

Worst-Case Running Times for Average-Case Algorithms

Luís Antunes
Faculdade de Ciências da U. Porto
Instituto de telecomunicações
Porto, Portugal
Email: lfa@dcc.fc.up.pt

Lance Fortnow
Northwestern University
EECS Department
Evanston, Illinois
Email: fortnow@eecs.northwestern.edu

Abstract—Under a standard hardness assumption we exactly characterize the worst-case running time of languages that are in average polynomial-time over all polynomial-time samplable distributions.

More precisely we show that if exponential time is not infinitely often in subexponential space, then the following are equivalent for any algorithm A :

- For all P-samplable distributions μ , A runs in time polynomial on μ -average.
- For all polynomial p , the running time for A is bounded by $2^{O(K^p(x) - K(x) + \log(|x|))}$ for all inputs x .

where $K(x)$ is the Kolmogorov complexity (size of smallest program generating x) and $K^p(x)$ is the size of the smallest program generating x within time $p(|x|)$.

To prove this result we show that, under the hardness assumption, the polynomial-time Kolmogorov distribution, $m^p(x) = 2^{-K^p(x)}$, is universal among the P-samplable distributions.

Index Terms—Kolmogorov Complexity, Average Polynomial Time

I. INTRODUCTION

Computation complexity usually considers the worst-case running time of an algorithm. In 1986, Levin [1] defined a notion of polynomial-time on average for a distribution μ . We give a tight characterization of the worst-case running time for problems solvable in polynomial-time on average over all polynomial-time samplable distributions.

We characterize the running time using the computational depth of the input. Computational depth was defined by Antunes, Fortnow and van Melkebeek [2] as the difference between the time-bounded and traditional Kolmogorov complexities for that input, i.e., $D^p(x) = K^p(x) - K(x)$ where we typically consider polynomial time bounds $p(|x|)$. Random strings have values for $K(x)$ and $K^p(x)$ both close to $|x|$ so the depth is small. Simple strings like 1^n or the first n digits of π have both $K(x)$ and $K^p(x)$ about $\log n$ so the depth is small. Antunes et. al. [2] show by construction that there exist an exponential number of strings of linear depth.

Informally we can state our main result as follows: If time does not have non-trivial space simulations then the following are equivalent for all algorithms A .

- A runs in time polynomial on average over all efficiently samplable distributions.
- For all x , A runs in time exponential in the depth of x .

Note that the running time for input x is a property of x and (up to polynomial factors) independent of the particular algorithm.

More formally if there exists a language L in $\text{DTIME}(2^{O(n)})$ where L does not have circuits of size $2^{o(n)}$ with Σ_2^p gates then the following are equivalent for any algorithm A :

- For all P-samplable distributions μ , A runs in time polynomial on μ -average.
- For all polynomial p , the running time for A is bounded by $2^{O(K^p(x) - K(x) + \log |x|)}$ for all inputs x .

Peter Bro Miltersen [3] shows that if $\text{DTIME}(2^{O(n)})$ is not contained in $\text{DSPACE}(2^{o(n)})$ then $\text{DTIME}(2^{O(n)})$ contains a language that does not have circuits of size $2^{o(n)}$ with Σ_2^p gates or even PSPACE gates.

To prove our main theorem we show the relationship between the polynomial-time bounded Kolmogorov distribution and the P-samplable distributions.

Levin [4] and Gcs [5] considered a semi-measure defined by $m(x) = 2^{-K(x)}$ where $K(x)$ is the prefix-free Kolmogorov complexity of x . They showed that $m(x)$ is universal among the semi-computable semi-measures.

In this paper we consider the polynomial-time bounded variation $m^p(x) = 2^{-K^p(x)}$. This measure sits somewhere between the polynomial-time computable and polynomial-time samplable distributions: Every polynomial-time computable distribution τ is dominated by m^p for some polynomial p , and for all polynomials q , m^q is dominated by a polynomial-time samplable distribution.

We show that under reasonable hardness assumptions $m^p(x)$ is universal among the polynomial-time samplable distributions, specifically

- 1) For any polynomial p , there is a polynomial-time samplable distribution τ and a constant c such that $\tau(x) \geq cm^p(x)$ for every x .
- 2) If $\mathbf{E} = \text{DTIME}(2^{O(n)})$ does not have size $2^{o(n)}$ circuits with Σ_2^p gates then for every polynomial-time samplable distribution τ there is a polynomial p such that $m^p(x) \geq \tau(x)/|x|^{O(1)}$.

The first item was proved earlier by Antunes, Fortnow and Vinodchandran [6] with sketch given in Section II-B. Our paper proves the second item.

Our proof uses another result by Antunes, Fortnow and Vinodchandran [6] relating worst-case running time with average case over the m^p distribution. In Section II-B we describe and, for completeness, give the proof of this result. We also use pseudorandom generators based on hard functions, specifically the work of Nisan and Wigderson [7], Impagliazzo and Wigderson [8] and Klivans and van Melkebeek [9].

In Section II we give background on Kolmogorov complexity, average-case complexity and pseudorandom generators. In Section III we show how to use the universality of m^p to give the characterization of worst-case complexity. In Section IV we give the proof of the universality of m^p .

II. PRELIMINARIES

We use binary alphabet $\Sigma = \{0, 1\}$ for encoding strings. Our computation model will be *prefix-free* Turing machines: Turing machines with a one-way input tape (the input head can only read from left to right), a one-way output tape and a two-way work tape. The function \log denote \log_2 . All explicit resource bounds we use in this paper are time-constructible.

A. Kolmogorov Complexity

We give essential definitions and basic result in Kolmogorov complexity and refer the reader to the textbook by Li and Vitnyi [10] for more details. Our main result, Theorem 3.1 holds for any reasonable model of Kolmogorov complexity but to be specific we will focus on prefix-free Kolmogorov complexity. A set of strings A is prefix-free if there are no strings x and y in A where x is a proper prefix of y . Kraft's inequality states that for any prefix-free set A

$$\sum_{x \in A} 2^{-|x|} \leq 1.$$

Definition 2.1: Let U be a fixed Turing machine with a prefix-free domain. Then for any string $x, y \in \{0, 1\}^*$, the Kolmogorov complexity of x given y is, $K(x|y) = \min_p \{|p| : U(p, y) = x\}$.

For any time constructible t , the t -time-bounded Kolmogorov complexity of x given y is, $K^t(x|y) = \min\{|p| : U(p, y) = x \text{ in at most } t(|x|) \text{ steps}\}$.

The default value for y is the empty string ϵ , we typically drop this argument in the notation. We can fix a universal Turing machine U whose program size $|p|$ is at most a constant additive factor worse, and the running time t at most a logarithmic multiplicative factor.

A function $f : S \rightarrow [0, 1]$ is called a *semi-measure* over the space S if

$$\sum_x f(x) \leq 1.$$

It is called a *measure* if equality holds. A semi-measure is called *constructive* if it is semi-computable from below, i.e., there is a computable function $g(x, t)$ monotone in t such that $f(x) = \lim_{t \rightarrow \infty} g(x, t)$.

Levin [11] and Gcs [5] show that the function $m(x) = 2^{-K(x)}$ is both a constructive semi-measure and for all other constructive semi-measures μ , there is a constant c_μ such that

for all x , $m(x) \geq c_\mu \mu(x)$. For this reason we often call $m(x)$ a universal distribution.

We can define a time bounded version of $m(x)$.

Definition 2.2: The t -time bounded universal distribution, m^t is given by $m^t(x) = 2^{-K^t(x)}$.

B. Average-Case Complexity

We give definitions from average case complexity theory necessary for our purposes. For more details readers can refer to the survey by Wang [12]. In average case complexity theory, a computational problem is a pair (L, μ) where $L \subseteq \Sigma^*$ and μ is a probability distribution. Levin [1] defined a notion of *polynomial on average* that is central to the theory of average case completeness.

Definition 2.3 (Levin): Let μ be a probability distribution function on $\{0, 1\}^*$. A function $f : \Sigma^+ \rightarrow \mathbf{N}$ is polynomial on μ -average if there exists an $\epsilon > 0$ such that $\sum_x \frac{f(x)^\epsilon}{|x|} \mu(x) < \infty$.

We can replace ∞ with 1 in Definition 2.3. One can also generalize the definition in the obvious way to allow μ to be a semi-measure.

Definition 2.4: Let μ be a probability distribution and $L \subseteq \Sigma^*$. Then the pair (L, μ) is in Average Polynomial time (denoted as Avg-P) if there is a Turing machine accepting L whose running time is polynomial on μ -average.

In this paper we consider average-case complexity over easily samplable distributions. Ben-David *et al.* in [13] introduced a family of natural distributions, P-samplable, consisting of distributions that can be sampled by randomized algorithms, working in time polynomial in the length of the sample generated.

Definition 2.5: A probability distribution σ on $\{0, 1\}^*$ is P-Samplable, if there is a polynomial-time computable function $F : \Sigma^m \rightarrow \Sigma^n$ with $n \geq m^{\Omega(1)}$ such that for y of length n ,

$$\sigma(y) = \frac{|\{w \in \Sigma^m \mid F(w) = y\}|}{2^m}.$$

We need the notion of *domination* for comparing distributions. The next definition formalizes this notion.

Definition 2.6: Let μ and ν be two distributions on Σ^* . Then μ *dominates* ν if there is a constant c such that for all $x \in \Sigma^*$, $\mu(x) \geq \frac{1}{|x|^c} \nu(x)$. We also say ν is dominated by μ .

Antunes, Fortnow and Vinodchandran [6] show that P-samplable distributions dominate the polynomial-time bounded universal distribution.

Theorem 2.7 (Antunes-Fortnow-Vinodchandran): For any polynomial p , there is a P-samplable distribution μ which dominates m^p .

Antunes, Fortnow and Vinodchandran also show a connection between average-case complexity and m^t .

Theorem 2.8 (Antunes-Fortnow-Vinodchandran): Let T be a computable time bound. Then for any computable t , the following statements are equivalent.

- 1) $T(x) \in 2^{O(K^t(x) - K(x))} |x|^{O(1)}$.
- 2) T is polynomial on m^t -average.

Our main result (Theorem 3.1) shows that under a reasonable hardness assumption one can replace m^t with P-samplable distributions in Theorem 2.8. Also, combining both theorems we get a new interpretation of universality in the sense that T is polynomial in the average for all P-samplable distributions if and only if T is polynomial on the average for all m^p (i.e., for every polynomial p).

For completeness we provide a proof of Theorem 2.8.

Proof: (1 \Rightarrow 2). We will show that the statement 1 implies that $T(x)$ is polynomial on m^t -average. Let $T(x) \in 2^{O(K^t(x)-K(x))}|x|^{O(1)}$. Because of the closure properties of functions which are polynomial on average, it is enough to show that the function $T'(x) = 2^{K^t(x)-K(x)}$ is polynomial on m^t -average. Consider the sum

$$\begin{aligned} \sum_{x \in \Sigma^*} \frac{T'(x)}{|x|} m^t(x) &= \sum_{x \in \Sigma^*} \frac{2^{K^t(x)-K(x)}}{|x|} 2^{-K^t(x)} \\ &= \sum_{x \in \Sigma^*} \frac{2^{-K(x)}}{|x|} \\ &< \sum_{x \in \Sigma^*} 2^{-K(x)} < 1 \end{aligned}$$

(2 \Rightarrow 1) Let $T(x)$ be a computable function which is polynomial on m^t -average. Then for some $\epsilon > 0$ we have

$$\sum_{x \in \Sigma^*} \frac{T(x)^\epsilon}{|x|} m^t(x) < 1$$

Define $S_{i,j,n} = \{x \in \Sigma^n \mid 2^i \leq T(x) < 2^{i+1} \text{ and } K^t(x) = j\}$. Let 2^r be the approximate size of $S_{i,j,n}$. Then the Kolmogorov complexity of elements in $S_{i,j,n}$ is at most r up to an additive $\log n$ factor. The following claim (proof omitted) states this fact more formally.

Claim 2.9: For $i, j \leq n^2$, let $2^r \leq |S_{i,j,n}| < 2^{r+1}$. Then for any $x \in S_{i,j,n}$, $K(x) \leq r + O(\log n)$.

Consider the above sum restricted to elements in $S_{i,j,n}$. Then we have

$$\sum_{x \in S_{i,j,n}} \frac{T(x)^\epsilon}{|x|} m^t(x) < 1$$

$T(x) \geq 2^i$, $m^t(x) = 2^{-j}$ and there are at least 2^r elements in the above sum. Hence the above sum is lower-bounded by the expression $\frac{2^r \cdot 2^{i\epsilon} \cdot 2^{-j}}{|x|}$. This gives us

$$\begin{aligned} 1 &> \sum_{x \in S_{i,j,n}} \frac{T(x)^\epsilon}{|x|} m^t(x) \\ &\geq \frac{2^r \cdot 2^{i\epsilon} \cdot 2^{-j}}{|x|} = 2^{i\epsilon+r-j-\log n} \end{aligned}$$

That is $i\epsilon + r - j - \log n < 0$. From Claim 2.9, it follows that there is a constant d , such that for all $x \in S_{i,j,n}$, $i\epsilon \leq K^t(x) - K(x) + d \log |x|$. Hence $T(x) \leq 2^{i+1} \leq 2^{\frac{d}{\epsilon}(K^t(x)-K(x)+\log |x|)}$. ■

C. Pseudorandom Generators

Pseudorandom generators are efficiently computable functions which stretch a seed into a long string so that for a random input the output looks random for a resource-bounded machine. We need some pseudorandom generators based on hard functions.

The following lemma is implicit in the work of Nisan and Wigderson [7].

Lemma 2.10 (Nisan-Wigderson): Consider a set \mathcal{H} of functions from Σ^ℓ to Σ^n with n bounded by a polynomial in ℓ with the following properties.

- 1) For every ℓ at least $3/4$ of all possible functions mapping Σ^ℓ to Σ^n are in \mathcal{H} .
- 2) For some k , there is a Σ_k^p machine with oracle access to a function H on input 1^ℓ will accept exactly when H is in \mathcal{H} .

Then there is a polynomial-time computable function $H'(x, r)$ with $x \in \Sigma^\ell$ and $|r|$ polynomial in ℓ such that for at least $2/3$ of the possible r , $H_r(x) = H'(x, r)$ is in \mathcal{H} .

Impagliazzo and Wigderson [8] strengthen the work of Nisan and Wigderson to show how to achieve full derandomization based on strong hardness assumptions. Klivans and van Melkebeek [9] generalize Impagliazzo-Wigderson by showing the results hold for relativized worlds in a strong way.

Lemma 2.11 (Impagliazzo-Wigderson, Klivans-van Melkebeek):

For any oracle A , suppose that there are languages in $\mathbf{DTIME}(2^{O(n)})$ that for some $\epsilon > 0$, cannot be computed by circuits of size $2^{\epsilon n}$ with access to an oracle for A . Then there is a k and a pseudorandom generator $g: \Sigma^{k \log n} \rightarrow \Sigma^n$ computable in time polynomial in n such that for all relativizable circuits C

$$\left| \Pr_{s \in \Sigma^{k \log n}} (C^A(g(s)) = 1) - \Pr_{r \in \Sigma^n} (C^A(r) = 1) \right| = o(1).$$

Peter Bro Miltersen [3] makes the following connection to uniform hardness.

Lemma 2.12 (Miltersen): If $\mathbf{DTIME}(2^{O(n)})$ is not contained in $\mathbf{DSPACE}(2^{o(n)})$ for infinitely many input lengths then for every language A in \mathbf{PSPACE} there is some $\epsilon > 0$ such that $\mathbf{DTIME}(2^{O(n)})$ contains a language that does not have circuits of size $2^{\epsilon n}$ with access to A .

Combining the above we have that if $\mathbf{DTIME}(2^{O(n)})$ is not contained in $\mathbf{DSPACE}(2^{o(n)})$ then the pseudorandom generator from Lemma 2.11 holds for any language A in \mathbf{PSPACE} , including the Σ_2^p -complete problems needed in our proofs.

III. MAIN THEOREM

Given the hardness assumption we can characterize the worst-case running time for all languages that are polynomial-time on average on P-samplable distributions.

Theorem 3.1: If \mathbf{E} is not contained in $\mathbf{DSPACE}(2^{\epsilon n})$ for some $\epsilon > 0$ and infinitely many n then the following are equivalent for every language L .

- L is polynomial-time on σ -average for all P-samplable σ .

- For all polynomials p the running time for some Turing machine $M(x)$ accepting L is

$$2^{O(K^p(x) - K(x) + \log |x|)}.$$

For our proof we can use the weaker (but more technical) assumption that \mathbf{E} does not have size $2^{o(n)}$ circuits with Σ_2^p gates. Dieter van Melkebeek [14] showed how to use the even weaker assumption that \mathbf{E} does not have nondeterministic circuits of size $2^{o(n)}$.

We apply Theorem 2.8 to prove Theorem 3.1 to show that, under the appropriate hardness assumption, the distribution m^p is in some sense universal over the P-samplable distribution.

Lemma 3.2: If \mathbf{E} is not contained in $\mathbf{DSPACE}(2^{\epsilon n})$ for some $\epsilon > 0$ and infinitely many n then for every polynomial-time samplable distribution σ , there is a polynomial p such that

$$m^p(x) \geq \frac{\sigma(x)}{|x|^{O(1)}}.$$

We prove Lemma 3.2 in Section IV.

To complete the proof of Theorem 3.1 we also need a technical lemma that if σ dominates τ and t is polynomial time on σ -average then t is polynomial-time on τ average. This lemma has appeared before in average-case complexity, see, for example, exercise 10.15 in [15].

Lemma 3.3: If there is a j such that for all x , $\sigma(x) \geq \tau(x)/|x|^j$ and t is polynomial time on σ -average then t is polynomial-time on τ average.

Proof: By assumption there is a k such that

$$\sum_{x \in \Sigma^*} \frac{t^{1/k}(x)}{|x|} \sigma(x) < \infty$$

Let A be the set of x such that $t(x) \geq |x|^{2jk}$. We have

$$\sum_{x \in \Sigma^*} \frac{t^{1/2jk}(x)}{|x|} \tau(x) = \sum_{x \in A} \frac{t^{1/2jk}(x)}{|x|} \tau(x) + \sum_{x \in \bar{A}} \frac{t^{1/2jk}(x)}{|x|} \tau(x)$$

For the first term we have

$$\begin{aligned} \sum_{x \in A} \frac{t^{1/2jk}(x)}{|x|} \tau(x) &\leq \sum_{x \in A} \frac{t^{1/2jk}(x) |x|^j}{|x|} \sigma(x) \\ &\leq \sum_{x \in A} \frac{t^{1/2jk}(x) t^{1/2k}(x)}{|x|} \sigma(x) \\ &\leq \sum_{x \in \Sigma^*} \frac{t^{1/k}(x)}{|x|} \sigma(x) < \infty. \end{aligned}$$

For the second term we have

$$\begin{aligned} \sum_{x \in \Sigma^* - A} \frac{t^{1/2jk}(x)}{|x|} \tau(x) &\leq \sum_{x \in \Sigma^* - A} \frac{|x|^{2jk/2jk}}{|x|} \tau(x) \\ &\leq \sum_{x \in \Sigma^* - A} \tau(x) \\ &\leq 1. \end{aligned}$$

We can now give the proof of Theorem 3.1.

Proof: Suppose L is polynomial-time on σ -average for all P-samplable σ . Fix a polynomial p . By Theorem 2.7 there is a P-samplable σ that dominates m^p . By Lemma 3.3, L is polynomial-time on m^p average. By Theorem 2.8 the running time of L is bounded by $2^{O(K^p(x) - K(x) + \log |x|)}$.

Now suppose the running time of L is bounded by $2^{O(K^p(x) - K(x) + \log |x|)}$ for all polynomials p . Let σ be a P-samplable distribution. By Lemma 3.2 there is a polynomial p such that $m^p(x) \geq \sigma(x)/|x|^{O(1)}$. By Theorem 2.8, L is polynomial time on m^p average and then by Lemma 3.3, L is polynomial time on σ -average. ■

IV. MAIN LEMMA

In this section we prove Lemma 3.2 that under an appropriate hardness assumption the universal time-bounded distribution dominates the P-samplable distributions. The lemma follows from the following theorem.

Theorem 4.1: Consider $F \in \mathbf{FP}$ such that $F : \Sigma^m \rightarrow \Sigma^n$ and $n \geq m$. Let $T_y = \{w \in \Sigma^m : F(w) = y\}$ and $V_k = \{y : |y| = n \text{ and } |T_y| \geq 2^k\}$. If there exists a language in $\mathbf{DTIME}(2^{O(n)})$ that does not have size $2^{o(n)}$ circuits with Σ_2^p gates then there is a function G computable in time polynomial in m

$$G : \Sigma^{m-k+O(\log n)} \rightarrow \Sigma^m$$

such that for all $y \in V_k$, $\text{range}(G) \cap T_y \neq \emptyset$.

Proof: We show that a random H satisfies the conclusion of Theorem 4.1 and then derandomize.

Since the T_y are pairwise disjoint, $|V_k| \leq 2^m/2^k$. Let $\ell = m - k + c \log n$ for a c to be chosen later. Consider a random $H : \Sigma^\ell \rightarrow \Sigma^m$. For any v in Σ^ℓ , $H(v)$ will hit each T_y for each y in V_k with probability at least 2^{k-m} . By a standard coupon collector argument for an appropriate choice of c , the range of H will intersect each of those T_y .

If H were polynomial-time computable we could just let $H = G$. But not only is H not polynomial-time computable, one needs an exponential number of bits just to describe H . We create a polynomial-time computable G in two steps.

- 1) We create a polynomial-time computable H' such that for most r of polynomial length and for all y in V_k , there will be an v such that $H'(v, r)$ is in T_y .
- 2) Using the assumption we derandomize $H'(v, r)$ to get a $G(v, s)$, $|s| = O(\log n)$ such that for all y for most s there is a v such that $G(v, s) = y$.

The total input length of G will be $m - k + O(\log n)$ and we will have proven Theorem 4.1.

We will use the Nisan-Wigderson Lemma 2.10 for the first step. Given y it is a simple \mathbf{NP}^H verification to see if the range of H intersects T_y . However to determine if T_y is in V_k , if $|T_y| \geq 2^k$ could be $\#P$ -complete. However, Stockmeyer [16] shows we can approximate $|T_y|$ in $\mathbf{P}^{\Sigma_2^p}$ and thus have a set A in $\mathbf{P}^{\Sigma_2^p}$ such that every y in V_k is in A and every y in A is in V_{k-1} . By adjusting c above the coupon collector argument shows the range of most H will intersect every T_y with y in V_{k-1} . ■

So we define \mathcal{H} to be the set of H that intersect T_y with y in A . We can verify whether H is in \mathcal{H} in $\mathbb{P}^{\Sigma_2^p}$ with an oracle for H and the existence of H' follows from Lemma 2.10.

For the second step we use the Klivans-van-Melkebeek derandomization Lemma 2.11. Consider the set B of strings r such that $H'(v, r)$ is in \mathcal{H} . By the same argument as above we can verify whether r is in B in $\mathbb{P}^{\Sigma_2^p}$. The existence of G follows now from Lemma 2.11. ■

We can now prove Lemma 3.2.

Proof:

Let σ be P-sampleable, i.e. there is a polynomial-time computable function $F : \Sigma^m \rightarrow \Sigma^n$ with $n \geq m^{\Omega(1)}$ such that for y of length n ,

$$\sigma(y) = \frac{|\{w \in \Sigma^m \mid F(w) = y\}|}{2^m}.$$

Fix a y and pick k maximal such that $\sigma(y) \geq 2^{k-m}$. Let T_y be the set of w such that $F(w) = y$ and note $|T_y| \geq 2^k$.

By the assumption, Lemma 2.12 and Theorem 4.1 we have a function G computable in time polynomial in m

$$G : \Sigma^{m-k+O(\log n)} \rightarrow \Sigma^m$$

such that for some s , $G(s)$ is in T_y , i.e., $F(G(s)) = y$. For some polynomial p we have

$$K^p(y) \leq |s| + O(\log n) = m - k + O(\log n)$$

and

$$m^p(y) = 2^{-K^p(y)} \geq 2^{k-m}/n^{O(1)} \geq \sigma(y)/n^{O(1)}.$$
■

V. CONCLUDING REMARKS

Eric Allender suggested defining a version of m^t using Kolmogorov measures that build the time into the complexity instead of as a parameter. We can define $mt(x) = 2^{-Kt(x)}$ and $mT(x) = 2^{-KT(x)}$ where $Kt(x) = \min_p\{|p| + \log t \mid U(p) \text{ outputs } x \text{ in } t \text{ steps}\}$ and $KT(x) = \min_p\{|p| + t \mid \text{for all } i, 1 \leq i \leq |x|, U(p, i) \text{ outputs the } i\text{th bit of } x \text{ in } t \text{ steps}\}$.

Lemma 3.2 holds for mt without the need for a polynomial t , since for all functions t , $mt(x) \geq m^t(x)/t(|x|)$. However we do not know if mt is dominated by a samplable distribution.

One can build a samplable distribution that dominates mT but we do not know if Lemma 3.2 holds for that distribution.

ACKNOWLEDGMENTS

We thank Eric Allender, Dieter van Melkebeek, Peter Bro Miltersen and Paul Vitányi for helpful discussions and comments.

REFERENCES

- [1] L. Levin, "Average case complete problems," *SIAM J. Computing*, vol. 15, no. 1, pp. 285–286, Feb. 1986.
- [2] L. Antunes, L. Fortnow, and D. V. Melkebeek, "Computational depth," in *CCC '01: Proceedings of the 16th Annual Conference on Computational Complexity*. Washington, DC, USA: IEEE Computer Society, 2001, p. 266.
- [3] P. Miltersen, "Derandomizing complexity classes," in *Handbook of Randomized Computing*. Kluwer, 2001.
- [4] L. Levin, "On the notion of a random sequence," *Soviet Math. Dokl.*, vol. 14, pp. 1413–1416, 1973.
- [5] P. Gacs, "On the symmetry of algorithmic information," *Soviet Math. Dokl.*, vol. 15, pp. 1477–1480, 1974.
- [6] L. Antunes, L. Fortnow, and V. Vinodchandran, "Using depth to capture average-case complexity," in *Fundamentals of Computation Theory, 14th International Symposium, FCT*, ser. Lecture Notes in Computer Science, vol. 2751. Springer, 2003, pp. 303–310.
- [7] N. Nisan and A. Wigderson, "Hardness vs. randomness," *Journal of Computer and System Sciences*, vol. 49, pp. 149–167, 1994.
- [8] R. Impagliazzo and A. Wigderson, "P=BPP unless E has subexponential circuits: derandomizing the xor lemma," in *Proceedings of the 29th STOC*, 1997, pp. 220–229.
- [9] A. R. Klivans and D. V. Melkebeek, "Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses," in *SIAM Journal on Computing*, 1999, pp. 659–667.
- [10] M. Li and P. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 1997.
- [11] L. Levin, "Laws of information conservation (nongrowth) and aspects of the foundation of probability theory," *Probl. Inform. Transm.*, vol. 10, pp. 206–210, 1974.
- [12] J. Wang, "Average-case computational complexity theory," in *Complexity Theory Retrospective II*, A. Selman and L. Hemaspaandra, Eds. Springer, 1997, pp. 295–328.
- [13] S. Ben-David, B. Chor, O. Goldreich, and M. Luby, "On the theory of average case complexity," *Journal of Computer and System Sciences*, vol. 44(2), pp. 193–219, 1992.
- [14] D. van Melkebeek, 2005, personal Communication.
- [15] O. Goldreich, *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, April 2008.
- [16] L. Stockmeyer, "On approximation algorithms for #P," vol. 14, no. 4, pp. 1–13, Nov. 1985.