Electronic Colloquium on Computational Complexity - Reports Series 1995 - available via:
ECCC
TR95-039
FTP: ftp.eccc.uni-trier.de:/pub/eccc/
WWW: http://www.eccc.uni-trier.de/eccc/
Email: ftpmail@ftp.eccc.uni-trier.de with subject 'help eccc'

Polynomial Time Samplable Distributions

Tomoyuki Yamakami

Department of Computer Science, University of Toronto
Toronto, Ontario, Canada M5S 1A4

July 11, 1995

Abstract

This paper studies distributions which can be "approximated" by sampling algorithms in time polynomial in the length of their outputs. First, it is known that if polynomial-time samplable distributions are polynomial-time computable, then NP collapses to P. This paper shows by a simple counting argument that every polynomial-time samplable distribution is computable in polynomial time if and only if so is every #P function. By this result, the class of polynomially samplable distributions contains no universal distributions if FP = #P. Second, it is also known that there exists polynomially samplable distributions which are not polynomially dominated by any polynomial-time computable distribution if strongly one-way functions exist. This paper strengthens this statement and shows that an assumption, namely, NP $\not\subseteq$ BPP leads to the same consequence. Third, this paper shows that P = NP follows from the assumption that every polynomial-time samplable distribution is polynomially equivalent to some polynomial-time computable distribution.

Key words: average-case complexity, sampling algorithm, domination

1 Introduction

Average-case analysis has been efficient used to show better upper and lower bounds on time and space for algorithms which work on instances distributed randomly according to known distributions. Here *a prior* knowledge of distributions are necessary to analyze the average behaviors of these algorithms. Most practical average-case analyses are made for simple distributions of instances; for example, the distributions which are computable by feasible deterministic algorithms or the distributions with which the instances are producible by feasible probabilistic algorithms. The algorithms of the latter case are simply called *sampling algorithms*, and distributions produced by those algorithms in time polynomial in the length of their outputs are called "P-samplable" by Ben-David et al. [3] (by definition, samplable distributions take only dyadic rational numbers with binary representation). In this paper, we tend to use a more accurate terminology "polynomial-time samplable."

Impagliazzo and Levin, however, took a different approach towards polynomial-time samplability by defining "polynomial-time samplable" distributions to be polynomial-time computable distributions of the pre-images of functions which are also computed in polynomial time [7]. As far as such functions are polynomially honest, the samplability of Impagliazzo and Levin implies that of Ben-David et al. since, otherwise, we can construct a polynomial-time computable function which the standard distribution of its pre-images is not computable in polynomial-time.

To handle broader classes of distributions whose ranges vary on real numbers between 0 and 1, Gurevich [6] proposed an approximation scheme to polynomial-time computable distributions: a distribution μ is called *polynomial-time computable* if we have a polynomial-time algorithm which, on a pair of inputs x and 0^i, outputs an approximation of the value $\mu(x)$ within a factor of 2^{-i}. In this paper, we use Gurevich's scheme to capture the notion of polynomial-time samplability and focus on the complexity of these samplable distributions.

All polynomial-time computable distributions are naturally polynomial-time samplable. However, Ben-David et al. show that, in their setting, polynomial-time samplable distributions are more complex than polynomial-time computable distributions unless P = NP [3]. This paper extends their result and shows by a simple counting argument that all polynomial-time samplable distributions are polynomial-time computable if and only if #P equals FP. In other words, polynomial-time samplable distributions are as hard as #P functions to compute deterministically. As a corollary, we can show that there is no universal, polynomial-time samplable distribution if FP = #P.

When a distribution μ dominates another distribution ν within a polynomial factor, we simply say that μ *polynomially dominates* ν. This type of domination relations between distributions are of great importance in average-case complexity theory. The first step along this line is due to Levin [11]. He introduced the notion of many-one reducibility between distributional problems (or randomized problems) by requiring an additional polynomial-domination relation between distributions. As proven in [6], distributional problems which can be solved in polynomial time on the average are invariant to polynomial-domination relations.

In Levin's definition, domination relations play a special role of "reducibility" between distributions in measuring the complexity of these distributions. For instance, every polynomial-time computable distribution is polynomially dominated by some polynomial-time samplable distribution; but whether the converse holds is not clear. Ben-David et al. show that, in their setting, if strongly one-way functions exist, then there

exists a polynomial-time samplable distribution which is not polynomially-dominated by any polynomial-time computable distributions [3]. This paper shows that the assumption NP $\not\subseteq$ BPP (weaker than the assumption that strongly one-way functions exist) leads to the same conclusion. In this proof, we use a family of hash function to randomize nondeterminism and an amplification technique for probabilistic algorithms to reduce their error probability (as has been used in, e.g., [8]).

When two distributions polynomially dominate each other, we simply call them *polynomially equivalent*. It seems unlikely that every polynomial-time samplable distribution is polynomially equivalent to some polynomial-time computable distribution, but it is still an open question. We note that the assumption FP $\neq$ #P may not sufficient for a negative answer. This paper shows that if NP differs from P, as is widely believed, then there is a polynomial-time samplable distribution which cannot be polynomially equivalent to any polynomial-time computable distribution.

2 Preliminaries

This section presents fundamental notions and notations used throughout this paper.

2.1 Basics

Denote by $\mathbb{N}$ and $\mathbb{R}^+$ the set of *nonnegative integers* and the set of *nonnegative real numbers*, respectively. Let $\mathrm{ilog}(m) = \lceil \log_2 m \rceil$. The notation $\log^k n$ stands for $(\log_2 n)^k$.

Fix $\Sigma = \{0,1\}$ and denote by λ the *empty string*. The length of a string x is denoted by $|x|$, and the cardinality of a set X is denoted by $\|X\|$. For $n \in \mathbb{N}$, Σ^n represents the collection of all strings of length n. The natural order on Σ^* (i.e., to order strings first by length and then lexicographically) is assumed. Let $x_{\leftarrow i}$ be the first i bits of string x. We assume that a pairing function $\langle , \rangle$ is computable and invertible in polynomial time.

Let $\mathbb{D}$ be the set of all *dyadic rational numbers* on the real interval $[0,1]$, i.e., $\{\frac{m}{2^n} \mid m, n \in \mathbb{N}\}$. We always identify a string $s_1 s_2 \cdots s_k$, where $s_i \in \{0,1\}$, with $\sum_{i=1}^{k} s_i \cdot 2^{-i}$ in $\mathbb{D}$.

By P and NP we denote the classes of sets which are deterministically and nondeterministically, respectively, computable in polynomial time. Denote by RP and BPP the classes of sets which are computed by probabilistic Turing machine in polynomial time with one-sided error and two-sided error bounded away from 1/2 by some constant, respectively.

Let FP be the class of functions on Σ^* which are polynomial-time computable. A function f from Σ^* to $\mathbb{N}$ is in #P [20] if there is a set $A \in \mathrm{P}$ and a polynomial p such that $f(x) = \|\{y \in \Sigma^{p(|x|)} \mid \langle x, y \rangle \in A\}\|$ for all x. A set S is in Few [4] if there exists a #P function f, a set $B \in \mathrm{P}$, and a polynomial p such that $S = \{x \mid \langle x, f(x) \rangle \in B\}$, and $f(x) \leq p(|x|)$ for all x.

A function f on Σ^* is *p-honest* if there exists a polynomial p such that $|x| \leq p(|f(x)|)$ for all x, and f is *exp-honest* if $|x| \leq 2^{c|f(x)|}$ for some constant $c \geq 0$. A function f from Σ^* to $\mathbb{R}^+$ is called *positive* if $f(x) > 0$ for all x, and f is *polynomially bounded* if there exists a polynomial p such that $f(x) \leq p(|x|)$ for all x.

A set A is *polynomial-time many-one reducible to* a set B if $A = \{x \mid f(x) \in B\}$ for some reduction f in FP. If f is also p-honest, then we say that A is *p-honest polynomial-time many-one reducible to* B.

A property $\mathcal{P}(x)$ holds for *almost all* x in S if the set $\{x \in S \mid \mathcal{P}(x) \text{ does not hold}\}$ is finite.

Let $\mathrm{Prob}[E]$ denote the probability that event E occurs. In particular, $\mathrm{Prob}_n[E(x)]$ denotes the conditional probability that $E(x)$ occurs when x is chosen from Σ^n at random.

2.2 Distributions

A *distribution* μ is a nondecreasing function from Σ^* to $[0,1]$ such that $\lim_{x\to\infty}\mu(x)=1$, and its associated *(probability) density function* $\hat{\mu}$ is defined by $\hat{\mu}(\lambda)=\mu(\lambda)$ and $\hat{\mu}(x)=\mu(x)-\mu(x^-)$, where x^- is the predecessor of x. For a practical reason, we use *semi-distributions* instead of distributions: a semi-distributions are obtained by replacing $\lim_{x\to\infty}\mu(x)=1$ by $\lim_{x\to\infty}\mu(x)\leq 1$.

For a set S, let $\hat{\mu}(S)=\sum_{x\in S}\hat{\mu}(x)$. For a function f on Σ^*, we simply write $\mu_{f^{-1}}$ to denote the distribution defined by $\hat{\mu}_{f^{-1}}(x)=\hat{\mu}(\{z\mid f(z)=x\})$.

In this paper, we use the *standard* distribution ν_{st}, whose values are dyadic rational numbers, which can be easily sampled by the following probabilistic Turing machine: pick a nonnegative integer n randomly. Precisely speaking, first define the translation by $tr(00)=0$, $tr(11)=1$, and $tr(01)=tr(10)=\#$, where $\#$ is the terminal symbol different from 0 and 1 and then let $\hat{\nu}_{\mathrm{st}}(x)=\mathrm{Prob}[\{z\mid\ |z| \text{ is even}, tr(z)=x\#\}]$ for all x. In other words, $\hat{\nu}_{\mathrm{st}}(x)=2^{-|x|-2\ell(|x|)-1}$, where $\ell(n)=\lfloor\log_2(n+1)\rfloor$.

Let $\mathcal{F}$ be any enumeration of semi-distributions, say $\mathcal{F}=\{\mu_0,\mu_1,\ldots\}$, and let k,s be functions from $\mathbb{N}$ to $\mathbb{R}^+$. A string x is *rare with respect to* $(k,s,\mathcal{F})$ if $\mu_i(x)\leq 2^{-s(|x|)}$ for all $i<k(|x|)$.

Lemma 2.2.1 *Let k be any nondecreasing function on $\mathbb{N}$ such that $0<k(n)\leq\frac{n}{9}$ for all n. Assume $s(n)<n+\log n-2\log k(9n)$. For all $n_0>0$, there exists an n with $n_0\leq n\leq 9n_0$ such that $\|\{x\in\Sigma^n\mid x$ is rare w.r.t. $(k,s,\mathcal{F})\}\|\geq 2^n-2^{s(n)-\log n+2\log k(9n)}$.*

Proof Sketch. We first show that, for any integer $n_0>0$, there exists an n with $n_0\leq n\leq 3n_0+6k(n_0)$ such that, for each $i<k(n_0)$, $\|\{x\in\Sigma^n\mid \mu_i(x)>2^{-s(n)}\}\|<\frac{k(n)\cdot 2^{s(n)}}{n}$. Assume otherwise. Let $r(n)=3n+6k(n)$ and define $A_n^i=\{x\in\Sigma^n\mid\mu_i(x)>2^{-s(n)}\}$. Take n_0 such that, for all n with $n_0\leq n\leq r(n_0)$, there exists an $i<k(n_0)$ satisfying $\|A_n^i\|\geq k(n)\cdot 2^{s(n)}/n$. Hence, at least $\lfloor\frac{r(n_0)-n_0+1}{k(n_0)}\rfloor$ many n's satisfy the condition $\|A_n^j\|\geq k(n)\cdot 2^{s(n)}/n$ for some $j<k(n_0)$. Let $c=\frac{(k(n_0)-1)(r(n_0)+1)+n_0}{k(n_0)}$. Since $r(n_0)\geq\frac{n_0+2k(n_0)-1}{e^{-1/k(n_0)}}$, we have $\left(\frac{r(n_0)}{c+1}\right)^{k(n_0)}\geq e$. Then,

$$\sum_x\mu_j(x)\geq\sum_{n=\lceil c\rceil}^{r(n_0)}\sum_{x\in A_n^i}\mu_i(x)\geq\sum_{n=\lceil c\rceil}^{r(n_0)}\frac{k(n)\cdot 2^{s(n)}}{n}\cdot 2^{-s(n)}=\sum_{c=\lceil c\rceil}^{r(n_0)}\frac{k(n)}{n}>\int_{c+1}^{r(n_0)}\frac{k(n)}{x}dx\geq 1.$$

The lemma immediately follows from the following inequation:

$$\|\{x\in\Sigma^n\mid\exists i<k(n)[\mu_i(x)>2^{-s(n)}]\}\|<k(n)\cdot\frac{k(9n)\cdot 2^{s(n)}}{n}\leq 2^{s(n)-\log n+2\log k(9n)}.$$

□

A function f from Σ^* to $\mathbb{R}^+$ is *polynomial on μ-average* [11] if $\sum_{x\neq\lambda}|x|^{-1}f(x)^\delta\hat{\mu}(x)$ is finite for some $\delta>0$; equivalently, there exists a polynomial p such that $\mathrm{Prob}[\{x\mid f(x)>p(|x|\cdot r)\}]<\frac{1}{r}$ for all real number $r>0$ [14, 18]. For a distribution μ, we say that a Turing machine M runs in *polynomial-time on μ-average* if $\lambda x.\mathrm{Time}_M(x)$ is polynomial on μ-average [6].

A distribution μ *polynomially dominates* a distribution ν [6] if there exists a polynomially bounded function p such that $p(x)\cdot\hat{\mu}(x)\geq\hat{\nu}(x)$ for all x.

Definition 2.2.2 [10, 6] A (semi-)distribution μ is *polynomial-time computable* if there exists a deterministic Turing machine, with two input tapes, one output tape, and one work tape, which works in polynomial time (i.e., on input (x, y), the running time of M is at most $p(|x|, |y|)$ for some polynomial p) such that $|\mu(x) - M(x, 0^i)| < 2^{-i}$ for all x and $i \in \mathbb{N}$. Denote by P-comp the set of all distributions which are polynomial-time computable.

Note that there is an effective enumeration of all polynomial-time computable semi-distributions [16, 19].

2.3 Hash functions

For $n, c \in \mathbb{N}$, let $H_{n,n+c}$ denote the family of pairwise independent universal *hash functions* from Σ^n to Σ^{n+c} which is defined as follows: a hash function h in $H_{n,n+c}$ is of the form $h = (M, b)$, where M is an $n+c$ by n bit matrix and b is a bit vector, and takes its value as $h(x) = Mx \oplus b$. Hence $H_{n,n+c}$ can be identified with the set of all $n+c$ by $n+1$ matrices over $\{0,1\}$, and h is encoded to a string of length $(n+1)(n+c)$.

It is known in [5] that if $x \neq y$ and $i \leq n+c$, then $\mathrm{Prob}[\{h \in H_{n,n+c} \mid h(x)_{\leftarrow i} = h(y)_{\leftarrow i}\}] = 2^{-i}$. Moreover $\mathrm{Prob}[\{h \in H_{n,n+c} \mid h(x)_{\leftarrow i} = w_{\leftarrow i}\}] = 2^{-i}$ for fixed x and w [5].

Fix n and c and assume $i \leq n$ and $\|X\| > 0$. We say that a hash function h in $H_{n,n+c}$ *i-distinguishes x on X* if $h(x)_{\leftarrow i+c} \neq h(w)_{\leftarrow i+c}$ for all $w \in X - \{x\}$. For every x and i with $\mathrm{ilog}(\|X\|) \leq i \leq n+c$, we have

$$\mathrm{Prob}[\{h \in H_{n,n+c} \mid h \text{ } i\text{-distinguishes } x \text{ on } X\}] \leq 1 - \frac{\|X\| - 1}{2^{i+c}} \leq 1 - 2^{-c}.$$

3 Polynomially samplable distributions

This section formally defines the notion of polynomial-time samplability of distributions and shows that polynomial-time samplable distributions are as hard as #P functions to compute deterministically.

3.1 Polynomially samplable distributions

Ben-David et al. [3] first formulated a notion of polynomial-time samplable distributions on dyadic rational numbers by using sampling algorithm. On a recent work on pseudo-random number generators, Håstad et al. [8] also used an ensemble of "polynomial samplable" probability distributions. Here we use an approximation scheme to cope with real-valued distributions and give a generalized definition of polynomial-time samplability.

Definition 3.1.1 A distribution μ is *polynomial-time samplable* if there exists a polynomial p and a probabilistic Turing machine M (it does not necessarily halt), called a *sampling algorithm*, such that $\left|\hat{\mu}(x) - \mathrm{Prob}[M(0^i) \text{ produces } x \text{ and halts within time } p(|x|, i)]\right| < 2^{-i}$ for all x and $i \in \mathbb{N}$. Denote by P-samp the set of all polynomial-time samplable distributions.

From a different point of view, Impagliazzo and Levin [7] defined "polynomial-time samplable" distributions to be of the form $\mu_{f^{-1}}$, where $\hat{\mu}_{f^{-1}}(x) = \mu(\{z \mid f(z) = x\})$, for some $\mu \in$ P-comp and some $f \in$ FP. This definition has major disadvantage: it is so broad that we can actually construct such a distribution that cannot belong to P-comp. Moreover we can construct a $\mu \in$ P-comp and an exp-honest $f \in$ FP such that

no distributions in P-comp polynomially dominate $\mu_{f^{-1}}$. Recall that μ polynomially dominates ν if there is a polynomially-bounded function p such that $p(x) \cdot \hat{\mu}(x) \geq \hat{\nu}(x)$ for all x. This notion will be thoroughly studied in Section 4.

Proposition 3.1.2 *There exists a positive distribution $\mu \in$ P-comp and a nondecreasing, exp-honest function $f \in$ FP such that $\mu_{f^{-1}}$ is not polynomially dominated by any ν in* P-comp.

Proof. We first define η as follows: $\hat{\eta}(x) = 2^{-2\ell(n-1)-1}$ if $x \in \{0\}^*$ and $|x| = n^6$ for some $n \geq 2$, or else $\hat{\eta}(x) = 0$. Let $\hat{\mu}(x) = \frac{1}{2}\hat{\nu}_{\mathrm{st}}(x) + \frac{1}{2}\hat{\eta}(x)$. This μ is positive and belongs to P-comp. For every $n \geq 2$ and for $x = 0^{n^6}$, we have

$$\hat{\mu}(x) > \frac{1}{2}\hat{\eta}(x) = \frac{1}{2 \cdot 2^{2\ell(n-1)}} \geq \frac{1}{2n^2} \geq \frac{1}{n^3} = \frac{1}{(n^6)^{1/2}} = \frac{1}{|x|^{1/2}}$$

since $2^{l(n-1)} \leq n$. Hence, $\hat{\mu}(x) > \frac{1}{|x|^{1/2}}$ holds for all $x = 0^{n^6}$ with $n \geq 2$.

To define the desired function f, we need an effective enumeration of all polynomial-time computable semi-distributions. Assume that $\mathcal{F} = \{\nu_i \mid i \in \mathbb{N}\}$ is such an enumeration (see [16, 19]). Let $f(x)$ be the minimum y such that $\log n \leq |y| \leq 9 \log n$ and $|y|^{k-1} \cdot \hat{\nu}_i(y) < \hat{\mu}(0^n)$ for all $i < \log n$ and all integers k with $1 \leq k \leq \frac{\log n}{5+2\log\log n}$, where $n = \min\{r \mid r^6 \leq |x| < (r+1)^6\}$ if $n \geq 2^{13}$; otherwise, let $f(x) = x$.

This f is well-defined. To see this, consider the case $|x| = n^6$ for some $n \geq 2^{13}$. By choosing $\log n$ as $k(n)$ and $n + \log n - 2\log\log n - 4$ as $s(n)$ in Lemma 2.2.1, we know that there exists at least one rare string y with respect to $(k, s, \mathcal{F})$ with $\log n \leq |y| \leq 9 \log n$, i.e., $\hat{\nu}_i(y) \leq 2^{-s(n)} = \frac{16\log^2 |y|}{|y| \cdot 2^{|y|}}$ for all $i < \log n$. For such a string y, we have

$$|y|^{k-1} \cdot \hat{\nu}_i(y) \leq \frac{16|y|^{k-2}\log^2 |y|}{2^{|y|}} \leq \frac{16 \cdot 9^{k-1}\log^{k-1} n \cdot \log^2(9\log n)}{n} \leq \frac{9^k \log^k n}{n} \leq \frac{1}{n^{1/2}} < \hat{\mu}(0^n)$$

since $\log n \leq |y| \leq 9 \log n$, $16\log^2(9\log n) \leq (9\log n)^2$ if $n \geq 4$, and $9^k \log^k n \leq \sqrt{n}$ if $k \leq \frac{\log n}{5+2\log\log n}$. Hence, $f(x)$ exists. It is easy to see that f is exp-honest and also polynomial-time computable.

By definition, for all k and i, $|y|^{k-1} \cdot \hat{\nu}_i(y) < \hat{\mu}_{f^{-1}}(y)$ for some y since $\hat{\mu}_{f^{-1}}(y) \geq \hat{\mu}(0^n)$. □

In this paper, we require f to be p-honest, and take the following weaker (than in [7]) definition:

Definition 3.1.3 A distribution μ is *weakly polynomial-time samplable* if there exists a distribution $\nu \in$ P-comp and a p-honest function $f \in$ FP such that $\mu = \nu_{f^{-1}}$. Denote by WP-samp the set of all weakly polynomial-time samplable distributions.

Proposition 3.1.4 P-comp $\subseteq$ WP-samp.

Proof. Take f to be the identity function. Then we have $\mu = \mu_{f^{-1}}$ for all distribution μ. □

As shown in [21], the feasible computability of $\mu_{f^{-1}}$, in general, does not imply that of μ, namely, there are distributions μ which are not in P-comp, but $\mu_{f^{-1}}$ is in P-comp for $f(x) = 0^{|x|}$. Moreover Wang and Belanger show that, for every $\mu \in$ P-comp and every nondecreasing, p-honest function $f \in$ FP, $\mu_{f^{-1}}$ belongs to P-comp [2].

Proposition 3.1.5 WP-samp $\subseteq$ P-samp.

Proof. To show the proposition, we modify the proof of Theorem 7 in [3]. Assume that ν is in WP-samp. By definition, there exist a p-honest function $f \in \mathrm{FP}$ and a deterministic polynomial-time Turing machine M such that $\nu = \mu_{f^{-1}}$ and $|\mu(x) - M(x, 0^i)| < 2^{-i}$ for all x and i. By [6] we can assume that $\lambda x.M(x, 0^k)$ is nondecreasing for each fixed k. We also assume that, for some polynomial p, $|x| \leq p(|f(x)|)$ and $|f(x)| \leq p(|x|)$ for all x.

For simplicity, write $\hat{M}(x, 0^k) = \sum_{z \in f^{-1}(x)} M'(z, 0^{p(|x|)+k-1})$, where $M'(x, 0^k) = M(x, 0^k) - M(x^-, 0^k)$. Note that $|\hat{\mu}(x) - M'(x, 0^k)| < 2^{-k+1}$. Since $|f(x)| \leq p(|x|)$, we have

$$|\hat{\nu}(x) - \hat{M}(x, 0^k)| \leq \sum_{z \in f^{-1}(x)} |\hat{\mu}(z) - M'(z, 0^{p(|x|)+k-1})| < 2^{p(|x|)} \cdot 2^{-p(|x|)-k} = 2^{-k}.$$

To complete the proof, we need to show that $\hat{M}(x, 0^k)$ can be computed by some sampling algorithm on input 0^k. Define the sampling algorithm N as follows:

begin sampling algorithm N
  input 0^k
  for $i = 1$ **to** ∞
    choose one bit b_i randomly
    let ρ_i be the real number identified with string $b_1 b_2 \cdots b_i$
    find the minimal string x by binary search such that
      $$M(x^-, 0^{p(|x|)+k-1}) < \rho_i \leq M(x, 0^{p(|x|)+k-1})$$
    if there is such an x **then** output $f(x)$ and halt
  end-for
end.

It is not difficult to see that $\hat{M}(x, 0^k)$ is equal to the probability $\mathrm{Prob}[N(0^k) = x \text{ in time } q(|y|, k)]$ for some polynomial q. □

3.2 The P-comp = P-samp question

This subsection shows that polynomial-time samplable distributions are computable in polynomial time if and only if FP = #P. So, it seems unlikely that P-comp = P-samp.

Toward the goal of this subsection, we first study another category of distributions, the so-called #P-computable distributions introduced by Schuler and Watanabe [17], which seems to have more computational power than polynomial-time samplable distributions. Again we modify their definition to fit our approximation scheme.

Definition 3.2.1 A distribution μ is #P-*computable* if there exists a function $f \in \#\mathrm{P}$ and a polynomial p such that $\left|\hat{\mu}(x) - \frac{f(x,0^i)}{2^{p(|x|,i)}}\right| < 2^{-i}$ for all x and $i \in \mathbb{N}$. Denote by #P-comp the set of all #P-computable distributions.

Proposition 3.2.2 P-samp $\subseteq$ #P-comp.

Proof. Assume that μ is polynomial-time samplable and is witnessed by a sampling algorithm M and a polynomial p. Without loss of generality, we assume that every path of $M(0^i)$ which outputs x halts in

exactly $p(|x|,i)$ steps. Let $f(x,0^i)$ be the number of computation paths y such that $M(0^i)$ outputs x and halts on path y in time $p(|x|,i)$. Clearly $f \in \#\mathrm{P}$ since each path of $M(0^i)$ is bounded by $p(|x|,i)$. It is easy to see that the probability that $M(0^i)$ outputs x and halts in time $p(|x|,i)$ equals $\frac{f(x,0^i)}{2^{p(|x|,i)}}$. Hence, μ turns out to be #P-computable. □

The converse inclusion, #P-comp $\subseteq$ P-samp, is an open question. The best-known result is due to Schuler and Watanabe [17] that, in their setting, every #P-computable distribution can be "approximated with a constant factor" by a sampling algorithm in time polynomial in the length of outputs with nonadaptive queries to an NP oracle.

The next lemma establishes a basic relationship between #P and #P-comp.

Lemma 3.2.3 FP = #P *implies* P-comp = #P-comp.

Proof. Assume that FP = #P. For an arbitrary $\mu \in \#\mathrm{P}$-comp, assume that there exists a function $f \in \#\mathrm{P}$ and a nondecreasing polynomial p such that $\left|\hat{\mu}(x) - \frac{f(x,0^i)}{2^{p(|x|,i)}}\right| < 2^{-i}$ for all x and $i \in \mathbb{N}$. Now we show that μ is computable by some deterministic Turing machine in polynomial time.

Define $g(x,0^i) = \sum_{z \le x} h(z,x,0^i)$, where $h(z,x,0^i) = f(z,0^{|x|+i}) \cdot 2^{p(|x|,|x|+i)-p(|z|,|x|+i)}$. Since $g \in \#\mathrm{P}$, it follows from our assumption that $g \in \mathrm{FP}$. We then have

$$\left|\mu(x) - \frac{g(x,0^i)}{2^{q(|x|,i)}}\right| = \left|\mu(x) - \sum_{z \le x} \frac{h(z,x,0^i)}{2^{q(|x|,i)}}\right| \le \sum_{z \le x} \left|\hat{\mu}(z) - \frac{f(z,0^{|x|+i})}{2^{p(|z|,|x|+i)}}\right| \le 2^{|x|} \cdot 2^{-|x|-i} = 2^{-i},$$

where $q(n,i) = p(n,n+i)$. Hence, $\mu \in$ P-comp. This completes the proof. □

It is known that P-samp = P-comp implies P = NP [3]. In the following lemma, we basically prove that WP-samp = P-comp implies FP = #P.

Lemma 3.2.4 *Assume that, for any* $\mu \in$ P-comp *and any one-one, p-honest, polynomial-time computable function* f, $\mu_{f^{-1}}$ *is in* P-comp. *Then,* FP = #P.

Proof. For any set A in P and any polynomial p, let $g(x) = \|\{y \in \Sigma^{p(|x|)} \mid xy \in A\}\|$. Without loss of generality, we assume that p is strictly increasing. We will show that $g \in \mathrm{FP}$.

Now take the standard distribution ν_{st} and define a one-one, polynomial-time computable function f as follows:

$$f(xy) = \begin{cases} 0xy & \text{if } xy \in A \text{ and } |y| = p(|x|), \\ 1xy & \text{if } xy \notin A \text{ and } |y| = p(|x|), \\ xy & \text{otherwise.} \end{cases}$$

We also define a weakly polynomial-time samplable distribution η by $\hat{\eta}(y) = \hat{\nu}_{\mathrm{st}}(\{z \mid f(z) = y\})$. By our assumption, η is computable in polynomial time. For g, we have the following simple equation:

$$2^{-r(|x|)-2\ell(r(|x|))-1} \cdot g(x) = \sum_{y:|y|=p(|x|)} \hat{\nu}_{\mathrm{st}}(f^{-1}(0xy)) = \eta(0x1^{p(|x|)}) - \eta(0x^-1^{p(|x|)}),$$

where $r(n) = n + p(n) + 1$. Therefore, g is polynomial-time computable. □

We combine the above lemmas and propositions and reach the desired conclusion.

Theorem 3.2.5 *The following statements are equivalent.*

1. FP = #P.
2. P-comp = #P-comp.
3. P-comp = P-samp.
4. P-comp = WP-samp.
5. *For any* $\mu \in$ P-comp *and any one-one, p-honest* $f \in$ FP, $\mu_{f^{-1}}$ *is in* P-comp.

Proof. The implication from (1) to (2) is due to Lemma 3.2.3. Clearly (2) implies (3) by Proposition 3.2.2, and (3) implies (4) by Proposition 3.1.5. By definition, (4) implies (5). The last implication from (5) to (1) immediately follows from Lemma 3.2.4. □

3.3 Universal distributions

We have seen in Theorem 3.2.5 that P-comp = P-samp exactly when FP = #P. This subsection applies this theorem to polynomial-time samplable universal distributions. Universal distributions are known to be *malign*, i.e., average-case complexity equals worst-case complexity [12].

Definition 3.3.1 Let $\mathcal{F}$ be a set of distributions and $\mathcal{T}$ a set of functions from Σ^* to $\mathbb{R}^+$. A distribution μ is called *$\mathcal{T}$-universal for $\mathcal{F}$* if $\mu \in \mathcal{F}$, and for all $\nu \in \mathcal{F}$ there exists a function $t \in \mathcal{T}$ such that $t(x) \cdot \hat{\mu}(x) \geq \hat{\nu}(x)$ for all strings x. Especially, if $\mathcal{T}$ is the set of polynomially-bounded functions, then μ is called *poly-universal*.

By a modification of the proof of Lemma 4.1 in [13], we can show that P-samp has no poly-universal distributions if FP = #P.

Lemma 3.3.2 *Assume that* FP = #P. *For every function* $f \in o(2^n)$, P-samp *has no* $O(f)$*-universal distribution. Hence, there is no poly-universal distribution for* P-samp.

Proof. Assume that FP = #P. Note that, under this assumption, NP collapses to P. We modify the proof of Lemma 4.1 in [13]. Assume that $f \in o(2^n)$, and μ_0 is $O(f)$-universal for P-samp. We note that $g(x) \cdot \hat{\mu}_0(x) \geq \hat{\nu}_{\mathrm{st}}(x)$ for some $g \in O(f)$ since μ_0 is universal. Hence, $\hat{\mu}_0(x) \geq \frac{\hat{\nu}_{\mathrm{st}}(x)}{g(x)} > 2^{-3|x|}$ for almost all x. Let x_{-1} be the minimal string x such that $\hat{\mu}_0(x) > 2^{-3|x|}$.

By Theorem 3.2.5, there is a polynomial-time Turing machine M which computes μ_0. Let $\nu(x) = M(x, 0^{3|x|+4})$ for all x and denote $\hat{\nu}(x) = \nu(x) - \nu(x^-)$. In general, ν is not a distribution since $\hat{\nu}$ does not always take nonnegative value. However, we have $\hat{\nu}(x) > 0$ for all $x \geq x_{-1}$. This is seen as follows: for all x,

$$|\hat{\mu}_0(x) - \hat{\nu}(x)| \leq |\mu_0(x) - \nu(x)| + |\mu_0(x^-) - \nu(x^-)| \leq 2^{-3|x|-4} + 2^{-3|x^-|-4} \leq 2^{-3|x|},$$

and thus $\hat{\nu}(x) \geq \hat{\mu}_0(x) - 2^{-3|x|} > 0$ if $x \geq x_{-1}$.

Now we define a series of strings $\{x_i | i \in \mathbb{N}\}$ as follows. For convenience sake, write $R(x, y)$ if $y \geq 2^{|x|}$ and $\nu(y) - \nu(x) \geq 2^{|y|} \cdot \hat{\nu}(y)$. Let x_{i+1} be the minimal string such that $R(x_i, x_{i+1})$ holds. This x_{i+1} exists since, otherwise, $\hat{\nu}(x_i^+) \leq \nu(y) - \nu(x_i) < 2^{|y|} \cdot \hat{\nu}(y)$ for all y of length $\geq 2^{|x_i|}$, and thus $\hat{\nu}(y) > \frac{1}{c \cdot 2^{|y|}}$ for some constant $c \geq 1$. For each integer $n > |x_i|$, $\sum_{|y|=n} \hat{\nu}(y) > 2^n \cdot \frac{1}{c \cdot 2^n} = \frac{1}{c}$, a contradiction.

The set $\{x_i \mid i \in \mathbb{N}\}$ is expressed by $\{y \mid \exists m < |y| \exists x_0, \ldots, x_m = y \forall i < m[\ x_0 \geq x_{-1}$ and x_{i+1} is the minimal string such that $R(x_i, x_{i+1})]\}$, and hence it belongs to NP. Since NP collapses to P, $\{x_i \mid i \in \mathbb{N}\}$ is in P. Note that

$$\sum_{i=0}^{\infty} 2^{|x_i|}\hat{\nu}(x_i) \leq \sum_{i=0}^{\infty}(\nu(x_{i+1}) - \nu(x_i)) \leq \lim_{i\to\infty} \nu(x_i) \leq 1.$$

Let $\hat{\eta}(x) = c \cdot 2^{|x|}(\hat{\nu}(x) + 2^{-3|x|})$ if $x \in \{x_i | i \in \mathbb{N}\}$; otherwise, 0, where c is an adequate positive constant. The distribution η is obviously computable in polynomial time, and thus $\eta \in$ P-comp.

By our definition, for any constant $d > 0$, there exists an i such that

$$\hat{\eta}(x_i) \geq 2^{|x_i|}(\hat{\nu}(x_i) + 2^{-3|x_i|}) \geq 2^{|x_i|} \cdot \hat{\mu}_0(x_i) \geq d \cdot f(x_i) \cdot \hat{\mu}_0(x_i).$$

This is a contradiction. □

4 Domination and equivalence relations

Domination relations were explicitly introduced by Levin [11] on his theory of average-case complexity as a certain type of "reducibility" between two distributions which measures the complexity of these distributions. In this sense, two distributions which dominate each other can be considered to have almost the same degree of complexity. So, we call them "equivalent." Equivalence relations capture the closeness of two distributions and also give rise to an appropriate "approximation" between them. In this section, we study the consequences of several types of conditions of these domination and equivalence relations.

4.1 Domination conditions

In this subsection, we first focus on polynomial-domination relations which were introduced by Levin [11]. Recall from §2.2 that μ *polynomially dominates* ν if $p(x) \cdot \hat{\mu}(x) \geq \hat{\nu}(x)$ for p a polynomially-bounded function. Polynomial-domination relations are useful in average-case complexity theory since they do not change the degree of average running time: namely, provided that μ polynomially dominates ν, if an algorithm requires polynomial-time on μ-average, then this algorithm also runs in polynomial-time on ν-average [6].

Consider the following condition:

Condition I. For every $\mu \in$ P-samp, there exists a distribution ν in P-comp such that ν polynomially dominates μ, i.e., $p(x) \cdot \hat{\nu}(x) \geq \hat{\mu}(x)$ for some polynomially bounded p.

By Theorem 3.2.5, Condition I is derived from the assumption FP = #P. Ben-David et al. further show as Theorem 8 in [3] that if Condition I holds, then no strong one-way function exists. In their proof, they actually used the following fact:

Lemma 4.1.1 [3] *Assume Condition I. For any set $B \in$ P and a polynomial p, let $S_B = \{x \mid \|B_x\| \leq p(|x|)\}$, where $B_x = \{z \in \Sigma^{|x|} \mid xz \in B\}$. There exists a deterministic Turing machine M such that, for each $n \in \mathbb{N}$, M on input x in $S_B \cap \Sigma^n$ lists all elements of B_x (whenever $B_x = \emptyset$, M outputs 0) in polynomial time if $\|S_B \cap \Sigma^n\| \geq 2^n/p(n)$.*

Let $\#\mathrm{P}_{\mathrm{few}}$ denote the set of all #P functions which are polynomially bounded. Lemma 4.1.1 immediately yields the following consequence.

Corollary 4.1.2 *Condition I implies* $\mathrm{FP} = \#\mathrm{P}_{\mathrm{few}}$, *and thus* $\mathrm{P} = \mathrm{Few}$.

Proof. Assume Condition I. Take a function $f \in \#\mathrm{P}_{\mathrm{few}}$ arbitrarily. Then, there exists a set $B \in \mathrm{P}$ and a polynomial p such that $f(x) = \|B_x\|$ and $f(x) \leq p(|x|)$ for all x. Let $S_B = \{x \mid \|B_x\| \leq p(|x|)\}$. Since $\|S_B \cap \Sigma^n\| = 2^n$, we can apply Lemma 4.1.1 to construct a deterministic polynomial-time Turing machine which, on input 0^n, lists all elements of $S_B \cap \Sigma^n$. Hence, f belongs to FP. □

Using the hash-function technique of [8] and the amplification technique for probabilistic algorithms from, e.g., [1], we can show that Condition I also leads to the consequence $\mathrm{NP} \subseteq \mathrm{BPP}$ which is stronger than the result stated in [3].

Theorem 4.1.3 *Condition I implies* $\mathrm{NP} \subseteq \mathrm{BPP}$.

Proof. Assuming Condition I, take any set A in NP to show that $A \in \mathrm{BPP}$. There exists a set $B \in \mathrm{P}$ such that $A = \{x \mid \exists z \in \Sigma^{|x|}[xz \in B]\}$. Let $g_B(x) = \|B_x\|$ and $B_x = \{z \in \Sigma^{|x|} \mid xz \in B\}$.

We take the set $H_{n,n+c}$ of hash functions. Define $\hat{B} = \{xz | x = x' s_k^n h(h(z')_{\leftarrow k+c})t, z = z'10^{|x|-|z'|}, x'z' \in B, x', z' \in \Sigma^n, t \in \Sigma^{n-k}, h \in H_{n,n+c}, c = \mathrm{ilog}(n)\}$, where s_k^n is the kth string in the set $\Sigma^{\mathrm{ilog}(n)}$.

Let $S_{\hat{B}} = \{x \mid \|\hat{B}_x\| \leq 1\}$. We first show that $\|S_{\hat{B}} \cap \Sigma^n\| \geq \frac{2^n}{n(n+2)}$ for almost all n. Now fix k and x' and assume that $n = |x'|$ is sufficiently large, and $\mathrm{ilog}(g_B(x')) \leq k \leq n$. Consider the case $g_B(x') > 0$. The probability $\rho_{k,x'} = \mathrm{Prob}[\{hw \mid g_{\hat{B}}(x' s_k^n hw) \leq 1, h \in H_{n,n+c}, w \in \Sigma^{n+c}\}]$ is larger than or equal to the sum of the probability over all hw that, for each $z' \in B_{x'}$, $h(z')_{\leftarrow k+c} = w_{\leftarrow k+c}$, and h k-distinguishes z' on $B_{x'}$. Thus, we have

$$\rho_{k,x'} > g_B(x') \cdot (1 - 2^{-c}) \cdot 2^{-(k+c)} \geq (1 - 2^{-c})2^{-c} = \frac{n-1}{n^2} \geq \frac{1}{n+2}.$$

For the case $g_B(x') = 0$, clearly $\rho_{k,x'} = 1$. Hence, $\mathrm{Prob}_n[g_{\hat{B}}(x) \leq 1] \geq 2^{-c} \cdot \min_{x'}\{\rho_{c,x'}\} \geq \frac{1}{n(n+2)}$. This yields the desired result.

By Lemma 4.1.1, there is a polynomial-time deterministic Turing machine N which recognizes $S_{\hat{B}}$. We define a probabilistic polynomial-time algorithm $\hat{M}$ as follows:

begin probabilistic algorithm $\hat{M}$
  input x (say, $n = |x|$)
  choose $w, h_1, \ldots, h_{n(n+2)}$ at random ($w \in \Sigma^n, h_i \in H_{n,n+c}, c = \mathrm{ilog}(n)$)
  let *Result* $= 0$
  for all j $(1 \leq j \leq n(n+2))$ and **all** k $(1 \leq k \leq n)$
    run N on $x'_{j,k} = x s_k^n h_j w$
    let *Result* $=$ OR of *Result* and $N(x'_{j,k})$
  end-for
  output *Result*
end.

Our goal is to prove that $\mathrm{Prob}[\hat{M}(x) = A(x)] \geq \frac{2}{3}$ for almost all x. Take any input x of length n. Let $\rho = \mathrm{Prob}[\{h_1 \cdots h_{n(n+2)} \mid A(x) = \mathrm{OR}_{j=1}^{n(n+2)} \mathrm{OR}_{k=1}^{n} N(x'_{j,k}))\}]$. Note that the probability $\mathrm{Prob}[\hat{M}(x) = A(x)]$

is at least ρ. To lead to the desired consequence, it suffices to show that $\rho \geq 1 - e^{-n}$ for almost all n since $1 - e^{-n} \geq \frac{2}{3}$ for all $n \geq 2$.

Now fix j. Assume $A(x) = 1$. Note that if $0 < g_{\tilde{B}}(x'_{j,k'}) \leq 1$ for some k', then $\mathrm{OR}^n_{k=1} N(x'_{j,k}) = 1$. The probability $\rho_j = \mathrm{Prob}[\{h_j \mid A(x) = \mathrm{OR}^n_{k=1} N(x'_{j,k})\}]$ is at least the sum of the probability over all h that, for each $z \in B_x$ and for some k with $\mathrm{ilog}(g_B(x)) \leq k \leq n$, $h(z)_{\leftarrow k+c} = w_{\leftarrow k+c}$ and h k-distinguishes z on B_x. Hence, $\rho_j > g_B(x) \cdot (1 - 2^{-c}) \cdot 2^{-(k+c)} \geq (1 - 2^{-c}) 2^{-c} \geq \frac{1}{n+2}$. For the other case $A(x) = 0$, $N(x s^n_k h w) = 0$ for all h, w, and k; thus $\rho_j = 1$. So, in general we have $\rho_j \geq \frac{1}{n+2}$ for all j. To calculate ρ, consider the error probability $1 - \rho_j$. After $n(n+2)$ independent trials for $h_1, \ldots, h_{n(n+2)}$, the error probability $1 - \rho$ is at most $\prod_{j=1}^{n(n+2)} (1 - \rho_j) \leq (1 - \frac{1}{n+2})^{n(n+2)} \leq e^{-n}$ since $(1 - \frac{1}{r})^r \leq \frac{1}{e}$ for almost all positive integers r. Thus, we have $\rho \geq 1 - e^{-n}$. This completes the proof. □

Polynomial-domination relations are useful but too tight to be considered as an effective measure of "approximation" or "reducibility" between distributions in average-case complexity theory. Gurevich [6] later introduced a weaker form of domination relations by requiring a function to be polynomially bounded "on the average." Following his definition, we also relax Condition I to allow p to be, instead, polynomial on the average in the following fashion:

Condition I'. For every p-honest function $f \in$ FP and every $\mu \in$ P-comp, there exists a distribution $\nu \in$ P-comp and a function p which is polynomial on μ-average such that $\hat{\nu}(y) \geq \sum_{x \in f^{-1}(y)} \frac{\hat{\mu}(x)}{p(x)}$ for all y.

Note that Condition I implies Condition I' since, for every p-honest function f in FP, the following two conditions are equivalent: (i) there exists a polynomially-bounded function p such that $p(y)\hat{\nu}(y) \geq \hat{\mu}_{f^{-1}}(y)$ for all y, and (ii) there exists a polynomially-bounded function p such that $\hat{\nu}(y) \geq \sum_{x \in f^{-1}(y)} \frac{\hat{\mu}(x)}{p(x)}$ for all y [6].

In the definition of Condition I', the p-honesty of f is essential since, without this condition, Condition I' fails to hold. This follows from a fact of the complexity class $\mathrm{P}_{\text{P-comp}}$. By $\mathrm{P}_{\text{P-comp}}$ we denote the collection of sets A such that, for all $\mu \in$ P-comp, there is a deterministic Turing machine M which computes A in polynomial-time on μ-average [18]. We need the fact that $\mathrm{P}_{\text{P-comp}}$ is not closed downward under polynomial-time many-one reductions [19]. The following lemma which was proven as Lemma 7.1 in [6] is also used to show the necessity of the p-honesty condition in Condition I'.

Lemma 4.1.4 [6] *Assume that $\hat{\nu}(y) \geq \sum_{x \in f^{-1}(y)} \frac{\hat{\mu}(x)}{p(x)}$ for all y, p is polynomial on μ-average, and A is many-one reducible to B via f in* FP. *If B is computable in polynomial-time on ν-average, so is A on μ-average.*

Proposition 4.1.5 *There exists some function $f \in$ FP and some $\mu \in$ P-comp such that, for every $\nu \in$ P-comp and every function p which is polynomial on μ-average, $\hat{\nu}(y) < \sum_{x \in f^{-1}(y)} \frac{\hat{\mu}(x)}{p(x)}$ for some y.*

Proof. Assuming to the contrary, we prove here that $\mathrm{P}_{\text{P-comp}}$ is closed under polynomial-time many-one reductions. This contradicts the fact that $\mathrm{P}_{\text{P-comp}}$ is not closed under polynomial-time many-one reductions [19].

Assume that A is polynomial-time many-one reducible to B via a reduction f, and B is in $\mathrm{P}_{\text{P-comp}}$. We show that $A \in \mathrm{P}_{\text{P-comp}}$. For every distribution μ in P-comp, by our assumption, there is a distribution

$\nu \in$ P-comp and a function p which is polynomial on μ-average such that $\hat{\nu}(y) \geq \sum_{x \in f^{-1}} \frac{\hat{\mu}(x)}{p(x)}$ for all y. Note that B is computable in polynomial-time on ν-average. By Lemma 4.1.4, A is also computable in polynomial-time on μ-average. Since μ is arbitrary, A belongs to $\mathrm{P}_{\text{P-comp}}$. □

As another consequence of Lemma 4.1.4, $\mathrm{P}_{\text{P-comp}}$ turns out to be closed under many-one reducibility with p-honest polynomial-time computable reductions.

Proposition 4.1.6 *If Condition I' holds, then* $\mathrm{P}_{\text{P-comp}}$ *is closed downward under p-honest polynomial-time many-one reducibility.*

Proof. Immediate from Lemma 4.1.4 and by the same argument in Proposition 4.1.5. □

4.2 Equivalence conditions

As seen in §4.1, domination relations can be viewed as an "approximation" or a "reducibility" between two distributions in average-case complexity theory. If two distributions dominate each other, in this paper, we call them "equivalent" since they are close to each other and have almost the same degree of complexity. Equivalence relations was first discussed in [17] under the terminology "approximation within constant factor" to show the closeness of two distributions.

In a particular case, we say that μ is *polynomially equivalent* to ν if μ polynomially dominates ν, and ν polynomially dominates μ. Now consider *flat* distributions which were defined in [6] (μ is flat if $\hat{\nu}(x) \leq 2^{|x|^\epsilon}$ for some $\epsilon > 0$). Note that most of natural distributions dealt with in average-case complexity theory are flat. These flat distributions are invariant to polynomial-equivalence relations. This is seen as follows. Since μ is polynomially equivalent to ν, $\frac{\hat{\nu}(x)}{p(x)} \leq \hat{\mu}(x) \leq p(x)\hat{\nu}(x)$ for some polynomially-bounded p. Then, we have

$$|\hat{\mu}(x) - \hat{\nu}(x)| \leq \max\left\{ |p(x)\hat{\nu}(x) - \hat{\nu}(x)|, \left|\frac{\hat{\nu}(x)}{p(x)} - \hat{\nu}(x)\right| \right\} \leq \frac{p(x)-1}{2^{|x|^\epsilon}}.$$

Hence, $|\hat{\mu}(x) - \hat{\nu}(x)| \leq 2^{-|x|^{\epsilon'}}$ for some $\epsilon' > 0$, and consequently μ is flat.

In this subsection, we study the following conditions of equivalence relations:

Condition II. For every $\mu \in$ P-samp, there exists a distribution ν in P-comp such that μ is polynomially equivalent to ν.

Condition II'. For every p-honest $f \in$ FP and every $\mu \in$ P-comp, there exists $\nu \in$ P-comp and functions p, q which are polynomial on μ-average such that $\sum_{x \in f^{-1}(y)} q(x)\hat{\mu}(x) \geq \hat{\nu}(y) \geq \sum_{x \in f^{-1}(y)} \frac{\hat{\mu}(x)}{p(x)}$ for all y.

Clearly Condition II implies Condition I, and Condition II' implies Condition I'. By Theorem 3.2.5, Condition II is true if FP = #P.

In what follows, we show that Condition II implies P = NP, and Condition II' implies P = RP. We prove the latter claim first. In this case, the amplification lemma to one-sided bounded-error probabilistic algorithms is effectively used to make its error probability exponentially small.

Proposition 4.2.1 *Condition II' implies* P = RP.

Proof. Take an arbitrary $A \in \mathrm{RP}$ and prove that A belongs to P. By the amplification lemma [15], there is a strictly increasing polynomial p and a set $B \in \mathrm{P}$ such that, for every $x \in \Sigma^n$, $\mathrm{Prob}[\{y \in \Sigma^{p(n)} \mid \langle x, y\rangle \notin B\}] \leq 2^{-n}$ if $x \in A$; otherwise, $\mathrm{Prob}[\{y \in \Sigma^{p(n)} \mid \langle x, y\rangle \in B\}] = 0$.

Take the distribution μ defined as $\hat{\mu}(xy) = \hat{\nu}_{\mathrm{st}}(x) \cdot 2^{-p(|x|)}$ if $|y| = p(|x|)$, or else $\hat{\mu}(xy) = 0$. Clearly μ is polynomial-time computable. Let

$$f(xy) = \begin{cases} x1^{p(|x|)} & \text{if } |y| = p(|x|) \text{ and } \langle x, y\rangle \in B, \\ x0^{p(|x|)} & \text{if } |y| = p(|x|) \text{ and } \langle x, y\rangle \notin B, \\ xy & \text{otherwise.} \end{cases}$$

By Condition II', we have a distribution $\nu \in \text{P-comp}$ and a function q which is polynomial on μ-average such that $\sum_{x \in f^{-1}(y)} q(x)\hat{\mu}(x) \geq \hat{\nu}(y) \geq \sum_{x \in f^{-1}(y)} \frac{\hat{\mu}(x)}{q(x)}$ for all y. Since $\sum_{x \neq \lambda} \frac{q(x)^{1/k}}{|x|}\hat{\mu}(x) \leq c$ for some constants $k, c \geq 1$, we have $q(x) \leq \left(\frac{c \cdot |x|}{\hat{\mu}(x)}\right)^k$ for all nonempty strings x with $\hat{\mu}(x) > 0$. Thus, for almost all x and all y of length $p(|x|)$,

$$q(xy) \leq \left(\frac{c \cdot |xy|}{\hat{\mu}(xy)}\right)^k \leq \left(c(|x| + p(|x|))(|x| + 1)2^{|x|+p(|x|)}\right)^k \leq \left(2^{|x|} \cdot 2^{|x|+p(|x|)}\right)^k = 2^{r(|x|)},$$

where $r(n) = (2n+p(n))^k$. Since $\nu \in \text{P-comp}$, there exists a deterministic polynomial-time Turing machine M such that $|\hat{\nu}(x) - M(x, 0^i)| < 2^{-i}$. Let $M'(x) = M(x, 0^{r(|x|)+2|x|})$. By definition, $|\hat{\nu}(x) - M'(x)| < 2^{-r(|x|)-2|x|}$ for all x.

Let $x \in \Sigma^n$. Assume that $x \in A$. Then, we have

$$\hat{\nu}(x1^{p(n)}) \geq \sum_{z \in f^{-1}(x1^{p(n)})} \frac{\hat{\mu}(z)}{q(z)} \geq \frac{\|f^{-1}(x1^{p(n)})\|}{2^{r(n)}} \cdot \frac{\hat{\nu}_{\mathrm{st}}(x)}{2^{p(n)}} \geq \frac{2^n - 1}{2^{r(n)+2n}}$$

since $\hat{\nu}_{\mathrm{st}}(x) \geq \frac{1}{2n^2 \cdot 2^n} \geq \frac{1}{2^{2n}}$ if $n \geq 7$. Hence, $M'(x) > \hat{\nu}(x1^{p(n)}) - 2^{-r(n)-2n} \geq 2^{-r(n)-2n}(2^n - 2)$. In the other case $x \notin A$, $\hat{\nu}(x1^{p(n)}) \leq \sum_{z \in f^{-1}(x1^{p(n)})} q(z) \cdot \hat{\mu}(z) = 0$. Hence, $M'(x) < \hat{\nu}(x1^{p(n)}) + 2^{-r(n)-2n} = 2^{-r(n)-2n}$. Now we have a complete characterization of A in terms of M'; namely, $A \cap \Sigma^n = \{x \in \Sigma^n \mid M'(x) \geq 2^{-r(n)-2n}(2^n - 2)\}$ for almost all n. Since M' halts in polynomial-time, A is also computable in polynomial-time. □

The above proposition may not simply achieved to conclude that P = PP since the amplification lemma may not hold for PP sets.

To close this subsection, we prove that Condition II leads to the consequence that NP collapses to P. This follows from a combination of Theorem 4.1.3 and Proposition 4.2.1.

Theorem 4.2.2 *Condition II implies* P = NP.

Proof. Assume Condition II. Theorem 4.1.3 implies $\mathrm{NP} \subseteq \mathrm{BPP}$. Ko [9] shows that $\mathrm{NP} \subseteq \mathrm{BPP}$ yields RP = NP. Thus, we have RP = NP. By Proposition 4.2.1, Condition II also leads to P = RP. Therefore, we conclude that P = NP. □

Acknowledgments

The author is grateful to Stephen A. Cook for discussions with him and to Rainer Schuler for giving him useful comments. He also thank Michael D. Hutton for his correcting errors in an early draft.

References

[1] J. L. Balcázar, J. Díaz and J. Gabarró, *Structural Complexity I, II*, Springer-Verlag, 1988(I), 1990(II).

[2] J. Belanger and J. Wang, Isomorphisms of NP complete problems on random instances, in: *Proceedings, 8th Conference on Structure in Complexity Theory*, 1993, pp.65–74.

[3] S. Ben-David, B. Chor, O. Goldreich, and M. Luby, On the theory of average case complexity, *J. Comput. System Sci.*, **44** (1992), pp.193–219.

[4] J. Cai and L. A. Hemachandra, On the power of parity polynomial time, *Math. Systems Theory*, **23** (1990), pp.95–106.

[5] J. Carter and M. Wegman, Universal classes of hash functions, *J. Comput. System Sci.*, **18** (1979), pp.143–154.

[6] Y. Gurevich, Average case complexity, *J. Comput. System Sci.*, **42** (1991), pp.346–398.

[7] R. Impagliazzo and L. A. Levin, No better ways to generate hard NP instances than picking uniformly at random, in: *Proceedings, 31st IEEE Conference on Foundation of Computer Science*, pp. 812–821, 1990.

[8] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, Construction of a pseudo-random generator from any one-way function, Technical Report, TR-91-068, International Computer Science Institute, Berkeley, California, 1991. Preliminary versions appeared in the proceedings of the 21st STOC, 1989, pp.12–24 and the 22nd STOC, 1990, pp.395–404.

[9] K. Ko, Some observations on the probabilistic algorithms and NP-hard problems, *Information Processing letters*, **14** (1982), pp.39–43.

[10] K. Ko and H. Friedman, Computational complexity of real functions, *Theoretical Computer Science*, **20** (1982), pp.323–352.

[11] L. Levin, Average case complete problems, *SIAM J. Comput.* **15** (1986), pp.285–286.

[12] M. Li and P. M.B. Vitányi, Average case complexity under the universal distribution equals worst-case complexity, *Information Processing Letters*, **42** (1992), pp.145–149.

[13] M. Li and P. Vitányi, *An introduction to Kolmogorov complexity and its applications*, Springer-Verlag, New York, 1993.

[14] R. E. Schapire, The emerging theory of average-case complexity, Technical Report MIT/LCS/TM-431, Massachusetts Institute of Technology, 1990.

[15] U. Schöning, Complexity and Structure, Lecture Notes in Computer Science, Vol.211, 1986.

[16] R. Schuler, On average polynomial time, Technical Report Nr.94-12, Universität Ulm, 1994.

[17] R. Schuler and O. Watanabe, Towards average-case complexity analysis of NP optimization problems, in: *Proceedings, 10th Conference on Structure in Complexity Theory*, 1995.

[18] R. Schuler and T. Yamakami, Structural average case complexity, in: *Proceedings, 12th Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science, Vol.652, 1992, pp.128–139, Springer-Verlag.

[19] R. Schuler and T. Yamakami, Sets computable in polynomial time on average, in: *Proceedings, 1st Annual International Computing and Combinatorics Conference*, August, Xi'an, China, 1995.

[20] L. Valiant, The complexity of computing the permanent, *Theoretical Computer Science*, **5** (1979), pp.189–201.

[21] J. Wang and J. Belanger, On average P vs. average NP, in *Complexity Theory – Current Research*, editors K. Ambos-Spies, S. Homer and U. Schöning, Cambridge University Press, pp.47–67, 1993.