$

  A sequence ω is said to be polynomially Martin-Löf random iff $\omega \notin \cap U_n$ where U_n is a polynomial Martin-Löf test.

It is easy to say that if a sequence is polynomially Martin-Löf random then it is p-unpredictably random and vice-versa which is implied by the way these two things have been defined. We have defined them in such a way because it was more appealing to the intuition and we cannot have a something like polynomial time thing in the measure theoretic paradigm of randomness because there is no algorithm, also it is more intuitively appealing that unpredictability is more close to the measure-theory which is more close to the probability which is a measure of uncertainty. Now the only thing left is to check that a sequence is *"1-k random for some k iff it is p-unpredictably random"*.

We state the following theorem:

Theorem 3.3.1 If a string is not p-unpredictably random then it is not 1-k random $\forall$ $k \geq 2$.

Proof: The proof is on the similar grounds as the proof of part 1 of Schnorr's theorem. As since $\exists$ a d_p such that $\omega S[d_p]$. We construct the sets $R_0, R_1, \ldots$ such that

$$[\![R_n]\!] = [\![\{\sigma : d_p(\sigma) > 2^k\}]\!].$$
$$R_n \subseteq \{\sigma : d_p(\sigma) > 2^n\},$$
$$R_n \text{ is constructed such that it is minimal (if } \sigma \epsilon R_n \text{ then } \sigma \upharpoonright \mathrm{l}\ \forall\ l < |\sigma| \notin \text{ in } R_n).$$

Hence

$$\sum_{n \geq 2} \sum_{\sigma \epsilon R_{n^2}} 2^{-|\sigma|+n} = \sum_{n \geq 2} 2^n \frac{d_p(\sigma)}{d_p(\sigma)}$$
$$\leq \sum_{n \geq 2} d_p(\epsilon) 2^{n-n^2}$$

without loss of generality lets assume $d(\epsilon) = 1$, hence

$$\sum_{n\geq 2} d(\epsilon)2^{n-n^2} \leq \sum_{m\geq 2} 2^{-m} < 1.$$

Now using the argument that even resource bounded complexity is minimal among all the resource bounded information content measures (bounded by the same amount of time). Hence we need to ensure the running time of the information measure $|\sigma| - n$ for the strings in R_{n^2} given n we can get to the R_{n^2} in $log(n)$ time also we know that $n < |\sigma|$. Now in a given R_{n^2} I can have at most $2^{|\sigma|/2}$ strings of same length such that $d_p(\sigma) > 2^{n^2}$ as if we have more then we will have more than $2^{|\sigma|/2}$ strings then we will not have R_{n^2} as minimal. Now we give ordinal numbers to the strings of the same length. Now it will take $< log(2^{|\sigma|/2}) + O(1)$ time to decode the string. Hence the measure $|\sigma| - n$ takes at most $|\sigma|$ time to get decoded. Hence $\forall\ k \geq 1$ there exists a constant c such that if $\sigma\epsilon R_{n^2}$ then for some $n \geq 2$, $K^{|\sigma|^k}(\sigma) \leq |\sigma| - n + c$. Hence $\omega\epsilon[\![R_{n^2}]\!]$, for each n $\exists$ a k' such that $K^{k'^k}(\omega \upharpoonright k') \leq k' - n + c$, hence ω is not 1-k random.

"This theorem says that the set of strings which are 1-k random form a subset of strings which are p-unpredictably random."

Now if we look at the converse then it does not seem to hold if we follow the constructional methodology of martingales used in proof of Schnorr's theorem,theorem 3.2.1 as we calculate d as $\sum_n d_n$. Also we claim that implication $1 \Rightarrow 2$ of theorem 3.1.1 should not hold in resource-bounds. We prove the implication $2 \Rightarrow 1$ of theorem 3.1.1 in resource bounds.

Theorem 3.3.2 *If a sequence is not p-unpredictably random then it is $\forall k\ \exists$ a computable function $f(n)$ such that $\forall c\ \exists n$ such that $C^{n^k}(\omega \upharpoonright n) < n - f(n) - c$*

Proof: The proof goes on the similar grounds as the proof of theorem 3.1.1 second part. Let d_p be c.e. martingale. Let us fix a k and let $\chi_c = \{\sigma : d_p(\sigma) \geq 2^c\}$. Now let ω be such that $\omega\ \epsilon\ S[d_p]$. Also without loss of generality assume these strings enter the set χ_c in lexicographic order. Hence for every $c\ \exists\ n_c$ such that $d(\omega \upharpoonright n_c) > 2^c$, hence this initial segment belongs to χ_c. Note that the measure of all the segments of length n_c= $m(n_c, c)$ in χ_c

$$= \frac{\#ofsegmentsoflengthn_c}{2^{-n_c}} = \sum_{\substack{\sigma\epsilon\chi_c\\ |\sigma|=n_c}} 2^{-n_c}\frac{d_p(\sigma)}{d_p(\sigma)}$$
$$\leq 2^{-c} \sum_{\substack{\sigma\epsilon\chi_c\\ |\sigma|=n_c}} 2^{-n_c} d_p(\sigma)$$
$$\leq \tfrac{d_p(\epsilon)}{2^c} \text{ using theorem 1.4.}$$

Also $\sum\limits_{n} m(n, c) \leq d(\epsilon)2^{-c}$ Now we consider the function f defined by the equation

$$2^{-f(n)} = \sum_c 2^{\frac{c}{2}} m(n, c).$$

Since each $m(n, c)$ and even the sum $\sum_n m(n, c)$ does not exceed 2^{-c}, the right hand side is a computably convergent series and f is computable. Hence,

$$\sum 2^{-f(n)} = \sum_{n,c} 2^{c/2} m(n, c) \leq \sum_c 2^{-c/2} < \infty.$$

We assign a lexicographic ordering to string of length n in χ_c and hence they can be uniquely and computably determined by c in $log(n) + log(2^n m(n, c)) < O(n)$ time using binary search type algorithm. Hence Kolmogorov complexity of the initial segment till n_c for any $k \leq 1$ using this algorithm say T.

$$C^{n^{k+1}}(\omega \upharpoonright n_c|n) \leq C_T{}^{n^k}(\omega \upharpoonright n_c|n) + O(1)$$
$$\leq 2log\ c + log(2^n m(c, n_c)) + O(1) \leq 2logc + n - f(n) - c + O(1)$$

Keep the constant $2logc - c$ as another constant say $c^{'}$. and this proves the theorem.

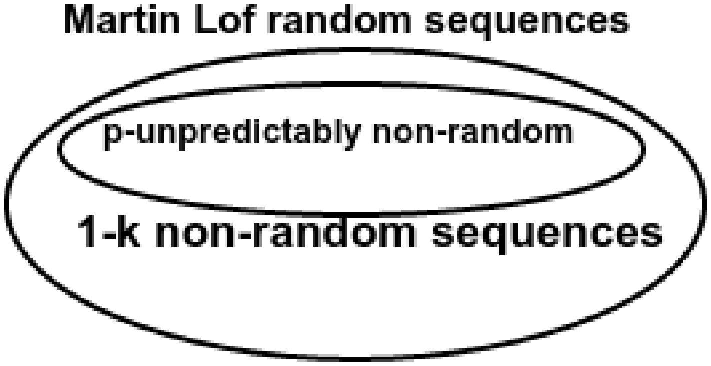

Figure 1: p-unpredictably non random $\subseteq$ 1-k non-random

3.4 Conjectures

We make the following conjectures.
Conjecture 1:

The set of 1-k random sequences forms a proper subset of set of p-unpredictably random sequences.

Conjecture 2:

The second implication of Miller-Yu theorem should not hold.

The justification for second conjecture is that the usual way of construction of martingales for the other implication uses $\sum_n d_n$ (theorem 3.1.1) which is not a polynomial. Hence this way does not work so we feel that the other implication should not hold.

4 Directions for future work

Future work in this area which we would be looking forward is:

- To prove the conjectures that we have made.
- There is famous theorem by Van Lambalgen on generation of random sequences from random sequences and we look forward to study its analogues in resource-bounded measures as generation of polynomial time random sequences from polynomial time random sequences.

References

[1] Laurent Bienvenu, Wolfgang Merkle, and Alexander Shen. A simple proof of miller-yu theorem. *Fundamenta Informaticae*, 83(1-2):21–24, 2008.

[2] Gregory J. Chaitin. A theory of program size formally identical to information theory. *J. ACM*, 22(3):329–340, 1975.

[3] Rodney G Downey and Denis R Hirschfeldt. *Algorithmic randomness and complexity.* Springer Science & Business Media, 2010.

[4] Ming Li and PMB Vitanyi. *An introduction to Kolmogorov complexity and its applications.* Springer, 2008.

[5] Per Martin Löf. Complexity oscillations in infinite binary sequences. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 19.

[6] Per Martin-Löf. The definition of random sequences. *Information and Control*, 9(6):602–619, 1966.

[7] Joseph Miller and Liang Yu. On initial segment complexity and degrees of randomness. *Transactions of the American Mathematical Society*, 360(6):3193–3210, 2008.