Examples of Derandomization

Samuel Epstein[*]

August 19, 2022

Contents

1 Tools

Proposition 1
For every $c, n \in \mathbb{N}$, if $x < y + c$ for some $x, y \in \mathbb{N}m$ then $x + n\mathbf{K}(x) < y + n\mathbf{K}(y) + O(n \log n) + 2c$.

Proof. $\mathbf{K}(x) <^+ \mathbf{K}(y) + \mathbf{K}(y-x)$ as x can be computed from y and $(y-x)$. Therefore $n\mathbf{K}(x) - n\mathbf{K}(y) < n\mathbf{K}(y-x) + dn$, for some $d \in \mathbb{N}$ dependent on U. We assume that this equation is not true; then, there exists $x, y, c \in \mathbb{N}$ where $x < y + c$, and $g \leq O(n \log n) + 2c$ where $y - x + g < n\mathbf{K}(x) - n\mathbf{K}(y) < n\mathbf{K}(y-x) + dn$, which is a contradiction for $g =^+ dn + 2c + \max_a\{2n \log a - a\} =^+ dn + 2c + 2n \log n$. □

Lemma 1 *For partial computable $f : \mathbb{N} \to \mathbb{N}$, for all $a \in \mathbb{N}$, $\mathbf{I}(f(a); \mathcal{H}) <^+ \mathbf{I}(a; \mathcal{H}) + \mathbf{K}(f)$.*

Proof.

$$\mathbf{I}(a; \mathcal{H}) = \mathbf{K}(a) - \mathbf{K}(a|\mathcal{H}) >^+ \mathbf{K}(a, f(a)) - \mathbf{K}(a, f(a)|\mathcal{H}) - \mathbf{K}(f).$$

[*]JP Theory Group. samepst@jptheorygroup.org

The chain rule ($\mathbf{K}(x,y) =^+ \mathbf{K}(x) + \mathbf{K}(y|x,\mathbf{K}(x))$) applied twice results in

$$\begin{aligned}
\mathbf{I}(a;\mathcal{H}) + \mathbf{K}(f) &>^+ \mathbf{K}(f(a)) + \mathbf{K}(a|f(a),\mathbf{K}(f(a))) - (\mathbf{K}(f(a)|\mathcal{H}) + \mathbf{K}(a|f(a),\mathbf{K}(f(a)|\mathcal{H}),\mathcal{H}) \\
&=^+ \mathbf{I}(f(a);\mathcal{H}) + \mathbf{K}(a|f(a),\mathbf{K}(f(a))) - \mathbf{K}(a|f(a),\mathbf{K}(f(a)|\mathcal{H}),\mathcal{H}) \\
&=^+ \mathbf{I}(f(a);\mathcal{H}) + \mathbf{K}(a|f(a),\mathbf{K}(f(a))) - \mathbf{K}(a|f(a),\mathbf{K}(f(a)),\mathbf{K}(f(a)|\mathcal{H}),\mathcal{H}) \\
&>^+ \mathbf{I}(f(a);\mathcal{H}).
\end{aligned}$$

□

Theorem 1 ([Lev16, Eps19])
For finite $D \subset \{0,1\}^*$, $-\log \max_{x\in D} \mathbf{m}(x) <^{\log} -\log \sum_{x\in D} \mathbf{m}(x) + \mathbf{I}(D;\mathcal{H})$.

We recall the definitions from the introduction. Random functions F over natural numbers are modeled by discrete stochastic processes indexed by $\mathbb{N}$, where each $F(t)$, $t \in \mathbb{N}$, is a random variable over $\mathbb{N}$. $\mathcal{F}$ is the set of all random functions. A random function $F \in \mathcal{F}$ is computable if there is a program that on input $(a_1,\dots,a_n)$ lower computes $\Pr[F(1) = a_1 \cap F(2) = a_2 \cap \cdots \cap F(n) = (a_n)]$. Put another way, a random function $F \in \mathcal{F}$ is computable if $X = \Pr[F(a_1) = b_1 \cap \cdots \cap F(a_n) = b_n]$ is uniformlly computble in $\{(a_i,b_i)\}_{i=1}^n$. The complexity $\mathbf{K}(F)$ of a random function $F \in \mathcal{F}$, is the smallest program that computes X. $\mathcal{G}$ is the set of all deterministic functions $G : \mathbb{N} \to \mathbb{N}$. A sample $S \in \mathcal{S}$ is a finite set of pairs $\{(a_i,b_i)\}_{i=1}^n$. The encoding of a sample is $\langle S\rangle = \langle\{(a_i,b_i\}_{i=1}^n\rangle$. $\mathcal{S}$ is the set of all samples. We say $G(S)$ if G is consistent with S, with $G(a_i) = b_i$, $i = 1,\dots,n$. For random functions, $F(S)$ is the event that F is consistent with S.

Theorem 2 *For $F \in \mathcal{F}$, $S \in \mathcal{S}$, if $s = \lceil -\log \Pr[F(S)]\rceil$ and $h = \mathbf{I}(\langle S\rangle;\mathcal{H})$, then*
$\min_{G\in\mathcal{G},G(S)} \mathbf{K}(G) < \mathbf{K}(F) + s + h + O(\mathbf{K}(s,h) + \log \mathbf{K}(F))$.

Theorem 2 can be readily extended to sets of samples $\mathfrak{S} = \{S_1,\dots,S_n\}$, where for deterministic function $G : \mathbb{N} \to \mathbb{N}$, $G(\mathfrak{S})$ if $\bigcup_{i=1}^n G(S_i)$. For random function $F \in \mathcal{F}$, $F(\mathfrak{S})$ is the union of events $F(S_i)$, $i = 1,\dots,n$.

Corollary 1 *For $F \in \mathcal{F}$, if $s = \lceil -\log \Pr[F(\mathfrak{S})]\rceil$ and $h = \mathbf{I}(\langle\mathfrak{S}\rangle;\mathcal{H})$, then $\min_{G\in\mathcal{G},G(\mathfrak{S})} \mathbf{K}(G) <^{\log}$ $\mathbf{K}(F) + s + h + O(\mathbf{K}(s,h) + \log \mathbf{K}(F))$.*

A continuous semi-measure Q is a function $Q : \{0,1\}^* \to \mathbb{R}_{\geq 0}$, such that $Q(\emptyset) = 1$ and for all $x \in \{0,1\}^*$, $Q(x) \geq Q(x0) + Q(x1)$. For prefix free set D, $Q(D) = \sum_{x\in D} Q(x)$. Let $\mathbf{M}$ be a largest, up to a multiplicative factor, lower semi-computable continuous semi-measure. That is, for all lower computable continuous semi-measures Q there is a constant $c \in \mathbb{N}$ where for all $x \in \{0,1\}^*$, $c\mathbf{M}(x) > Q(x)$. Thus for any lower computable continuous semi-measure W and open set S, $-\log \mathbf{M}(S) <^+ \mathbf{K}(W) - \log W(S)$, where $\mathbf{K}(W)$ is the size of the smallest program that lower computes W.

The monotone complexity of a finite prefix-free set G of finite strings is $\mathbf{Km}(G) \stackrel{\text{def}}{=} \min\{\|p\| : U(p) \in x \sqsupseteq y \in G\}$. Note that this differs from the usual definition of $\mathbf{Km}$, in that our definition requires U to halt.

Theorem 3 *For finite prefix-free set $G \subset \{0,1\}^*$, $i = \lceil -\log \mathbf{M}(G)\rceil$, $h = \mathbf{I}(G;\mathcal{H})$, we have*
$\mathbf{Km}(G) < i + h + O(\mathbf{K}(i,h))$.

Corollary 2 *For (potentially infinite) prefix-free set $G \subset \{0,1\}^*$, where if $i = \lceil -\log \mathbf{Km}(G)\rceil$, $h = \mathbf{I}(\langle G\rangle : \mathcal{H})$, then $\mathbf{Km}(G) < i + h + O(\mathbf{K}(i,h))$.*

Theorem 3 can also be applied to clopen sets $C \subseteq \{0,1\}^\infty$. In this case $\mathbf{M}(C) = \sum\{\mathbf{M}(x) : \Gamma_x \text{ is maximal in } C\}$. In addition $\mathbf{Km}(C)$ is the shortest program that will produce a string $x \in \{0,1\}^*$ such that $\Gamma_x \subseteq C$. This also applies to Corollary 2 and open sets.

For the first model, the agent $\mathbf{p}$ and environment $\mathbf{q}$ are defined as follows. The agent is a function $\mathbf{p} : (\mathbb{N}\times\mathbb{N})^* \to \mathbb{N}$, where if $\mathbf{p}(w) = a$, $w \in (\mathbb{N}\times\mathbb{N})^*$ is a list of the previous actions of the agent and the environment, and $a \in \mathbb{N}$ is the action to be performed. The environment is of the form $\mathbf{q} : (\mathbb{N}\times\mathbb{N})^* \times \mathbb{N} \to \mathbb{N} \cup \{\mathbf{W}\}$, where if $\mathbf{q}(w,a) = b \in \mathbb{N}$, then b is $\mathbf{q}$'s response to the agent's action a, given history w, and the game continues. If $\mathbf{q}$ responds $\mathbf{W}$ then the agents wins and the game halts. The agent can be randomized. The game can continue forever, given certain agents and environments. This is called a win/no-halt game.

Theorem 4 *If probabilistic agent $\mathbf{p}'$ wins against environment $\mathbf{q}$ with at least probability p, then there is a deterministic agent $\mathbf{p}$ of complexity $<^{\log} \mathbf{K}(\mathbf{p}') - \log p + \mathbf{I}(\langle p, \mathbf{p}', \mathbf{q}\rangle; \mathcal{H})$ that wins against $\mathbf{q}$.*

The second game is modified such that the environment gives a nonnegative rational penalty term to the agent at each round. Furthermore the environment specifies an end to the game without specifying a winner or loser. This is called a penalty game.

Corollary 3 *If given probabilistic agent $\mathbf{p}$, environment $\mathbf{q}$ halts with probability 1, and $\mathbf{p}$ has expected penalty less than $n \in \mathbb{N}$, then there is a deterministic agent of complexity $<^{\log} \mathbf{K}(\mathbf{p}) + \mathbf{I}(\langle \mathbf{p}, n, \mathbf{q}\rangle; \mathcal{H})$ that receives penalty $< 2n$ against $\mathbf{q}$.*

2 Examples

2.1 Function-Minimization

Given computable functions $\{f_i\}_{i=1}^n$, where each $f_i : \mathbb{N} \to \mathbb{N}\cup\infty$, the goal of Function-Minimization is to find numbers $\{x_i\}_{i=1}^n$, that minimizes $\sum_{i=1}^n f_i(x_i)$. Let $p : \mathbb{N} \to \mathbb{R}_{\geq 0}$ be a lower semi-computable semi-measure, where $\mathbf{E}_p[f_i] \in \mathbb{R}$ for all $i = 1,\dots,n$. We define a computable probability $P : \{0,1\}^* \to \mathbb{R}_{\geq 0}$ where $P(\langle a_1\rangle\langle a_2\rangle \dots \langle a_n\rangle) = \prod_{i=1}^n p(a_i)$. $\mathbf{K}(P) <^+ \mathbf{K}(p,n)$. Let D' be a (potentially infinite) set of strings where $x \in D$ iff $x = \langle a_1\rangle\langle a_2\rangle\dots\langle a_n\rangle$ and

$$\sum_{i=1}^n f_i(a_i) \leq \left\lceil 2\sum_{\{b_i\}}\left(\prod_{i=1}^n p(b_i)\right)\sum_{i=1}^n f_i(b_i)\right\rceil = \left\lceil 2\sum_{i=1}^n \mathbf{E}_p[f_i]\right\rceil.$$

Let $\tau = \lceil 2\sum_{i=1}^n \mathbf{E}_p[f_i]\rceil$. By the Markov inequality, let the finite set $D \subseteq D'$ be constructed from $\langle p, \{f_i\}, \tau\rangle$, such that $P(D) > 1/2$ and $\mathbf{K}(D|\langle p, \{f_i\}, \tau\rangle) = O(1)$. By Theorem 1 and Lemma 1, there a string $x \in D$ such that

$$\begin{aligned}
\mathbf{K}(x) &<^{\log} -\log \mathbf{m}(D) + \mathbf{I}(D;\mathcal{H})\\
&<^{\log} \mathbf{K}(P) - \log P(D) + \mathbf{I}(\langle p, \{f_i\}, \tau\rangle;\mathcal{H})\\
&<^{\log} \mathbf{K}(p,n) + \mathbf{I}(\langle p, \{f_i\}, \tau\rangle;\mathcal{H}).
\end{aligned}$$

Thus given any computable probability p and functions $\{f_i\}_{i=1}^n$, there are numbers $\{x_i\}_{i=1}^n$ such that $\sum_{i=1}^n f(x_i) \leq \lceil 2\sum_{i=1}^n \mathbf{E}_p[f_i]\rceil = \tau$ and $\mathbf{K}(\{x_i\}_{i=1}^n) <^{\log} \mathbf{K}(n,P) + \mathbf{I}(\langle p, \{f_i\}, \tau\rangle;\mathcal{H})$.

Note that if the functions are uncomputable, then using the argument $\mathbf{I}(a;\mathcal{H}) <^+ \mathbf{I}(\beta:\mathcal{H})+\mathbf{K}(a|\beta)$, the information term is $\mathbf{I}(\langle p,\{f_i\},\tau\rangle:\mathcal{H})$, where $\langle p,\{f_i\},\tau\rangle\in\{0,1\}^\infty$.

An instance of this formulation is as follows. Let $n=1$ and $f_1(a)=[a>2^m]\infty+[a\leq 2^m]2^{m-\mathbf{K}(a|m)}$. Let $p(a)=[a\leq 2^m]2^{-m}$. Thus this example proves there exists a number x such that $f_1(x)\leq\lceil 2\mathbf{E}_p[f_1]\rceil\leq 2$. Furthermore

$$\mathbf{K}(x) <^{\log} \mathbf{K}(p)+\mathbf{I}(\langle p,f_1\rangle;\mathcal{H}) <^{\log} \mathbf{K}(m)+\mathbf{I}(\langle m,f_1\rangle;\mathcal{H}).$$

But if $f_1(x)\leq 2$, by the definition of f_1, this means $\mathbf{K}(x)\geq m-1$. This means $m <^{\log} \mathbf{I}(\langle m,f_1\rangle;\mathcal{H}) <^{\log} \mathbf{I}(f_1;\mathcal{H})$. This makes sense because f_1 is a deficiency of randomness function and therefore $m <^{\log} \mathbf{K}(f_1)$ and $\mathbf{K}(f_1|\mathcal{H}) <^+ \mathbf{K}(m)$.

2.2 Randomized-QuickSort

The goal of Randomized-QuickSort is to sort a list of n numbers. The algorithm is as follows. At the start of each round, a picot location is chosen at random. Then the array is sorted put numbers smaller than the pivot to the left and larger than the pivot to the right. For any starting array, the expected number of comparisons $<2n\ln n$. Furthermore it is

Theorem 5 *Given an array A of n numbers, there is a list x of pivots $\{v_i\}_{i=1}^m$ $m\leq n$ for which* Randomizes-QuickSort *can use to sort the array with less than $4n<\ln n$ comparisons. This list has complexity $\mathbf{K}(x) <^{\log} 4\log n+\mathbf{I}(\langle A\rangle;\mathcal{H})$.*

Proof. Let $D\subset\mathbb{N}^n$ consist of all permutations of the numbers $\{1,\ldots,n\}$, such that when applied to the Randomized-QuickSort algorithm, produces a sorted array in $<4n\ln n$ comparisons. $\mathbf{K}(D|A)=O(1)$. Let P be a computable probability over $\{0,1\}^*$ that gives equal probability to each permutation to encoded numbers $\{1,\ldots,n\}$. Thus $P(D)\geq 1/2$. Thus by Theorem 1 and Lemma 1, there is an $x'\in D$, with $\mathbf{K}(x') <^{\log} \mathbf{K}(P)-\log P(D)+\mathbf{I}(D;\mathcal{H}) <^{\log} \mathbf{K}(n)+\mathbf{I}(\langle A\rangle;\mathcal{H})$. Thus there is a cut-off point $m\leq n$ in x' where the algorithm halts. Let $x\sqsubseteq x'$ be the pivot points that are used by the Randomized-QuickSort algorithm. Thus

$$\mathbf{K}(x) <^{\log} 4\log n+\mathbf{I}(\langle A\rangle;\mathcal{H}).$$

2.3 Independent-Set

An independent seet in a graph G is a set of vertices with no edges between then. The Independent-Set problem consists of an undirected graph G and the goal is to find the largest independent set of that G.

Theorem 6 *For a graph G on n vertices with m edges, there exists an independent set S of size $0.75\sqrt{n}-2m/n$ and complexity $<^{\log} \mathbf{K}(n,m)+4(\log n)(m/n)+\mathbf{I}(G;\mathcal{H})$.*

Proof. We use a modification of the algorithm in the proof of Theorem 6.5 in [MU05]. The randomized algorithm A is as follows.

1. Delete each vertex (along with its incident edges) independently with probability $1-p$.

2. For each remaining edge, remove it and one of its adjacent vertices.

For X, the number of vertices that survive the first round $\mathbf{E}[X] = np$. Let Y be the number of edges that survive the first step, $\mathbf{E}[Y] = mp^2$. The second steps removes at most Y vertices. The output is an independent set size of at least $\mathbf{E}[X - Y] = np - mp^2$.

Let $p = 1/\sqrt{n}$. Thus $\mathbf{E}[X] = \sqrt{n}$, $\mathbf{E}[Y] = m/n$, and $\mathbf{E}[X - Y] = \sqrt{n} - m/n$. By the Markov inequality, $\Pr[Y < 2m/n] > 1/2$. By the Hoeffding's inequality,

$$\Pr[X \leq 0.75\sqrt{n}] \leq e^{-2*(0.75)^2(np)^2/n} \leq e^{-2(.75^2)(n*n^{-.5})^2/n} \leq e^{-2*0.5} = e^{-1}.$$

For a sequence $x \in \{0,1\}^*$, $x_1 = |\{i : x[i] = 1\}|$ and $x_0 = \|x\| - x_1$. Let $P : \{0,1\}^* \to \mathbb{R}_{\geq 0}$ be a computable probability, where for a string $x \in \{0,1\}^n$, $P(x) = (x_1)^{1/\sqrt{n}}(x_0)^{1-1/\sqrt{n}}$. Thus each x represents a selection of vertices selected according to the randomized algorithm A. Let $D \subseteq \{0,1\}^n$ be the set consists of all sequences x such that the X variable resultant from x is $X_x > 0.75\sqrt{n}$ and the Y variable resultant from algorithm A is $Y_x \leq 2m/n$. Thus $P(D) \geq (1 - e^{-1}) + 1/2 - 1 > 1/10$. Furthermore D can be constructed from G, with $\mathbf{K}(D|G) = O(1)$. By Theorem 1 and Lemma 1, there exists an $x \in D$, with

$$\begin{aligned}
\mathbf{K}(x) &<^{\log} \mathbf{K}(P) - \log P(D) + \mathbf{I}(D; \mathcal{H}) \\
&<^{\log} \mathbf{K}(n) + \mathbf{I}(G; \mathcal{H}).
\end{aligned}$$

In order for x to represent an independent set, the second step of algorithm A needs to be applied. In this case there are $< 2m/n$ vertices that needs to be removed. Thus a modification x' that has these vertices deleted represents an independent set.

$$\mathbf{K}(x') <^{\log} \mathbf{K}(x|n) + \mathbf{K}(n,m) + (2\log n)(2m/n) <^{\log} \mathbf{K}(n,m) + (4\log n)(m/n) + \mathbf{I}(D; \mathcal{H}).$$

This independent set has $X_x > 0.75\sqrt{n}$ and $Y_x < 2m/n$, it size is $\geq 0.75\sqrt{n} - 2m/n$.

2.4 Set-Membership

For a set $G \subseteq \{0,1\}^\ell$, a function $f : \{0,1\}^* \to \{0,1\}$ is a partial checker for G, if $f(x) = 1$ if $x \in G$. We use $\mathcal{U}$ to denote the uniform distribution over $\{0,1\}^\ell$. $\text{Error}(G,f) = \Pr_{x \sim \mathcal{U}}[f(x) = 1, x \notin G]$. The goal of Set-Membership, is given a set $G \subseteq \{0,1\}^\ell$, what is the simpliest partial checker f for G that reduces $\text{Error}(G,f)$.

Theorem 7 *For $n > O(1)$, given $G \subseteq \{0,1\}^\ell$, $|G| = m$, there is a partial checker f such that $\text{Error}(f,G) \leq 0.878^{n/m}$ and $\mathbf{K}(f) <^{\log} \mathbf{K}(n,m,\ell) + n + \mathbf{I}(\langle G, n\rangle; \mathcal{H})$.*

Proof. We derandomize the Bloom filter algorithm [Blo70]. Let there be k random functions $h_i : \{0,1\}^\ell \to \{1,\ldots,n\}$, where each h_i maps each input $x \in \{0,1\}^\ell$ to its range with uniform probability. We start with a string $v = 0^n$. For each member $x \in G$, and $i \in \{1,\ldots,k\}$, $v[h_i(x)]$ is set to 1. Thus the functions h_i serve as a way to test membership of G. If $x \in G$, then all the indicator functions h_i would be one. The probability that a specific bit is 0 is

$$p' = \left(1 - \frac{1}{n}\right)^{km}.$$

Let X be the number of bins that are 0. Due to [MU05],

$$\Pr(|X - np'| \geq \epsilon n) \leq 2e\sqrt{n}e^{-n\epsilon^2/3p'}.$$

For $\epsilon = p'/10$, we get

$$\Pr(X/n \geq p'9/10) \leq 2e\sqrt{n}e^{-np'/300}.$$

Thus for proper choice of k determined later, the right hand side of the above inequality is less than 0.5. Thus with probability $> .5$, the expected false positive rate, r that is $x \in \{0,1\}^\ell$, $x \notin G$, $h_i(x) = 1$, for all $i \in \{1,\ldots,k\}$ is less than

$$\begin{aligned} r \leq & (1 - .9p')^k \\ = & \left(1 - .9\left(1 - \frac{1}{n}\right)^{km}\right)^k \\ \leq & \left(1 - .9e^{-km/n}\right)^k. \end{aligned}$$

Setting $k = \lceil n/m \rceil$, with probability $\geq 1/2$, $r \leq (1 - .5e^{-2})^{m/n} \leq 0.878^{m/n}$.

Let F' consist of all encodings of k hash functions $h_i : \{0,1\}^\ell \to \{1,\ldots,n\}$. Let $F \subseteq F'$ consist of all hash functions such that the probability of error $\leq 0.878^{m/n}$. Let P be the uniform distribution over F'. $P(F) > 1/2$. $\mathbf{K}(F|G,n) = O(1)$. By Theorem 1 and Lemma 1, there is an $h \in F$ such that

$$\mathbf{K}(h) <^{\log} \mathbf{K}(P) - \log P(F) + \mathbf{I}(\langle F\rangle;\mathcal{H}) <^{\log} \mathbf{K}(n,m,\ell) + \mathbf{I}(\langle G,n\rangle;\mathcal{H}).$$

Thus h represents a set of k deterministic hash functions. Let x be the Bloom filter using h on G. Using x and h, one can define a partial checker f that is a Bloom filter such that $\mathrm{Error}(f,G) \leq 0.669^{n/m}$. Furthermore,

$$\mathbf{K}(f) <^{\log} \mathbf{K}(x|n) + \mathbf{K}(h,n) <^{\log} \mathbf{K}(n,m,\ell) + n + \mathbf{I}(\langle G,n\rangle;\mathcal{H}).$$

□

2.5 Parallel-Routing

The Parallel-Routing problem consists (G,d) an directed graph $G = (N,V)$ and a set of destinations $d : N \to N$. Each node represents a processor i in a network containing a packet v_i destined for another processor $d(i)$ in the network. The packet moves along a route represented by a path in G. During its transmission, a packet may have to wait at an intermediate node because the node is busy transmitting another packet. Each node contains a separate queue for each of its links and follows a FIFO queuing disciple to route packets, with ties handled arbitrarily. The goal of Parallel-Routing is to provide N routes from $i \in N$ to $d(i)$ that minimize lag time.

We restrict graphs to *Boolean Hypercube* networks, which is popular for parallel processing. The cube network contains $N = 2^n$ processing elements/nodes and is connected in the following manner. if $(i_0,\ldots,i_{n-1})$ and $(j_0,\ldots,j_{n-1})$ are binary representation of node i and node j, then there exist directed edges (i,j) and (j,i) betweeen the nodes if and only if the binary representation differ in exactly one position. One set of solutions, called *oblivious algrithms* satisfies the following property: a route followed by v_i depends on $d(i)$ alone, and not on $d(j)$ for any $j \neq i$. We focus our attention on a 2 phase oblivious routing algorithm, Two-Phase. Under this scheme, packet v_i executes the following two phases independently of all the other packets.

1. Pick a intermediate destination $\sigma(i)$. Packet v_i travels to node $\sigma(i)$.

2. Packet v_i travels from $\sigma(i)$ to destination $d(i)$.

The method that the routes use for each phase is the *bit-fixing* routing strategy. Its description is as follows. To go from i to $\sigma(i)$: one scans the bits of $\sigma(i)$ from left to right, and compares them with i. One sends v_i out of the current node along the edge corresponding to the left-most bit in which the current position and $\sigma(i)$ differ. Thus going from (1011) to (0000), the packet would pass through (0011) and then (0001).

Theorem 8 *Given a* PARALLEL-ROUTING *instance* (G, d)*, there is a set of intermediate destinations* $\sigma : \mathbb{N} \to \{0,1\}^n$ *for each* i *such that every packet* i *using* $\sigma(i)$ *and the* TWO-PHASE *algorithm reaches its destination in at most* $14n$ *steps and* $\mathbf{K}(\sigma) <^{\log} \mathbf{I}(\langle G, d\rangle; \mathcal{H})$.

Proof. By Theorem 47 in [MR95], if the intermediate destinations are chosen randomly, with probability least $1 - (1/N)$, every packet reaches its destination in $14n$ or fewer steps. Let $D \subset \{0,1\}^{Nn}$ be the set of all intermediate destinations $\sigma \in D$ such the lag time of instance (G, d) using σ is $\leq 14n$. Thus $\mu(D) > 0.5$, where μ is the uniform measure over the Cantor set. $\mathbf{K}(D|(G,d) = O(1)$. Theorem 3 and Lemma 1 results in

$$\mathbf{Km}(D) <^{\log} -\log \mathbf{M}(D) + \mathbf{I}(\langle D\rangle; \mathcal{H}) <^{\log} -\log \mu(D) + \mathbf{I}((G,d); \mathcal{H}) <^{\log} \mathbf{I}((G,d); \mathcal{H}).$$

Thus using $y \sqsupseteq x \in D$ that realizes $\mathbf{Km}(D)$, one can construct a function $\sigma : \mathbb{N} \to \{0,1\}^n$ which produces the desired intermediate destinations, and $\mathbf{K}(\sigma) <^{+} \mathbf{K}(y) <^{\log} \mathbf{I}((G,d); \mathcal{H})$.

2.6 CLASSIFICATION

In machine learning, CLASSIFICATION is the task of learning a binary function c from $\mathbb{N}$ to bits $\{0,1\}$. The learner is given a sample consisting of pairs (x, b) for string x and bit b and outputs a binary classifier $h : \mathbb{N} \to \{0,1\}$ that should match c as much as possible. Occam's razor says that "the simplest explanation is usually the best one." Simple hypothesis are resilient against overfitting to the sample data. In The question is, given a particular problem in machine learning, how simple can the hypotheses be?

We use a probabilistic model. The target concept is modeled by a random variable $\mathcal{X}$ with distribution p over ordered lists of natural numbers. The random variable $\mathcal{Y}$ models the labels, and has a distribution over lists of bits, where the distribution of $\mathcal{X} \times \mathcal{Y}$ is $p(x, y)$ with conditional probability requirement $p(y|x) = \prod_{i=1..|x|} p(y_i|x_i)$. Each such (x_i, y_i) is a labeled sample. A binary classifier f is consistent with labelled samples (x, y), if for all i, $f(x_i) = y_i$. Let $\Gamma(x, y)$ be the minimum Kolmogorov complexity of a classifier consistent with (x, y). $\mathcal{H}(\mathcal{Y}|\mathcal{X})$ is the conditional entropy of $\mathcal{Y}$ given $\mathcal{X}$.

Theorem 9

1. $\mathcal{H}(\mathcal{Y}|\mathcal{X}) \leq \mathbf{E}[\Gamma(\mathcal{X}, \mathcal{Y})] <^{\log} \mathcal{H}(\mathcal{Y}|\mathcal{X}) + \mathbf{K}(p)$.
2. *For each* $c, b \in \mathbb{N}$*, there exists random labeled samples* $\mathcal{X} \times \mathcal{Y}$ *with distribution* p*, such that, up to precision* $O(\log cb)$*,* $\mathbf{E}[\Gamma(\mathcal{X}, \mathcal{Y})] = b + c$*,* $\mathcal{H}(\mathcal{Y}|\mathcal{X}) = b$*, and* $\mathbf{K}(p) = c$.

Proof. We start with the lower bound of part 1. $\mathbf{E}[\Gamma(\mathcal{X},\mathcal{Y})] = \sum_x p(x)\sum_y p(y|x)\Gamma(x,y)$. Each $\Gamma(x,y)$ represents a self-delimiting program to compute a classifier f such that $f(x_i) = y_i$. Thus if $y \neq y'$, $\Gamma(x,y)$ and $\Gamma(x,y')$ represents two programs v and v' such that $v \not\sqsubseteq v'$ and $v' \not\sqsubseteq v$. Thus for a fixed x, ranged over y, $\Gamma(x,y)$ represents the length of a self-delimiting code. Due to properties of conditional entropy, which is minimal over all self-delimiting codes,

$$\mathbf{E}[\Gamma(\mathcal{X},\mathcal{Y})] = \sum_x p(x)\sum_y p(y|x)\Gamma(x,y) \geq \sum_x p(x)\sum_y p(y|x)(-\log p(y|x)) = \mathcal{H}(\mathcal{Y}|\mathcal{X}).$$

We now prove the upper bound of part 2. To do so, we need the following lemma. The following lemma is perhaps surprising because it shows that the $\mathbf{I}(\cdot;\mathcal{H})$ terms in inequalities can be removed by averaging over a computable probability.

Lemma 2 *For computable probability p, $\sum_x p(x)\mathbf{I}(x;\mathcal{H}) <^+ \mathbf{K}(p)$.*

Proof. This follows from Theorem 3.1.3 in [G2́1], and we will reproduce its arguments. Since $\mathbf{K}(x/\mathcal{H})$ is the length of a self delimiting code,

$$\sum_x p(x)\mathbf{K}(x/\mathcal{H}) \geq \mathcal{H}(p),$$

where $\mathcal{H}(p)$ is the entropy of p. Furthermore, for all $x \in \{0,1\}^*$, $\mathbf{K}(x) <^+ -\log p(x) + \mathbf{K}(p)$. Therefore

$$\sum_x p(x)\mathbf{K}(x) <^+ \sum_x p(x)(-\log p(x)) + \mathbf{K}(p) <^+ \mathcal{H}(p) + \mathbf{K}(p).$$

So

$$\sum_x p(x)\mathbf{I}(x;\mathcal{H}) = \sum_x p(x)\left(\mathbf{K}(x) - \mathbf{K}(x/\mathcal{H})\right) <^+ \mathcal{H}(p) + \mathbf{K}(p) - \sum_x p(x)\mathbf{K}(x/\mathcal{H}) <^+ \mathbf{K}(p).$$

□

Binary classifiers are identified by infinite sequences $\alpha \in \{0,1\}^\infty$. We define the computable measure $S : \{0,1\}^* \to \mathbb{R}_{\geq 0}$ over $\{0,1\}^\infty$, where $S(x) = \prod_{n=1..|x|} p(x_n|n)$, where $\mathbf{K}(S|p) = O(1)$. Let $\{(x_i, y_i)\}$ be a set of labelled samples and we define clopen set $C_{x,y} = \{\alpha : \alpha \in \{0,1\}^\infty, \alpha[x_i] = y_i\}$. Then $S(C_{x,y}) = p(y|x)$. By Theorem 3, relativized to p,

$$\begin{aligned}\min_{\alpha\in C_{x,y}} \mathbf{K}(\alpha|p) &<^{\log} \mathbf{K}(S|p) - \log S(C_{x,y}) + \mathbf{I}(C_{x,y};\mathcal{H}|p)\\ &<^{\log} -\log S(C_{x,y}) + \mathbf{I}(C_{x,y};\mathcal{H}|p)\\ &<^{\log} -\log p(y|x) + \mathbf{I}(C_{x,y};\mathcal{H}|p)\end{aligned}$$

Averaging over all x and y using probability p, one gets

$$\sum_{x,y} p(x,y) \min_{\alpha\in C_{x,y}} \mathbf{K}(\alpha|p) <^{\log} \sum_{x,y} p(x,y)(-\log p(y|x)) + \sum_{x,y} p(x,y)\mathbf{I}(C_{x,y};\mathcal{H}|p). \tag{1}$$

Applying Lemma 2 relative to p, we get

$$\sum_{x,y} p(x,y)\mathbf{I}(\langle C_{x,y}\rangle;\mathcal{H}|p) = \sum_{x,y} p(\langle C_{x,y}\rangle)\mathbf{I}(\langle C_{x,y}\rangle;\mathcal{H}|p) <^+ \mathbf{K}(p|p) = O(1). \tag{2}$$

Combining equations 1 and 2,

$$\sum_{x,y} p(x,y) \min_{\alpha \in C_{x,y}} \mathbf{K}(\alpha|p) <^{\log} \sum_{x,y} p(x,y)(-\log p(y|x))$$

$$\left(\sum_{x,y} p(x,y) \min_{\alpha \in C_{x,y}} \mathbf{K}(\alpha)\right) - \mathbf{K}(p) <^{\log} \sum_{x,y} p(x,y)(-\log p(y|x))$$

$$\mathbf{E}[\Gamma(\mathcal{X},\mathcal{Y})] <^{\log} \mathcal{H}(\mathcal{Y}|\mathcal{X}) + \mathbf{K}(p).$$

We now prove part 2. We ignore all $O(\log cd)$ terms. So equality $=$ is equivalent to $= \pm O(\log cd)$. We define a probability $p(x,y)$ over the first $n = 2c+2b+2$ numbers and corresponding bits. Thus we can describe p as a probability measure over strings of size n, making sure to maintain p's conditional probability restriction described in the introduction.

Let $z \in \{0,1\}^c$ be a random string of size c, with $c <^+ \mathbf{K}(z)$. For all strings $w \in \{0,1\}^b$ of size b, $p(\langle z\rangle\langle w\rangle) = 2^{-b}$, with $\|\langle z\rangle\langle w\rangle\| = n$. $\mathcal{H}(\mathcal{Y}|\mathcal{X}) = -\sum_{w\in\{0,1\}^b} 2^{-b}(\log p(\langle z\rangle\langle w\rangle)) = -\sum_{w\in\{0,1\}^b} 2^{-b}(\log 2^{-b}) = b$. Furthermore $\mathbf{K}(p) = c$. The infinite sequence $\alpha = \langle z\rangle\langle w\rangle 0^\infty$ realizes $\Gamma(\langle z\rangle\langle w\rangle)$ up to an additive constant for each $w \in \{0,1\}^b$. Thus $\mathbf{K}(\alpha) = \mathbf{K}(z,w)$.

$$\mathbf{E}[\Gamma(\mathcal{X},\mathcal{Y})] = 2^{-b} \sum_{w\in\{0,1\}^b} \mathbf{K}(\langle z\rangle\langle w\rangle) = \mathbf{K}(z) + 2^{-b} \sum_{w\in\{0,1\}^b} \mathbf{K}(w/z, \mathbf{K}(z)).$$

Using Theorem 3.1.3 in [G2́1] conditioned on $\langle z, \mathbf{K}(z)\rangle$, we get that $\sum_{w\in\{0,1\}^b} 2^{-b}\mathbf{K}(w/z,\mathbf{K}(z)) = \mathcal{H}(\mathcal{U}_b) \pm \mathbf{K}(b/z,\mathbf{K}(z)) = b$, where $\mathcal{U}_b$ is the uniform measure over strings of size b. So $\mathbf{E}[\Gamma(\mathcal{X},\mathcal{Y})] = \mathbf{K}(z) + b = b + c$.

2.7 Cover-Time

We define the following interactive penalty game. Let $G = (E,V)$ be a graph consisting of n vertices V and undirected edges E. The environment $\mathbf{q}$ consists of (G,s,ℓ). $G = (E,V)$ is a non-bipartite graph with undirected edges, $s \in V$ is the starting vertex. ℓ is a mapping from numbers to edges to be described later.

The agent starts at $s \in V$. At round 1, the environment gives the agent the degree $s \in V$, $\mathrm{Deg}(s)$. The agent picks an number between 1 and $\mathrm{Deg}(s)$ and sends it to $\mathbf{q}$. The agent moves along the edge the number is mapped to and is given the degree of the next vertex it is on. Each round's mapping of numbers to edges, ℓ, is a computable function of the current vertex, round number, and the agent's past actions. The game stops if the agent has visited all vertices and the penalty is the number of turns the agents takes.

Theorem 10 *There is a deterministic agent* $\mathbf{p}$ *that can play against* Cover-Time *instance* (G,S,ℓ), $|G| = n$, *and achieve penalty* $\frac{4}{27}n^3 + o(n^3)$ *and* $\mathbf{K}(\mathbf{p}) <^{\log} \mathbf{I}((G,s,\ell);\mathcal{H})$.

Proof. A probabilistic agent $\mathbf{p}'$ is defined as selecting each edge with equal probability. Thus the agent performs a random walk. The game halts with probability 1. Due to [Fei95], the expected time (i.e. expected penalty) it takes to reach all vertices is $\frac{4}{27}n^3 + o(n^3)$. Thus by Corollary 3 there is a deterministic agent $\mathbf{p}$ that can reach each vertex with a penalty of $\frac{8}{27}n^3 + o(n^3)$ and has complexity

$$\mathbf{K}(\mathbf{p}) <^{\log} \mathbf{K}(\mathbf{p}') + \mathbf{I}((G,s);\mathcal{H}) <^{\log} \mathbf{I}((G,s,\ell);\mathcal{H}).$$

2.8 Super-Set

Given a finite set $S \subseteq \{0,1\}^n$, the goal of Super-Set is to find a set $T \supseteq S$, $T \subseteq \{0,1\}^n$ that minimizes $|T|$.

Theorem 11 *Given $m \leq n$, $S \subseteq \{0,1\}^n$, $|S| < 2^{n-m-1}$ there exists a $T \supseteq S$, $T \subseteq \{0,1\}^n$ $|T| = 2^{n-m}$, $\mathbf{K}(T) <^{\log} \mathbf{K}(n,m) + (m+1)|S| + \mathbf{I}(\langle S, m\rangle; \mathcal{H})$.*

Proof. Let $P : \{0,1\}^* \to \mathbb{R}_{\geq 0}$ be the the uniform distribution over all sequences of size 2^n that have exactly 2^{n-m} 1s. Let $D \subset \{0,1\}^{2^n}$ consist of all sequences $x_R \in \{0,1\}^{2^n}$ that encode sets $R \subseteq \{0,1\}^n$ in the natural way such that $R \supseteq S$ and $|R| = 2^{m-n}$. Thus if $x \in D$ then x has 2^{n-m} 1s. $P(D) =$

$$\left(\frac{2^{n-m}}{2^n}\right)\left(\frac{2^{n-m-1}}{2^n - 1}\right)\cdots\left(\frac{2^{n-m} - |S|}{2^n - |S|}\right) \geq \left(\frac{2^{n-m} - |S|}{2^n - |S|}\right)^{|S|} \geq \left(\frac{2^{n-m-1}}{2^n}\right)^{|S|} = 2^{-(m+1)|S|}.$$

$\mathbf{K}(D|\langle S, m\rangle) = O(1)$. Thus by Theorem 1, there exists a $t \in D$, such that $\mathbf{K}(t) <^{\log} \mathbf{K}(P) - \log P(D) + \mathbf{I}(D; \mathcal{H}) <^{\log} \mathbf{K}(n,m) + (m+1)|S| + \mathbf{I}(\langle S, m\rangle; \mathcal{H})$. This t encodes a set $T \supseteq S$, $T \subseteq \{0,1\}^n$ such tha $|T| = 2^{n-m}$.

□

2.9 Even-Odds

The sum of $\sqrt{n}\sum_{i=1}^n X_i \sim \mathcal{N}(0,1)$. $\Phi(x) > \frac{1}{2\pi}\frac{x}{x^2+1}e^{-x^2/2}$. $\Phi(1) = (1/4\pi)e^{-1/2}?1/8\pi$.

We define the following win/no-halt game, entitled Even-Odds-N. There are N rounds. At round 1, the enviroment $\mathbf{q}$ secretly records bit $e_1 \in \{0,1\}$. It sends an empty message to the agent who responds with bit $a_1 \in \{0,1\}$. The agent gets a point if $e_1 \oplus b_1 = 1$. Otherwise the agent loses a point. For round i, the environment selects a bit b_i that is a computable function of the previous agent's actions $\{a_j\}_{j=1}^{i-1}$ and sends an empty message to the agent, which responds in a_i and the agent gets a point if $e_i \oplus b_i = 1$, otherwise it loses a point. The agent wins after N rounds if it has a score of at least $\sqrt{N}$.

Theorem 12 *For large enough N, there is a deterministic agent $\mathbf{p}$ that can win* Even-Odds-N *with complexity $\mathbf{K}(\mathbf{p}) <^{\log} \mathbf{I}(\mathbf{q}; \mathcal{H})$.*

Proof. We describe a probabilistic agent $\mathbf{p}'$. At round i, $\mathbf{p}'$ submits 0 with probability 1/2. Otherwise it submits 1. By the central limit theorem, for large enough N, the score of the probabilistic agent divided by $\sqrt{N}$ is $S \sim \mathcal{N}(0,1)$. Let $\Phi(x) > \Pr[S > x]$. A common bound for $\Phi(x)$ is

$$\Phi(x) > \frac{1}{2\pi}\frac{x}{x^2+1}e^{-x^2/2}$$
$$\Phi(1) > \frac{1}{4\pi}e^{-1/2} > \frac{1}{8\pi}.$$

Thus when $S \geq 1$, the score is at least $\sqrt{N}$. Thus $\mathbf{p}'$ wins with probability at least $p = \frac{1}{8\pi}$. Thus by Theorem 4, there exists a deterministic agent $\mathbf{p}$ that can beat $\mathbf{q}$ with complexity

$$\mathbf{K}(\mathbf{p}) <^{\log} \mathbf{K}(\mathbf{p}') - \log p + \mathbf{I}(\langle p, \mathbf{p}', \mathbf{q}\rangle; \mathcal{H}) <^{\log} \mathbf{I}(\mathbf{q}; \mathcal{H}).$$

□

2.10 Binary-Matrix

For a vector $v = (v_1, \ldots, v_n) \in \mathbb{R}^n$, $\|v\|_\infty = \max_i |V_i|$. Binary matrix M is a matrix whose values are either 0 or 1. The goal of Binary Matrix, is given M, to find a vector $b \in \{-1,+1\}^n$ that minimizes $\|Mb\|_\infty$.

Theorem 13 *Given $n \times n$ binary matrix M, there is a vector $b = \{-1,+1\}^n$ such that $\|Mb\|_\infty \leq 4\sqrt{n \ln n}$ and $\mathbf{K}(b) <^{\log} \mathbf{K}(n) + \mathbf{I}(\langle M \rangle; \mathcal{H})$.*

Proof. Let $v = (v_1, \ldots, v_n)$ be a row of M. Choose a random $b = (b_1, \ldots, b_n) \in \{-1,+1\}^n$. Let $i_1, \ldots, i_m$ be the indices such that $v_{i_j} = 1$. Thus

$$Y = \langle v, b \rangle = \sum_{i=1}^n v_i b_i = \sum_{j=1}^m v_{i_j} b_{i_j} = \sum_{j=1}^m b_{i_j}.$$

$$\mathbf{E}[Y] = \mathbf{E}[\langle v, b \rangle] = \mathbf{E}\left[\sum_i v_i b_i\right] = \sum_i \mathbf{E}[v_i b_i] = \sum v_i \mathbf{E}[b_i] = 0.$$

By the Chernoff inequality and the symmetry Y, for $\tau = 4\sqrt{n \ln n}$,

$$\Pr[|Y| \geq \tau] = 2\Pr[v \cdot b \geq \tau] = 2\Pr\left[\sum_{j=1}^m b_{i_j} \geq \tau\right] \leq 2\exp\left(-\frac{\tau^2}{2m}\right) = 2\exp\left(-8\frac{n \ln n}{m}\right) \leq 2n^{-8}.$$

Thus, the probability that any entry in Mb exceeds $4\sqrt{n \ln n}$ is smaller than $2n^{-7}$. Thus, with probability $1 - 2n^{-7}$, all the entries of Mb have value smaller than $4\sqrt{n \ln n}$.

Let $P : \{0,1\}^* \to \mathbb{R}_{\geq 0}$, be the uniform measure over string of length n, with $P(x) = [\|x\| = n]2^{-n}$. Let D consist of all strings that encode vectors $b_x \in \{-1,+1\}^n$ in the natural way such that $\|Mb_x\|_\infty \leq 4\sqrt{n \ln n}$. $\mathbf{K}(b|M) = O(1)$. By Theorem 1 and Lemma 1, there exists an $x \in D$, such that

$$\mathbf{K}(x) <^{\log} \mathbf{K}(P) - \log P(D) + \mathbf{I}(D; \mathcal{H}) <^{\log} \mathbf{K}(n) + \mathbf{I}(M; \mathcal{H}).$$

Thus there exists a $b_x \in \{-1,+1\}^n$ that satisfies the theorem statement.

2.11 Min-Cut

We define the following win/no-halt game, entitled Min-Cut. The game is defined by an undirected graph G and a mapping ℓ from numbers to edges. At round i, the enviroment $\mathbf{q}$ sends the number of edges of G. The player responds with a number. The environment maps the number to an edge, and this mapping can be a function of the round number and player's previous actions. The enviroment then contracts the graph G along the edge. The game halts when the graph G has contracted into two vertices. The player wins if the cut represented by the contractions is a min cut

Theorem 14 *There is a deterministic agent $\mathbf{p}$ that can play against* Cover-Time *instance (G, S, ℓ), $|G| = n$, such that $\mathbf{K}(\mathbf{p}) <^{\log} 2\log n + \mathbf{I}((G, \ell); \mathcal{H})$.*

Proof. We define the following randomized agent $\mathbf{p}'$. At each round, $\mathbf{p}'$ chooses an edge at random. Thus the interactions of $\mathbf{p}'$ and $\mathbf{q}$ represent an implementation of Karger's algorithm. Karger's algorithm has an $\Omega(1/n^2)$ probabability of returning a min-cut. Thus $\mathbf{p}'$ has an $\Omega(1/n^2)$ chance of winning. By Theorem 4, there exist a deterministic agent $\mathbf{p}$ and c where $\mathbf{p}$ can beat $\mathbf{q}$ and has complexity $\mathbf{K}(\mathbf{p}) <^{\log} \mathbf{K}(p') - \log c/n^2 + \mathbf{I}(\langle \mathbf{q} \rangle; \mathcal{H}) <^{\log} 2\log n + \mathbf{I}((G, \ell); \mathcal{H})$.

2.12 Hypergraph-Coloring

In this section we show how to compress colorings of k-uniform hypergraph. A *hypergraph* is a pair $J = (V, E)$ of vertices V and edges $E \subseteq \mathcal{P}(V)$. Thus each edge can connect ≥ 2 vertices. A hypergraph is *k-uniform* of the size $|e| = k$ for all edges $e \in E$. A 2-uniform hypergraph is just a simple graph. A valid *C-coloring* f a hypergraph (V, E) is a mapping $f : V \to |C|$ where every edge $e \in E$ is not *monochromatic* $|\{f(v) : v \in e\}| > 1$. The goal of Hypergraph-Coloring-k is given a k uniform hypergraph, produce a coloring using the smallest amount of colors.

Theorem 15 *Every k-uniform hypgraph $J = (V, E)$, $|E| = n$, $|V| = m$ has a $\lceil \sqrt[k-1]{2m} \rceil$ coloring g where $\mathbf{K}(g) <^{\log} \mathbf{K}(k, n, m) + \mathbf{I}(J; \mathcal{H})$.*

Proof. We randomly color every vertex $v \in V$ using $C = \lceil \sqrt[k-1]{2m} \rceil$ colors. Let A_e be the bad event that edge e is monochromatic. This event has probability:

$$\Pr[A_e] = C \cdot (1/C)^k = (1/C)^{k-1} < 1/2m,$$

because there are C possible colors and each vertex has a $1/C$ chance of getting a particular color. We can get a union-bound over all m edges to find the bad probability.

$$\Pr\left[\bigcup_{e \in E} A_e\right] < \sum_{e \in E} \Pr[A_e] < m \cdot (1/2m) = 1/2.$$

We Let $D \subset \{0,1\}^{v\lceil logC \rceil}$ be the set of all encodings of C colorings (so no edge in monochromatic. $\mathbf{K}(D|J) = O(1)$. Let $P : \{0,1\}^* \to \mathbb{R}_{\geq 0}$ be a probability measure over $\{0,1\}^*$, uniformly distributed over all $x \in BT^{v\lceil \log C \rceil}$ that encode a C-coloring. $P(D) > .5$. By Theorem 1 and Lemma 1, there is a graph coloring $g \in D$ where

$$\mathbf{K}(g) <^{\log} \mathbf{K}(P) - \log P(D) + \mathbf{I}(D; \mathcal{H}) <^{\log} \mathbf{K}(k, m, n) + \mathbf{I}(J; \mathcal{H}).$$

References

[Blo70] B. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, page 422–426, 1970.

[Eps19] S. Epstein. On the algorithmic probability of sets. *CoRR*, abs/1907.04776, 2019.

[Fei95] U Feige. A tight upper bound on the cover time for random walks on graphs. *Random Struct. Algorithms*, 6(1):51–54, 1995.

[G2́1] Peter Gács. Lecture notes on descriptional complexity and randomness. *CoRR*, abs/2105.04704, 2021.

[Lev16] L. A. Levin. Occam bound on lowest complexity of elements. *Annals of Pure and Applied Logic*, 167(10):897–900, 2016. And also: S. Epstein and L.A. Levin, Sets have simple members, arXiv preprint arXiv:1107.1458, 2011.

[MR95] R Motwani and P Raghavan. *Randomized Algorithms.* Cambridge University Press, Cambridge; NY, 1995.

[MU05] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, 2005.