On the Algorithmic Probability of Sets

Samuel Epstein
samepst@icloud.com

August 8, 2022

Abstract

The combined universal probability $\mathbf{m}(D)$ of strings x in sets D is close to max $\mathbf{m}(x)$ over x in D: their logs differ by at most D's information $\mathbf{I}(D:\mathcal{H})$ about the halting sequence $\mathcal{H}$. As a result of this, given a binary predicate P, the length of the smallest program that computes a complete extension of P is less than the size of the domain of P plus the amount of information that P has with the halting sequence.

1 Introduction

In this paper, we present two main results. The first result consists of statements about the algorithmic probability of set. The second result is about completions of binary predicates.

We show that sets D with large algorithmic probability, $\mathbf{m}(D)$, will have simple members. Sets which do not, have high mutual information with the halting sequence, $\mathbf{I}(D:\mathcal{H})$. This result has applications in algorithmic rate distortion theory. For a (partial computable) distortion function $d:\Sigma^*\times\Sigma^*\rightarrow\mathbb{R}_{\geq 0}$, if a (non-exotic) code word $\mathbf{y}$ has an approximation codeword $\mathbf{x}$, with $d(\mathbf{x},\mathbf{y})<R$, then there is another approximation $\mathbf{x}'$, with distortion $d(\mathbf{x}',\mathbf{y})<R$, of complexity $\mathbf{K}(\mathbf{x}')\lesssim\mathbf{I}(\mathbf{x};\mathbf{y})$.

The second result involves binary predicates. A binary predicate is a set of pairs $\{(x_i,b_i)\}$ where $x_i\in\mathbb{N}$ and $b_i\in\{0,1\}$. Binary predicates are used in learning theory to repesent samples of a target concept which a learning algorithm must approximate with a hypothesis. A complete extension to a binary predicate P is another binary predicate over all $\mathbb{N}$ that is consistent with P, where P is defined.

In this paper, we prove upper bounds on the size of the smallest program that computes a complete extension of a given binary predicate P. We prove that for non-exotic predicates, this size is not more than the number of elements of P. Exotic predicates have high mutual information with the halting sequence, and thus no algorithm can generate such predicates. To prove this, we first show new properties about the universal lower-semicomputable continuous semi-measure, $\mathbf{M}$. In particular, for a non-exotic prefix free set of strings G, the monotone complexity of G, $\mathbf{Km}(G)$, is less than the negative logarithm of $\mathbf{M}(G)$. See Section 3 for a formal definition of $\mathbf{Km}$ and $\mathbf{M}$.

2 Related Work

For information relating to the history of Algorithmic Information Theory and Kolmogorov complexity, we refer the readers to the textbooks [LV08] and [DH10]. A survey about the shared information between strings and the halting sequence is in the work [VV04]. Work on the deficiency of randomness can be found in [She83, KU87, V'Y87, She99]. Stochasticity of objects can be found in the works [She83, She99, V'Y87, V'Y99]. More information on stochasticity and algorithmic statistics are in the works [GTV01, VS17, VS15]. Section 6 and Lemma 2 are taken and adapted from [EL11], representing joint work with L. A. Levin. Lemma 3 also appears in [Eps13].

3 Conventions and Context

We use $\mathbb{Q}$, $\mathbb{N}$, $\mathbb{W}$, $\mathbb{R}$, Σ, Σ^*, and Σ^∞ to denote rationals, natural numbers, whole numbers, reals, bits, finite strings, and infinite strings. The notation $D_{>0}$ and $D_{\geq 0}$ is used to denote the positive and nonnegative members of D. If mathematical statement X is true, then $[X]=1$, otherwise $[X]=0$. Natural numbers and other elementary objects will be used reciprocally with finite strings. The empty string is denoted by $\emptyset$. For a string x, x^- is equal to x with the last bit removed. $\Sigma^{*\infty}=\Sigma^\infty\cup\Sigma^*$. For (finite or infinite) strings x, y, we say $x\sqsubseteq y$ iff $x=y$ or x is a prefix of y. We say $x\sqsubset y$ if $x\sqsubseteq y$ and $x\neq y$. The bit length of a string $x\in\Sigma^*$ is $\|x\|$. The ith bit of $\alpha\in\Sigma^{*\infty}$ is represented with α_i. The first n bits of $\alpha\in\Sigma^{*\infty}$ is represented by $\alpha_{\leq n}$.

We use $\langle x\rangle$, to represent a self delimiting code for $x\in\Sigma^*$, such as $1^{\|x\|}0x$. The self delimiting code for a finite set of strings $\{a_1,\dots,a_n\}$ is $\langle\{a_1,\dots,a_n\}\rangle=\langle n\rangle\langle a_1\rangle\langle a_2\rangle\dots\langle a_n\rangle$. For $X\subseteq\Sigma^*$, $X\Sigma^*=\{xy : x\in X, y\in\Sigma^*\}$. The number of elements of a set D is denoted to be $|D|$.

A measure over natural numbers is a nonnegative function $Q:\mathbb{N}\to\mathbb{R}_{\geq 0}$. The support of a measure Q is denoted by $\mathrm{supp}(Q)$, and it is equal to $Q^{-1}(\mathbb{R}_{>0})$. An elementary measure is a measure with finite support and a range of $\mathbb{Q}_{\geq 0}$. Elementary measures are elementary objects and can be encoded by finite strings. We say a measure Q is a semimeasure iff $\sum_a Q(a)\leq 1$. We say Q is a probabilty measure iff $\sum_a Q(a)=1$. For a set of natural numbers $D\subseteq\mathbb{N}$, its measure with respect to Q is equal to $Q(D)=\sum_{x\in D}Q(x)$. For semimeasure Q, the function $s:\mathbb{N}\to\mathbb{R}_{\geq 0}$ is a Q-test, if $\sum_{x\in N}2^{s(x)}Q(x)\leq 1$.

For positive real functions g, we denote $\leq g{+}O(1)$, $\geq g{-}O(1)$, $=g{\pm}O(1)$ with the notation $<^+ g$, $>^+ g$, $=^+ g$. In addition $\overset{*}{<}g$, $\overset{*}{>}g$, and $\overset{*}{=}$ denote $<g/O(1)$, $>g/O(1)$ and $=g*/O(1)$, respectively. Furthermore, we denote $\leq g{+}O(\log(g{+}1))$, $\geq g-O(\log(g{+}1))$, $=g{\pm}O(\log(g{+}1))$, by $<^{\log} g$, $>^{\log} g$, $\sim g$, respectively.

We say algorithm T is defined for input x and auxilliary input $\alpha\in\Sigma^{*\infty}$, if given x on the input tape, and α on the auxilliary tape for $\alpha\in\Sigma^\infty$, $\alpha\#^\infty$ on the auxilliary tape for $\alpha\in\Sigma^*$, T reads exactly $\|x\|$ bits on the input tape, and reads any amount of text on the auxilliary tape and outputs some finite string $T_\alpha(x)$ and halts. Otherwise $T_\alpha(x)$ is undefined. By this definition T is prefix-free, since for all auxillary inputs $\alpha\in\Sigma^{*\infty}$, there are no two strings $x\sqsubset y$, such that $T_\alpha(x)$ is defined and $T_\alpha(y)$ is defined. There is a universal prefix free algorithm U, such that for all algorithms T, there is a string $t\in\Sigma^*$, where for all $\alpha\in\Sigma^{*\infty}$ and $x\in\Sigma^*$, $U_\alpha(tx)=T_\alpha(x)$. We define Kolmogorov complexity with respect to this universal machine, where $\mathbf{K}(x|y)=\min\{\|p\| : U_y(p)=x\}$. When we say that the universal Turing machine is relativized to an elementary object a, this means that an encoding of the object, $\langle a\rangle$, is provided to the universal Turing machine on an auxilliary tape.

Let $\{f_i\}$ be an enumeration of partial computable functions $f_i:\mathbb{N}\to\mathbb{N}$. For a partial computable function g, let $D_g\subset\mathbb{N}$ be the indices of g in $\{f_i\}$. Then the complexity of g is defined to be $\mathbf{K}(g)=\min_{i\in D_g}\mathbf{K}(i)$. We say that a function $f:\mathbb{N}\to\mathbb{R}_{\geq 0}$ is lower computable if there exists an enumeration for the set $\{(a,b), a\in\Sigma^*, f(a)>b\in\mathbb{Q}\}$. Let $\{h_i\}$ be an enumeration of all enumerations that output elements of $\Sigma^*\times\mathbb{Q}$. For lower computable function h, let D_h be the indices of the enumerations of $\{(a,b), a\in\Sigma^*, h(a)>b\in\mathbb{Q}\}$ in the list $\{h_i\}$. Then the complexity of h is $\mathbf{K}(h)=\min_{i\in D_h}\mathbf{K}(i)$.

The halting sequence $\mathcal{H}\in\Sigma^\infty$ is the characteristic sequence of the domain of U, where $\mathcal{H}_i=[U(i)\text{ halts}]$. For $x,y\in\Sigma^*$, $\mathbf{I}(x:\mathcal{H}|y)=\mathbf{K}(x|y)-\mathbf{K}(x|\langle y,\mathcal{H}\rangle)$ is the amount of information that $\mathcal{H}$ has about string x, conditional to y. For strings x and y, the chain rule states that $\mathbf{K}(x)+\mathbf{K}(y|x,\mathbf{K}(x))=^+\mathbf{K}(x,y)$. The universal probability of a set D is $\mathbf{m}(D|y)=\sum_z[U_y(z)\in D]2^{-\|z\|}$. The universal probability of a string is $\mathbf{m}(x|y)=\mathbf{m}(\{x\}|y)$. By the coding theorem, we have that $-\log\mathbf{m}(x|y)=^+\mathbf{K}(x|y)$. The amount of information that y has about x is $\mathbf{I}(x;y)=\mathbf{K}(x)-\mathbf{K}(x|y)$.

4 Left-Total Machines

An string $x \in \Sigma^*$ is total with respect to algorithm T_α iff T_α will halt on all expansions of x that are long enough. Another way to define the concept is a string x is total with respect to T_α iff there exists a finite set of strings G, such that $\mu(G) = 1$ and T_α halts on each element in the set $\{xy : y \in G\}$. For sequences $x, y \in \Sigma^{*\infty}$, x is to the left of y, denoted by $x \lhd y$, if there is a string $z \in \Sigma^*$ such that $z0 \sqsubseteq x$ and $z1 \sqsubseteq y$. We say that a machine T is left total if for auxilliary inputs $\alpha \in \Sigma^{*\infty}$ and all $x, y \in \Sigma^*$, if $T_\alpha(y)$ halts, and $x \lhd y$, then x is total for T_α. An example of the domain of a left total machine can be seen in Figure 1. This example also illustrates the reason for using "left" in the definition.

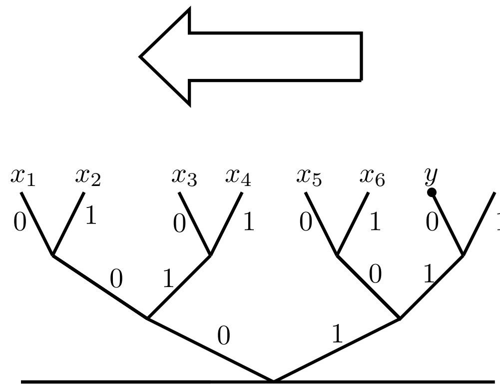

Figure 1: The above diagram represents the domain of a left total machine with the 0 bits branching to the left and the 1 bits branching to the right, with $y = 110$. For $i \in \{1, \ldots, 5\}$, $x_i \lhd x_{i+1}$ and $x_i \lhd y$. Assuming $T(y)$ halts, each x_i is total. This also implies each x_i^- is total as well.

We give a construction of a universal left-total algorithm U', in that its corresponding complexity functions are minimal. This construction has access to the universal algorithm U, which is not assumed to be left-total. The algorithm U' enumerates all strings $p \in \Sigma^*$ such that $U(p)$ is defined and orders them by the convergence time of $U(p)$, with the first string in the (infinite) list being the program with the shortest U-running time. For each ordered program p, U' successively assigns them consecutive open intervals $i_p \subset [0,1]$ of width $2^{-\|p\|}$. Then U' outputs $U(p)$ on input p' if the *open interval corrsponding to* p' and not that of $(p')^-$ is strictly contained in i_p. The open interval corresponding to p' is $\left([p']2^{-\|p'\|}, ([p']+1)2^{-\|p'\|}\right)$. The term $[p']$ presents the binary value of p'. For example, the binary value of both strings 011 and 0011 is 3 and the value of 0100 is 4, etc. The same definition applies for algorithms U'_α and U_α, over all auxilliary inputs $\alpha \in \Sigma^{*\infty}$. It is easy to verify that U' is left-total.

For the rest of the paper, we set U equal to U', and assume that the universal algorithm is left-total. Without loss of generality, the complexity terms, including $\mathbf{K}$, $\mathbf{Km}$, etc, are defined with respect to a left-total universal Turing machine.

Let $\mathcal{B} \in \Sigma^\infty$ be the border sequence, defined as the unique sequence where if $x \in \Sigma^*$ is a prefix of $\mathcal{B}$, $x \sqsubset \mathcal{B}$, then x has total and non-total expansions. If for $x \in \Sigma^*$, $x \lhd \mathcal{B}$, then x is total. If $\mathcal{B} \lhd x$, then U will diverge on all expansions of x. This is why $\mathcal{B}$ was given the terminology "border". The sequence $\mathcal{B}$ is computable relative to the halting sequence $\mathcal{H}$. Figure 2, visually shows the properties of $\mathcal{B}$.

For total string b, let $\mathbf{bb}(b) = \max\{\|y\| : U(p) = y : p \lhd b \text{ or } p \sqsupseteq b\}$ be the length of the

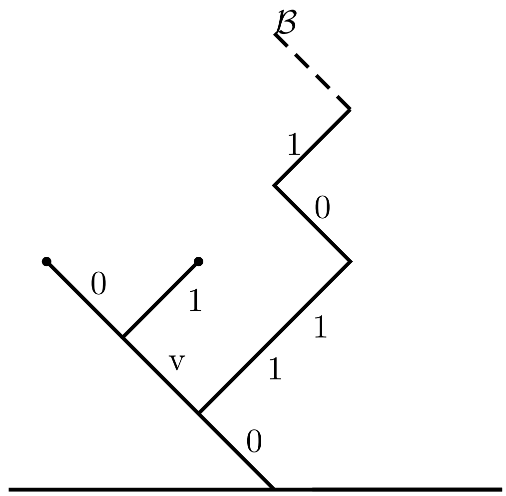

Figure 2: The above diagram represents the domain of the universal left-total algorithm U, with the 0 bits branching to the left and the 1 bits branching to the right. The strings in the above diagram, $0v0$ and $0v1$, are halting inputs to U. This implies that $0v$ is a total string. The infinite border sequence $\mathcal{B} \in \Sigma^\infty$ represents the unique infinite sequence such that all its finite prefixes have total and non-total extensions. All finite strings branching to the right of $\mathcal{B}$ will cause U to diverge.

longest output of a string from a program to the left of b or that extends b. $\mathbf{bb}(b)$ is 0 if b is not total. If b is total, then $\mathbf{bb}(b)$ is computable. Furthermore, for total string b, strings $x, y \in \Sigma^*$, let $\mathbf{m}_b(x|y) = \sum\{2^{-\|p\|} : U_y(p) = x,\ p \lhd b \text{ or } p \sqsupseteq b\}$ be the algorithmic weight of x, conditional to y, using programs that extend b or are to the "left" of b. Let $\mathbf{m}_b(x|y) = 0$ for non-total b. Thus for total b, $\mathbf{m}_b$ is computable. We note that if b and b^- are total, then $\mathbf{bb}(b^-) \geq \mathbf{bb}(b)$ and $\mathbf{m}_{b^-} \geq \mathbf{m}_b$.

Proposition 1 *The border sequence $\mathcal{B}$ is Martin Löf random, where for $b \sqsubset \mathcal{B}$, $n = \|b\| <^+ \mathbf{K}(b)$. Furthermore if $b \in \Sigma^*$ is total and b^- is not, then $b^- \sqsubset \mathcal{B}$.*

Proof. Let $\Omega = \sum_x \mathbf{m}(x)$ be Chaitin's Omega, the probability that U will halt. Given $b \sqsubset \mathcal{B}$, $b \in \Sigma^n$ of length n, one can compute $\hat{\Omega} = \sum\{2^{-\|y\|} :\ U(y) \text{ is defined and } y \lhd b\}$ which differs from Ω in the summation of programs which branch from $\mathcal{B}$ at positions n or higher. Thus $\Omega - \hat{\Omega} \leq 2^{-n}$. So the binary expansion of Ω and $\hat{\Omega}$ share the first $>^+ n$ bits. So from b, one can compute the first $>^+ n$ bits of the binary expansion of Ω. It is well known that the binary expansion of Chaitin's Ω is Martin Löf random, so this implies $\mathbf{K}(b) >^+ n$.

If $b \in \Sigma^*$ is total and b^- is not, then b^- has a total extension b^-0 and a non total extension b^-1, thus by the definition of the border sequence, $b^- \sqsubset \mathcal{B}$. □

5 Stochasticity

We use notions from algorithmic statistics, most notably the deficiency of randomness of a string $a \in \Sigma^*$ with respect to (necessarily elementary) probability measure W and string $y \in \Sigma^*$, denoted by $\mathbf{d}(a|W, y) = \lfloor -\log W(a) \rfloor - \mathbf{K}(a|y)$. By definition, the function $\mathbf{d}$ is a W-test. In addition, for any elementary probability measure W, for any lower computable W-test d, and for any string $y \in \Sigma^*$, over all $a \in \Sigma^*$, we have that $d(a) <^+ \mathbf{d}(a|W, y) + \mathbf{K}(d|y) + \mathbf{K}(W|y)$. For more information about $\mathbf{d}$, we refer the readers to [GÍ13]. The stochasticity of string $a \in \Sigma^*$, conditional to $y \in \Sigma^*$ is denoted

$$\Lambda(a|y) = \min\{j + 3\log k : \exists v \in \{0,1\}^j, U_y(v) = \langle W\rangle, \mathbf{d}(a|W, \langle v, y\rangle) \leq k \in \mathbb{N}\}.$$

In fact, the theorems in this paper hold if the $(3\log k)$ term above is replaced with k. The conditional term in the stochasticity measure can be removed at a cost of a logarithm of the complexity of that term, as shown in the following proposition.

Proposition 2 *For strings $x, y \in \Sigma^*$, $\Lambda(x|y) <^+ \Lambda(x) + 3\log \mathbf{K}(y)$.*

Proof. Let W, $v \in \Sigma^*$ be the elementary probability measure and program that minimizes $\Lambda(x)$, with $U(v) = \langle W\rangle$, $\|v\| + 3\log\max\{\mathbf{d}(x|W, v), 1\} = \Lambda(x)$. Since $\mathbf{K}(x|v) <^+ \mathbf{K}(x|v, y) + \mathbf{K}(y)$, we have that $\mathbf{d}(x|W, \langle v, y\rangle) <^+ \mathbf{d}(x|W, v) + \mathbf{K}(y)$. So

$$\begin{aligned}\Lambda(x|y) &\leq \|v\| + 3\log\max\{\mathbf{d}(x|W, \langle v, y\rangle), 1\}\\ &<^+ \|v\| + 3\log\max\{\mathbf{d}(x|W, v), 1) + 3\log \mathbf{K}(y\}\\ &<^+ \Lambda(x) + 3\log \mathbf{K}(y).\end{aligned}$$

□

A total computable function cannot increase the stochasticity of a sequence by more than constant factor of its complexity. This notion is captured in Proposition 5 of [VS17]. Another expression of this idea can be found in the following lemma.

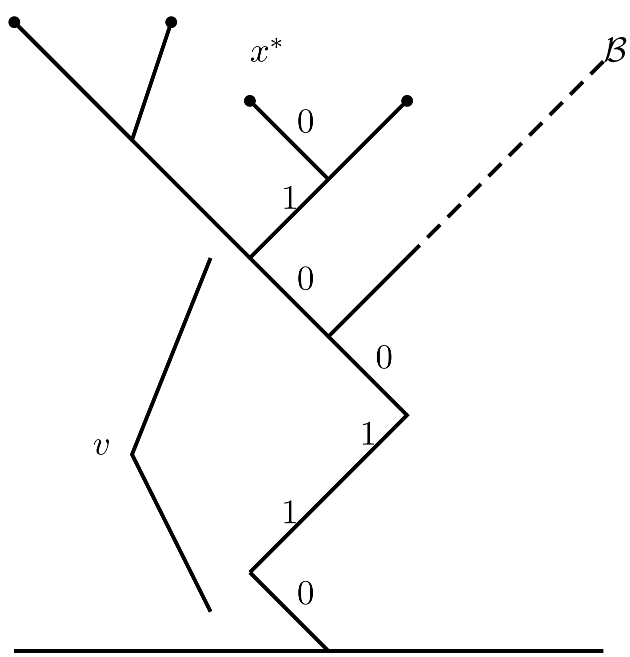

Figure 3: A graphical depicton of the terms used in Lemma 2. The shortest program for $x \in \Sigma^*$ is $x^* = 0110010$, with $U(x^*) = x$. The shortest total prefix of x^* is $v = 01100$, with $v^- = 0110$ being a prefix of border $\mathcal{B}$. Assuming x^* is the only extension of v that is an program for x, then $Q(x) = 2^{-\|x\|+\|v\|} = 2^{-2}$.

Lemma 1 *Given total recursive function $g : \Sigma^* \to \Sigma^*$, $\Lambda(g(a)) < \Lambda(a) + \mathbf{K}(g) + O(\log \mathbf{K}(g))$.*

Proof. Let $v \in \Sigma^*$ and W be the program and elementary probability measure that realize $\Lambda(a)$, where $U(v) = \langle W \rangle$ and $\|v\| + 3 \log \max\{\mathbf{d}(a|W, v), 1\} = \Lambda(a)$. The lemma is proven using the image probability measure of W with respect to g, denoted by W_g, where $W_g(x) = \sum\{W(y) : g(y) = x\}$. Since W is elementary, W_g is elementary. The function $\mathbf{d}(g(\cdot)|W_g, v)$ is a W-test, because

$$\begin{aligned}\sum_x 2^{\mathbf{d}(g(x)|W_g,v)} W(x) &= \sum_y 2^{\mathbf{d}(y|W_g,v)} W_g(y) \\ &\leq \sum_y 2^{-\mathbf{K}(y|v)} \\ &\leq 1.\end{aligned}$$

The function $\mathbf{d}(g(\cdot)|W_g, v)$ is lower computable and it has complexity (conditioned on v)

$$\mathbf{K}(\mathbf{d}(g(\cdot)|W_g, v)|v) <^+ \mathbf{K}(g|v).$$

Since $\mathbf{d}$ is a universal lower computable W-test, we have the inequality

$$\begin{aligned}&\mathbf{d}(g(a)|W_g, v) <^+ \mathbf{d}(a|W, v) + \mathbf{K}(\mathbf{d}(g(\cdot)|W_g, v)|v) + \mathbf{K}(W|v), \\ &\mathbf{d}(g(a)|W_g, v) <^+ \mathbf{d}(a|W, v) + \mathbf{K}(g). \qquad (1)\end{aligned}$$

Let z be a program for W_g that contains v and a shortest program for g, with $\|z\| <^+ \|v\| + \mathbf{K}(g)$ and $\mathbf{K}(z|v) <^+ \mathbf{K}(g)$. Because $\mathbf{K}(a|v) <^+ \mathbf{K}(z|v) + \mathbf{K}(a|z)$, we have $-\mathbf{K}(a|z) <^+ -\mathbf{K}(a|v) + \mathbf{K}(g)$. This gives us

$$\begin{aligned}\mathbf{d}(g(a)|W_g, z) &= \lfloor -\log W_g(g(a)) \rfloor - \mathbf{K}(g(a)|z) \\ &<^+ \lfloor -\log W_g(g(a)) \rfloor - \mathbf{K}(g(a)|v) + \mathbf{K}(g) \\ &=^+ \mathbf{d}(g(a)|W_g, v) + \mathbf{K}(g).\end{aligned}$$

Combined with inequality 1, we get

$$\mathbf{d}(g(a)|W_g, z) < \mathbf{d}(a|W, v) + O(\mathbf{K}(g)).$$

So we have that

$$\begin{aligned}\Lambda(g(a)) &\leq \|z\| + 3 \log \max\{\mathbf{d}(g(a)|W_g, z), 1\} \\ &<^+ \|v\| + 3 \log \max\{\mathbf{d}(g(a)|W_g, z), 1\} + \mathbf{K}(g) \\ &< \|v\| + 3 \log \max\{\mathbf{d}(a|W, v), 1\} + \mathbf{K}(g) + O(\log \mathbf{K}(g)) \\ &< \Lambda(a) + \mathbf{K}(g) + O(\log \mathbf{K}(g)).\end{aligned}$$

□

Strings with high stochasticity measure are *exotic*, in that they have high mutual information with the halting sequence, as shown in the following lemma.

Lemma 2 *For $x \in \Sigma^*$, $\Lambda(x) <^{\log} \mathbf{I}(x:\mathcal{H})$.*

Proof. Let $x^* \in \Sigma^*$ be any string such that $U(x^*) = x$ and $\|x^*\| = \mathbf{K}(x)$. Let $v \in \Sigma^*$ be the shortest total prefix of x^*. Thus by definition v^- is not total. We define the elementary probability measure Q such that $Q(y) = \sum_w 2^{-\|w\|}[U(vw)=y]$. Thus Q is computable relative to v. In addition, since $v \sqsubseteq x^*$, one has the lower bound $Q(x) \geq 2^{-\|x^*\|+\|v\|} = 2^{-\mathbf{K}(x)+\|v\|}$. So

$$-\log Q(x) \leq \mathbf{K}(x) - \|v\|. \tag{2}$$

This is not always an equality, because there could exist $w \in \Sigma^*$, $vw \neq x^*$ such that $U(vw) = x$. A graphical description of these terms can be found in Figure 3. So

$$\mathbf{d}(x|Q,v) = \lfloor -\log Q(x) \rfloor - \mathbf{K}(x|v).$$

Using Equation 2, and $\mathbf{K}(x) <^+ \mathbf{K}(v) + \mathbf{K}(x|v)$ we have

$$\begin{aligned}\mathbf{d}(x|Q,v) &<^+ \mathbf{K}(x) - \|v\| - \mathbf{K}(x|v)\\ &<^+ (\mathbf{K}(v) + \mathbf{K}(x|v)) - \|v\| - \mathbf{K}(x|v).\end{aligned}$$

Since $\mathbf{K}(v) <^+ \|v\| + \mathbf{K}(\|v\|)$,

$$\begin{aligned}\mathbf{d}(x|Q,v) &<^+ (\|v\| + \mathbf{K}(\|v\|) + \mathbf{K}(x|v)) - \|v\| - \mathbf{K}(x|v),\\ \mathbf{d}(x|Q,v) &<^+ \mathbf{K}(\|v\|).\end{aligned} \tag{3}$$

We have

$$\mathbf{K}(x|\mathcal{H}) <^+ \mathbf{K}(x|\langle Q\rangle) + \mathbf{K}(\langle Q\rangle|\mathcal{H}).$$

Since $\langle Q\rangle$ is computable from v, $\mathbf{K}(\langle Q\rangle|\mathcal{H}) <^+ \mathbf{K}(v|\mathcal{H})$, so

$$\mathbf{K}(x|\mathcal{H}) <^+ \mathbf{K}(x|\langle Q\rangle) + \mathbf{K}(v|\mathcal{H}).$$

Since x is in the support of Q, $\mathbf{K}(x|\langle Q\rangle) <^+ -\log Q(x)$, by using standard Shannon-Fano encoding. So

$$\mathbf{K}(x|\mathcal{H}) <^+ -\log Q(x) + \mathbf{K}(v|\mathcal{H})$$

Since v is total and v^- is not total, by Proposition 1, v^- is a prefix of the border sequence $\mathcal{B}$. Since $\mathcal{B}$ is computable from the halting sequence $\mathcal{H}$, v^- is simple relative to $\mathcal{H}$ and $\|v\|$, with $\mathbf{K}(v|\mathcal{H}) <^+ \mathbf{K}(\|v\|)$, so

$$\mathbf{K}(x|\mathcal{H}) <^+ -\log Q(x) + \mathbf{K}(\|v\|).$$

Using Equation 2,

$$\begin{aligned}\mathbf{K}(x|\mathcal{H}) &<^+ \mathbf{K}(x) - \|v\| + \mathbf{K}(\|v\|)\\ \|v\| &<^+ \mathbf{K}(x) - \mathbf{K}(x|\mathcal{H}) + \mathbf{K}(\|v\|)\\ \|v\| &<^{\log} \mathbf{I}(x:\mathcal{H}).\end{aligned} \tag{4}$$

Since Q is computable from v (and helper code of size $O(1)$), one gets

$$\Lambda(x) <^+ \mathbf{K}(v) + 3\log(\max\{\mathbf{d}(x|Q,v), 1\}).$$

Since $\mathbf{K}(v) <^+ \|v\| + \mathbf{K}(\|v\|)$,

$$\Lambda(x) <^+ \|v\| + \mathbf{K}(\|v\|) + 3\log(\max\{\mathbf{d}(x|Q,v),1\}).$$

Due to Equation 3, we have

$$\begin{aligned}\Lambda(x) &<^+ \|v\| + \mathbf{K}(\|v\|) + 3\log\mathbf{K}(\|v\|)\\ &<^{\log} \|v\|.\end{aligned}$$

Due to Equation 4, we have the final form of the lemma,

$$\Lambda(x) <^{\log} \mathbf{I}(x:\mathcal{H}).$$

□

6 Algorithmic Probability of Sets

The following theorem states that sets with very large m-measure, for some computable semi-measure m, containing no simple member (conditional to m), will be exotic, that is have high stochasticity.

Theorem 1 *For finite set $D \subset \Sigma^*$, relativized to elementary semi-measure m,* $\min_{x\in D}\mathbf{K}(x) <^+ -\log m(D) + 2\mathbf{K}(\lceil -\log m(D)\rceil) + \Lambda(D)$.

Proof. We note that by the term "relativized", we mean that the universal Turing machine has access to a finite encoding of m on an auxilliary tape. Let $i = \lceil -\log m(D)\rceil$ be the log length of the m-weight of D. Let Q' and $v' \in \Sigma^*$ be the elementary probability measure and program that minimize the i-conditional stochasticity of $\langle D\rangle$, where $U_i(v') = \langle Q'\rangle$, and $\|v'\| + 3\log\max\{\mathbf{d}(\langle D\rangle|Q', \langle v', i\rangle), 1\} = \Lambda(\langle D\rangle|i)$.

We focus on encoded finite sets in the support of Q' that have large m-measure. Let $S = \{\langle F\rangle : F \subset \Sigma^*, \langle F\rangle \in \mathrm{supp}(Q), i \geq -\log m(F)\}$ be the set of encoded sets in the support of Q that are i-heavy. Let Q be the probability of Q' conditioned on S, with $Q(\langle F\rangle) = [\langle F\rangle \in S]Q'(\langle F\rangle)/Q'(S)$.

Using Q instead of Q' will not affect the bounds needed in this theorem, as shown in the following reasoning. Let $v = v_0 v' \in \Sigma^*$ be a program that uses i on the auxilliary tape, helper code v_0, and v' to output an encoding of the (elementary) probability measure Q, with $U(v) = \langle Q\rangle$. Let $d = \max\{\mathbf{d}(\langle D\rangle|Q, \langle v, i\rangle), 1\}$. Thus the following inequality justifies working with Q and v instead of Q' and v'.

$$\begin{aligned}\|v\| &<^+ \|v'\|\\ \|v\| + 3\log d &<^+ \|v'\| + 3\log d\\ &<^+ \|v'\| + 3\log\max\{-\log Q(\langle D\rangle) - \mathbf{K}(\langle D\rangle|\langle v,i\rangle), 1\}\\ &<^+ \|v'\| + 3\log\max\{-\log Q'(\langle D\rangle) - \mathbf{K}(\langle D\rangle|\langle v,i\rangle), 1\}\\ &<^+ \|v'\| + 3\log\max\{-\log Q'(\langle D\rangle) - \mathbf{K}(\langle D\rangle|\langle v',i\rangle) + \mathbf{K}(v|\langle v',i\rangle), 1\}\\ &<^+ \|v'\| + 3\log\max\{-\log Q'(\langle D\rangle) - \mathbf{K}(\langle D\rangle|\langle v',i\rangle), 1\}\\ \|v\| + 3\log d &<^+ \Lambda(\langle D\rangle|i). \end{aligned} \tag{5}$$

Using the probabilistic method, we can guarantee a large enough vector of strings has a non empty intersection with a large Q-percentage of sets. Assign m measure $1 - m(\Sigma^*)$ to the empty string, so that m can be assumed to be an elementary probability measure. Let $c \in \mathbb{N}$ be constant solely determined by U to be specified later. Let δ be a random vector of finite strings of size $cd2^{i+1}$ where each element of δ is chosen independently according to m.

Let t_z, be a non-negative function over sets of strings, parameterized by a vector z over strings. $t_z(F) = 0$ if $F \cap z \neq \emptyset$ and $t_z(F) = e^{cd}$ otherwise. Using the fact that $(1-a) \leq e^{-a}$ for $a \in [0,1]$, we have that

$$\begin{aligned}
\mathbf{E}_{z\sim\delta}\mathbf{E}_{\langle F\rangle\sim Q}[t_z(F)] &= \sum_{F\subset\Sigma^*} Q(\langle F\rangle)\,(1-m(F))^{cd2^{i+1}}\,e^{cd} \\
&\leq \sum_{F\subset\Sigma^*} Q(\langle F\rangle)(1-2^{-i})^{cd2^{i+1}}e^{cd} \\
&\leq \sum_{F\subset\Sigma^*} Q(\langle F\rangle)e^{-2^{-i}cd2^{i+1}}e^{cd} \\
&\leq e^{-2cd}e^{cd} < 1.
\end{aligned}$$

Thus by the probabilistic method, there exists a vector z of size $cd2^{i+1}$ such that $\mathbf{E}_{\langle F\rangle\sim Q}[t_z(F)] \leq 1$. This vector is simple relative to a program for $\langle Q\rangle$ and the relevant parameters. More specifically, z can be found given brute search given v (and thus Q), i, c and d, with

$$\mathbf{K}(t_z|v,i) <^+ \mathbf{K}(c,d). \tag{6}$$

The sets which have an empty intersection with z are atypical and have a high Q-deficiency of randomness, higher than d, due to sizing of the vector. A graphical depiction of the atypical sets can be seen in Figure 4.

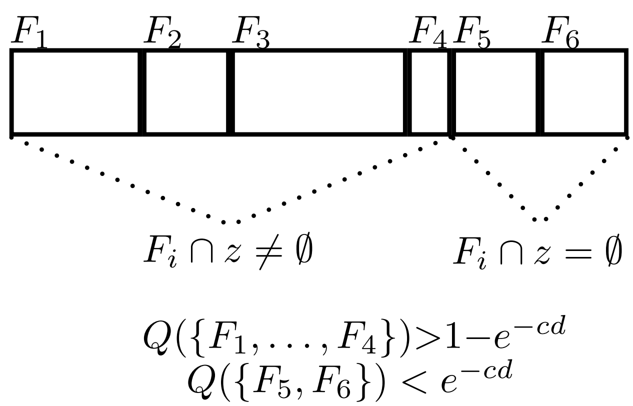

Figure 4: A graphical depiction of the properties of the vector z. A majority of encoded sets in the support of Q have a non-empty intersection with z. The two set F_5 and F_6 which are separate from z have very low combined Q measure of $< e^{-cd}$, and are atypical.

It must be that $z \cap D \neq \emptyset$ and thus $t_z(D) = 0$. Otherwise $t_z(D) = e^{cd}$, and we have

$$d > -\log Q(\langle D\rangle) - \mathbf{K}(D|v,i) - O(1).$$

Using the fact that $t_z(\cdot)Q(\cdot)$ is a semi-measure, and thus objects in its support can be identified by Shannon-Fano code of size $=^+ -\log t_z(\cdot)Q(\cdot)$, we have that $\mathbf{K}(D|v,i) <^+ -\log t_z(D)Q(\langle D\rangle) +$

$\mathbf{K}(t_z, Q|v, i)$.

$$\begin{aligned} d &> -\log Q(\langle D\rangle) - (-\log t_z(D)Q(\langle D\rangle) + \mathbf{K}(t_z, Q|v, i)) - O(1) \\ &> cd(\log e) - \mathbf{K}(t_z|v, i) - \mathbf{K}(Q|v, i) - O(1). \end{aligned}$$

Combined with Equation 6, we get the contradiction

$$\begin{aligned} d &> cd(\log e) - \mathbf{K}(c, d) - O(1) \\ &> d. \end{aligned} \tag{7}$$

Equation 7 is due to choice of large enough c, which is entirely dependent on the choice of the universal algorithm U used. Since c is an additive constant, we remove it from consideration for the rest of the proof. The contradiction implies $t_z(D) = 0$ which means that $z \cap D \neq \emptyset$. Thus for some $x \in z \cap D$,

$$\mathbf{K}(x|i) <^+ \log |z| + \mathbf{K}(z|i).$$

Using the fact that $\mathbf{K}(z|i) <^+ \mathbf{K}(d, v|i)$, and also that v is a program that uses i on the auxilliary tape, and thus $\mathbf{K}(v|i) <^+ \|v\|$, we have

$$\begin{aligned} \mathbf{K}(x|i) &<^+ i + \log d + \mathbf{K}(d, v|i) \\ &<^+ i + 3\log d + \|v\|. \end{aligned}$$

Using inequality 5, and Proposition 2, we get

$$\begin{aligned} \mathbf{K}(x|i) &<^+ i + \Lambda(\langle D\rangle|i) \\ \mathbf{K}(x) &<^+ i + \mathbf{K}(i) + \Lambda(\langle D\rangle) + 3\log \mathbf{K}(i) \\ &<^+ i + 2\mathbf{K}(i) + \Lambda(\langle D\rangle). \end{aligned}$$

□

Theorem 1, is a reworking of Lemma 6 in [VV04], shown below. This lemma is in terms of plain Kolmogorov complexity $\mathbf{C}$, which we refer readers to [LV08]. This differs from Theorem 1 which deals with a computable semi-measure m and the m-weight of sets D. Whereas Lemma 6 of [VV04] deals with combinatorial arguments over sets of sets, Theorem 1 focuses on the stochasticity, using elementary probability measures of sets.

Lemma 6: [VV04] *Let n, m, k be natural numbers and x a string of length n. Let $\mathcal{B}$ be a family of subsets of Σ^n and $\mathcal{B}(x) = \{B \in \mathcal{B} : x \in B\}$. If $\mathcal{B}(x)$ has at least 2^m elements (that is, sets) of Kolmogorov complexity less than k, then there is an element in $\mathcal{B}(x)$ of Kolmogorov complexity at most $k - m + O(\mathbf{C}(\mathcal{B}) + \log n + \log k + \log m)$.*

Theorem 1 is used to prove the main Theorem 2. Theorem 1 requires a computable m whereas Theorem 2 is in terms of the lower computable $\mathbf{m}$. For reasoning described in the proof of Theorem 2, the way to bridge the gap is to invoke Theorem 1 conditioned on a prefix b of the border sequence and let $m = \mathbf{m}_b$. The following lemma is used in Theorem 2 to remove the conditioning on b. It states that if a prefix of border b is simple relative to a string x (and its own length $\|b\|$), then it will be common information shared between x and the halting sequence $\mathcal{H}$.

Lemma 3
If string b is total and b^- is not, then for all strings x, $\mathbf{K}(b)+\mathbf{I}(x:\mathcal{H}|b) <^{\log} \mathbf{I}(x:\mathcal{H})+\mathbf{K}(b|x,\|b\|)$.

Proof. The chain rule gives the additive equality

$$\mathbf{K}(x,b) =^+ \mathbf{K}(b) + \mathbf{K}(x|b,\mathbf{K}(b)) =^+ \mathbf{K}(x) + \mathbf{K}(b|x,\mathbf{K}(x)). \tag{8}$$

We have the inequality $\mathbf{K}(x|b) <^+ \mathbf{K}(x|b,\mathbf{K}(b))+\mathbf{K}(\mathbf{K}(b))$ from the fact that there is an x program conditional to b that uses any x program conditional to $\langle b,\mathbf{K}(b)\rangle$ and an encoding of $\mathbf{K}(b)$. We also have the inequaltiy $\mathbf{K}(b|x,\mathbf{K}(x)) <^+ \mathbf{K}(b|x)$, since the conditional term on the left hand side contains more information. These inequalities combined with Equation 8, results in the inequality,

$$\mathbf{K}(b) + \mathbf{K}(x|b) <^+ \mathbf{K}(x) + \mathbf{K}(b|x) + \mathbf{K}(\mathbf{K}(b)). \tag{9}$$

Subtracting $\mathbf{K}(x|\langle b,\mathcal{H}\rangle)$ from both sides of Equation 9 results in

$$\mathbf{K}(b) + \mathbf{K}(x|b) - \mathbf{K}(x|\langle b,\mathcal{H}\rangle) <^+ \mathbf{K}(x) + \mathbf{K}(b|x) + \mathbf{K}(\mathbf{K}(b)) - \mathbf{K}(x|\langle b,\mathcal{H}\rangle). \tag{10}$$

The inequality $\mathbf{K}(x|\mathcal{H}) <^+ \mathbf{K}(x|\langle b,\mathcal{H}\rangle) + \mathbf{K}(b|\mathcal{H})$, rewritten as $-\mathbf{K}(x|\langle b,\mathcal{H}\rangle) <^+ -\mathbf{K}(x|\mathcal{H}) + \mathbf{K}(b|\mathcal{H})$, and combined with Equation 10 gives us,

$$\begin{aligned}\mathbf{K}(b) + \mathbf{K}(x|b) - \mathbf{K}(x|\langle b,\mathcal{H}\rangle) &<^+ \mathbf{K}(x) + \mathbf{K}(b|x) + \mathbf{K}(\mathbf{K}(b)) - \mathbf{K}(x|\mathcal{H}) + \mathbf{K}(b|\mathcal{H})\\ \mathbf{K}(b) + \mathbf{I}(x:\mathcal{H}|b) &<^+ \mathbf{K}(b|x) + \mathbf{K}(\mathbf{K}(b)) + \mathbf{K}(b|\mathcal{H}) + \mathbf{I}(x:\mathcal{H}).\end{aligned} \tag{11}$$

By Proposition 1, $b^- \sqsubset \mathcal{B}$ and thus $\|b\| <^+ \mathbf{K}(b)$. Since $\mathcal{B}$ is computable relative to the halting sequence $\mathcal{H}$, and $b^- \sqsubset \mathcal{B}$, one can compute b given $\mathcal{H}$ and b's length, $\|b\|$. This gives us the inequality, which, using Proposition 1, is of the form

$$\mathbf{K}(b|\mathcal{H}) <^+ \mathbf{K}(\|b\|) = O(\log\|b\|) = O(\mathbf{K}(b)). \tag{12}$$

Combining Equations 11 and 12, we get

$$\mathbf{K}(b) + \mathbf{I}(x:\mathcal{H}|b) <^+ \mathbf{I}(x:\mathcal{H}) + \mathbf{K}(b|x) + O(\log\mathbf{K}(b)). \tag{13}$$

Applying again Proposition 1,

$$\begin{aligned}\mathbf{K}(b|x) &<^+ \mathbf{K}(b|\langle x,\|b\|\rangle) + \mathbf{K}(\|b\|)\\ &<^+ \mathbf{K}(b|\langle x,\|b\|\rangle) + O(\log\|b\|)\\ &<^+ \mathbf{K}(b|\langle x,\|b\|\rangle) + O(\log\mathbf{K}(b)),\end{aligned}$$

and combined with Equation 13, we have

$$\begin{aligned}\mathbf{K}(b) + \mathbf{I}(x:\mathcal{H}|b) &<^+ \mathbf{I}(x:\mathcal{H}) + \mathbf{K}(b|\langle x,\|b\|\rangle) + O(\log\mathbf{K}(b))\\ &<^{\log} \mathbf{I}(x:\mathcal{H}) + \mathbf{K}(b|\langle x,\|b\|\rangle).\end{aligned}$$

This form proves the lemma. □

Theorem 2 states that sets with large $\mathbf{m}$ measure will contain a simple element. Otherwise the set is exotic, containing high information with the halting sequence. An example is the set S_n, the set of all random strings x of length n, where $n <^+ \mathbf{K}(x) <^+ n + \mathbf{K}(n)$. There are at least $2^{n-O(1)}$ such strings, so

$$-\log \mathbf{m}(S_n) < -\log |S_n| 2^{-n-\mathbf{K}(n)-O(1)} <^+ -\log \left(2^n 2^{-n-\mathbf{K}(n)}\right) <^+ \mathbf{K}(n).$$

However $\min_{x \in S_n} \mathbf{K}(x) >^+ n$. By Theorem 2, this implies that S_n is exotic, with $n <^{\log} \mathbf{I}(\langle S_n \rangle : \mathcal{H})$. This can be verified by noting the following. Let x be the first element of the encoding $\langle S_n \rangle$. Then $\mathbf{K}(\langle S_n \rangle) >^+ \mathbf{K}(x) >^+ n$. In addition, given the halting sequence $\mathcal{H}$ and n, one can compute all random strings of length n, so $\mathbf{K}(\langle S_n \rangle | \mathcal{H}) <^+ \mathbf{K}(n)$. This implies $\mathbf{I}(\langle S_n \rangle : \mathcal{H}) = \mathbf{K}(\langle S_n \rangle) - \mathbf{K}(\langle S_n \rangle | \mathcal{H}) >^{\log} n$.

Theorem 2
For finite set $D \subset \Sigma^*$, $\min_{x \in D} \mathbf{K}(x) < -\log \mathbf{m}(D) + \mathbf{I}(D : \mathcal{H}) + O(\mathbf{K}(\lceil -\log \mathbf{m}(D) \rceil) + \log \mathbf{I}(D : \mathcal{H}))$.

Proof. Let $i = 1 + \lceil -\log \mathbf{m}(D) \rceil$. Let $b \in \Sigma^*$ be the shortest total string such that $i \geq -\log \mathbf{m}_b(D)$. The string b is unique. Otherwise there exists a total string $b' \neq b$, $\|b'\| = \|b\|$, and $-\log \mathbf{m}_{b'}(D) \leq i$. If $b' \lhd b$, this would imply b'^- is total, and since $\mathbf{m}_{b'^-} \geq \mathbf{m}_{b'}$, b'^- would be a total string shorter than b such that $i \geq -\log \mathbf{m}_{b'^-}(D)$. Similar arguments show that it can't be that $b \lhd b'$. Thus a contradiction occurs over the existence of b'. A graphical depiction of this argument can be seen in Figure 5.

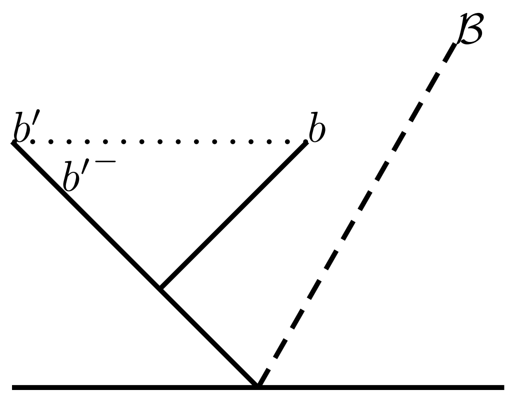

Figure 5: A graphical argument for why the total string b in the proof of Theorem 2 is unique. Each path repsents a string, with 0s branching to the left and 1s branching to the right. If another string b' exists with the desired $\mathbf{m}_{b'}$ property, and it is to the left of b, then its prefix b'^- will also be total and have the desired $\mathbf{m}_{b'^-}$ property, causing a contradiction.

This b can be constructed in the following manner given D, i, $\|b\|$, leveraging the property that b is unique. The algorithm enumerates all total strings of length $\|b\|$. Once it finds a total $y \in \Sigma^{\|b\|}$ such that $\mathbf{m}_y(D) \geq 2^{-i}$, it halts and outputs that y, which is equal to b. Thus $\mathbf{K}(b | \langle D, \|b\| \rangle) <^+ \mathbf{K}(i)$.

We invoke Theorem 1, relativized to b. We recall that this means the universal Turing machine has access to b on an auxilliary tape. We set $m(\cdot) = \mathbf{m}_b(\cdot)$, which is computable because we relativized to b. This proves the existence of an $x \in D$, with

$$\mathbf{K}(x|b) < -\log \mathbf{m}_b(D) + \Lambda(D|b) + O(\mathbf{K}(\lceil -\log \mathbf{m}_b(D)\rceil)).$$

Since $\lceil -\log \mathbf{m}_b(D))\rceil =^+ \log \mathbf{m}(D) =^+ i$,

$$\begin{aligned}\mathbf{K}(x|b) &< i + \Lambda(D|b) + O(\mathbf{K}(i))\\ \mathbf{K}(x) &< i + \mathbf{K}(b) + \Lambda(D|b) + O(\mathbf{K}(i)).\end{aligned}$$

Lemma 2, relativized to b, produces

$$\begin{aligned}\mathbf{K}(x) &< i + \mathbf{K}(b) + \mathbf{I}(D:\mathcal{H}|b) + O(\mathbf{K}(i) + \log \mathbf{I}(D:\mathcal{H}|b)) \\ &< i + \mathbf{K}(b) + \mathbf{I}(D:\mathcal{H}|b) + O(\mathbf{K}(i) + \log(\mathbf{I}(D:\mathcal{H}|b) + \mathbf{K}(b))).\end{aligned} \tag{14}$$

Most terms containing b can be removed using Lemma 3, with

$$\mathbf{K}(x) < i + \mathbf{I}(D:\mathcal{H}) + \mathbf{K}(b|D, \|b\|) + O(\mathbf{K}(i) + \log \mathbf{I}(D:\mathcal{H}) + \log \mathbf{K}(b|D, \|b\|)).$$

Recalling that b is simple relative to its length $\|b\|$, D, and i, we prove the theorem

$$\mathbf{K}(x) < i + \mathbf{I}(D:\mathcal{H}) + O(\mathbf{K}(i) + \log \mathbf{I}(D:\mathcal{H})).$$

□

Theorem 3 shows another characterization of exotic sets. Sets D whose members are complex and have members containing low information about the set, (i.e. low $\min_{x\in D}\mathbf{I}(x;D)$) are exotic. We continue with our example S_n introduced above. The set S_n has at least $2^{n-O(1)}$ elements, so there exists an element $y \in S_n$, where $\mathbf{K}(y|\langle S_n\rangle) >^+ n$. Thus since y is a string of length n, $\mathbf{K}(y) <^+ n + \mathbf{K}(n)$.

$$\min_{x\in S_n} \mathbf{I}(x;S_n) \le \mathbf{I}(y;S_n) = \mathbf{K}(y) - \mathbf{K}(y|\langle S_n\rangle) <^+ \mathbf{K}(n).$$

Since $\min_{x\in S_n}\mathbf{K}(x) >^+ n$, by Theorem 3, S_n is proven to be exotic, with $\mathbf{I}(\langle S_n\rangle : \mathcal{H}) >^{\log} n$.

Theorem 3 *For finite set $D \subset \Sigma^*$, $\min_{x\in D}\mathbf{K}(x) <^{\log} \min_{x\in D}\mathbf{I}(x;D) + \mathbf{I}(D:\mathcal{H})$.*

Proof. Let $i = \lceil -\log \mathbf{m}(D)\rceil$. We define the following semi-measure $\tau(x) = 2^{i-1}\mathbf{m}(x)[x \in D]$. This semi-measure is 0 for all strings not in D. Since τ is lower computable, for all $x \in D$,

$$\begin{aligned}\mathbf{K}(x|D) &<^+ -\log \tau(x) + \mathbf{K}(\tau|D)\\ &<^+ \mathbf{K}(x) - i + \mathbf{K}(i)\\ i &<^+ \mathbf{I}(x;D) + \mathbf{K}(i)\\ i &<^{\log} \min_{x\in D}\mathbf{I}(x;D).\end{aligned}$$

An invocation of Theorem 2 gives us $x \in D$, with

$$\begin{aligned}\mathbf{K}(x) &<^{\log} i + \mathbf{I}(D:\mathcal{H})\\ &<^{\log} \min_{x\in D}\mathbf{I}(x;D) + \mathbf{I}(D:\mathcal{H}).\end{aligned}$$

□

7 Distortion of Individual Codewords

An equivalent form to Theorem 3 is the statement: *If there exists a element $y \in D$ then there exists another element $x \in D$, with $\mathbf{K}(x) <^{\log} \mathbf{I}(y : D) + \mathbf{I}(D : \mathcal{H})$.* The theorem in this form is a reworking of Theorem 2 in [VV04], which is a statement about sets of sets. In [VV04], Theorem 2 was shown to be an algorithmic version of rate-distortion theory.

In this section we provide another variant of the algorithmic rate distortion theory with the following setup. Alice wants to communicate a single message $\mathbf{y}$ to Bob, and they both share the same reference universal Turing machine U. Alice sends a program p to Bob, who decompresses it to a codeword $\mathbf{x} = U(p)$, and measures its accuracy with a distortion function $d(\mathbf{x}, \mathbf{y})$. A distortion function $d : \Sigma^* \times \Sigma^* \to \mathbb{R}_{\geq 0}$, is a non-negative, partial computable function. The following theorem is a characterization of the amount of information required to send for a particular distortion $R \in \mathbb{R}_{>0}$.

Theorem 4 *Let $\mathbf{y}$ be a codeword. Relativized to distortion function d and $R \in \mathbb{Q}_{>0}$, if there is a codeword $\mathbf{x}$ such that $d(\mathbf{x}, \mathbf{y}) < R$, then there exists a codeword $\mathbf{x}'$ such that $d(\mathbf{x}', \mathbf{y}) < R$ and $\mathbf{K}(\mathbf{x}') <^{\log} \mathbf{I}(\mathbf{x}; \mathbf{y}) + \mathbf{I}(\mathbf{y} : \mathcal{H})$.*

Proof. We recall that relativization to elementary objects means that the universal Turing machine have access to their encodings on auxilliary tapes and the complexity terms implicitly have the encoded objects in the conditional terms. The main substance of this proof is to modify the arguments of the previous section to apply to enumerable sets. Let $D_\infty = \{\mathbf{x} : d(\mathbf{x}, \mathbf{y}) < R\}$ be the finite or infinite set of codewords that have distortion measure less than R with $\mathbf{y}$. The set D_∞ is enumerable, and for total string $b \in \Sigma^*$, let D_b be the finite subset of D_∞, enumerated after $\mathbf{bb}(b)$ steps. Let $i = 1 + \lceil -\log \mathbf{m}(D_\infty) \rceil$, and b be the shortest total string where $i \geq -\log \mathbf{m}_b(D_b)$. Arguments similar to those in the proof of Theorem 2 show that b is unique with $\mathbf{K}(b|\mathbf{y}, \|b\|) <^+ \mathbf{K}(i)$. The invocation of Theorem 1, relative to b, with $m = \mathbf{m}_b$ and $D = D_b$, results in $\mathbf{x}' \in D_b$, with

$$\mathbf{K}(\mathbf{x}'|b) < i + \Lambda(D_b|b) + O(\mathbf{K}(i)). \tag{15}$$

Let $g : \Sigma^* \to \Sigma^*$, be total computable function between strings, defined as follows. For input $\mathbf{y}'$, g outputs an encoding of all strings $\mathbf{x}'$ such that $d(\mathbf{x}', \mathbf{y}') < R$, after enumeration of $\mathbf{bb}(b)$ steps. Thus $g(\mathbf{y}) = \langle D_b \rangle$. So Lemma 1, relativized to b, applied to g and D_b, results in

$$\begin{aligned}\Lambda(\langle D_b \rangle|b) &< \Lambda(\mathbf{y}|b) + O(\mathbf{K}(g|b)) \\ &<^+ \Lambda(\mathbf{y}|b).\end{aligned}$$

Combined with Equation 15,

$$\mathbf{K}(\mathbf{x}'|b) < i + \Lambda(\mathbf{y}|b) + O(\mathbf{K}(i)).$$

Lemma 2, conditional to b, results in

$$\begin{aligned}\mathbf{K}(\mathbf{x}'|b) &< i + \mathbf{I}(\mathbf{y} : \mathcal{H}|b) + O(\mathbf{K}(i) + \log \mathbf{I}(\mathbf{y} : \mathcal{H}|b)). \\ \mathbf{K}(\mathbf{x}') &< i + \mathbf{K}(b) + \mathbf{I}(\mathbf{y} : \mathcal{H}|b) + O(\mathbf{K}(i) + \log \mathbf{I}(\mathbf{y} : \mathcal{H}|b)).\end{aligned}$$

Using reasoning starting with Equation 14 in the proof of Theorem 2, we get

$$\mathbf{K}(\mathbf{x}') <^{\log} i + \mathbf{I}(\mathbf{y} : \mathcal{H}). \tag{16}$$

Let $\tau(x) = 2^{i-1}\mathbf{m}(x)[x \in D_\infty]$. This semi-measure is lower computable, and if $\mathbf{x} \in D_\infty$, then $\mathbf{K}(\mathbf{x}|\mathbf{y}) <^+ -\log\tau(\mathbf{x}) + \mathbf{K}(\tau|\mathbf{y})$. Using reasoning similar to that in Theorem 3, we get that for all $\mathbf{x} \in D_\infty$.

$$i <^{\log} \mathbf{I}(\mathbf{x};\mathbf{y}).$$

This, combined with Equation 16, results in

$$\mathbf{K}(\mathbf{x}') <^{\log} \mathbf{I}(\mathbf{x};\mathbf{y}) + \mathbf{I}(\mathbf{y}:\mathcal{H}).$$

□

Corollary 1 *Let $\mathbf{y}$ be a codeword and let the universal Turing machine be relativized to a distortion function d and $R \in \mathbb{Q}_{>0}$. If $\mathbf{x}$ is the simplest codeword where $d(\mathbf{x},\mathbf{y}) < R$, then $\mathbf{K}(\mathbf{x}|\mathbf{y}) < \mathbf{I}(\mathbf{y}:\mathcal{H}) + O(\log(\mathbf{I}(\mathbf{y}:\mathcal{H}) + \mathbf{K}(\mathbf{x})))$.*

8 String-Monotonic Machines

In this section, we relate continuous semi-measures with so-called string-monotonic programs. A continuous semi-measure Q is a function $Q : \Sigma^* \to \mathbb{R}_{\geq 0}$, such that $Q(\emptyset) = 1$ and for all $x \in \Sigma^*$, $Q(x) \geq Q(x0) + Q(x1)$. The function μ is the uniform measure, with $\mu(x) = 2^{-\|x\|}$. For continuous semi-measure Q, prefix free set $G \subset \Sigma^*$, $Q(G) = \sum_{x \in G} Q(x)$. For an open set B of the Cantor space, $Q(B) = Q(\{x : \text{ interval } \Gamma_x \text{ is maximal in } B\})$. Let $\mathbf{M}$ be a largest, up to a multiplicative factor, lower semi-computable continuous semi-measure. For more information about $\mathbf{M}$, we refer readers to [LV08]. Note that in our definition, $\mathbf{M}(\{x0, x1\})$ may differ from $\mathbf{M}(\{x\})$. $\mathbf{KM}(G)$ is used to denote $1 - \lceil \log \mathbf{M}(G) \rceil$. The notation $\mathbf{KM}(x)$ is used to denote $\mathbf{KM}(\{x\})$.

Informally speaking, a string-monotonic program is a total recursive Turing machine with an input tape, a work tape, and an output tape, where the tape heads of input tape and the output tape can only move in one direction. A total computable function $\nu : \Sigma^* \to \Sigma^*$ is string-monotonic iff for all strings x and y, $\nu(x) \sqsubseteq \nu(xy)$. Let $\overline{\nu} : \Sigma^{*\infty} \to \Sigma^{*\infty}$ be used to represent to the unique extension of ν to infinite sequences. Its definition for all $\alpha \in \Sigma^{*\infty}$ is $\overline{\nu}(\alpha) = \sup\{\nu(\alpha_{\leq n}) : n \leq \|\alpha\|\}$, where the supremum is respect to the partial order derived with the $\sqsubseteq$ relation. The following theorem relates prefix monotone machines and continuous semi-measures. It is similar to Theorem 4.5.2 in [LV08], with the additional property that the string-monotonic machine be total computable.

Theorem 5 *For each lower-semicomputable continuous semi-measure σ over Σ^∞, there is a string-monotonic function ν_σ such that for all $x \in \Sigma^*$, $\lceil -\log\sigma(x)\rceil =^+ \lceil -\log\mu\{\alpha : x \sqsubseteq \overline{\nu_\sigma}(\alpha),\ \alpha \in \Sigma^\infty\}\rceil$.*

Proof. We prove this theorem by an explicit construction of ν_σ. Since σ is lower-semicomputable, there exists a total computable function $\theta : \Sigma^* \times \mathbb{N} \to \mathbb{Q}_{\geq 0}$, such that $\theta(x, k+1) \geq \theta(x,k)$ and $\lim_{k\to\infty}\theta(x,k) = \sigma(x)$. Without loss of generality, we can assume, for all $k \in \mathbb{N}$, $\theta(x,k) \geq \theta(x0,k) + \theta(x1,k)$ and also for all $k \in \mathbb{N}$, $|\{x : \theta(x,k) > 0\}| < \infty$.

For a finite set of strings $S \subset \Sigma^*$, such that for all $x \in S$, $\|x\| < n$, we define $\delta(S,n) = \{xy : \|xy\| = n, x \in S\}$. If S contains a string of length not less than n, then $\delta(S,n)$ is undefined. For each string $x \in \Sigma^*$ and $k \in \mathbb{N}$, we define the finite prefix-free sets $S_{x,k} \subset \Sigma^*$ and $T_{x,k} \subset \Sigma^*$. For each $x \in \Sigma^*$, $k \in \mathbb{N}$, we define $\xi(x,k) = \lceil -\log \sum \left\{2^{-\|y\|} : y \in S_{x,k} \cup T_{x,k}\right\}\rceil$.

For each $k \in \mathbb{W}$, we will use natural numbers $N_k \in \mathbb{N}$, to be defined later. ν_σ starts by setting N_0 equal to some constant c, $S_{\emptyset,0} = \Sigma^c$, and $T_{\emptyset,0} = \{\}$. Also for $x \in \Sigma^*\backslash\emptyset$, $S_{x,0} = T_{x,0} = \{\}$. The variable k starts at 0.

The algorithm for ν_σ iterates in a loop, where at the beginning of the loop, k is incremented by 1. Next, the variable N_k is set to $\max\{N_{k-1}+1, \max\{(\lceil -\log\theta(x,k)\rceil + 2) : x \in \Sigma^*, \theta(x,k) > 0\}\}$. Starting with $\emptyset$, we perform the following operation on each string x where $\theta(x,k) > 0$, with the operation being performed on x before $x0$ and $x1$. We set $S_{x,k} = \delta(S_{x,k-1}, N_k)$ and $T_{x,k} = T_{x,k-1}$. This operation is defined because $S_{x,k-1} \subset \Sigma^{N_{k-1}}$ and $N_{k-1} < N_k$. The string x may have received a finite number of strings $D \subset \Sigma^{N_k}$ from its parent x^-. The string x adds these strings D to $S_{x,k}$. For $b \in \Sigma$, if $\xi(xb,k) > \lceil -\log\theta(xb,k)\rceil$, then the string x will gift enough strings from $S_{x,k}$ into $S_{xb,k}$ such that $\xi(xb,k) = \lceil -\log\theta(xb,k)\rceil$. The gifted strings are removed from $S_{x,k}$ and also put into $T_{x,k}$. After this step is completed, the algorithm for ν_σ restarts the loop, starting with the incrementing of k again.

On input of y, $\nu_\sigma(y)$ is defined to be x, where x is equal to first occurrence of a string in the looping algorithm described above, i.e. smallest k, with one of the following properties:

- $y \in S_{x,k}$
- there exists a $z \in S_{x,k}$, $z' \in S_{x,k+1}$, with $z \sqsubset y \sqsubset z'$
- there exists a $z \in S_{x,k}$, $b \in \{0,1\}$, $z' \in S_{xb,k+1}$, with $z \sqsubset y \sqsubset z'$.

From the construction, it can be seen that the algorithm for ν_σ is total computable. This construction satisfies the properties of the theorem. This is because for any $k \in \mathbb{N}$, if for $y \in \Sigma^*$, there exists an $x, z \in \Sigma^*$, and $k \in \mathbb{N}$ such that $y \sqsupseteq z \in S_{x,k} \cup T_{x,k}$, then $x \sqsubseteq \nu_\sigma(y)$. This combined with the fact that for all $k \in \mathbb{N}$, $\xi(x,k) = \lceil -\log\theta(x,k)\rceil$, ensures the theorem. □

Theorem 5 can be extended from strings x to finite prefix free sets of a strings G.

Corollary 2 *For each lower-computable continuous semi-measure σ over Σ^∞, there is a string-monotonic function ν_σ, where for prefix free $G \subset \Sigma^*$, $\lceil -\log\sigma(G)\rceil =^+ \lceil -\log\mu\{\alpha{:}\overline{\nu_\sigma}(\alpha) \sqsupseteq x \in G\}\rceil$.*

Since there is a universal lower-semicomputable continuous semi-measure $\mathbf{M}$, there exists a string-monotonic function $\nu_{\mathbf{M}}$, with the following property.

Corollary 3 *For finite prefix free set G, $\mathbf{KM}(G) =^+ -\log\mu\{\alpha : x \sqsubseteq \overline{\nu_{\mathbf{M}}}(\alpha),\, \alpha \in \Sigma^\infty, x \in G\}$.*

The string-monotonic function $\nu_{\mathbf{M}}$ is universal, due to the universality properties of $\mathbf{M}$, with respect to every ν_σ, for lower-computable continuous semi-measure σ. The uniform measure of sequences whose $\nu_{\mathbf{M}}$-image extends a particular string x is greater (up to a multiplicative constant) than the measure of sequences whose ν_σ-image extends that finite string. More formally for any string x, $\mu\{\alpha : x \sqsubseteq \overline{\nu_{\mathbf{M}}}(\alpha), \alpha \in \Sigma^\infty\} \overset{*}{>} \mu\{\alpha : x \sqsubseteq \overline{\nu_\sigma}(\alpha), \alpha \in \Sigma^\infty\}$.

Note that this is different than the universality properties of the (partial) algorithm U, defined in Section 3. Whereas the universal U uses an encoding of each algorithm to simulate it, $\nu_{\mathbf{M}}$ is constructed (by the proof of Theorem 5) with respect to $\mathbf{M}$, and so the universal properties of $\nu_{\mathbf{M}}$ are derived from the universal properties of $\mathbf{M}$. Since the algorithm for $\nu_{\mathbf{M}}$ requires the lower computation of $\mathbf{M}$ on all finite strings, this requires the lower computation of each lower-computable continuous semi-measure σ on all finite strings.

Definition 1 *For a given string-monotonic function ν, prefix free set $G \subset \Sigma^*$, and $N \in \mathbb{N}$, let $G^N_\nu = \Sigma^N \cap \nu^{-1}(G\Sigma^*) = \{y : y \in \Sigma^N, \nu(y) \sqsupseteq x \in G\}$. Thus G^N_ν represents the strings of length N, when given as input to ν, produces a string whose prefix is in G.*

Lemma 4 describes properties of G^N_ν, which is represented graphically in Figure 6

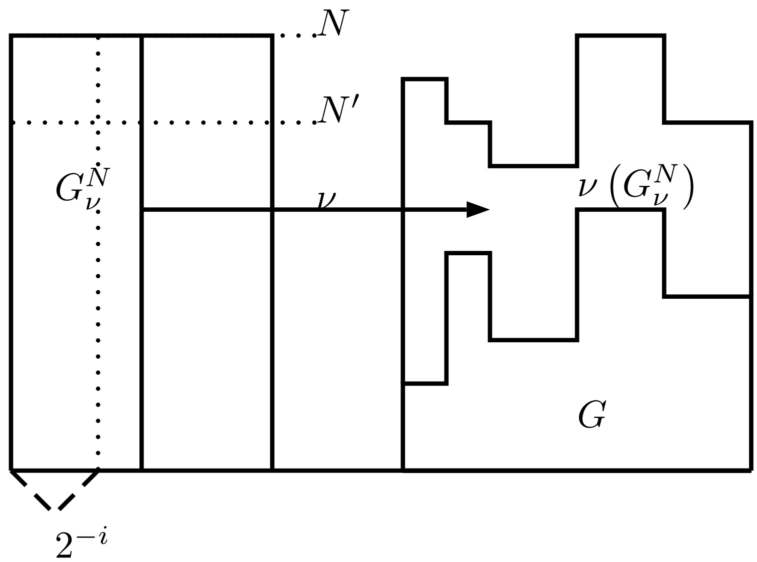

Figure 6: A graphical depiction of Lemma 4. The set $G_v^N \subseteq \Sigma^N$ has more than 2^{N-i} members, and each member has length N and an ν image with a prefix in G.

Lemma 4 *Let ν be a string-monotonic function and $G \subset \Sigma^*$ a finite prefix free set. If $i = 1 + \lceil -\log \mu\{\alpha : \overline{\nu}(\alpha) \sqsupseteq x \in G, \alpha \in \Sigma^\infty\}\rceil$ then there is some N', where*

1. *for all $N \geq N'$, $2^{-i+2} > |G_\nu^N| 2^{-N} > 2^{-i}$,*
2. *for all $N'' < N'$, $2^{-i} \geq |G_\nu^{N''}| 2^{-N''}$.*

Proof. We first show the existence of a number M, where $|G_\nu^M|2^{-M} > 2^{-i}$. Let $S = \{\alpha : \overline{\nu}(\alpha) \sqsupseteq x \in G, \alpha \in \{0,1\}^\infty\}$ be the open set of infinite sequences whose $\overline{\nu}$ images contains prefixes in G. Since S is open, there exists a finite or infinite prefix-free set of finite strings $F \subset \Sigma^*$, $S = \bigcup_{y \in F} y\Sigma^\infty$. For all $y \in F$, $\nu(y) \sqsupseteq x \in G$ and $\nu(y^-) \not\sqsupseteq x'$, for all $x' \in G$.

Furthermore, by the definition of i, $\sum_{y \in F} 2^{-\|y\|} \geq 2^{1-i}$. So there is some finite subset $F' \subseteq F$, where $\sum_{y \in F'} 2^{-\|y\|} > 2^{-i}$. Let $M \in \mathbb{N}$ be length of the longest string in F'. Then the set $F'' = \{xw : x \in F', \|w\| = M - \|x\|\}$ has the property $F'' \subseteq \Sigma^M$, $|F''|2^{-M} > 2^{-i}$, and for all $y \in F''$, $\nu(y) \sqsupseteq x \in G$. Thus since $G_\nu^M \supseteq F''$, $|G_\nu^M|2^{-M} > 2^{-i}$.

We show that for all $N \in \mathbb{N}$, $|G_\nu^N|2^{-N} < 2^{-i+2}$. Assume not, and there exists $N \in \mathbb{N}$ where $|G_\nu^N| \geq 2^{N-i+2}$. Thus $S \supseteq \{\alpha : \alpha \sqsupset y \in G_\nu^N, \alpha \in \Sigma^\infty\}$, which implies $-\log \mu(S) \leq i - 2$, causing a contradiction of the definition of i.

The statements of the lemma follow from the reasoning that for any $M \in \mathbb{N}$, $G_\nu^{M+1} \supseteq \{yb : y \in G_\nu^M, b \in \Sigma\}$, since if $\nu(y) \sqsupseteq x \in G$, then $\nu(yb) \sqsupseteq x$, for $b \in \Sigma$. This implies $|G_\nu^{M+1}|2^{-M-1} \geq |G_\nu^M|2^{-M}$, so for every N' where $|G_\nu^{N'}|2^{-N'} > 2^{-i}$, for all $N \geq N'$, $|G_\nu^N|2^{-N} > 2^{-i}$. The existence of such an N' is proven above. The smallest such N' proves the Lemma. $\square$

9 Complexity of Completing Predicates

In addition to the standard definition of Kolmogorov complexity, we introduce a monotonic variant. The monotone complexity of a finite prefix-free set $G \subset \Sigma^*$ of finite strings is $\mathbf{Km}(G) = \min_{U(p)\in G\Sigma^*} \|p\|$. This is larger than the usual definition of monotone complexity, see for example [LV08]. This is due to the requirement of U halting and being a standard universal program (instead of a monotone operator). However since Theorem 6 is an upper bound on $\mathbf{Km}(G)$, it applies to smaller definitions of monotonic complexity.

Theorem 6 *For any finite prefix-free set G of strings,*
$\mathbf{Km}(G) < \mathbf{KM}(G) + \mathbf{I}(G:\mathcal{H}) + O(\mathbf{K}(\mathbf{KM}(G)) + \log \mathbf{I}(G:\mathcal{H}))$.

Proof. Let $i = 1 + \lceil -\log \mu\{\alpha : x \sqsubseteq \overline{\nu_{\mathbf{M}}}(\alpha),\ \alpha \in \Sigma^\infty, x \in G\}\rceil$. By invoking Lemma 4 with $\nu_{\mathbf{M}}$, there exists a number $N' \in \mathbb{N}$, where for all $N \geq N'$, $2^{N-i+2} > |G^N_{\nu_{\mathbf{M}}}| > 2^{N-i}$ and for all $N'' < N'$, $2^{-i} \geq 2^{-N''}|G^{N''}_{\nu_{\mathbf{M}}}|$.

The next step is to invoke Theorem 1, using $D = G^N_{\nu_{\mathbf{M}}}$ for some $N \geq N'$. For reason discussed later, it is beneficial to have N represent common information between G and the halting sequence $\mathcal{H}$. To this end we let $b \in \Sigma^*$ be the shortest total string such that $N = \mathbf{bb}(b) \geq N'$. We note that b^- is not total (otherwise $\mathbf{bb}(b^-) \geq \mathbf{bb}(b)$).

To show that b is simple relative to G, we first show that it is unique. Otherwise there exists a $b' \neq b$, $\|b'\| = \|b\|$, such that $\mathbf{bb}(b') \geq N'$. It cannot be that $b' \lhd b$, otherwise b'^- would be total (see Figure 5). This implies $\mathbf{bb}(b'^-) \geq \mathbf{bb}(b') \geq N'$, contradicting the definition that b is the shortest total string with $\mathbf{bb}(b) \geq N'$. Similar reasoning can be used to show that it cannot be that $b \lhd b'$. Thus b is unique.

This string b can be found given G, i, and $\|b\|$ by the following algorithm. Each total string y of length $\|b\|$ is enumerated and the algorithm returns the first $y = b$ such that $|G^{\mathbf{bb}(y)}_{\nu_{\mathbf{M}}}|2^{-\mathbf{bb}(y)} > 2^{-i}$. Thus by this algorithm

$$\mathbf{K}(b|G, \|b\|) <^+ \mathbf{K}(i). \tag{17}$$

Hence b simple relative to G and some coded parameters. Let $N = \mathbf{bb}(b)$. By Lemma 4, since $|G^N_{\nu_{\mathbf{M}}}|2^{-N} > 2^{-i}$, this guarantees that $N \geq N'$. Let the finite set $D = G^N_{\nu_{\mathbf{M}}} = \{y : \|y\| = N, \nu_{\mathbf{M}}(y) \sqsupseteq x \in G\}$. Let $m(y) = [y \in \Sigma^N]2^{-N}$ be a computable elementary probability measure, uniform over strings of length N. Theorem 1, relativized to b, produces x with

$$\mathbf{K}(x|b) <^+ -\log m(D) + 2\mathbf{K}(\lceil -\log m(D)\rceil) + \Lambda(D|b). \tag{18}$$

Due to Lemma 4, since every string in D has length $N \geq N'$,

$$\begin{aligned} -\log m(D) &= -\log |D|2^{-N} \\ &= -\log |G^N_{\nu_{\mathbf{M}}}|2^{-N} \\ &=^+ i. \end{aligned}$$

Combined with Equation 18, we get

$$\mathbf{K}(x|b) < i + \Lambda(D|b) + O(\mathbf{K}(i)). \tag{19}$$

However Equation 2 still is in terms of D, whereas we want the final result to be in terms of G. Fortunately, D can be removed from the Λ term using Lemma 1. Let $g : \Sigma^* \to \Sigma^*$ be a total computable function defined as follows. The input to g can be assumed to be an encoding $\langle F\rangle$ of a finite prefix free set $F \subset \Sigma^*$. For all other inputs, g outputs the empty string. Otherwise, the output is an encoding of the finite set $F^{\mathbf{bb}(b)}_{\nu_{\mathbf{M}}} \subset \Sigma^*$, which is $\{y : \|y\| = \mathbf{bb}(b), \nu_{\mathbf{M}}(y) \sqsupseteq x \in F\}$, the set of all strings y of length $\mathbf{bb}(b)$, which has a $\nu_{\mathbf{M}}$ image whose prefix is in F. The function g is total recursive, because g can create the set $\Sigma^{\mathbf{bb}(b)}$ and test whether whether each string $y \in \Sigma^{\mathbf{bb}(b)}$ has an $\nu_{\mathbf{M}}$-image whose prefix is in F. We note that $\nu_{\mathbf{M}}$ is a string-monotonic function, and thus total computable. Thus

$$g(\langle G\rangle) = \left\langle G^{\mathbf{bb}(b)}_{\nu_{\mathbf{M}}}\right\rangle = \langle G^N_{\nu_{\mathbf{M}}}\rangle = \langle D\rangle.$$

Thus applying Lemma 1, using total function g, with b on the auxilliary tape results in

$$\begin{aligned}\Lambda(D|b) &<^+ \Lambda(G|b) + O(\mathbf{K}(g|b))\\ &<^+ \Lambda(G|b),\end{aligned}$$

where the second inequality is due to the fact that g is simple relative to b. So combined with Equation 19, we get

$$\mathbf{K}(x|b) < i + \Lambda(G|b) + O(\mathbf{K}(i)). \tag{20}$$

Due to Lemma 2, relativized to b, $\Lambda(G|b) <^{\log} \mathbf{I}(G : \mathcal{H}|b)$. So combined with Equation 20, we get

$$\mathbf{K}(x|b) < i + \mathbf{I}(G : \mathcal{H}|b) + O(\mathbf{K}(i) + \log \mathbf{I}(G : \mathcal{H}|b)). \tag{21}$$

The last step involves removing b from the conditionals, and we use the same reasoning starting at Equation 14 in the proof of Theorem 2. From Equation 21, and the fact that $\mathbf{K}(x) - \mathbf{K}(b) <^+ \mathbf{K}(x|b)$, we get

$$\mathbf{K}(x) < i + \mathbf{K}(b) + \mathbf{I}(G : \mathcal{H}|b) + O(\mathbf{K}(i) + \log(\mathbf{I}(G : \mathcal{H}|b) + \mathbf{K}(b))). \tag{22}$$

The total string b represents common information between G and $\mathcal{H}$, and since b^- is not total, we can apply Lemma 3 to Equation 22, resulting in

$$\mathbf{K}(x) < i + \mathbf{I}(G : \mathcal{H}) + \mathbf{K}(b|G, \|b\|) + O(\mathbf{K}(i) + \log \mathbf{I}(G : \mathcal{H}) + \log \mathbf{K}(b|G, \|b\|)). \tag{23}$$

Applying the inequality in Equation 17, we get

$$\mathbf{K}(x) < i + \mathbf{I}(G : \mathcal{H}) + O(\mathbf{K}(i) + \log \mathbf{I}(G : \mathcal{H})).$$

Due to the definition of i and Corollary 3, we have $i =^+ \mathbf{KM}(G)$, so

$$\mathbf{K}(x) < \mathbf{KM}(G) + \mathbf{I}(G : \mathcal{H}) + O(\mathbf{K}(\mathbf{KM}(G)) + \log \mathbf{I}(G : \mathcal{H})). \tag{24}$$

There exists a program $p \in \Sigma^*$, of length $<^+ \mathbf{K}(x)$, that produces x from a shortest x-program then outputs $\nu_{\mathbf{M}}(x)$ and halts. since $x \in D = G^N_{\nu_{\mathbf{M}}}$, x has a $\nu_{\mathbf{M}}$ image that has a prefix in G. So by the definition of $\mathbf{Km}$, $\mathbf{Km}(G) \leq \|p\| <^+ \mathbf{K}(x)$. So combined with Equation 24, we get the form of the theorem,

$$\mathbf{Km}(G) < \mathbf{KM}(G) + \mathbf{I}(G : \mathcal{H}) + O(\mathbf{K}(\mathbf{KM}(G)) + \log \mathbf{I}(G : \mathcal{H})).$$

□

A binary predicate is defined to be a function of the form $f : D \to \Sigma$, where $D \subseteq \mathbb{N}$. We say that binary predicate λ is an extension of γ, if $\mathrm{Dom}(\gamma) \subseteq \mathrm{Dom}(\lambda)$, and for all $i \in \mathrm{Dom}(\gamma)$, $\gamma(i) = \lambda(i)$. If binary predicate λ has a domain of $\mathbb{N}$ and is an extension of binary predicate γ, then we say it is a complete extension of γ. The self-delimiting code for a binary predicate γ with a finite domain is $\langle\{x_1, \lambda(x_1), \ldots, x_n, \lambda(x_n)\}\rangle$. The Kolmogorov complexity of a binary predicate λ with an infinite sized domain is $\mathbf{K}(\lambda) = \mathbf{K}(f)$, where $f : \mathbb{N} \to \mathbb{N}$ is a partial computable function where $f(i) = \lambda(i)$ if $i \in \mathrm{Dom}(\lambda)$ and $f(i)$ is undefined otherwise. If there is no such partial computable function, then $\mathbf{K}(\lambda) = \infty$.

Theorem 7 *For binary predicate γ and the set Γ of complete extensions of γ,*
$\min_{g \in \Gamma} \mathbf{K}(g) <^{\log} |\mathrm{Dom}(\gamma)| + \mathbf{I}(\langle\gamma\rangle : \mathcal{H})$.

Proof. The theorem is meaningless if $|\mathrm{Dom}(\gamma)| = \infty$, so we can assume $\mathrm{Dom}(\gamma)$ is finite. Let

$$N = \max\{i \in \mathrm{Dom}(\gamma)\} = \max\{i : \gamma(i) \text{ is defined}\}.$$

Let $G \subseteq \Sigma^N$ be the largest set of strings x of length N such that $x \supseteq \gamma$. Thus for every $x \in G$ and for every $i \in \mathrm{Dom}(\gamma)$, $x_i = \gamma(i)$. Since all the strings in G have the same values at indices specified by $\mathrm{Dom}(\gamma)$, their meaure is equal to

$$\mu(G) = 2^{-|\mathrm{Dom}(\gamma)|}.$$

For example, if $\gamma = (2, 0), (4, 0)$, then

$$G = \begin{Bmatrix} 0000, \\ 0010, \\ 1000, \\ 1010 \end{Bmatrix},$$

and $\mu(G) = 2^{-|\mathrm{Dom}(G)|} = 2^{-2}$.

Theorem 6, applied to G, has $\mathbf{Km}(G) <^{\log} \mathbf{KM}(G) + \mathbf{I}(G : \mathcal{H})$. Thus by the definition of $\mathbf{Km}$, there is some program p with

$$\|p\| <^{\log} \mathbf{KM}(G) + \mathbf{I}(G : \mathcal{H}), \tag{25}$$

where $U(p) \in G\Sigma^*$, that is p outputs some string $U(p)$ who has a prefix in G.

From an encoding of G, one can produce γ (see the above example) and from an encoding of γ, one can produce G. Thus

$$\begin{aligned}\mathbf{K}(G|\langle\gamma\rangle) &= O(1),\\ \mathbf{K}(\langle\gamma\rangle|G) &= O(1).\end{aligned}$$

This implies

$$\begin{aligned}\mathbf{I}(G : \mathcal{H}) &= \mathbf{K}(G) - \mathbf{K}(G|\mathcal{H})\\ &<^+ \mathbf{K}(\langle\gamma\rangle) + \mathbf{K}(G|\langle\gamma\rangle) - \mathbf{K}(G|\mathcal{H})\\ &<^+ \mathbf{K}(\langle\gamma\rangle) - \mathbf{K}(G|\mathcal{H})\\ &<^+ \mathbf{K}(\langle\gamma\rangle) - \mathbf{K}(\langle\gamma\rangle|\mathcal{H}) + \mathbf{K}(\langle\gamma\rangle|G)\\ &<^+ \mathbf{I}(\langle\gamma\rangle : \mathcal{H}).\end{aligned}$$

So combined with Equation 25, we get

$$\|p\| <^{\log} \mathbf{KM}(G) + \mathbf{I}(\langle\gamma\rangle : \mathcal{H}). \tag{26}$$

In addition, since $\mathbf{M}$ is a universal semi-computable continuous semi-measure, it majorizes the uniform measure, with $\mathbf{M} \overset{*}{>} \mu$. Hence

$$\mathbf{KM}(G) <^{+} -\log \mu(G).$$

So combined with Equation 26,

$$\begin{aligned} \|p\| &<^{\log} -\log \mu(G) + \mathbf{I}(\langle\gamma\rangle : \mathcal{H}), \\ \|p\| &<^{\log} |\mathrm{Dom}(\gamma)| + \mathbf{I}(\langle\gamma\rangle : \mathcal{H}). \end{aligned} \tag{27}$$

Thus there exists a complete extension $g' \in \Gamma$, of γ, that is equal to $U(p)_i$ for all $i \leq \|U(p)\|$, and 0 otherwise. This g' can be computed with a program of size $<^{+} \|p\|$, thus combined with Equation 27,

$$\min_{g\in\Gamma} \mathbf{K}(g) \leq \mathbf{K}(g') <^{+} \|p\| <^{\log} |\mathrm{Dom}(\gamma)| + \mathbf{I}(\langle\gamma\rangle : \mathcal{H}).$$

□

References

[DH10] R. G. Downey and D.R. Hirschfeldt. *Algorithmic Randomness and Complexity*. Theory and Applications of Computability. Springer New York, 2010.

[EL11] Samuel Epstein and Leonid Levin. On sets of high complexity strings. *CoRR*, abs/1107.1458, 2011.

[Eps13] Samuel Epstein. All sampling methods produce outliers. *CoRR*, abs/1304.3872, 2013.

[Ǵ13] P. Gács. Lecture notes on descriptional complexity and randomness, 2013.

[GTV01] P. Gács, J. Tromp, and P. Vitányi. Algorithmic Statistics. *IEEE Transactions on Information Theory*, 47(6):2443–2463, 2001.

[KU87] A. N. Kolmogorov and V. A. Uspensky. Algorithms and Randomness. *SIAM Theory of Probability and Its Applications*, 32(3):389–412, 1987.

[LV08] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Publishing Company, Incorporated, 3 edition, 2008.

[She83] A. Shen. The concept of (alpha,beta)-stochasticity in the Kolmogorov sense, and its properties. *Soviet Mathematics Doklady*, 28(1):295–299, 1983.

[She99] A. Shen. Discussion on Kolmogorov Complexity and Statistical Analysis. *The Computer Journal*, 42(4):340–342, 1999.

[VS15] Nikolai K. Vereshchagin and Alexander Shen. Algorithmic statistics revisited. *CoRR*, abs/1504.04950, 2015.

[VS17] Nikolay K. Vereshchagin and Alexander Shen. Algorithmic statistics: Forty years later. In *Computability and Complexity*, pages 669–737, 2017.

[VV04] N. Vereshchagin and P. Vitányi. Kolmogorov's Structure Functions and Model Selection. *IEEE Transactions on Information Theory*, 50(12):3265 – 3290, 2004.

[V'Y87] V.V. V'Yugin. On Randomness Defect of a Finite Object Relative to Measures with Given Complexity Bounds. *SIAM Theory of Probability and Its Applications*, 32:558–563, 1987.

[V'Y99] V.V. V'Yugin. Algorithmic complexity and stochastic properties of finite binary sequences, 1999.