Lectures Notes : Algorithmic Information Theory

Itisan Halias

itha.mail0@gmail.com

September 2025

Contents

Chapter 1

Algorithmic "Löf" Randomness

1.1 Measure theoretical typicalness - Effectively Null Set

1.1.1 Effective Null sets

We will first establish a convention that will be used throughout this chapter.

Convention:

For a partial function ϕ and $x \notin dom(\phi)$, we set by convention $\Gamma_{\phi(x)} := \emptyset$.
It is crucial to note that this convention cannot be used in an effective procedure. Indeed, to apply it, it would be necessary to be able to determine whether $x \notin \text{dom}(\phi)$, which is undecidable according to the halting problem.

Definition: Covering Algorithm / Effectively Null Set

We say that the partial computable function $\phi : \langle \mathcal{N}, \mathcal{N} \rangle \mapsto \mathbb{B}^*$ is a covering algorithm for X if for all $c \in \mathcal{N}$ we have:

1. $X \subset \bigcup_{i \in \mathcal{N}} \Gamma_{\phi(c,i)}$ [Covering condition]
2. $\sum_{i \in \mathcal{N}} \lambda \left(\Gamma_{\phi(c,i)}\right) \leq 2^{-c}$ [Measure condition]

We then say that X is an effectively null set.

Remark. The notion of effectively null sets can be characterized by several equivalent definitions. We present them here for the reader's information. Although we will not use these other definitions subsequently, it can be shown that they are all equivalent to the one we stated above, thus designating the same class of effectively null sets:

- We say that $X \subset \mathbb{B}^\infty$ is an effectively Null set if there exists a partial computable function $\phi : \langle \mathcal{Q}^{>0}, \mathcal{N} \rangle \mapsto \mathbb{B}^*$ such that for all $\varepsilon \in \mathcal{Q}^{>0}$ we have

  1. $X \subset \bigcup_{i \in \mathcal{N}} \Gamma_{\phi(\varepsilon,i)}$
  2. $\sum_{i \in \mathcal{N}} \lambda(\Gamma_{\phi(\varepsilon,i)}) \leq \varepsilon$

- A Martin-Löf test is an effectively enumerable set $W \subseteq \langle \mathcal{N}, \mathcal{N} \rangle$ such that, by setting

$W_n := \{w \in \mathbb{B}^* \mid \langle n, w \rangle \in W\}$, we have for all $n \in \mathbb{N}$:

$$\sum_{w \in W_n} 2^{-|w|} \leq 2^{-n}$$

We then say that W is an effectively null set.

◇

We will now give an equivalent reformulation of a covering algorithm that we will use extensively in the remainder of this chapter for practical reasons.

Property: Reformulation of a covering algorithm

Let β, k be strictly positive integers. There exists a covering algorithm ϕ for X if and only if there exists a partial computable function $\phi' : \langle \mathcal{N}, \mathcal{N} \rangle \mapsto \mathbb{B}^*$ such that for all $c \in \mathcal{N}$:

1. $X \subset \bigcup_{i \in \mathcal{N}} \Gamma_{\phi'(c,i)}$
2. $\sum_{i \in \mathcal{N}} \lambda(\Gamma_{\phi'(c,i)}) \leq \beta \cdot 2^{-k \cdot c}$

Proof. ($\implies$) Let ϕ be a covering algorithm for X. We set $\phi'(c, i) := \phi(k \cdot c, i)$. The covering condition is trivially verified. For the measure, since $\beta \geq 1$:

$$\sum_{i \in \mathbb{N}} \lambda(\Gamma_{\phi'(c,i)}) = \sum_{i \in \mathbb{N}} \lambda(\Gamma_{\phi(k \cdot c,i)}) \leq 2^{-k \cdot c} \leq \beta \cdot 2^{-k \cdot c}$$

($\impliedby$) Conversely, let ϕ' be a function satisfying the property for integers $\beta, k \geq 1$. Let $c_0 \in \mathbb{N}$ be a constant integer such that $2^{c_0} \geq \beta$. We define $\phi(c, i) := \phi'(c + c_0, i)$. The covering condition is immediate. For the measure, we have for all $c \in \mathbb{N}$:

$$\sum_{i \in \mathbb{N}} \lambda(\Gamma_{\phi(c,i)}) = \sum_{i \in \mathcal{N}} \lambda(\Gamma_{\phi'(c+c_0,i)}) \leq \beta \cdot 2^{-k(c+c_0)} = (\beta \cdot 2^{-kc_0}) \cdot 2^{-kc}$$

Now, $2^{c_0} \geq \beta$ and $k \geq 1$ imply $2^{kc_0} \geq \beta$, therefore $\beta \cdot 2^{-kc_0} \leq 1$. Moreover, $k \geq 1$ implies $2^{-kc} \leq 2^{-c}$. It follows that:

$$\sum_{i \in \mathbb{N}} \lambda(\Gamma_{\phi(c,i)}) \leq 1 \cdot 2^{-kc} \leq 2^{-c}$$

Thus, ϕ is a covering algorithm, which concludes the proof. ■

We can then show that the union of a sequence of Null Sets whose covering algorithms can be enumerated is stable as stated in the following property.

Property: Stability under union of effectively null sets

Let $\phi_0, \phi_1, \phi_2, \ldots$ be covering algorithms for $X_0, X_1, X_2, \ldots$ respectively and let $\varrho : i \in \mathcal{N} \mapsto \langle \phi_i \rangle$ be a totally computable function. Then, there exists a covering algorithm for $\bigcup_{i \in \mathcal{N}} X_i$.

Proof idea. We want a covering of $\bigcup_{i \in \mathcal{N}} X_i$ with precision 2^{-c}. To do this, we ask ϕ_0 to provide a covering of X_0 with measure $\leq 2^{-(c+1)}$, ϕ_1 a covering of X_1 with measure $\leq 2^{-(c+2)}$, and so on. The collection of all these coverings indeed covers the union, and the total measure is bounded by $\sum_{i=0}^{\infty} 2^{-(c+i+1)} = 2^{-c}$. ■

Proof. Let $\pi : \langle \mathcal{N}, \mathcal{N} \rangle \to \mathcal{N}$ be the Cantor pairing function. The goal is to construct a covering algorithm Φ for the set $X := \bigcup_{i\in\mathbb{N}} X_i$. To do this, let us define the partial computable function $\Phi : \langle \mathcal{N}, \mathcal{N} \rangle \to \mathbb{B}^*$ by:

$$\Phi(c,k) = \phi_i\,(c+i+1, j) \qquad \text{where } \pi^{-1}(k) = \langle\, i, j \,\rangle$$

Let us justify that the function Φ is partial computable: For an input $\langle\, c, k \,\rangle$ 1) Compute $\pi^{-1}(k) = \langle\, i, j \,\rangle$; 2) Retrieve the algorithm ϕ_i by applying $\varrho(i)$; 3) Compute $\phi_i(\langle\, c+i+1, j \,\rangle)$ and write it to the output.

Let us verify that Φ is indeed a covering algorithm for X. Let $c \in \mathcal{N}$.

1. Covering condition: Let us show that $X \subset \bigcup_{k\in\mathbb{N}} \Gamma_{\Phi(c,k)}$. Let $x \in X$. By definition of X, there exists an index $i_0 \in \mathcal{N}$ such that $x \in X_{i_0}$. Since ϕ_{i_0} is a covering algorithm for X_{i_0}, for the integer $c' = c + i_0 + 1$, we have $X_{i_0} \subset \bigcup_{j\in\mathcal{N}} \Gamma_{\phi_{i_0}(c',j)}$. There exists therefore a $j_0 \in \mathcal{N}$ such that $x \in \Gamma_{\phi_{i_0}(c',j_0)}$. Let us set $k_0 = \pi(i_0, j_0)$. By definition of Φ, we have $\Phi(c, k_0) = \phi_{i_0}(c + i_0 + 1, j_0)$. Thus, $x \in \Gamma_{\Phi(c,k_0)} \subset \bigcup_{k\in\mathbb{N}} \Gamma_{\Phi(c,k)}$.

2. Measure condition: Let us show that $\sum_{k\in\mathcal{N}} \lambda(\Gamma_{\Phi(c,k)}) \leq 2^{-c}$. By the bijection of π we can reindex the sum:

$$\sum_{k\in\mathcal{N}} \lambda(\Gamma_{\Phi(c,k)}) = \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \lambda(\Gamma_{\Phi(c,\pi(i,j))}) = \sum_{i\in\mathcal{N}} \left(\sum_{j\in\mathcal{N}} \lambda\left(\Gamma_{\phi_i(c+i+1,j)}\right) \right)$$

By definition of ϕ_i, for each i, we have the upper bound:

$$\sum_{j\in\mathcal{N}} \lambda\left(\Gamma_{\phi_i(c+i+1,j)}\right) \leq 2^{-(c+i+1)}$$

By substituting, we therefore obtain:

$$\sum_{k\in\mathcal{N}} \lambda(\Gamma_{\Phi(c,k)}) \leq \sum_{i=0}^{\infty} 2^{-(c+i+1)} = 2^{-c} \sum_{i=0}^{\infty} \frac{1}{2^{i+1}} = 2^{-c} \cdot 1 = 2^{-c}$$

Both conditions being satisfied, Φ is a covering algorithm for X, which is therefore an effectively null set. ■

1.1.2 Maximal Effectively Null Set

Algorithm: Mapping $\phi \xrightarrow{pc \Rightarrow cv} \phi'$

Let ϕ be a partial computable function. We then define the partial computable function ϕ' as follows:

Input: $\langle\, c, i \,\rangle \in \langle \mathcal{N}, \mathcal{N} \rangle$

Step 1. Assign to the sequence:

- $j \leftarrow 0$
- $S_0 \leftarrow 0$

Step 2. Call via dovetailing $\phi(c,t)$ for $t = 0, 1, 2, \ldots$ For each computation $\phi(c,t)$ that halts, go to step 3.

Step 3. If $S_j + 2^{-|\phi(c,t)|} \leq 2^{-c}$ then do: [$\diamond$]

- Assign $k_j \leftarrow t$
- Assign $S_{j+1} \leftarrow S_j + 2^{-|\phi(c,k_j)|}$
- If $j == i$ then [$\otimes$]
  - Write $\phi(c,k_j)$ to output and accept.
- Assign $j \leftarrow j + 1$

Resume the dovetailing call at step 2

Lemma: Mapping

Let ϕ be a partial computable function and ϕ' the function obtained after mapping ($\phi \xrightarrow{pc \Rightarrow cv} \phi'$). We then have:

1. ϕ' is a covering algorithm.
2. If ϕ is a covering algorithm for X, then ϕ' is also a covering algorithm for X.

Proof. Let $c \in \mathcal{N}$ and let $\phi_\otimes$ be the procedure ϕ' without its halting condition [$\otimes$]. The computation of $\phi_\otimes(c,\cdot)$ assigns the sequences $(S_j)_{j<j_{\max}}$, $(k_j)_{j<j_{\max}}$, and $(t_j)_{j<j_{\max}}$, where $j_{\max} \in \mathcal{N} \cup \{\infty\}$ is the maximum value that j takes.

1) Let us set the loop invariant $I(j)$ for $0 \leq j \leq j_{max}$:

$$\text{"At the start of step 2, } S_j = \sum_{l=0}^{j-1} \lambda(\Gamma_{\phi(c,k_l)}) \leq 2^{-c}\text{."}$$

Base case ($j = 0$): The call $\phi^\otimes(c,\cdot)$ initializes $j = 0$ and $S_0 = 0$ at step 1. We thus have $S_0 = \sum_{l=0}^{0-1} \lambda(\Gamma_{\phi(c,k_l)}) = 0 < 2^{-c}$. That is, $I(0)$ is satisfied.

Maintenance ($j \to j+1$): Assume that Inv_j is true for some $j < j_{max}$. By definition of j_{max}, there still exists a $\phi(c,t)$ that halts in the dovetailing call and that satisfies $\diamond$ during the passage through step 3 (otherwise j never reaches j_{max}). For such a $\phi(c,t)$ at step 3:

- We have the assignment of $k_j \leftarrow t$ and $S_{j+1} \leftarrow S_j + 2^{-|\phi(c,k_j)|}$
- Knowing that we assumed $I(j)$ to be true, we then obtain

$$S_{j+1} = \sum_{l=0}^{j-1} \lambda(\Gamma_{\phi(c,k_l)}) + 2^{-|\phi(c,k_j)|} = \sum_{l=0}^{j} \lambda(\Gamma_{\phi(c,k_l)}) \leq 2^{-c}$$

- We then have $j \leftarrow j+1$ and a return to step 2. The invariant $I(j+1)$ is therefore satisfied.

Termination: We reconsider the condition $\otimes$ now. By $\otimes$, we notice that $\phi'(c,i) = \phi(c,k_i)$ for $i \in \{0, 1, \ldots, j_{max}\}$. Moreover, by definition of j_{max} we observe that $i > j_{max}$ implies

$\phi'(c,i)$ does not halt. We thus have for $i \in dom(\phi') = \{0, 1, \dots, j_{max}\}$ that $\phi'(c,i) = \phi(c,k_i)$. We can then write

$$\sum_{n\in\mathcal{N}} \lambda(\Gamma_{\phi'(c,n)}) = \sum_{\langle c,n\rangle \in dom(\phi')} \lambda(\Gamma_{\phi'(c,n)}) = \sum_{n\in\{0,1,\dots,j_{max}-1\}} \lambda(\Gamma_{\phi(c,k_n)}) \leq 2^{-c}$$

with the last inequality by $I(j_{max})$. Since this is true for all c in $\mathcal{N}$, we have that ϕ' is a covering algorithm.

2) Assume ϕ is a covering algorithm for X. We know by the point we just proved that ϕ' is a covering algorithm. It remains to show that $X \subset \bigcup_{i\in\mathcal{N}} \Gamma_{\phi'(c,i)}$. For the call $\phi_{\otimes}(c,\cdot)$, let $(t_n)_{n=1,2,\dots}$ be the sequence, finite or infinite, of integers such that $\phi(c,t_n)$ is the n-th computation to halt in the dovetailing.

Let us first prove that for all $j \in \mathcal{N}$ we have $k_j = t_j$. Let us proceed by contradiction and assume that this is not the case. There thus exists a smallest j_0 such that $k_{j_0} \neq t_{j_0}$. This necessarily means that the condition $\diamond$ failed, $S_{j_0} + 2^{-|\phi(c,t_{j_0})|} > 2^{-c}$. Moreover, by $\sharp$, we necessarily have that S_{j_0} is a finite sum of the form $\sum_{z\in K} 2^{-|z|}$ with $K = \{\phi(c,t_{l_0}), \phi(c,t_{l_1}), \dots, \phi(c,t_{l_{|K|}})\}$ and each element of $t_{l_0}, \dots, t_{l_{|K|}}$ pairwise distinct and all strictly less than t_{j_0}. We would therefore have $(\sum_{t\in\{t_0,\dots,t_{|K|}\}} 2^{-|\phi(c,t_{l_z})|}) + 2^{-|\phi(c,t_{j_0})|} > 2^{-c}$, which is absurd because by definition of a covering algorithm $\sum_{t=0}^{\infty} 2^{-|\phi(c,t)|} \leq 2^{-c}$. We thus indeed have for all $j \in \mathcal{N}$ that $k_j = t_j$.

We will now show that $X \subset \bigcup_{i\in\mathcal{N}} \Gamma_{\phi'(c,i)}$. Let $x \in X$. By definition of a covering algorithm for X, there necessarily exists an i_0 such that $x \in \Gamma_{\phi(c,i_0)}$ with $\langle c, i_0 \rangle \in dom(\phi)$. Since $\langle c, i_0 \rangle \in dom(\phi)$ there exists an l such that $i_0 = t_l$ (because the dovetailing call halts on all words of $dom(\phi)$). And therefore by the result we just showed $i_0 = k_l$. Thus we have $\phi'(c,i_0) = \phi'(c,k_l)$. Now, by the halting condition $\otimes$ we have $\phi'(c,k_l) = \phi(c,k_l)$. We therefore have that $\phi'(c,k_l) = \phi(c,i_0)$. Thus $x \in \Gamma_{\phi'(c,k_l)}$, which shows that ϕ' is a covering algorithm for X. ■

Theorem: Maximal Effectively Null Set

There exists a covering algorithm ϕ_{max} for X_{max} such that for every effectively null set X we have $X \subset X_{max}$. We then say that

- X_{max} is the maximal effectively null set.
- ϕ_{max} is a universal covering algorithm.

Proof. Let $\pi : i \in \mathcal{N} \mapsto \langle \phi_i \rangle \in \langle \mathcal{T} \rangle$ be an enumeration of the partial computable functions. Let us then define the partial computable function ϱ with the following operation

> Input: $i \in \mathcal{N}$; 1) Assign $\langle \phi_i \rangle \leftarrow \pi(i)$; 2) Apply the mapping $\phi_i \xrightarrow{pc\Rightarrow cv} \phi'_i$; 3) Write $\langle \phi'_i \rangle$ to output and then accept.

Let Ξ be the set of effectively null sets. Let us show that for all $X \in \Xi$ there exists an i_0 such that $\varrho(i) = \langle \phi'_{i_0} \rangle$ with ϕ'_{i_0} being a covering algorithm for X. Let $X \in \Xi$. Since the enumeration π enumerates all partial computable functions, the enumeration also contains all covering algorithms. Thus there exists an i_0 such that $\pi(i_0) = \langle \phi_{i_0} \rangle$ is a covering algorithm for X. By the mapping lemma this implies that $\phi_{i_0} \xrightarrow{pc\Rightarrow cv} \phi'_{i_0}$ with ϕ'_{i_0} being a covering algorithm for X. Now $\varrho(i_0) = \langle \phi'_{i_0} \rangle$.

By the stability property of the union of effectively null sets, we then have that there exists a covering algorithm ϕ_{max} for $X_{max} := \bigcup_{X \in \Xi} X$. ■

1.1.3 Martin-Löf Randomness

Definition: Martin-Löf Random

We say that $\alpha \in \mathbb{B}^\infty$ is ML-Random if for every effectively null set X we have $\alpha \notin X$.

Remark. Equivalently, this means that if there exists a covering algorithm ϕ for X such that $\alpha \in X$, then α is not ML-Random. ◇

We can then directly state the following trivial property.

Property: ML-random Characterization

Let $\alpha \in \mathbb{B}^\infty$, we have equivalence between

1. α is ML-Random.
2. $\alpha \notin X_{max}$

Proof. $1 \Rightarrow 2$: By contraposition: Assume that $\alpha \in X_{max}$, then α belongs to an Effective Null Set, that is, α is non-ML-Random.

$2 \Rightarrow 1$: Also by contraposition: Assume that α is non-ML-Random, then there exists an Effective Null Set X such that $\alpha \in X$, therefore $\alpha \in X_{max}$ ■

1.2 Incompressibility - Kolmogorov complexity

1.2.1 Characterization by K - Schnorr

Theorem: Schnorr's Theorem

Let $\alpha \in \mathbb{B}^\infty$, we have α is ML-Random if and only if there exists $c \in \mathcal{N}$ such that

$$\forall n \in \mathcal{N}, \quad K(\alpha \upharpoonright n) \geq n - c$$

Proof. ($\implies$) Let us proceed by contraposition, that is, let us assume that α possesses c-defective prefixes for all $c \in \mathcal{N}$, ($\forall c \in \mathcal{N}, \exists n \in \mathcal{N}$ such that $K(\alpha \upharpoonright n) \leq n - c$), and let us show that α is non-ML-Random.

To do this, let us define the set of sequences of $\mathbb{B}^\infty$ that are c-defective, i.e., formally

$$\theta := \{\omega \in \mathbb{B}^\infty \mid \forall c \in \mathbb{B}^*, \exists N_c \in \mathcal{N}, K(w \upharpoonright N_c) < N_c - c\}$$

We notice immediately that $\alpha \in \Theta$. It is therefore sufficient for us to prove that θ is an effectively null set. For all $c \in \mathcal{N}$, let us set the set $\mathcal{D}_c := \{u \in \mathcal{B}^* \mid K(u) < |u| - c\}$ and let us show that $\mathcal{D}_c$ is semi-decidable, which the following procedure proves

> For an input $u \in \mathbb{B}^*$. Search by calling via dovetailing $\phi_{uc}(u, t)$ on $t = 0, 1, 2, \ldots$ for a computation such that $\phi_{uc}(u, t) < |u| - c$. If such a computation is found then accept. (With here, $\phi_{uc} \xrightarrow{uc} K$ because K is semi-computable from above)

The procedure semi-decides $\mathcal{D}_c$; for $u \in \mathcal{D}_c$ there necessarily exists t_0 such that $\phi_{uc}(u, t_0) < |u| - c$. Thus for all $c \in \mathcal{N}$ there exists a bijective enumeration of $\mathcal{D}_c$. We can then set the partial computable function $\phi : \langle \mathcal{N}, \mathcal{N} \rangle \mapsto \mathbb{B}^*$ such that for all $c \in \mathcal{N}$ we have $\phi(c, \cdot)$ is a bijective enumeration of $\mathcal{D}_c$. Let us now show that ϕ is a covering algorithm for θ:

- Let us show that for all $c \in \mathcal{N}$, we have $\theta \subset \bigcup_{i \in \mathcal{N}} \Gamma_{\phi(c,i)}$. Let $c \in \mathcal{N}$ and $\omega \in \theta$. By definition of θ, there necessarily exists a $u \in \mathbb{B}^*$ such that $K(u) < |u| - c$ and $\omega = uz$ with $z \in \mathbb{B}^\infty$. By bijectivity in $\mathcal{D}_c$, we then have that there exists an i_0 such that $\phi(c, i_0) = u$. We then have $\omega \in \Gamma_{\phi(c,i_0)}$, that is $\omega \in \bigcup_{i \in \mathcal{N}} \Gamma_{\phi(c,i)}$.

- Let us show that for all $c \in \mathcal{N}$ we have $\sum_{i \in \mathcal{N}} \lambda(\Gamma_{\phi(c,i)}) \leq 2^{-c}$. Let $c \in \mathcal{N}$. We then have that

  $$\sum_{i \in \mathcal{N}} \lambda(\Gamma_{\phi(c,i)}) = \sum_{i \in \{1,2,\ldots,|\mathcal{D}_c|\}} 2^{-|\phi(c,i)|}$$

  Now by definition of a c-defective word we have for all $i \in \{1, 2, \ldots, |\mathcal{D}_c|\}$ that $K(\phi(c, i)) < |\phi(c, i)| - c$ which gives $2^{-|\phi(c,i)|} < 2^{-K(\phi(c,i))} \cdot 2^{-c}$. We deduce,

  $$\sum_{i \in \mathcal{N}} \lambda(\Gamma_{\phi(c,i)}) < \sum_{i \in \{1,2,\ldots,|\mathcal{D}_c|\}} 2^{-K(\phi(c,i))} \cdot 2^{-c} = 2^{-c} \sum_{i \in \{1,2,\ldots,|\mathcal{D}_c|\}} 2^{-K(\phi(c,i))} \leq 2^{-c}$$

  with the last inequality by Kraft's inequality.

We therefore have that ϕ is a covering algorithm for θ. Since $\alpha \in \theta$ we have α non-ML-random.

($\Longleftarrow$) Let us show this by contraposition and assume for this that α is non-ML-Random. Let X_{max} be the maximum Effective Null Set. There exists a covering algorithm $\phi : \langle \mathcal{N}, \mathcal{N} \rangle \mapsto \mathbb{B}^*$ such that for all $c \in \mathcal{N}$

- $X_{max} \subset \bigcup_{i \in \mathcal{N}} \Gamma_{\phi(c,i)}$
- $\sum_{i \in \mathcal{N}} \lambda(\Gamma_{\phi(c,i)}) \leq 2^{-2c}$

Let us then set the function $\upsilon : \langle x, i \rangle \in \langle \mathcal{N}, \mathcal{N} \rangle \mapsto |\phi(c, i)| - c$ which is partial computable because ϕ is. We then notice that for all c in $\mathcal{N}$

$$\sum_{i \in \mathcal{N}} 2^{-\upsilon(c,i)} = \sum_{i \in \mathcal{N}} 2^{-\phi(c,i)} \cdot 2^c \leq 2^{-2c} \cdot 2^c \leq 1$$

Let us set the function $A : x \in \mathbb{B}^* \mapsto \sum_{\langle c,i \rangle | \phi(c,i) = x} 2^{-n(c,i)}$. We can then justify that A is lower semicomputable by the following procedure

> For an input $\langle x, k \rangle \in \langle \mathbb{B}^*, \mathcal{N} \rangle$. Call via dovetailing ϕ for p iterating through $\langle \mathcal{N}, \mathcal{N} \rangle$. Assign to P the inputs p of the first k computations that halt with x as output (If there are strictly fewer than k, the procedure loops). Write $\sum_{p \in P} 2^{-n(p)}$ to output then accept.

Let $\mathbf{m}$ be the universal measure (that is, dominant). Since A is lower semicomputable there exists $\gamma \in \mathcal{N}$ such that for all $x \in \mathbb{B}^*$ we have $\mathbf{m}(x) \geq \gamma \cdot A(x)$ which by Levin's Coding Theorem gives $K(x) \leq -\log(A(x)) + \eta$ with $\eta \in \mathcal{N}$ a constant (independent of x).

For all $\langle c, i \rangle$ in $dom(\phi)$, we notice that $A(\phi(c,i)) \geq 2^{-v(c,i)}$ which, by taking $x = \phi(c,i)$ in the previous relation, gives

$$\begin{aligned} K(\phi(c,i)) &\leq -\log(A(\phi(c,i))) + \eta \\ &\leq -\log(2^{-v(c,i)}) + \eta \qquad \text{[since -log is decreasing]} \\ &\leq v(c,i) + \eta \\ &\leq |\phi(c,i)| - c + \eta \end{aligned}$$

We had assumed that α is non-ML-Random, which implies $\alpha \in X_{max}$. Let $c \in \mathcal{N}$. We then have $\alpha \in \bigcup_{i\in\mathcal{N}} \Gamma_{\phi(c,i)}$. There exists therefore an $i_0 \in \mathcal{N}$ such that $\alpha \in \Gamma_{\phi(c,i_0)}$. Thus, we necessarily deduce that α contains the prefix $\phi(c,i_0)$, that is by setting $n_c = |\phi(c,i_0)|$ we have $K(\alpha \upharpoonright n_c) \leq n_c - c + \eta$. In summary, for all $c \in \mathcal{N}$ there exists n_c such that $K(\alpha \upharpoonright n_c) \leq n_c - c + \eta$, which concludes the proof by contraposition. ■

1.2.2 Characterization by C - Miller and Yu

Theorem: Miller-Yu (If)

There exists a computable function $g : \mathcal{N} \to \mathcal{N}$ satisfying $\sum_{n=0}^{\infty} 2^{-g(n)} < \infty$ and satisfying for all $\alpha \in \mathbb{B}^\infty$ the equivalence

$$\alpha \text{ is ML-Random} \quad \Longrightarrow \quad \exists \eta \in \mathcal{N}, \quad \forall n \in \mathcal{N}, \quad \mathscr{C}(\alpha \upharpoonright n) \geq n - g(n) - \eta$$

Proof. Let $\alpha \in \mathbb{B}^\infty$. Let us proceed by contraposition, that is, let us assume that for all $\eta \in \mathcal{N}$ there exists an $n \in \mathcal{N}$ such that $\mathscr{C}(\alpha \upharpoonright n) < n - g(n) - \eta$ and let us show that α is non-ML-Random. Let us define the set

$$\theta := \{\omega \in \mathbb{B}^\infty \mid \forall \eta \in \mathcal{N}, \quad \exists n \in \mathcal{N}, \quad \mathscr{C}(\omega \upharpoonright n) < n - g(n) - \eta\}$$

We notice $\alpha \in \theta$. It is thus sufficient to show that θ is an effective Null Set. Let us define the sets $\mathcal{F}_\eta := \{u \in \mathbb{B}^* \mid \mathscr{C}(u) < |u| - g(|u|) - \eta\}$ for all $\eta \in \mathcal{N}$. We can then show that $\mathcal{F}_\eta$ is effectively enumerable by showing that it is semi-decidable. $\mathscr{C}$ is computable from above $\phi_{uc} \xrightarrow{uc} \mathscr{C}$. We then have the set $\mathcal{F}_\eta$ which is semi-decidable by the following procedure

> For an input $u \in \mathbb{B}^*$, loop over $t = 0, 1, 2, \ldots$ until obtaining $\phi_{uc}(u,t) < |u| - g(u) - \eta$ then accept the input.

For an input u in $\mathcal{F}_\eta$ the procedure necessarily halts because there exists a t_0 such that $\phi_{uc}(u,t_0) < |u| - g(u) - \eta$. We can then set the partial computable function $\phi : \langle \mathcal{N}, \mathcal{N} \rangle \mapsto \mathbb{B}^*$ such that, for each $\eta \in \mathcal{N}$, we have $\phi(\eta, \cdot)$ is a bijective effective enumeration of $\mathcal{F}_\eta$. Let us now show that ϕ is a covering algorithm for θ.

Covering condition: Let us show that for all η we have $\theta \subset \bigcup_{i\in\mathcal{N}} \Gamma_{\phi(\eta,i)}$. Let $\eta \in \mathcal{N}$ and $\omega \in \theta$. By definition of θ there exists $u \in \mathbb{B}^*$ such that $\mathscr{C}(u) < |u| - g(u) - \eta$ and $u \leq_p \omega$. By bijectivity there exists i_0 such that $\phi(\eta, i_0) = u$. Thus $\omega \in \Gamma_{\phi(\eta,i_0)}$ which implies $\theta \subset \Gamma_{\phi(\eta,i_0)}$.

Measure condition: Let $\eta \in \mathcal{N}$. We can decompose by the bijectivity of $\phi(c,\cdot)$ then by splitting on the lengths of $u \in \mathcal{F}_\eta$ write

$$\sum_{i\in\mathcal{N}} \lambda\left(\Gamma_{\phi(\eta,i)}\right) = \sum_{u\in\mathcal{F}_\eta} \lambda\left(\Gamma_u\right) = \sum_{n\in\mathcal{N}} \sum_{u\in\mathcal{F}_\eta \text{ and } |u|=n} 2^{-|u|}$$

Now for a fixed $n \in \mathcal{N}$ we have by an incompressibility property that $|\{x \in \mathbb{B}^* \mid \mathscr{C}(x) < n - g(n) - \eta\}| \leq 2^{n-g(n)-\eta}$, that is $|\{u \in \mathcal{F}_\eta \mid |u| = n\}| \leq 2^{n-g(n)-\eta}$. This allows us to write, still for an $n \in \mathcal{N}$,

$$\sum_{u \in \mathcal{F}_c \text{ and } |u|=n} 2^{-|u|} = |\{u \in \mathcal{F}_\eta \mid |u| = n\}| \cdot 2^{-n} \leq 2^{n-g(n)-\eta} \cdot 2^{-n} \leq 2^{-g(n)-\eta}$$

By substituting we then obtain

$$\sum_{n \in \mathcal{N}} \sum_{u \in \mathcal{F}_\eta \text{ and } |u|=n} 2^{-|u|} = \sum_{n \in \mathcal{N}} 2^{-g(n)-\eta} \leq \beta \cdot 2^{-\eta}$$

with $\beta = \sum_{n \in \mathcal{N}} 2^{-g(n)}$ independent of η. We have thus shown the measure condition.

Conclusion: ϕ is a covering algorithm for θ. Now $\alpha \in \theta$, which proves that α is non-ML-Random and completes the proof by contraposition. ■

Theorem: Miller-Yu (Only if)

There exists a computable function $g : \mathcal{N} \to \mathcal{N}$ satisfying $\sum_{n=0}^{\infty} 2^{-g(n)} < \infty$ and satisfying for all $\alpha \in \mathbb{B}^\infty$ the equivalence

$$\alpha \text{ is ML-Random} \quad \Longleftarrow \quad \exists \eta \in \mathcal{N}, \quad \forall n \in \mathcal{N}, \quad \mathscr{C}(\alpha \upharpoonright n) \geq n - g(n) - \eta$$

Proof. Let ϕ be a covering algorithm for X_{max}, a maximal effective Null Set. Let us set the sets $W = \{\langle c, u \rangle \mid \exists i \in \mathcal{N}, \phi(c, i) = u\}$. We notice that the set W is enumerable by the following effective procedure

> For an input $n \in \mathcal{N}$. Call ϕ via dovetailing on the inputs $\langle \mathcal{N}, \mathcal{N} \rangle$. Write the n-th computation that halts to the output and accept.

Moreover we notice that W is necessarily infinite. Indeed assume that $|W| < \infty$, then there necessarily exists a c such that $X_{max} \subset \bigcup_{i \in \mathcal{N}} \Gamma_{\phi(c,i)} = \emptyset$ which is absurd (we would have $X_{max} = \emptyset$). We can therefore set an effective enumeration $\pi_W : i \in \mathcal{N} \mapsto \langle c_i, u_i \rangle \in W$ which is bijective.

Let us now define two effectively enumerable sequences $(n_i)_{i \in \mathcal{N}}$ and $(l_i)_{i \in \mathcal{N}}$ as follows

$$\forall i \in \mathcal{N}, \quad \begin{cases} l_i = min\{l \in \mathcal{N} \mid \ell(u_i) + l > n_{i-1} \text{ and } l > l_{i-1}\} \\ n_i = \ell(u_i) + l_i \end{cases} \qquad \text{with } n_{-1} = l_{-1} = -1$$

The sequences thus defined are indeed computable because all necessary operations (obtaining u_i by π_Λ, searching for a minimum...) are. We notice that $(n_i)_{i \in \mathcal{N}}$ is a strictly increasing sequence because for all $i \in \mathcal{N}$ we have $\ell(u_i) + l_i > n_{i-1}$ is equivalent to $n_i > n_{i-1}$. Moreover we notice immediately that $(l_i)_{i \in \mathcal{N}}$ is also strictly increasing.

Construction of g: Let us set the set consisting of the elements of the sequence $(n_i)_i$, that is $R := \{n_i \mid i \in \mathcal{N}\}$. Let us define the function g as follows

$$g(n) = \begin{cases} \ell(u_i) - \lceil \log(c_i + 1) \rceil & \text{if } n \in R \\ n & \text{otherwise.} \end{cases}$$

We can then show that g is totally computable by the following effective procedure:

Input: $n \in \mathcal{N}$
Initialize $j \leftarrow 0$
While $n_j \leq n$ loop:

- If $n = n_i$ then
  - Write $\ell(u_i) - \lceil \log(c_i + 1) \rceil$ to output and accept.
- Assign $j \leftarrow j + 1$

Write n to output and accept

Let us justify that this procedure computes g. Consider an input $n \in \mathcal{N}$. Let j_0 be the smallest integer such that $n_{j_0} > n$. If there exists i such that $n_i = n$ then, by strict increasingness of the n_j, we have $i < j_0$, which implies that the condition will be satisfied when $j = i$, hence writing $\ell(u_i) - \lceil \log(c_i + 1) \rceil$ to output. Otherwise, if $n \notin R$, the condition will never be satisfied so n_j increases strictly until exceeding n, hence exiting the loop and writing n to output. We thus indeed have in both cases that the output equals $g(n)$.

Let us now verify that $\sum_{n\in\mathcal{N}} 2^{-g(n)} < \infty$. The series is split into two parts. The sum over the integers $n \notin R$ is bounded by $\sum_{n=0}^{\infty} 2^{-n} = 2$. The sum over the integers $n \in R$ is bounded as follows:

$$\sum_{i=0}^{\infty} 2^{-g(n_i)} = \sum_{i=0}^{\infty} 2^{-(|u_i| - \lceil \log(c_i+1) \rceil)} \leq \sum_{i=0}^{\infty} 2^{-|u_i|} \cdot 2(c_i + 1)$$
$$= \sum_{c=0}^{\infty} 2(c+1) \sum_{u \mid \langle c,u \rangle \in W} 2^{-|u|} \leq 2 \sum_{c=0}^{\infty} (c+1) \cdot 2^{-c} < \infty.$$

The second to last inequality is justified because $\sum_{i\in\mathcal{N}} \lambda(\Gamma_{\phi(c,i)}) = \sum_{u \mid \langle c,u \rangle \in W} 2^{-|u|} \leq 2^{-c}$ by definition of a covering algorithm. Since both series converge, the series converges.

Proof by contraposition: Assume that α is not ML-Random and let us prove that for all $\eta \in \mathcal{N}$ there exists an $n \in \mathcal{N}$ such that $\mathscr{C}(\alpha \upharpoonright n) \leq n - g(n) - \eta$. Let $\eta \in \mathcal{N}$.

Let us set the sets $W_c^\alpha := \{u \mid u \leq_p \alpha \text{ and } \langle c, u \rangle \in W\}$ for all $c \in \mathcal{N}$. We necessarily have for all c that $W_c^\alpha \neq \emptyset$ (otherwise we would have for all i that $\alpha \notin \Gamma_{\phi(c,i)}$ which implies $\alpha \in X_{max} \subset \bigcup_{i\in\mathcal{N}} \Gamma_{\phi(c,i)}$ which is absurd). We then notice

- Consider a $u \in W_c^\alpha$ for any c. Since $\langle u, c \rangle \in W$ there exists i such that $u_i = u$ and $c_i = c$. We recall that $n_i = \ell(u_i) + l_i$ which implies $u_i \leq_p (\alpha \upharpoonright n_i)$. There exists therefore $v \in \mathbb{B}^*$ such that $(\alpha \upharpoonright n_i) = u_i v$ with $\ell(v) = l_i$. Let us set the machine M

  Input: $v \in \mathbb{B}^*$

  1) Find q such that $l_q = \ell(v)$, (this operation is computable because $(l_i)_{i\in\mathcal{N}}$ is strictly increasing. It is then sufficient to enumerate $l_0, l_1, \ldots$ until obtaining $l_q = \ell(v)$)

  2) Write $u_q v$ to output then accept.

  We then notice by construction that $M(v) = u_i v = (\alpha \upharpoonright n_i)$. Thus by the invariance theorem there exists $\delta \in \mathcal{N}$ a constant such that $\mathscr{C}(\alpha \upharpoonright n_i) \leq l_i + \delta$.

- Since $W_c^\alpha \neq \emptyset$ for c as large as we want, there therefore exists a $c \in \mathcal{N}$ and $u \in W_c^\alpha$ such that $\lceil \log(c+1) \rceil \geq \eta + \delta$. We can then set in the sequence k such that $\langle c_k, u_k \rangle = \langle c, u \rangle$ satisfying

$$\begin{cases} \lceil \log(c_k + 1) \rceil \geq \eta + \delta \\ \mathscr{C}(\alpha \upharpoonright n_k) \leq l_k + \delta \end{cases}$$

We then obtain the desired upper bound:

$$\begin{aligned} n_k - g(n_k) - \eta &= \ell(u_k) + l_k - (\ell(u_k) - \lceil log(c_k+1)\rceil) - \eta \\ &= l_k + \lceil \log(c_k+1)\rceil - \eta \\ &\geq l_k + \delta \\ &\geq \mathscr{C}(\alpha \upharpoonright n_k) \end{aligned}$$

Which concludes the proof by contraposition for $n = n_k$. ■

1.3 Unpredictability - Effective Martingales

We adopt the usual notation $\limsup_{n\to\infty} A(n) := \lim_{n\to\infty} \sup_{k\geq n} A(n)$ for A an arbitrary function with domain and codomain in $\mathcal{N}$.

1.3.1 Martingales

Definition: Effective (super-)martingale

A function $f : \mathbb{B}^* \to \mathbb{R}^+ \cup \{0\}$ is a super-martingale if, for all $\sigma \in \mathbb{B}^*$, it satisfies the condition

$$f(\sigma) \geq \frac{f(\sigma 0) + f(\sigma 1)}{2}$$

It is a martingale if this condition is an equality. Moreover, such a (super-)martingale f:

- succeeds on $\alpha \in \mathbb{B}^\infty$ if $\limsup_{n\to\infty} f(\alpha \upharpoonright n) = \infty$;
- is said to be effective if it is lower semicomputable.

1.3.2 Characterization by Martingales - Schnorr

Lemma: Kolmogorov's Inequality

Let $f : \mathbb{B}^* \to \mathbb{R}^+$ be a super-martingale. Let $\nu \in \mathbb{B}^*$ and a set $X \subset \{x \in \mathbb{B}^* \mid \nu \leq_p x\}$ such that X is prefix-free. Then,

$$f(\nu) \geq \sum_{x\in X} 2^{|\nu|-|x|} f(x)$$

Proof. Let us establish a reformulation of f being a super-martingale. To do this, let us set the function $g : \mathbb{B}^* \mapsto \mathbb{R}^+$ such that for all $\sigma \in \mathbb{B}^*$ we have $g(\sigma) = 2^{-|\sigma|} f(\sigma)$. We then notice for all $\sigma \in \mathbb{B}^*$ that

$$\begin{aligned} f(\sigma) \geq \frac{f(\sigma 1) + f(\sigma 0)}{2} &\iff 2^{-|\sigma|} f(\sigma) \geq 2^{-|\sigma|-1} f(\sigma 1) + 2^{-|\sigma|-1} f(\sigma 0) \\ &\iff g(\sigma) \geq g(\sigma 0) + g(\sigma 1) \end{aligned}$$

Let $\nu \in \mathbb{B}^*$. Let us prove by induction on $S_X := \sum_{x\in X} |x| - |\nu|$ that $g(\nu) \geq \sum_{x\in X} g(x)$:

Base ($S_X = 0$): We then have for all $x \in X$ that $|x| - |\nu| = 0$. Since X is prefix-free, this necessarily implies that X contains a single element which is ν. We therefore indeed have $g(\nu) \geq \sum_{x \in X} g(x) = g(\nu)$.

Maintenance ($S_X \geq 0$): Assume the property is true for all $Y \subset \{x \mid \nu \leq_p x\}$ with Y prefix-free such that $S_Y < S_X$. If $S_X = 0$ it is the base case, we then assume $S_X > 0$. There therefore exists an $x \in X$ such that $|x| - |\nu| > 0$. Thus, by definition of X we have that $x >_p \nu$. Since X is assumed to be prefix-free, we therefore have $\nu \notin X$. We then notice that the sets $X_0 := \{x \in X \mid \nu 0 \leq_p x\}$ and $X_1 = \{x \in X \mid \nu 1 \leq_p x\}$ form a partition of X. We can then prove $g(\nu 0) \geq \sum_{x \in X_0} g(x)$ by distinguishing two cases depending on whether X_0 is empty or not:

- If $X_0 \neq \emptyset$. Knowing that X_0 is a subset of the prefix-free set X, we have that X_0 is prefix-free. Moreover, we notice that

$$S_{X_0} := \sum_{x \in X_0} |x| - |\nu 0| = \sum_{x \in X_0} (|x| - |\nu| - 1) < S_X$$

  We can then apply the induction hypothesis (for $Y = X_0$) and assert that $g(\nu 0) \geq \sum_{x \in X_0} g(x)$.

- If $X_0 = \emptyset$. Since $\sum_{x \in X_0} g(x) = 0$ and $g(\nu 0) \geq 0$ then trivially we obtain $g(\nu 0) \geq \sum_{x \in X_0} g(x)$

We can apply the exact same reasoning for the set X_1 by replacing $\nu 0$ with $\nu 1$ and prove the inequality $g(\nu 1) \geq \sum_{x \in X_1} g(x)$.

By the equivalence that we have proved we already have $g(\sigma) \geq g(\sigma 0) + g(\sigma 1)$. By substituting we find:

$$g(\nu) \geq g(\nu 0) + g(\nu 1) \geq \sum_{x \in X_1} g(x) + \sum_{x \in X_0} g(x) = \sum_{x \in X} g(x)$$

We have therefore proved by induction the desired result. Knowing that $\forall \sigma, g(\sigma) = 2^{-|\sigma|} f(\sigma)$ we therefore obtain $f(\nu) \geq \sum_{x \in X} 2^{|\nu| - |x|} f(x)$ which is the desired expression. ■

Lemma: Kolmogorov's Inequality

Let f be a super-martingale and let $S^k(f) := \{\sigma \in \mathbb{B}^* \mid f(\sigma) \geq k\}$ for a $k \in \mathcal{N}$, then

$$\lambda \left(\bigcup_{x \in S^k(f)} \Gamma_x \right) \leq \frac{1}{k} f(\varepsilon)$$

Proof. Let us set the set $X = \lfloor S^k(f) \rfloor$, that is, the minimal prefix language of $S^k(f)$. We then obtain

$$k \cdot \lambda \left(\bigcup_{x \in S^k(f)} \Gamma_x \right) = k \cdot \lambda \left(\bigcup_{x \in X} \Gamma_x \right) = k \cdot \sum_{x \in X} 2^{-|x|} \leq \sum_{x \in X} 2^{-|x|} f(x) \leq f(x)$$

The first inequality being by definition of X: if $\sigma \in \mathbb{B}^*$ then $f(\sigma) \geq k$. The second inequality is obtained by using the previous lemma for $\nu = \varepsilon$. ■

Theorem: Characterization by Martingale - Schnorr

$\alpha \in \mathbb{B}^\infty$ is ML-Random $\iff$ α succeeds for no effective super-martingale.

Proof. ($\implies$) Let us proceed by contraposition. Assume that f, a super-martingale, succeeds on $\alpha \in \mathbb{B}^\infty$ and let us show that α is not ML-Random. To do this, let us set the sets $S^k(f) := \{\sigma \mid f(\sigma) \geq 2^n\}$ for all integers k.

We notice, as f is effective, that $S^k(f)$ is enumerable for all fixed k. Indeed, for $\phi_{lc} \stackrel{uc}{\to} f$, the set is semi-decidable by the following procedure

> For an input $\sigma \in \mathbb{B}^*$. Loop over $t = 0, 1, 2, \ldots$ until verifying $\phi_{uc}(\sigma, t) \geq 2^n$ then accept.

There therefore exists an effective enumeration of the set $S^k(f)$ for all k. We can then set $\phi : \langle \mathcal{N}, \mathcal{N} \rangle \mapsto \mathbb{B}^*$ such that $\phi(k, \cdot)$ is a bijective enumeration of $S^k(f)$ for all k. By the previous lemma for all $k \in \mathcal{N}$,

$$\lambda\left(\bigcup_{x \in S^k(f)} \Gamma_x\right) = \sum_{i \in \mathcal{N}} \lambda(\Gamma_{\phi(k,i)}) \leq f(\varepsilon) \cdot 2^{-i} \leq 2^{-i}$$

Thus we notice that ϕ is a covering algorithm for $\bigcup_{k \in \mathcal{N}} S^k(f)$. It is therefore sufficient for us to show that $\alpha \in S^k(f)$ for all k:

Let k be given. We assumed that f succeeds for α. Thus there exists n such that $f(\alpha \upharpoonright n) \geq 2^k$. We therefore have, by the bijectivity of $\phi(k, \cdot)$ with $S^k(f)$, that there exists an i_0 such that $\phi(k, i_0) = (\alpha \upharpoonright n)$. In other words, $\alpha \in \Gamma_{\phi(k,i_0)} \subset \bigcup_{i \in \mathcal{N}} \Gamma_{\phi(k,i)}$. Since this is for all k, then $\alpha \in \bigcup_{k \in \mathcal{N}} S^k(f)$. α therefore belongs to an effectively null set, that is, it is not ML-Random.

($\impliedby$) Let us proceed by contraposition. Assume that α is non-ML-Random and let us show that there exists a super-martingale that succeeds for α. By definition, there exists a covering algorithm ϕ such that $\alpha \in \bigcup_{i \in \mathcal{N}} \Gamma_{\phi(c,i)}$ and $\sum_{i \in \mathcal{N}} \lambda(\Gamma_{\phi(c,i)}) \leq 2^{-2c}$ for all $c \in \mathcal{N}$.

Let us set, for all integers c, s and the words $\sigma_0, \sigma_1, \ldots$ arranged in lexicographical order, the set $W_{c,s} := \{\phi(c, i) \mid \phi(c, i)$ halts in at least t transitions and $0 \leq i, t \leq s\}$. Let us then set the function ϕ_{lc} such that for all $\langle \sigma, s \rangle \in \langle \sigma, \mathcal{N} \rangle$,

$$\phi_{lc}(\sigma, s) := \sum_{c=0}^{s} 2^c \sum_{\tau \in W_{c,s} \mid \sigma \leq_p \tau} 2^{-(|\tau| - |\sigma|)}$$

For all σ we notice that $\phi_{lc}(\sigma, \cdot)$ is increasing. We can justify that ϕ_{lc} is totally computable by the following effective procedure:

> Input: $\langle \sigma, s \rangle$
> Initialize $S \leftarrow 0$
> For all $c = 0, 1, \ldots, s$ assign:
> - $\theta \leftarrow \{\phi(c, \sigma_j)$ halts in at least t transitions and $\sigma \leq_p$ with $0 \leq j, t \leq s\}$
> - $\theta' \leftarrow \{\tau \in \theta \mid \sigma \leq_p \tau\}$
> - $S \leftarrow S + 2^c \sum_{\tau \in \theta'} 2^{-(|\tau| - |\sigma|)}$
>
> Write S to output then accept.

We then set the function for all $\sigma \in \mathbb{B}^*$ such that $f(\sigma) := \lim_{s\to\infty} \phi_{lc}(\sigma, s)$. By definition f is lower semicomputable. Let us set the set $W_c := \bigcup_{s\in\mathcal{N}} W_{c,s}$. We then notice that $W_c = \{\phi(c,j) \mid \langle c,j \rangle \in dom(\phi)\}$. For all $\sigma \in \mathbb{B}^*$,

$$\begin{aligned} f(\sigma) &= \sum_{c=0}^{\infty} 2^c \left(\sum_{\tau\in W_c \mid \sigma <_p \tau} 2^{-(|\tau|-|\sigma|)} + \sum_{\tau\in W_c \mid \tau=\sigma} 2^{-(|\tau|-|\sigma|)} \right) \\ &= \frac{1}{2} \sum_{c=0}^{\infty} 2^c \left(\sum_{\tau\in W_c \mid \sigma 0 \leq_p \tau} 2^{-(|\tau|-|\sigma 0|)} + \sum_{\tau\in W_c \mid \sigma 1 \leq_p \tau} 2^{-(|\tau|-|\sigma 1|)} \right) + \sum_{c \mid \sigma\in W_c} 2^c \\ &= \frac{f(\sigma 0) + f(\sigma 1)}{2} + \sum_{c \mid \sigma\in W_c} 2^c. \end{aligned}$$

which gives $f(\sigma) \geq \frac{1}{2}(f(\sigma 0) + f(\sigma 1))$. We can now show that

$$f(\varepsilon) = \sum_{c=0}^{\infty} 2^c \sum_{\tau\in W_c} 2^{-|\tau|} \leq \sum_{c=0}^{\infty} 2^c \cdot 2^{-2c} = \sum_{c=0}^{\infty} 2^{-c} = 2$$

with the inequality following by definition of a covering algorithm. This also ensures that for all σ we have $f(\sigma) < \infty$. We therefore indeed have f a super-martingale. It is therefore sufficient for us now to show that f succeeds on α.

Since ϕ is a covering algorithm for $\{\alpha\}$, for all $c \in \mathcal{N}$, we have $\alpha \in \bigcup_{i\in\mathcal{N}} \Gamma_{\phi(c,i)}$ which implies the existence of an $i \in \mathcal{N}$ such that $\beta_c := \phi(c,i) \leq_p \alpha$ verifying

$$f(\sigma) \geq 2^c \cdot (2^{-(|\sigma|-|\sigma|)}) = 2^c$$

The sequence $(f(\beta_c))_{c\in\mathcal{N}}$ is therefore unbounded. Moreover $(f(\beta_c))_{c\in\mathcal{N}}$ is a subsequence of $(f(\alpha \upharpoonright n))_{n\in\mathcal{N}}$. We therefore have that $\limsup_{n\to\infty} f(\alpha \upharpoonright n) = \infty$, which shows that f is a super-martingale that succeeds on α. ■

1.4 Chaitin's Omega

In this section, we fix a reference universal prefix Turing machine U_{pf}.

Definition: Chaitin's Omega

We call Chaitin's Omega number the real number equal to

$$\Omega = \sum_{p \mid U_{pf}(p)\downarrow} 2^{-\ell(p)}$$

Remark. Note that the value of Ω depends on the choice of the universal prefix Turing machine. We adopt the convention that Ω is written with its proper expansion, that is, Ω does not contain an infinite sequence of consecutive 1s. We will denote by $\Omega_{1:n}$ the first n bits of Ω. ◇

Lemma:

Let $H_n := \{p \in \mathbb{B}^{\leq n} \mid U_{pf}(p) \downarrow\}$ be the sets for all $n \in \mathcal{N}$. There exists a partial computable function ϕ such that for all $n \in \mathcal{N}$ and $p \in \mathbb{B}^{\leq n}$,

$$\phi(\Omega_{1:n}, p) = \begin{cases} 1 & \text{if } p \in H_n \\ 0 & \text{otherwise} \end{cases}$$

In other words, knowing $\Omega_{1:n}$, it is possible to decide for p of length at most n whether $U_{pf}(p)$ halts.

Proof. Let $n \in \mathcal{N}$. Let us define the partial computable function ϕ with the operation

For an input $\langle \Omega', p \rangle$ in $\langle \mathbb{B}^*, \mathbb{B}^{\leq n} \rangle$

Initialize $S \leftarrow 0$

Call U_{pf} via dovetailing on inputs $q \in \mathbb{B}^*$. For a computation $U_{pf}(q)$ that halts:

- Assign $S \leftarrow S + 2^{-|q|}$
- If $q = p$ then accept with 1 as output. [$\sharp$]
- If $S \geq \Omega'$ then accept with 0 as output. [$\dagger$]

Let us consider a $p \in \mathbb{B}^{\leq n}$ in the following.

Let us justify that $\phi(\Omega_{1:n}, p)$ halts. If condition $\sharp$ is verified: For a step of the dovetailing we will have termination. If condition $\sharp$ is never verified: At each computation $U_{pf}(q)$ that halts in the dovetailing, S will be incremented by $2^{-|q|}$. Thus S tends towards Ω. Knowing that $\Omega \geq \Omega_{1:n}$, we will necessarily have that S will eventually exceed or equal $\Omega_{1:n}$, therefore condition $\dagger$ will be verified, hence the termination.

Let us now justify that $\phi(\Omega_{1:n}, p)$ indeed corresponds to the function described in the statement of the lemma. Since the call $\phi(\Omega_{1:n}, p)$ halts, two cases are possible:

- Case $\phi(\Omega_{1:n}, p) = 1$: This necessarily means that condition $\sharp$ was verified, which implies $p \in L_{\downarrow}(U_{pf})$.

- Case $\phi(\Omega_{1:n}, p) = 0$: This necessarily means that condition $\dagger$ was verified with S reaching a value S_0 such that $S_0 \geq \Omega_{1:n}$. Moreover p did not contribute to S_0 (otherwise condition $\sharp$ would have been verified), that is

$$S_0 \leq \sum_{q \in L_{\downarrow}(U_{pf}) \setminus \{p\}} 2^{-q} = \Omega - 2^{-|p|} \quad \text{or equivalently} \quad \Omega \geq S_0 + 2^{-|p|}$$

Assume for the sake of contradiction that $\downarrow U(p)$. Since $p \in \mathbb{B}^{\leq n}$ we have that $|p| \leq n$, which implies $2^{-|p|} \geq 2^{-n}$. By substituting the inequalities:

$$\Omega \geq S_0 + 2^{-|p|} \geq \Omega_{1:n} + 2^{-n}$$

Now we have $\Omega - \Omega_{1:n} < 2^{-n}$. The hypothesis $\downarrow U_{pf}(p)$ is therefore false, that is $p \notin H_n$.

We indeed have termination and correctness of ϕ, which concludes the proof. ■

Theorem: Ω is ML-Random

Ω is ML-Random

Proof. Our goal is to present a procedure taking as input $\Omega_{1:n}$, for n an integer, and which returns an x such that $K(x) \geq n$. Let ϕ be the function defined as in the previous lemma. Let us define ϕ' such that

Input: $\Omega' \in \mathbb{B}^*$.
Assign $n \leftarrow \ell(\Omega')$
For each $z \in \mathbb{B}^{=n}$ do:

- If the condition $\mathcal{P}(z) \equiv \left[\forall p \in \mathbb{B}^{<n}, \left(\phi(\Omega', p) = 1 \implies U_{pf}(p) \neq z\right)\right]$ is verified do:
  * Write z to output and accept.

Let $n \in \mathcal{N}$.

We notice that for a fixed z, $\mathcal{P}(z)$ is a computable operation: for each $p \in \mathbb{B}^{\leq n}$, if $\phi(\Omega_{1:n}, p) = 1$, the lemma guarantees that the computation $U_{pf}(p)$ halts, therefore $U_{pf}(p)$ halts, which allows the comparison with z.

By an incompressibility result, the set $\Lambda_n := \{x \in \mathbb{B}^l \mid K(x) \geq n\}$ is non-empty. For all $x \in \Lambda_n$, there exists by definition no $p \in \mathbb{B}^{<n}$ such that $U_{pf}(p) = x$. Consequently, every $x \in \Lambda_n$ satisfies the condition $\mathcal{P}(x)$. Since $\Lambda_n \neq$ there will exist an iteration where $z \in \Lambda_n$, hence the termination of $\phi'(\Omega_{1:n})$.

Let z be the string returned by $\phi'(\Omega_{1:n})$. By construction, z satisfies $\mathcal{P}(z)$. Let us show that $K(z) \geq n$. Assume for the sake of contradiction that $K(z) < n$. There would then exist a string $p_0 \in \mathbb{B}^{<n}$ such that $U_{pf}(p_0) = z$. According to the lemma, we would have $\phi(\Omega_{1:n}, p_0) = 1$. The implication $(\phi(\Omega_{1:n}, p_0) = 1 \implies U_{pf}(p_0) \neq z)$ would then be false, which contradicts that z satisfies $\mathcal{P}(z)$. The hypothesis is therefore false, and we indeed have $K(z) \geq n$, that is $K(\phi'(\Omega_{1:n})) \geq n$.

By non-increase of information, there exists a constant $c \in \mathcal{N}$ (independent of n) such that for all $n \in \mathcal{N}$ we have $K(\Omega_{1:n}) + c \geq n$, or equivalently $K(\Omega_{1:n}) \geq n - c$. By characterization via Kolmogorov complexity, we have Ω is ML-Random. ■