Electronic Colloquium on Computational Complexity - Reports Series 1995 - available via:

Revision 02 of
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†

Abstract

This paper studies the complexity of the polynomial-time samplable (**P**-samplable) distributions, which can be approximated within an exponentially small factor by sampling algorithms in time polynomial in the length of their outputs. The paper shows common assumptions in complexity theory that yield the separation of polynomial-time samplable distributions from the polynomial-time computable distributions with respect to polynomial domination, average-polynomial domination, polynomial equivalence, and average-polynomial equivalence.

Key words: average-case complexity, sampling algorithm, domination condition, strong one-way functions

1 Introduction

Average-case complexity theory has provided a rich source of analysis that shows better upper and lower bounds on time and space for randomized algorithms that work on instances distributed randomly according to underlying "natural" distributions. In the course of early studies, we have found randomized algorithms that solve even some **NP**-complete problems in average polynomial time. Levin's discovery of the existence of average-case **NP**-complete problems, however, not only changed the course of our attempt to solve all **NP** problems fast on the average but also highlighted the importance of research on average-case analysis of **NP** problems as well as underlying distributions.

Most of "natural" distributions dealt in average-case complexity theory are *computable* (or more loosely approximable) or *samplable*—the distributions with which the instances are producible by feasible, probabilistic algorithms. The algorithms that "generate" samplable distributions are called *sampling algorithms* or *generators* and the distributions produced (or more loosely, approximated) by those algorithms in time polynomial in the length of their outputs are particularly called

*A preliminary version of the paper appeared in the Proceedings of the 21st International Symposium on Mathematical Foundations of Computer Science, Lecture Note in Computer Science Vol.1113, pp.566–578, 1996.

†Current affiliation: Department of Computer Science, Princeton University, Princeton, NJ 08544.

P-*samplable* (polynomial-time samplable), whose name was attributed to Ben-David et al.[2], in contrast to **P**-*computable* distributions.

Toward the complexity of **P**-samplable distributions, an early study shows that **P**-samp—the set of all **P**-samplable distributions—contains **P**-comp—the set of all **P**-computable distributions [3, 2]. In early 1990's, Ben-David et al. showed that **P**-samp differs from **P**-comp unless **NP** collapses to **P** [2]. This result was soon followed by the final answer that **P** = **PP** is a sufficient and necessary condition for **P**-comp = **P**-samp [9].

Along the line of average-case complexity theory, the notion of *p-domination* (that is, every probability of a distribution exceeds that of another distribution with a multiple of a polynomial factor) [8] is more appropriate on a complexity-theoretic discussion among distributions since the average-case complexity measure, *polynomial on the average*, is indeed invariant to the p-domination; in other words, any two distributions which are *p-equivalent* (that is, the two distributions p-dominate each other) [19] preserve this measure for any distributional problems. It thus seems natural to ask whether every distribution in **P**-samp is p-dominated by some distribution in **P**-comp and more proper to discuss the computational complexity of the set **P**-samp$/_{\approx^{\mathrm{p}}}$—the equivalence classes of distributions modulo the above p-equivalence $\approx^{\mathrm{p}}$—than **P**-samp itself.

Another notion more suitable in average-case complexity theory is the *avp-domination* (average-polynomial domination) [4], which is an average-case version of p-domination, and its *avp-equivalence* $\approx^{\mathrm{avp}}$. It is also known that any two avp-equivalent distributions preserve the measure "polynomial on the average" no matter what distributional problems are chosen.

In this paper, we study two questions of whether there exists a **P**-samplable distribution which no **P**-computable distribution p-dominates (or avp-dominates) and whether the set **P**-samp$/_{\approx^{\mathrm{p}}}$ differs from **P**-comp$/_{\approx^{\mathrm{p}}}$ (or **P**-samp$/_{\approx^{\mathrm{avp}}} \not\subseteq$ **P**-comp$/_{\approx^{\mathrm{avp}}}$). Notice that, for the separation **P**-comp$/_{\approx^{\mathrm{p}}} \neq$ **P**-samp$/_{\approx^{\mathrm{p}}}$, for example, even the assumption **P** $\neq$ **PP** may not suffice. Thus we must ask what type of assumption suffices to lead to our desired consequences. In this paper, we shall extensively focus on this question and throughout Sections 3 and 4 we shall give the following answers: if **P** $\neq$ **RP** then **P**-samp$/_{\approx^{\mathrm{avp}}}$ differs from **P**-comp$/_{\approx^{\mathrm{avp}}}$; if **P** $\neq$ **NP** then **P**-samp$/_{\approx^{\mathrm{p}}}$ differs from **P**-comp$/_{\approx^{\mathrm{p}}}$; and if **NP** $\not\subseteq$ **Nearly-BPP** then there exists a distribution in **P**-samp that is not p-dominated by any distribution in **P**-comp. A further discussion is presented in Section 5.

2 Basic Notions and Notation

Denote by $\mathbb{N}$ and $\mathbb{R}^+$ the set of *nonnegative integers* and the set of *nonnegative real numbers*, respectively. Let $\mathrm{ilog}(m) = \lceil \log_2 m \rceil$ and $\mathrm{llog}(m) = \lfloor \log_2(m+1) \rfloor$. The notation $\log^k n$ stands for $(\log_2 n)^k$. A property $\mathcal{P}(x)$, is said to hold *for almost all* x in an infinite set S if the set $\{x \in S \mid \mathcal{P}(x) \text{ does not hold}\}$ is finite.

Fix our alphabet $\Sigma = \{0,1\}$ and denote by λ the *empty string*. Let $\Sigma^+ = \Sigma^* - \{\lambda\}$. For each $n \in \mathbb{N}$, A^n denotes $A \cap \Sigma^n$ for a set $A \subseteq \Sigma^*$, where $\Sigma^n = \{x \in \Sigma^* \mid |x| = n\}$. Denote by s_k^n the k-th string of the set $\Sigma^{\mathrm{ilog}(n)}$ with respect to the standard order on Σ^* (i.e., to order strings first by length and then lexicographically); for instance, $s_1^n = 0^{\mathrm{ilog}(n)}$. By $x \sqsubseteq y$, we mean that x is an *initial segment* of y, i.e., $xs = y$ for some s. For any set $A \subseteq \Sigma^*$ and $x \in \Sigma^*$, let $A(x) = 1$ if $x \in A$, or else $A(x) = 0$. Let $x_{\leftarrow i}$ be the first i bits of string x.

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

A function f from Σ^* to Σ^* is *p-bounded* (polynomially bounded) if there exists a polynomial p such that $|f(x)| \leq p(|x|)$ for all x. Moreover, f is said to be *p-honest* (polynomially honest) if there exists a polynomial p such that $|x| \leq p(|f(x)|)$ for all x; similarly, f is *exp-honest* if $|x| \leq 2^{c|f(x)|}$ for some constant $c \geq 0$.

A real-valued function f, from Σ^* to $\mathbb{R}^+$, is called *positive* if $f(x) > 0$ for all x, and f is *p-bounded* if there exists a polynomial p such that $f(x) \leq p(|x|)$ for all x. A function f from $\mathbb{N}$ to $\mathbb{R}^+$ is *negligible* if, for every polynomial p, $f(n) \leq 1/p(n)$ holds for almost all $n \in \mathbb{N}$.

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

We assume the reader's familiarity with Turing machines, central complexity classes **P**, **NP**, **RP**, **BPP**, **PP**, and **E** (liner-exponential time), and two function classes **FP** and #**P**. (For more details, see, e.g., [10].)

The notation $\mathrm{Prob}[E]$ in general stands for the probability that event E occurs and $\mathrm{Prob}_{x \in A}[E(x)]$ denotes the conditional probability that $E(x)$ occurs when x is chosen from finite set A at random. A *distribution* μ is a nondecreasing function from Σ^* to $[0,1]$ such that $\mu(x)$ converges to 1 as $|x|$ grows 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 set $S \subseteq \Sigma^*$, $\hat{\mu}(S)$ denotes $\sum_{x \in S} \hat{\mu}(x)$. For a function f from Σ^* to Σ^*, we write $\mu_{f^{-1}}$ to denote the distribution defined by its probability $\hat{\mu}_{f^{-1}}(x) = \hat{\mu}(\{z \mid f(z) = x\})$. In this paper, we use Regan's pairing function $\langle , \rangle$ [11]; however, we often write $\hat{\mu}(x,y)$ for $\hat{\mu}(\langle x, y \rangle)$ for brevity. For convenience, let ν_{st} (the *standard* distribution) be defined by its probability $\hat{\nu}_{\mathrm{st}}(x) = 2^{-|x| - 2\mathrm{llog}(|x|) - 1}$.

Let μ be a distribution. A function f from Σ^* to $\mathbb{R}^+ \cup \{\infty\}$ is called *polynomial on μ-average*[‡]

[‡]Equivalently, there exists a polynomial p such that $\hat{\mu}(\{x \mid f(x) > p(|x|r)\}) < 1/r$ for all positive real numbers r [12, 17].

if the expectation $\sum_{x\in\Sigma^+} |x|^{-1} f(x)^\delta \hat{\mu}(x)$ converges for some constant $\delta > 0$ [8, 4].

For any two distributions μ and ν, μ *p-dominates* (polynomially dominates) ν, symbolically $\nu \preceq^{\mathrm{p}} \mu$, if there exists a p-bounded function p from Σ^* to $\mathbb{R}^+$ such that $p(x)\hat{\mu}(x) \geq \hat{\nu}(x)$ for all x [8]; similarly, μ *avp-dominates* (average-polynomially dominates) ν, denoted by $\nu \preceq^{\mathrm{avp}} \mu$, if there exists a function p from Σ^* to $\mathbb{R}^+$ which is polynomial on ν-average such that $p(x)\hat{\mu}(x) \geq \hat{\nu}(x)$ for all x [4]. Moreover, μ is *p-equivalent* (polynomially equivalent) *to* ν, symbolically $\mu \approx^{\mathrm{p}} \nu$, if both μ and ν p-dominate each other [19]. Similarly, μ is *avp-equivalent* (average-polynomially equivalent) *to* ν, denoted by $\mu \approx^{\mathrm{avp}} \nu$, if both μ and ν avp-dominate each other.

A distribution μ is said to be **P**-*computable* (polynomial-time computable)[§] if there exists a deterministic polynomial-time Turing machine M which "approximates" μ, i.e., $|\mu(x) - M(x, 0^i)| \leq 2^{-i}$ for all $x \in \Sigma^*$ and $i \in \mathbb{N}$ [7, 4]. Denote by **P**-comp the set of all **P**-computable distributions. In a similar fashion, we can define **E**-comp, the set of **E**-computable distributions. In contrast, μ is called *strictly* **P**-*computable* if there exists a polynomial-time Turing machine M such that $M(x) = \mu(x)$ for all x [2].

A distribution μ is called #**P**-*computable* if there exist a function $f \in \#\mathbf{P}$ and a polynomial p such that $\left|\hat{\mu}(x) - \frac{f(x,0^i)}{2^{p(|x|,i)}}\right| \leq 2^{-i}$ for all $x \in \Sigma^*$ and $i \in \mathbb{N}$ [16] and the set of all #**P**-computable distributions is denoted by #**P**-comp.

A distribution μ is **P**-*samplable* (polynomial-time samplable) if there exists a polynomial p and a randomized Turing machine M (which does not necessarily halt), called a *sampling machine* or *generator*, which "approximates" $\hat{\mu}$, i.e.,

$$\left|\hat{\mu}(x) - \mathrm{Prob}_M[M(0^i) \text{ produces } x \text{ and halts within time } p(|x|, i)]\right| \leq 2^{-i}$$

for all x and $i \in \mathbb{N}$. In contrast, we call μ *strictly* **P**-*samplable* if a probabilistic polynomial-time algorithm generates strings x with probabilities $\hat{\mu}(x)$ [2]. An algorithm used for a sampling machine is called a *sampling algorithm*. Let **P**-samp denote the set of all **P**-samplable distributions.

Another type of "polynomial-time samplable" distribution, introduced in [5] as is of the form $\mu_{f^{-1}}$ for some $\mu \in \mathbf{P}$-comp and some $f \in \mathbf{FP}$, is of importance[¶] in average-case analysis. This definition, nevertheless, allows us to construct a positive distribution $\mu \in \mathbf{P}$-comp and a nondecreasing, exp-honest function $f \in \mathbf{FP}$ such that $\mu_{f^{-1}}$ is not p-dominated by any ν in **P**-samp. Hence we must restrict our interest and require f be *p-honest*. We call such samplable distributions *invertibly* **P**-*samplable* (**IP**-samplable, for short) for clarity. Let **IP**-samp denote the set of all **IP**-samplable distributions and furthermore let $\mathbf{IP_1}$-samp be the set of all distributions of the form $\mu_{f^{-1}}$ for a

[§]In the theory of average-case **NP**-completeness, a distribution is sometimes called *polynomial-time computable* if it is p-dominated by one that is computable in polynomial time.

[¶]These distributions play an important role in average-case complexity theory; for example, any **NP** problem under these distributions is shown to be "probabilistically" reduced to a single **NP** problem under the standard distribution [5].

distribution $\mu \in \mathbf{P}$-comp and a p-honest, *one-one* function f in $\mathbf{FP}$.

At the end, we note that, by extending a result in [9] that $\mathbf{P} = \mathbf{PP}$ exactly when $\mathbf{P}$-comp = $\mathbf{P}$-samp, the following five statements are shown to be equivalent: (1) $\mathbf{P} = \mathbf{PP}$; (2) $\mathbf{P}$-comp = $\#\mathbf{P}$-comp; (3) $\mathbf{P}$-comp = $\mathbf{P}$-samp; (4) $\mathbf{P}$-comp = $\mathbf{IP}$-samp; and (5) $\mathbf{P}$-comp = $\mathbf{IP_1}$-samp.

3 Domination Relation

This section focuses on the $\mathbf{P}$-comp versus $\mathbf{P}$-samp question from the viewpoint of domination relation.

For brevity, we say that $\mathcal{G}$ p-dominates (avp-dominates, resp.) $\mathcal{F}$ if every distribution in $\mathcal{F}$ is p-dominated (avp-dominated, resp.) by some distribution in $\mathcal{G}$ for two sets $\mathcal{F}$ and $\mathcal{G}$ of distributions. We then re-use the symbols $\preceq^{\mathrm{p}}$ and $\preceq^{\mathrm{avp}}$ as *set relations* between two sets of distributions: we write $\mathcal{F} \preceq^{\mathrm{p}} \mathcal{G}$ ($\mathcal{F} \preceq^{\mathrm{avp}} \mathcal{G}$, resp.) to mean that $\mathcal{G}$ p-dominates (avp-dominates, resp.) $\mathcal{F}$. Clearly the set inclusion $\mathcal{F} \subseteq \mathcal{G}$ implies the p-domination $\mathcal{F} \preceq^{\mathrm{p}} \mathcal{G}$; furthermore, the set relations $\preceq^{\mathrm{p}}$ and $\preceq^{\mathrm{avp}}$ are as reflexive and transitive as the set inclusion $\subseteq$.

The following proposition exemplifies the difference between $\subseteq$ and $\preceq^{\mathrm{p}}$: although we do not know whether $\mathbf{P}$-samp $\subseteq$ $\mathbf{IP}$-samp, the domination enables us to show that $\mathbf{P}$-samp $\preceq^{\mathrm{p}}$ $\mathbf{IP}$-samp.

Proposition 3.1 $\mathbf{P}$-samp $\preceq^{\mathrm{p}}$ $\mathbf{IP}$-samp. *More strongly, for every* $\mu \in \mathbf{P}$-samp *and every p-honest function* $f \in \mathbf{FP}$, *there exists a distribution* $\nu \in \mathbf{P}$-samp *such that* $\mu_{f^{-1}} \preceq^{\mathrm{p}} \nu$.

Proof. We first demonstrate that $\mathbf{P}$-samp $\preceq^{\mathrm{p}}$ $\mathbf{IP}$-samp. Let μ be a $\mathbf{P}$-samplable distribution, and let M be a randomized Turing machine witnessing μ with a time-bound polynomial p. We assume that, at every configuration of M, M flips a fair coin. Without loss of generality, p is assumed to be increasing.

We define a function g as follows: Let $g(z)$ be the output x of M on input $0^{2|x|}$ on path z' and in time $p(3|x|)$ if $z = z'1$ and such x exists; let $g(z) = \lambda$ if $z = z'1$ but no such x exists; let $g(z) = z'$ if $z = z'0$. Obviously g is $\mathbf{P}$-computable. On the other hand, since M is a sampling algorithm, we have $\hat{\mu}(x) \leq 2^{-2|x|} + \sum_w \frac{A_x(w)}{2^{|w|}}$, where A_x is the set of all strings w such that, on input $0^{2|x|}$ on the computation path encoded by w, M halts in time $p(3|x|)$ and produces x. Let $\hat{\nu}(x) = \hat{\nu}_{\mathrm{st}}(\{w \mid g(w) = x\})$ and let $q(z) = 8(p(3z)+1)^2 + c_0$, where c_0 is the minimal positive integer such that $c_0\hat{\nu}(\lambda) \geq \hat{\mu}(\lambda)$. It is not difficult to show that $q(|x|)\hat{\nu}(x) \geq \hat{\mu}(x)$. We thus have $\mu \preceq^{\mathrm{p}} \nu$.

For the second part of the proposition, assume that $\mu \in \mathbf{P}$-samp. Following the previous argument, we can choose a distribution μ' from $\mathbf{IP}$-samp such that $\mu \preceq^{\mathrm{p}} \mu'$. It is easy to see that $\mu \preceq^{\mathrm{p}} \mu'$ implies $\mu_{f^{-1}} \preceq^{\mathrm{p}} \mu'_{f^{-1}}$. Now let $\nu = \mu'_{f^{-1}}$. Since f is p-honest, $\mu'_{f^{-1}}$ also belongs to $\mathbf{IP}$-samp.

Thus, we conclude that ν belongs to **IP**-samp. □

We have known that the domination **P**-samp $\preceq^{\rm p}$ **P**-comp is derived immediately from the assumption **P** = **PP**. It is nevertheless possible that **P**-comp p-dominates **P**-samp even if **PP** differs from **P**. In the rest of this section, we discuss the possibility of **P**-samp $\not\preceq^{\rm p}$ **P**-comp.

We begin with the next lemma, which lists several different statements that are equivalent to **P**-samp $\preceq^{\rm p}$ **P**-comp.

Lemma 3.2 *The following statements are equivalent: (1)* **P**-samp $\preceq^{\rm p}$ **P**-comp*; (2)* **IP**-samp $\preceq^{\rm p}$ **P**-comp*; and (3) for every p-honest function* $f \in$ **FP** *and every* $\mu \in$ **P**-comp*, there exists a distribution* ν *in* **P**-comp *and a p-bounded function* p *from* Σ^* *to* $\mathbb{R}^+$ *such that* $\hat{\nu}(y) \geq \sum_{x \in f^{-1}(y)} \frac{\hat{\mu}(x)}{p(x)}$ *for all strings* y*.*

Proof. Since **IP**-samp $\subseteq$ **P**-samp, (1) implies (2). It follows from [4, Lemma 3.3] that (2) is equivalent to (3). The implication from (2) to (1) follows from Proposition 3.1. □

It is known that **P**-samp $\not\preceq^{\rm p}$ **P**-comp if strong one-way functions[‖] exist [2]. We improve this result by showing that a much weaker assumption suffices to reach the same conclusion. To be more precise, it is enough to assume the existence of **NP** sets which are not nearly-**BPP**, which is defined as follows.

Definition 3.3 A set A is *nearly*-**BPP** if, for every polynomial p, there exists a set S and a polynomial-time randomized Turing machine M such that, for each x, (i) $x \in \Sigma^* - S$ implies $\mathrm{Prob}_M[M(x) \neq A(x)] \leq 1/3$ and (ii) $\mathrm{Prob}_{x \in \Sigma^n}[x \in S] < 1/p(n)$ for almost all n. Let **Nearly-BPP** denote the collection of all nearly-**BPP** sets.

The relationship between the strong one-way functions and nearly-**BPP** sets is stated in the following proposition.

Proposition 3.4 *If strong one-way functions exist, then* **NP** $\not\subseteq$ **Nearly-BPP**.

Proof Sketch. A key idea of the proof is to construct a length-regular[**], strong one-way function f which is one-one on most instances with the property $|f(x)| \geq |x|$ for all x [2]. Let A be the set of all strings of the form $xs_i^{|x|}$ such that $f(z) = x$ and the ith bit of z is 1 for some z. Clearly A

[‖] A *(uniform) strong one-way* function is a **P**-computable function f such that, for every polynomial-time randomized Turing machine M, the function $\lambda n.\mathrm{Prob}_{(x,s) \in \Gamma_M^n}[f(M(1^n, f(x); s)) = f(x)]$ is negligible, where Γ_M^n is the set of all pairs (x, s) for which, along with random seed s, M on input x of length n halts.

[**] A function f from Σ^* to Σ^* is *length-regular* if $|f(x)| = |f(y)|$ for all pairs (x, y) for which $|x| = |y|$.

belongs to **NP**. Toward the conclusion, it suffices to show that if A is in nearly-**BPP**, then there exists a randomized Turing machine that "inverts" f with nonnegligible probability. ■

Lemma 3.5, given below, is a crucial lemma to our main theorem. It shows that the assumption **P**-samp $\preceq^{\mathrm{P}}$ **P**-comp helps find all elements of the inverse image $f^{-1}(y)$ of any **P**-computable function f which does not decrease significantly, whenever the set $f^{-1}(y)$ is relatively small.

Lemma 3.5 *Assume* **P**-samp $\preceq^{\mathrm{P}}$ **P**-comp. *Let f be any function in* **FP**, *let k be a positive integer, and let q and q' be any two polynomials with $q(n) \geq 1$ for all n. Assume that $|x| \leq |f(x)| + k \log |f(x)|$ for almost all x. There exist a set S and a deterministic Turing machine M such that (i) $S \subseteq range(f)$; (ii) $|S \cap \Sigma^n| < \frac{2^n}{q(n)}$ for each $n \in \mathbb{N}$; and (iii) M on input $y \in \overline{S}$ correctly lists all elements of $f^{-1}(y)$ (whenever $f^{-1}(y) = \emptyset$, M outputs 0) in polynomial time when $|f^{-1}(y)| \leq q'(|y|)$.*

Proof. A crucial point of the following proof is to define a coding function h which, on input $(1^n, x)$, encodes the output $f(x)$ together with the first n bits, say z, of x if x is in the domain of f so that with the help of the inverse function h^{-1} we can find from y an element x of $f^{-1}(y)$ by depth-first search (asking whether $z0$ or $z1$ is a coded word) with polynomially-many steps.

Formally, the desired function h is defined as follows. Let $h(\langle w, x\rangle)$ be $\langle y, z1\rangle$ if $w \in \{1\}^*$, $|z| = |w|$, $z \sqsubseteq x$, and $f(x) = y$; otherwise, set $h(\langle w, x\rangle) = \langle w, x0\rangle$. Notice that h is defined on all strings and p-honest. Take a distribution μ defined as follows: $\hat{\mu}(w, x) = \hat{\nu}_{\mathrm{st}}(x) \cdot 2^{-2\mathrm{llog}(|w|)-1}$ if $w \in \{1\}^*$, or else 0. Clearly, $\mu_{h^{-1}} \in$ **IP**-samp. Recall that $\hat{\nu}_{\mathrm{st}}(x) \geq \frac{2^{-|x|}}{8(|x|+1)^2}$.

Let n_0 be the minimal positive integer n such that $k \log n \leq n$ and $|x| \leq |y| + k \log |y|$ for all $y \in \mathrm{range}(f)$ of length at least n and all x for which $f(x) = y$.

By our assumption **IP**-samp $\preceq^{\mathrm{P}}$ **P**-comp (equivalent to **P**-samp $\preceq^{\mathrm{P}}$ **P**-comp by Lemma 3.2), there are an $\eta \in$ **P**-comp and a polynomial r with $r(n) \geq q'(n)$ for all n such that $r(|y|)\hat{\eta}(y, z1) \geq \hat{\mu}_{h^{-1}}(y, z1)$ for all y and z. Without loss of generality, we assume that η is strictly **P**-computable. For each $y \in \mathrm{range}(f)$ of length at least n_0, we thus have $\hat{\eta}(y, z1) \geq \frac{2^{-|y|}}{s(|y|)}$ for any initial segment z of each element in $f^{-1}(y)$, where $s(n) = 256r(n)n^{k+4}$. For each y of length at least n_0, let $C_y = \{z \mid \hat{\eta}(y, z1) \geq \frac{2^{-|y|}}{s(|y|)}, |z| \leq |y| + k \log |y|\}$. Notice that C_y, when $|y| \geq n_0$, consists of all initial segments of each element in $f^{-1}(y)$; in particular, $f^{-1}(y) \subseteq C_y$. By the computability of η, there exists a polynomial-time algorithm which recognizes the set $\{(y, z) \mid z \in C_y\}$.

For the desired S, define $S = \{y \mid |C_y| > q(|y|)s(|y|), y \in \mathrm{range}(f)\}$. Clearly we have $S \subseteq \mathrm{range}(f)$. We show that, for all n, $|S^n| < \frac{2^n}{q(n)}$. Assume otherwise, and let y' be an element of S. Let $n = |y'|$. Then we have

$$\sum_{y \in S^n} \sum_{z \in C_y} \hat{\eta}(y, z1) \geq \frac{|S^n| \cdot |C_{y'}| \cdot 2^{-n}}{s(n)} > 1,$$

a contradiction.

Define a Turing machine M as follows. On input y of length at least n_0, by a depth-first search, M computes at most $q(|y|)s(|y|)$ elements z of C_y and lists all these elements z, if any, which satisfy $f(z) = y$, or else M outputs 0. For concreteness, when the length of input y is less than n_0, we design M so that all elements of $f^{-1}(y)$ are encoded into M's program. It is not difficult to show that if $|f^{-1}(y)| \leq q'(|y|)$, $|y| \geq n_0$, and $y \in \overline{S}$, then all elements of $f^{-1}(y)$ are retrieved in polynomial time. This completes the proof. □

Note that the set S in the above proof may not be **P**-computable. To avoid the introduction of S we must assume a stronger assumption. For a further discussion, see Lemma 4.4.

Using the hash function technique, we can show the desired result that **P**-comp cannot p-dominate **P**-samp unless every **NP** set is nearly-**BPP**.

Theorem 3.6 **P**-samp $\not\preceq^{\mathrm{p}}_{\mathrm{D}}$ **P**-comp *unless* **NP** $\subseteq$ **Nearly-BPP**. *More strongly, the following holds: Assume* **P**-samp $\preceq^{\mathrm{p}}$ **P**-comp *and let A be any set in* **NP**. *For every polynomial p with $p(n) \geq 1$ for all $n \in \mathbb{N}$, there exist a set D and a polynomial-time randomized Turing machine M such that $D \subseteq A$, and, for each x, $x \in A - D$ implies* $\mathrm{Prob}_M[M(x) \neq A(x)] < 1/2$, *$x \notin A$ implies* $\mathrm{Prob}_M[M(x) \neq A(x)] = 0$, *and* $\mathrm{Prob}_{x \in \Sigma^n}[x \in D] < 1/p(n)$ *for almost all n.*

Proof. Assume **P**-samp $\preceq^{\mathrm{p}}$ **P**-comp. Take any set A in **NP** and any polynomial q. We want to show that A satisfies the claim. It is sufficient to consider the case that there exists a set $B \in \mathbf{P}$ satisfying $A = \{x \mid \exists z \in \Sigma^{|x|}[xz \in B]\}$. For each x, let B_x be the set of witnesses, $\{z \mid xz \in B \cap \Sigma^{2|x|}\}$, for "$x \in A$." Assume that there exists a nondecreasing polynomial p such that $\mathrm{Prob}_{x \in \Sigma^n}[x \in A] \geq \frac{1}{p(n)}$ for almost all n since, otherwise, the theorem is trivial by choosing $D = \emptyset$.

In order to apply Lemma 3.5, we want to define a function f that maps any element in xB_x to $x0^{|x|}$ (and the others to $x1^{|x|}$) so that, by help of the inverse $f^{-1}(x0^{|x|})$, we can retrieve all witnesses B_z for "$x \in A$" if they exist. In the case that there are always at most polynomially-many witnesses for A, Lemma 3.5 guarantees the existence of an algorithm that computes all witnesses in polynomial time. However, this attempt fails if A has many witnesses in general.

Instead, we use hash function to make such a function f one-one on most inputs. Take the set $H_{n,n+c}$ of hash functions. Define $f(x') = 1xs^n_k hh(y)_{\leftarrow k+c}0^{n-k}$ if $x' = xys^n_k h$ and $y \in B_x$; otherwise $0x'$, where $x \in \Sigma^n$, $h \in H_{n,n+c}$, and $c = \mathrm{ilog}(n)$. Notice that $|x'| \leq |f(x')|$ for all x'. For brevity, write $t(n) = 1 + n + \mathrm{ilog}(n) + (n+1)(n + \mathrm{ilog}(n)) + n + \mathrm{ilog}(n)$.

We show that f is one-one on the fraction of each input set $\Sigma^{t(n)}$. For each k and x of length n, let $g(x) = |f^{-1}(x)|$ and $\rho_{k,x} = \mathrm{Prob}_{(h,w) \in H_{n,n+c} \times \Sigma^{n+c}}[g(1xs^n_k hw_{\leftarrow k+c}0^{n-k}) = 1]$. It suffices to show that $\rho_{k,x} \geq 1/2n$ for almost all n. Now we fix k and x, and assume that $n = |x|$ is sufficiently

large and $\mathrm{ilog}(g(x)) \leq k \leq n$. Consider the case $|B_x| > 0$. The probability $\rho_{k,x}$ is larger than or equal to the probability over all pairs (h, w) that, for each y in B_x, $h(y)_{\leftarrow k+c} = w_{\leftarrow k+c}$, and h k-distinguishes y on B_x. Thus, since $c \leq \log n + 1$, we have

$$\rho_{k,x} \geq |B_x| \cdot 2^{-(k+c)} \cdot (1 - 2^{-c}) \geq (1 - 2^{-c}) \cdot 2^{-c} \geq 2^{-c+1} \geq \frac{1}{2n}.$$

For the case $|B_x| = 0$, clearly $\rho_{k,x} = 1$ since $g(1xs_k^n hw_{\leftarrow k+c}0^{n-k}) = 1$ for all k, h, and w. This yields the desired result $\rho_{k,x} \geq 1/2n$.

Now we apply Lemma 3.5. It follows by this lemma that there are a set S and a polynomial-time deterministic Turing machine N which recognizes $\overline{S}$ such that $S \subseteq \mathrm{range}(f)$ and $|S \cap \Sigma^{t(n)}| < \frac{2^{t(n)}}{4n \cdot q(n)}$. For the desired randomized polynomial-time algorithm M, we define it as follows:

begin randomized algorithm for M
  input x (say, $n = |x|$)
  choose w and h at random ($w \in \Sigma^{n+c}$, $h \in H_{n,n+c}$, $c = \mathrm{ilog}(n)$)
  let *Result* $:= 0$
  for all k $(1 \leq k \leq n)$
    run N on $x'_k = 1xs_k^n hw_{\leftarrow k+c}0^{n-k}$
    let *Result* $:=$ OR of *Result* and $N(x'_k)$
  end-for
  output *Result* and **halt**
end.

For the desired set D, we first write $\delta_{k,x} = \mathrm{Prob}_{(h,w) \in H_{n,n+c} \times \Sigma^{n+c}}[1xs_k^n hw_{\leftarrow k+c}0^{n-k} \in S]$, where $c = \mathrm{ilog}(n)$. Using this $\delta_{k,x}$, we then define $D = \{x \in \Sigma^+ \cap A \mid \exists k[\mathrm{ilog}(|B_x|) \leq k \leq n \wedge \delta_{k,x} \geq \frac{1}{4n} \wedge |x| = n]\}$. We must show that $\mathrm{Prob}_{x \in \Sigma^n}[x \in D] < 1/q(n)$. Assume otherwise. Thus, we have $\delta_{k,x} \geq 1/4n$ for some k $(\mathrm{ilog}(|B_x|) \leq k \leq n)$ and $x \in D^n$. Since

$$\max\{\delta_{k,x} \mid \mathrm{ilog}(|B_x|) \leq k \leq n, x \in \Sigma^n\} \cdot \mathrm{Prob}_{x \in \Sigma^n}[x \in D] \leq \frac{|S^{t(n)}|}{2^{t(n)}} < \frac{1}{4nq(n)},$$

we have $\max\{\delta_{k,x} \mid \mathrm{ilog}(|B_x|) \leq k \leq n, x \in \Sigma^n\} < 1/4n$. This is a contradiction. Therefore, $\mathrm{Prob}_{x \in \Sigma^n}[x \in D] < 1/q(n)$.

Now our final task is to prove that (i) $\mathrm{Prob}_M[M(x) = A(x)] \geq 1/4n$ for all x in $A - D$, and (ii) $\mathrm{Prob}_M[M(x) \neq A(x)] = 0$ for all $x \notin A$. This is enough to establish the theorem because we can amplify its success probability. Take any input x of length n. Let $\rho_x = \mathrm{Prob}_{(h,w) \in H_{n,n+c} \times \Sigma^{n+c}}[A(x) = \mathrm{OR}_{k=1}^n N(x'_k))]$. Note that the probability $\mathrm{Prob}_M[M(x) = A(x)]$ is at least ρ_x. Assume $A(x) = 1$ for a string $x \in \overline{D}$. Note that if $x'_{k'} \in \overline{S}$ and $g(x'_{k'}) = 1$ for some k', then $\mathrm{OR}_{k=1}^n N(x'_k) = 1$. Hence,

$$\rho_x \geq \max\{\rho_{k,x} - \delta_{k,x} \mid \mathrm{ilog}(|B_x|) \leq k \leq n\} \geq \frac{1}{4n}.$$

For the other case $A(x) = 0$, $N(xs_k^n hw_{\leftarrow n+c}0^{n-k}) = 0$ for all h, w, and k; and thus, $\rho_x = 1$. This completes the proof. □

Different from the p-domination, there is no clear evidence for the separation between **P**-comp and **P**-samp with respect to the avp-domination. See Section 5 for more discussion. It is, however,

easy to show that $\mathbf{E}\text{-comp} \not\preceq^{\text{avp}} \mathbf{P}\text{-comp}$. Note that the average-case complexity class $\mathbf{P}_{\mathcal{F}}$ consists of sets recognized in polynomial-time on μ-average for every distribution μ in set $\mathcal{F}$ [17].

Proposition 3.7 $\mathbf{E}\text{-comp} \not\preceq^{\text{avp}} \mathbf{P}\text{-comp}$.

Proof. Assume to the contrary that $\mathbf{E}\text{-comp} \preceq^{\text{avp}} \mathbf{P}\text{-comp}$. This yields the consequence $\mathbf{P}_{\mathbf{P}\text{-comp}} \subseteq \mathbf{P}_{\mathbf{E}\text{-comp}}$. Since $\mathbf{P}_{\mathbf{E}\text{-comp}} = \mathbf{P}$ [17], we obtain $\mathbf{P}_{\mathbf{P}\text{-comp}} = \mathbf{P}$, which clearly contradicts another result in [14, 18] that $\mathbf{P}_{\mathbf{P}\text{-comp}}$ differs from $\mathbf{P}$. Therefore, $\mathbf{E}\text{-comp} \not\preceq^{\text{avp}} \mathbf{P}\text{-comp}$. □

4 Equivalence Relation

In this section, we study the *equivalence classes* of distributions modulo the equivalence relations $\approx^{\text{p}}$ and $\approx^{\text{avp}}$ for which the notion "polynomial on the average" is invariant.

Formally, for a set $\mathcal{F}$ of distributions, $\mathcal{F}/_{\approx^{\text{p}}}$ denotes the collection of all equivalence classes $[\mu]$ for every $\mu \in \mathcal{F}$, where $[\mu] = \{\xi \mid \xi \approx^{\text{p}} \mu\}$, and among such collections the notions of *p-inclusion* $\subseteq^{\text{p}}$ and *p-equality* $\cong^{\text{p}}$ can be naturally introduced: $\mathcal{G} \subseteq^{\text{p}} \mathcal{F}$ means $\mathcal{G}/_{\approx^{\text{p}}} \subseteq \mathcal{F}/_{\approx^{\text{p}}}$ and $\mathcal{F} \cong^{\text{p}} \mathcal{G}$ means $\mathcal{F}/_{\approx^{\text{p}}} = \mathcal{G}/_{\approx^{\text{p}}}$. Similarly, the *avp-inclusion* $\subseteq^{\text{avp}}$ and *avp-equality* $\cong^{\text{avp}}$ are defined by the use of $\approx^{\text{avp}}$ instead of $\approx^{\text{p}}$. Note that these new relations are reflective and transitive; moreover, $\cong^{\text{p}}$ and $\cong^{\text{avp}}$ are symmetric. Obviously, $\mathcal{F} \subseteq \mathcal{G}$ implies $\mathcal{F} \subseteq^{\text{avp}} \mathcal{G}$, and $\mathcal{F} = \mathcal{G}$ implies $\mathcal{F} \cong^{\text{avp}} \mathcal{G}$. Furthermore, $\mathcal{F} \subseteq^{\text{p}} \mathcal{G}$ ($\mathcal{F} \subseteq^{\text{avp}} \mathcal{G}$, resp.) implies $\mathcal{F} \preceq^{\text{p}} \mathcal{G}$ ($\mathcal{F} \preceq^{\text{avp}} \mathcal{G}$, resp.).

As a main theorem of this section, we show that, under the assumption $\mathbf{P} \neq \mathbf{RP}$, $\mathbf{P}\text{-comp} \not\cong^{\text{avp}} \mathbf{IP_1}\text{-samp}$; thus, if $\mathbf{RP}$ differs from $\mathbf{P}$, then $\mathbf{P}$-comp cannot avp-equal $\mathbf{P}$-samp. The following lemma is useful to show our theorem.

Lemma 4.1 *The following two conditions are equivalent: (1)* $\mathbf{P}\text{-comp} \cong^{\text{avp}} \mathbf{IP_1}\text{-samp}$*; and (2) for every p-honest* $f \in \mathbf{FP}$ *and every* $\mu \in \mathbf{P}\text{-comp}$*, there exists* $\nu \in \mathbf{P}\text{-comp}$ *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.

Theorem 4.2 *If* $\mathbf{P} \neq \mathbf{RP}$*, then* $\mathbf{P}\text{-comp} \not\cong^{\text{avp}} \mathbf{IP_1}\text{-samp}$.

Proof. Consider an arbitrary set $A \in \mathbf{RP}$. We want to prove that A belongs to $\mathbf{P}$. By the amplification lemma [13], there is a strictly increasing polynomial p and a set $B \in \mathbf{P}$ such that, for every $x \in \Sigma^n$, $\text{Prob}_{y \in \Sigma^{p(n)}}[xy \notin B] \leq 2^{-n}$ if $x \in A$, and otherwise, $\text{Prob}_{y \in \Sigma^{p(n)}}[xy \in B] = 0$. For each x, let B_x be the set of witnesses for "$x \in A$"; that is, $B_x = \{y \in \Sigma^{p(|x|)} \mid xy \in B\}$.

The key idea of the proof is to define the function f that assigns each witness xy with the value $xA(x)^{p(|x|)}$ so that the probability $\hat{\mu}_{f^{-1}}(x1^{p(|x|)})$ measures the cardinality of the set B_x. Our

assumption ensures $\mu_{f^{-1}}$ can be "approximated" deterministically in polynomial time. Since B_x is either large or empty, we can determine in polynomial time whether $\hat{\mu}_{f^{-1}}(x1^{p(|x|)}) > 0$, which is equivalent to $A(x) = 1$.

We formally define μ by $\hat{\mu}(xy) = \hat{\nu}_{\rm st}(x) \cdot 2^{-p(|x|)}$ if $|y| = p(|x|)$, or else $\hat{\mu}(xy) = 0$. Clearly μ is **P**-computable. Let

$$f(xy) = \begin{cases} xA(x)^{p(|x|)} & \text{if } |y| = p(|x|), \\ xy & \text{otherwise.} \end{cases}$$

By the assumption $\mathbf{P} \neq \mathbf{RP}$ and Lemma 4.1, we have a distribution $\nu \in \mathbf{P}$-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 $\sum_{x \neq \lambda} |x|^{-1} q(x)^{1/k} \hat{\mu}(x) \leq c$ 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 for all y of length $p(|x|)$,

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

where $r(n) = (2n + p(n))^k$. Since $\nu \in \mathbf{P}$-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

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

since $\hat{\nu}_{\rm st}(x) \geq \frac{1}{2n^2 \cdot 2^n} \geq \frac{1}{2^{2n}}$ 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 case that $x \notin A$, $\hat{\nu}(x1^{p(n)}) \leq \sum_{z \in f^{-1}(x1^{p(n)})} q(z) \cdot \hat{\mu}(z) = 0$. 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. □

As another main theorem, we shall show that the assumption $\mathbf{P} \neq \mathbf{NP}$ suffices to reach the desired conclusion $\mathbf{P}$-comp $\not\cong^{\rm p}$ $\mathbf{P}$-samp. We start with the following lemma, which follows from Lemma 3.2.

Lemma 4.3 *The following three conditions are equivalent: (1)* $\mathbf{P}$-comp $\cong^{\rm p}$ $\mathbf{P}$-samp*; (2)* $\mathbf{P}$-comp $\cong^{\rm p}$ $\mathbf{IP}$-samp*; and (3) for every p-honest* $f \in \mathbf{FP}$ *and every* $\mu \in \mathbf{P}$-comp*, there exist a distribution* ν *in* $\mathbf{P}$-comp *and p-bounded functions* p *and* q *from* Σ^* *to* $\mathbb{R}^+$ *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*.*

In addition to the above lemma, we can show that $\mathbf{P}$-comp $\cong^{\rm p}$ $\mathbf{P}$-samp if and only if $\mathbf{P}$-comp $\cong^{\rm p}$

#**P**-comp. To prove this claim, however, we need several results and we shall see the proof at the end of this section.

To show the desired theorem, we would like to have a lemma, stronger than Lemma 3.5, which can be obtained under the stronger assumption that **P**-samp $\subseteq^{\mathrm{p}}$ **P**-comp. Under this assumption, the statement regarding the set S in Lemma 3.5 are eliminated because S turns out to be **P**-computable.

Lemma 4.4 *Assume that* **P**-samp $\subseteq^{\mathrm{p}}$ **P**-comp. *For any set* $B \in \mathbf{P}$ *and any 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* N *such that, for each* $n \in \mathbb{N}$, N *on input* x *in* $S_B \cap \Sigma^n$ *lists all elements of* B_x *(whenever* $B_x = \emptyset$, N *outputs* 0*) in polynomial time.*

Proof. Assume that **P**-samp $\subseteq^{\mathrm{p}}$ **P**-comp. Until the introduction of a set C_y, the proof is similar to that of Lemma 3.5.

We define a p-honest function h as follows: Let $h(\langle w, yx\rangle) = \langle 1y, z\rangle$ if $w = s_k^{|xy|}$ for some k, $|x| = |y|$, $|z| = k$, $z \sqsubseteq x$, and $yx \in B$; otherwise, let $h(\langle w, yx\rangle) = \langle 0y, x\rangle$. Let $\hat{\mu}(w, x) = \hat{\nu}_{\mathrm{st}}(x) \cdot 2^{-2\mathrm{llog}^2(|x|)-1}$ if $w \in \Sigma^{\mathrm{llog}(|x|)}$ and 0 otherwise, where $\mathrm{llog}^2(n) = \mathrm{llog} \circ \mathrm{llog}(n)$.

Since $\mu_{h^{-1}} \in$ **P**-samp, it follows by Lemma 4.3(3) that the assumption **P**-samp $\subseteq^{\mathrm{p}}$ **P**-comp ensures that there are an $\eta \in$ **P**-comp and a nondecreasing polynomial r such that $r(|y| + |z|) \cdot \hat{\mu}_{h^{-1}}(y, z) \geq \hat{\eta}(y, z) \geq \hat{\mu}_{h^{-1}}(y, z)/r(|y| + |z|)$ for all y and z. Denote by $D_{y,z}$ the collection of x such that $z \sqsubseteq x$ and $x \in B_y$. Note that if $y \in S_B$ then $|D_{y,z}| \leq p(|y|)$. Since $\hat{\mu}_{h^{-1}}(y, z) = \frac{|D_{y,z}|}{2^{2\mathrm{llog}^2(2|y|)+1}} \cdot \frac{2^{-2|y|}}{2^{2\mathrm{llog}(2|y|)+1}}$, we obtain $\hat{\eta}(1y, z) \geq \frac{2^{-2|y|}}{r(2|y|+1) \cdot 2^{q(|y|)}}$ when $D_{y,z} \neq \emptyset$, where $q(n) = 2\mathrm{llog}^2(2n) + 2\mathrm{llog}(2n) + 2$.

Let M be a polynomial-time Turing machine which approximates $\hat{\eta}$. Let d be the minimal positive integer such that $3r(2n+1)2^{q(n)} \leq 2^{d \cdot \mathrm{llog}(n)}$ for almost all n. We define a new machine M' as $M'(\langle y, z\rangle) = M(\langle y, z\rangle, 0^{2|y|+d \cdot \mathrm{llog}(|y|)+1})$. Hence, $M'(\langle 1y, z\rangle) > \hat{\eta}(1y, z) - 2^{-2|y|-d \cdot \mathrm{llog}(|y|)-1} > \frac{2 \cdot 2^{-2|y|}}{2^{d \cdot \mathrm{llog}(|y|)}}$. For each $y \in \Sigma^n$, let $C_y = \{z \mid M'(\langle 1y, z\rangle) > \frac{2 \cdot 2^{-2n}}{2^{d \cdot \mathrm{llog}(n)}}, |z| \leq n\}$. Notice that $B_y \subseteq C_y$ if $y \in S_B$ and that the set $\{(z, y) \mid z \in C_y\}$ is in **P**.

To avoid the introduction of a set S, as in Lemma 3.5, we use the other inequality $r(|y| + |z| + 1) \cdot \hat{\mu}_{h^{-1}}(1y, z) \geq \hat{\eta}(1y, z)$, which enables us to prove the following claim: For each $n > 0$ and any $y \in S_B \cap \Sigma^n$, $|C_y| \leq 2^d r(2n+1)^2 p(n)^3 n$. This claim guarantees that if $y \in S_B$ then all elements of B_y are printable in polynomial time by depth-first search for the set C_y. In what follows, we must show the claim above.

Note that if $z \in C_y$, then $\hat{\eta}(1y, z) > \frac{2^{-2|y|}}{2^{d \cdot \mathrm{llog}(n)}}$. For each $n > 0$ and any $y \in S_B \cap \Sigma^n$,

$$\frac{r(2n+1)^2 \cdot 2^{-2n}}{2^{q(n)}} \cdot \sum_z |D_{y,z}| \geq \sum_z r(|y| + |z| + 1) \hat{\mu}_{h^{-1}}(1y, z) \geq \sum_z \hat{\eta}(1y, z) \geq \frac{2^{-2n}}{2^{d \cdot \mathrm{llog}(n)}} |C_y|.$$

Therefore, $|C_y| \leq 2^{d \cdot \mathrm{ilog}(n)-q(n)} \cdot r(2n+1)^2 \cdot \sum_z |D_{y,z}| \leq 2^d r(2n+1)^2 p(n)^3 n$. This completes the proof. □

Theorem 4.5 **P**-comp $\not\cong^{\mathrm{p}}$ **P**-samp *unless* $\mathbf{P} = \mathbf{NP}$.

Proof. Assume that **P**-comp $\cong^{\mathrm{p}}$ **P**-samp. Let A be an arbitrary set in **NP**. It is enough to prove that $A \in \mathbf{RP}$ since $\mathbf{P} = \mathbf{RP}$ follows by Theorem 4.2 from our assumption. Notice that it also suffices to consider a set A only of the form $A = \{x \mid \exists z \in \Sigma^{|x|}[xz \in B]\}$ for some $B \in \mathbf{P}$. Let $B_x = \{z \in \Sigma^{|x|} \mid xz \in B\}$.

The most crucial part of the proof is to randomize by hash functions a witness set B_x so that the density of its corresponding witness set $\hat{B}_{x'}$ is small on most x'. Lemma 4.4 then guarantees the existence of an algorithm that determines whether there is a witness (i.e., $\hat{B}_{x'} \neq \emptyset$). If we run this algorithm on random input x', with high probability we can hit a witness and thus, we can conclude that $x \in A$.

Formally we define:

$$\hat{B} = \{x'z' \mid \exists khxz[x, z \in \Sigma^n \wedge x' = xs_k^n hh(z)_{\leftarrow k+c}0^{n-k} \\ \wedge z' = z10^{|x'|-|z|} \wedge xz \in B \wedge h \in H_{n,n+c} \wedge c = \mathrm{ilog}(n)]\}.$$

Since $B \in \mathbf{P}$, $\hat{B}$ is also in **P**. Let $S_{\hat{B}} = \{x' \mid |\hat{B}_{x'}| \leq 1\}$, where $\hat{B}_{x'} = \{z' \in \Sigma^{|x'|} \mid x'z' \in \hat{B}\}$.

We define $\rho_{k,x} = \mathrm{Prob}_{(h,w) \in H_{n,n+c} \times \Sigma^{n+c}}[xs_k^n hw_{\leftarrow k+c}0^{n-k} \in S_{\hat{B}}]$. Similar to the proof of Theorem 3.6, we can prove that $\rho_{k,x} > 1/2n$ holds for almost all n and for all x with $|B_x| > 0$.

We then apply Lemma 4.4 to the set $S_{\hat{B}}$, and we obtain a polynomial-time deterministic Turing machine N that recognizes $S_{\hat{B}}$. We define the randomized polynomial-time algorithm M as follows:

begin randomized algorithm for M
  input x (say, $n = |x|$)
  choose w, h at random ($w \in \Sigma^n, h \in H_{n,n+c}, c = \mathrm{ilog}(n)$)
  let *Result* $= 0$
  for all k $(1 \leq k \leq n)$
    run N on $x'_k = xs_k^n h_j w$
    let *Result* = OR of *Result* and $N(x'_k)$
  end-for
  output *Result*
end.

To see that M recognizes A, it suffices to prove that $\mathrm{Prob}_M[M(x) = A(x)] \geq 1/2n$ for almost all x since we can amplify its success probability. Take any input x of length n and let $\rho_x = \mathrm{Prob}_{(h,w) \in H_{n,n+c} \times \Sigma^n}[A(x) = \mathrm{OR}_{k=1}^n N(x'_k))]$. Note that the probability $\mathrm{Prob}_M[M(x) = A(x)]$ is at least ρ_x. Our goal now is to prove that $\rho_x \geq 1/2n$.

We must consider two separate cases. First we consider the case $A(x) = 1$. Note that if

$0 < g_{\hat{B}}(x'_{k'}) \leq 1$ for some k', then $\mathrm{OR}_{k=1}^{n} N(x'_k) = 1$. The probability ρ_x is at least the sum of the probability over all pairs (h, w) that, for each $z \in B_x$ and for some k with $\mathrm{ilog}(g_{\hat{B}}(x)) \leq k \leq n$, $h(z)_{\leftarrow k+c} = w_{\leftarrow k+c}$ and h k-distinguishes z on B_x. Hence, $\rho_x > g_B(x) \cdot (1 - 2^{-c}) \cdot 2^{-(k+c)} \geq (1 - 2^{-c})2^{-c} \geq 1/2n$. For the other case $A(x) = 0$, $N(xs_k^n hw_{\leftarrow k+c}0^{n-k}) = 0$ for all triplets (h, w, k); thus $\rho_x = 1$. This completes the proof. □

Toward the end of this section, we show that **P**-samp $\not\cong^{\mathrm{p}}$ #**P**-comp implies **NP** $\not\subseteq$ **BPP**. For its proof, we need so-called $\mathbf{P}_{\mathrm{tt}}^{\mathbf{NP}}$-samplable distributions. A distribution μ is called $\mathbf{P}_{\mathrm{tt}}^{\mathbf{NP}}$-*samplable*[††] if there exist a sampling oracle machine M, a deterministic Turing machine N, and a set $A \in \mathbf{NP}$ such that (i) M with oracle A "approximates" $\hat{\mu}$ in time polynomial in the length of outputs and that (ii) on each input $(0^i, s)$, N lists in polynomial time all query strings of M^A on input 0^i along with computation path p if s is a correct code of path p of M [16]. It is known that #**P**-comp $\subseteq^{\mathrm{p}}$ $\mathbf{P}_{\mathrm{tt}}^{\mathbf{NP}}$-samp [16], where $\mathbf{P}_{\mathrm{tt}}^{\mathbf{NP}}$-samp is the set of all $\mathbf{P}_{\mathrm{tt}}^{\mathbf{NP}}$-samplable distributions.

We are now ready to show the following theorem.

Theorem 4.6 **P**-samp $\not\cong^{\mathrm{p}}$ #**P**-comp *implies* **NP** $\not\subseteq$ **BPP**.

Proof. We note that **P**-samp $\subseteq$ #**P**-comp. It thus suffices to show that #**P**-comp $\subseteq^{\mathrm{p}}$ **P**-samp under the assumption **NP** $\subseteq$ **BPP**. Let us assume **NP** $\subseteq$ **BPP**. Take an arbitrary distribution μ in #**P**-comp. By the result mentioned above, there is a distribution $\nu \in \mathbf{P}_{\mathrm{tt}}^{\mathbf{NP}}$-samp such that μ is p-equivalent to ν. Under our assumption, ν belongs to $\mathbf{P}^{\mathbf{BPP}}$-samp. It is not difficult to show that $\mathbf{P}^{\mathbf{BPP}}$-samp $\cong^{\mathrm{p}}$ **P**-samp, which has a similar flavor to the result that $\mathbf{BPP}^{\mathbf{BPP}} = \mathbf{BPP}$ (see, e.g., [21]). Therefore, there is a distribution ξ in **P**-samp such that $\nu \approx^{\mathrm{p}} \xi$. Hence, $\mu \approx^{\mathrm{p}} \xi$. □

As a corollary, we can show an extension of Lemma 4.3.

Corollary 4.7 **P**-comp $\cong^{\mathrm{p}}$ **P**-samp *if and only if* **P**-comp $\cong^{\mathrm{p}}$ #**P**-comp.

Proof. It suffices to show the "only if" part of the corollary. Assume **P**-samp $\subseteq^{\mathrm{p}}$ **P**-comp. By Theorem 4.5, we have **P** = **NP**. Remember that **NP** $\subseteq$ **BPP** if and only if **NP** = **RP** [6]. Thus, we have **NP** $\subseteq$ **BPP**. By Theorem 4.6, every #**P**-computable distribution is p-equivalent to some distribution that can be sampled by a randomized Turing machine in time polynomial in its output. □

[††]Originally Schuler and Watanabe used an ensemble of conditional distributions but we can easily modify their proof to accommodate a distribution on the infinite set.

5 Further Discussion

We have shown that certain reasonable complexity-theoretic assumptions lead to the separation of **P**-samp from **P**-comp with respect to *domination* and *equivalence*. In this section we shall discuss further results related to our subjects.

We begin with a discussion on the possibility of **P**-samp $\not\preceq^{\rm avp}$ **P**-comp. Since the avp-domination inherently embodies average-case complexity measure, there is a close connection to the average-case complexity class $\mathbf{P}_{\mathcal{F}}$. We present three assumptions that yield the desired conclusion **P**-samp $\not\preceq^{\rm avp}$ **P**-comp. Firstly, if $\mathbf{P} = \mathbf{P}_{\mathbf{P}\text{-samp}}$, then **P**-comp cannot avp-dominate **P**-samp (because **P**-samp $\preceq^{\rm avp}$ **P**-comp together with $\mathbf{P} = \mathbf{P}_{\mathbf{P}\text{-samp}}$ leads to the conclusion $\mathbf{P}_{\mathbf{P}\text{-comp}} = \mathbf{P}$, which contradicts a result in [14, 18]). Regarding the $\mathbf{P} = \mathbf{P}_{\mathbf{P}\text{-samp}}$ question, we note in addition that Theorem 4.6 enables us to prove that, assuming $\mathbf{P}_{\mathbf{P}\text{-samp}} \neq \mathbf{P}$, either $\mathbf{FP}^{\mathbf{E}} \not\subseteq \#\mathbf{P}$ or $\mathbf{NP} \not\subseteq \mathbf{BPP}$ holds.

Secondly, the non-closure property of $\mathbf{P}_{\mathbf{P}\text{-comp}}$ under p-m-reduction (polynomial-time many-one reduction) suffices to conclude that **P**-samp $\not\preceq^{\rm avp}$ **P**-comp. We note that $\mathbf{P}_{\mathbf{P}\text{-comp}}$ is closed downward under increasing p-m-reductions but not under exp-honest p-m-reductions [18]. In contrast, $\mathbf{P}_{\mathbf{P}\text{-samp}}$ is indeed closed downward under p-honest p-m-reductions.

Thirdly, the existence of avp-universal **P**-samplable distribution yields the desired consequence (because **P**-samp $\preceq^{\rm avp}$ **P**-comp implies the existence of avp-universal **P**-computable distributions, which contradicts a result in [15]). Here, a distribution is called *avp-universal for* $\mathcal{F}$ if it is in $\mathcal{F}$ and avp-dominates every distribution in $\mathcal{F}$. We can further weaken this notion and ask whether **P**-comp has an $\mathcal{F}$-universal distribution μ (that is, $\mu \in$ **P**-comp and for every $\nu \in$ **P**-comp, there exists an $f \in \mathcal{F}$ such that $\hat{\nu}(x) \leq f(x)\hat{\mu}(x)$ for all x). If there exists an $O(f)$-universal distribution for **P**-comp, where f is any function in the set $o(2^n)$, then we are able to draw a conclusion that **P** and **NP** are truly different.

At the end, we note that nearly-**BPP** sets are also closely related to the average-case complexity measure in the following fashion: $\mathbf{BPP}_{\mathbf{P}\text{-comp}} \subseteq$ **Nearly-BPP**, where $\mathbf{BPP}_{\mathbf{P}\text{-comp}}$ is the collection of sets A such that, for every $\mu \in$ **P**-comp, there exists a randomized Turing machine M which recognizes A with bounded-error probability in *polynomial-time on μ-average* (i.e., $\sum_x \sum_{s \in \Gamma_M(x)} \frac{\mathrm{Time}_M(x;s)^\delta}{|x|} \cdot \frac{\hat{\mu}(x)2^{-|s|}}{\sum_{s' \in \Gamma_M(x)} 2^{-|s'|}} < \infty$, where $\Gamma_M(x)$ consists of all random seeds s for which, along with s, M on input x halts [1]). Hence, if a strong one-way function exists, then $\mathbf{NP} \not\subseteq \mathbf{BPP}_{\mathbf{P}\text{-comp}}$.

Acknowledgments

The author is grateful to Stephen A. Cook for discussions with him and thanks Osamu Watanabe for pointing out an erroneous statement in an early draft and Ker-I Ko for his helpful comments

on the draft. He also thanks anonymous referees who referred him to Milterson's paper.

References

[1] A. Blass and Y. Gurevich, Randomized reductions of search problems, *SIAM J. Comput.*, **22** (1993), pp.949–975.

[2] 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.

[3] Y. Gurevich, The challenger-solver game: variations on the theme of **P** =?**NP**, *Bulletin of the ETACS*, **39** (1989), pp.112–121.

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

[5] 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.

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

[7] K. Ko and H. Friedman, Computational complexity of real functions, *Theor. Comput. Sci.*, **20** (1982), pp.323–352.

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

[9] P. B. Milterson, The complexity of malign ensembles, *SIAM J. Comput.*, **22** (1993), pp.147–156.

[10] C. H. Papadimitriou, *Computational Complexity*, Addison and Wesley, 1994.

[11] K. W. Regan, Minimum-complexity pairing functions, *J. Comput. System Sci.*, **45** (1992), 285–295.

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

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

[14] R. Schuler, Some properties of sets tractable under every polynomial-time computable distribution, *Information Processing Letters*, **55** (1995), pp.179-184.

[15] R. Schuler, A note on universal distributions for polynomial-time computable distributions, in: *Proceedings, 12th Conference on Structure in Complexity Theory Conference*, pp.69–73, 1997.

[16] R. Schuler and O. Watanabe, Towards average-case complexity analysis of **NP** optimization problems, in: *Proceedings, 10th Conference on Structure in Complexity Theory Conference*, pp.148–159, 1995.

[17] R. Schuler and T. Yamakami, Structural average case complexity, *J. Comput. System Sci.*, **52** (1996), pp.308–327.

[18] R. Schuler and T. Yamakami, Sets computable in polynomial time on average, in: *Proceedings, 1st Annual International Computing and Combinatorics Conference*, Lecture Notes in Computer science, Vol.959, pp.400–409, 1995, Springer-Verlag.

[19] J. Wang and J. Belanger, On the NP-isomorphism problem with respect to random instances, *J. Comput. System Sci.*, **50** (1995), pp.151–164.

[20] T. Yamakami, *Average case computational complexity*, Ph.D. Dissertation, University of Toronto, 1997.

[21] S. Zachos, Probabilistic quantifiers and games, *J. Comput. System Sci.*, **36** (1988), pp.433–451.