Electronic Colloquium on Computational Complexity, Report No. 174 (2015)

Complexity of distributions and average-case hardness*

Dmitry Itsykson † Alexander Knop† Dmitry Sokolov†

Abstract

We address a natural question in average-case complexity: does there exist a language L such that for all easy distributions D the distributional problem (L, D) is easy on the average while there exists some more hard distribution D' such that (L, D') is hard on the average? We consider two complexity measures of distributions: complexity of sampling and complexity of computing the distribution function. The most interesting measure is the complexity of sampling. We prove that for every $0 < a < b$ there exists a language L, an ensemble of distributions D samplable in $n^{\log^b n}$ steps and a linear-time algorithm A such that for every ensemble of distribution F that samplable in $n^{\log^a n}$ steps, A correctly decides L on all inputs from $\{0,1\}^n$ except of a set that has infinitely small F-measure, and for every algorithm B there are infinitely many n such that the set of all elements of $\{0,1\}^n$ for which B correctly decides L has infinitely small D-measure.

In case of complexity of computing the distribution function we prove the following tight result: for every $a > 0$ there exists a language L, an ensemble of polynomial-time computable distributions D, and a linear-time algorithm A such that for every computable in n^a steps ensemble of distributions F, A correctly decides L on all inputs from $\{0,1\}^n$ except a set that has F-measure at most $2^{-n/2}$, and for every algorithm B there are infinitely many n such that the set of all elements of $\{0,1\}^n$ for which B correctly decides L has D-measure at most 2^{-n+1}.

1 Introduction

This paper is devoted to average-case complexity. In the average case settings every computational problem is supplied with a ensemble of distributions on inputs. The problem is easy on the average if it can be solved efficiently on all but a small fraction (according to the distribution) of the inputs.

The paper [6] gave an example of noncomputable ensemble of distributions such that every language with that ensemble of distributions is easy on the average iff it is easy in the worst case. This explains why the average-case complexity studies not all but only feasible ensembles of distributions. The most natural class of ensembles of distributions is the class of polynomial-time samplable distributions. Such distributions are distributions of outputs of polynomial-time randomized algorithms. The second important

*The research is partially supported by the RFBR grant 14-01-00545, by the President's grant MK-2813.2014.1 and by the Government of the Russia (grant 14.Z50.31.0030).

†Steklov Institute of Mathematics at St. Petersburg, 27 Fontanka, St.Petersburg, 191023, Russia, dmitrits@pdmi.ras.ru, aaknop@gmail.com, sokolov.dmt@gmail.com.

ISSN 1433-8092

class of ensembles of distributions is the class of polynomial-time computable ensembles distributions. An ensemble of distributions is computable in polynomial time if its cumulative distribution function is computable in polynomial time. It is known that every polynomial-time computable ensemble of distribution is polynomial-time samplable but the opposite is not true if one-way functions exist [2].

It is well known that several hard problems can be efficiently solved on almost all inputs for some natural distributions. For example the **NP**-complete problem Hamiltonian Path is decidable in a linear time for the ensemble of the uniform distributions on the graphs [3]. Other interesting example is the Graph Isomorphism problem that is solvable in linear time in the case of uniform distribution on the inputs [1], while there exists much more tricky distribution (see for example [8]) such that there are no known efficient algorithms that solve graph isomorphism problem with high probability.

In this paper we study questions of how the complexity of distributions may affect the complexity of the problems. By the complexity of polynomial-time samplable distributions we mean the time complexity of the sampling algorithm. For polynomial-time computable distributions we also consider another complexity measure: the time complexity of the algorithm that computes the distribution function.

Consider the following variant of the time hierarchy theorem: For all $a > 0$ there is a language that can be solved in polynomial time on almost all inputs according to some polynomial-time samplable distribution, while it cannot be solved in time n^a on almost all inputs for all distributions that are samplable in time n^a. For deterministic computations the following statement can be proved by a straightforward diagonalization: For all $a > 0$ and $\epsilon > 0$ there exists a language $L \in \mathbf{P}$ such that $(L, D) \notin \mathrm{Heur}_{1-\epsilon}\mathbf{DTime}[n^a]$ for all ensembles of distributions D.

For randomized algorithms with bounded error the following extension of Pervyshev's [7] heuristic **BPP** hierarchy was noted in [5]:

Theorem ([5])**.** For all $a > 0$ and $\epsilon > 0$ there exists a language L such that the following holds:

- $(L, U) \in \mathrm{Heur}_{\epsilon}\mathbf{BPP}$, where U is an ensemble of uniform distributions;
- $(L, D) \notin \mathrm{Heur}_{\frac{1}{2}-\epsilon}\mathbf{BPTime}[n^a]$ for all ensembles of distributions D that are samplable in $O(n^a)$ steps.

Informally speaking this theorem says that for some languages it is hard to sample easy instances.

In this paper we consider an opposite problem: the complexity of sampling hard instances. Namely we study the following question: does there exist a language L such that for all distributions F of complexity $g(n)$ the distributional problem (L, F) is easy on the average, but there exist an ensemble of distributions D of complexity $f(n)$ such that the distributional problem (L, D) is hard on the average?

- We consider two complexity measures of distributions:
    1. time complexity for sampling;
    2. time complexity of computing the distribution function.

- We say that a distributional problem (L, D) is easy on the average if $(L, D) \in \mathrm{Heur}_{\alpha(n)}\mathbf{P}$, where $\alpha(n) = o(1)$.

- We consider two variants of the notion that (L, D) is hard on the average:

  1. Strong hardness: (L, D) is not in $\mathrm{Heur}_{1-\beta(n)}\mathbf{P}$ for $\beta(n) = o(1)$. We may also consider the randomized analogue of this class: $\mathrm{Heur}_{1-\beta(n)}\mathbf{BPP}$, but we will see that if there exist an appropriate L and D such that (L, D) is not in $\mathrm{Heur}_{1-\beta(n)}\mathbf{P}$ then there exist an appropriate L' and D' such that (L, D) is not in $\mathrm{Heur}_{1-\beta(n)}\mathbf{BPP}$.
  2. Weak hardness: (L, D) is not in $\mathrm{Heur}_{\beta(n)}\mathbf{P}$. In this case it is reasonable to assume that $\alpha(n) = o(\beta(n))$.

- It is desirable for $f(n)$ to be not much larger than $g(n)$. In tight results $g(n)$ would be at most polynomial in $f(n)$; in other results $g(n)$ is bounded by a quasipolynomial in $f(n)$.

1.1 Our results

In Section 4 we consider a complexity of a distribution as the complexity of computing the distribution function. We prove the result for the strong hardness and $f(n) = poly(g(n))$. Namely we prove the theorem:

Theorem 1.1. For every $a > 0$ there exists a language L and an ensemble of polynomial-time computable distributions D such that

- There exists a linear-time algorithm A such that $\Pr_{x \leftarrow F_n}[A(x) \neq L(x)] = O(2^{-n})$ for all F that are computable in $O(n^a)$ steps;

- For every algorithm A and for all n, $\Pr_{x \leftarrow D_n}[A(x) \neq L(x)] > 1 - \frac{1}{2^{n-1}}$.

The most interesting complexity measure of distributions is the complexity of sampling.

In Section 3.1 we consider the statement with the strong notion of hardness. We show that in this case the positive answer to our question is equivalent to the following *hierarchy for sampling distributions*: there exists a distribution D that is samplable in $f(n)$ steps such that for every distribution F that is samplable in $g(n)$ steps, the statistical distance between D and F is at least $1 - o(1)$.

Watson [9] recently proved the similar (but weaker for our goals) theorem:

Theorem ([9]). For all $a > 0$, $\epsilon > 0$ and $k \in \mathbb{N}$ there exists an ensemble $D \in \mathbf{PSamp}$ such that:

- for all n the distribution D_n is concentrated on $\{1, 2, \ldots, k\}$;

- for every ensemble of distributions $F \in \mathbf{Samp}[n^a]$ there exist infinitely many n such that statistical distance between D_n and F_n is at least $1 - \frac{1}{k} - \epsilon$.

Watson left an open problem to prove the following conjecture:

Conjecture 1.1. *For every $a > 0$ there exists $b > 0$ such that the hierarchy for sampling distributions holds for $f(n) = n^b$ and $g(n) = n^a$.*

We prove the hierarchy for sampling distributions for $f(n) = n^{\log^b n}$ and $g(n) = n^{\log^a n}$ for all $0 < a < b$. And we get the following corollary:

Corollary 1.1. For all $\epsilon > 0$ and $c > 0$ there exist a language L and a linear-time algorithm A such that for every polynomial-time samplable ensemble of distributions F and all n, $\Pr_{x \leftarrow F_n}[A(x) = L(x)] \geq 1 - \frac{1}{2^{(\log\log\log n)^c}}$ and there exists $D \in \mathbf{Samp}[n^{\log^\epsilon n}]$ such that $(L, D) \notin \mathrm{Heur}_{1-\frac{1}{2^{(\log\log\log n)^c}}}\mathbf{R}$.

It is interesting to compare Corollary 1.1 with the result of Gutfreund, Shaltiel and Ta-Shma [4]; they proved that for every $\alpha(n) = o(1)$ there is a distribution D that is samplable in quazipolynomial-time such that for every **NP**-complete language L every polynomial-time randomized algorithm fails to compute L with probability at least $\alpha(n)$ for infinitely many n unless $\mathbf{NP} \subseteq \mathbf{BPP}$. In contrast to [4] Corollary 1.1 is unconditional, uses strong notion of hardness and additionally states that L is easy for all polynomial-time samplable distributions, on the other hand the distribution from [4] is the same for all **NP**-complete languages and **NP**-complete languages are important while a language from Corollary 1.1 is an artificial language based on the tricky diagonalization.

In Section 3.2 we consider the weak notion of hardness and $f(n) = poly(g(n))$. We show that in this case the positive answer to our question is equivalent to the following conjecture:

Conjecture 1.2. *There exist infinitely small functions $\beta(n)$ and $\alpha(n) = o(\beta(n))$ such that for all integer $a > 0$ and $b > 0$ there exist an ensemble of distributions $D \in$ **PSamp**, an increasing sequence of integers l_n and a sequence of sets $S_n \subseteq \{0,1\}^{l_n}$ such that the following holds:*

- *$D(S_n) > \beta(l_n)$ for all n;*
- *for all $F \in \mathbf{Samp}[n^a]$, $F(S_n) \leq \alpha(l_n)$ for infinitely many n.*

We prove only the weaker statement, then where $\alpha(n) = \beta(n)$. Namely we prove the following theorem:

Theorem 1.2. For all integer $a > 0$ and $b > 0$ there exist an ensemble of distributions $D \in \mathbf{PSamp}$, a sequence of integers l_n and a sequence of sets $S_n \subseteq \{0,1\}^{l_n}$ such that the following holds:

- $D(S_n) > \frac{1}{l_n^b}$ for all n;
- for all $F \in \mathbf{Samp}[n^a]$, $F(S_n) \leq \frac{1}{l_n^b}$ for infinitely many n.

2 Preliminaries

An ensemble of distributions is a sequence $\{D_n\}_{n=1}^\infty$, where D_n is a probability distribution on $\{0,1\}^n$. Sometimes it is convenient to assume that D_n is concentrated on $\{0, 1, \ldots, 2^n - 1\}$.

For two distributions A and B on $\{0,1\}^n$ the statistical distance between them is $\Delta(A,B) = \max_{S \subseteq \{0,1\}^n} |\Pr_{x \leftarrow A}[x \in S] - \Pr_{x \leftarrow B}[x \in S]|$.

A distributional problem is a pair (L, D) that consists of the language L and the ensemble of distributions D.

Let $\delta : \mathbb{N} \to [0,1]$ be a function. We say that a distributional problem (L, D) is heuristically decidable it time $t(n)$ with error $\delta(n)$ if there exists an algorithm A such that A runs in $O(t(n))$ steps on the inputs on length n and the following holds: $\Pr_{x \leftarrow D_n}[A(x) \neq L(x)] \leq \delta(n)$ for all n. We denote it as $(L, D) \in \text{Heur}_{\delta(n)}\mathbf{DTime}[t(n)]$. We also define a class of distributional problems $\text{Heur}_{\delta(n)}\mathbf{P} = \bigcup_{c>0} \text{Heur}_{\delta(n)}\mathbf{DTime}[n^c]$.

We also define a class $\text{Heur}_{\delta(n)}\mathbf{R}$ that consists all distributional problems (L, D) such that there exists an algorithm A such that $\Pr_{x \leftarrow D_n}[A(x) \neq L(x)] \leq \delta(n)$ for all n.

We say that an ensemble of distributions D is samplable in time $t(n)$ if there exists a randomized algorithm S that on the input 1^n runs in at most $O(t(n))$ steps and $S(1^n)$ is distributed accordingly D_n. The set of all ensembles that are samplable in time $t(n)$ we denote as $\mathbf{Samp}[t(n)]$. We consider the set $\mathbf{PSamp} = \bigcup_{c>0} \mathbf{Samp}[n^c]$ of all polynomial-time samplable ensembles.

3 Samplable distributions

3.1 Strong hardness

Definition 3.1. We say that time constructible functions f and g satisfy the *hierarchy property of sampling distributions* with parameter $\lambda(n)$ if there exists an ensemble of distributions $D \in \mathbf{Samp}[f(n)]$ such that for every ensemble of distributions $F \in \mathbf{Samp}[g(n)]$, there exist infinitely many numbers n such that the statistical distance between D_n and F_n is at least $1 - \lambda(n)$.

Definition 3.2. We say that time constructable functions f and g satisfy the *hierarchy property on complexity of distributional problems* with parameters $\alpha(n) > 0$ and $\beta(n) > 0$ if there exist a language L and an ensemble of distributions $D \in \mathbf{Samp}[f(n)]$ steps such that:

- $(L, F) \in \text{Heur}_{\alpha(n)}\mathbf{P}$ for all $F \in \mathbf{Samp}[g(n)]$;
- $(L, D) \notin \text{Heur}_{1-\beta(n)}\mathbf{P}$.

We say that f and g satisfy *strong* hierarchy property on complexity of distributional problems if the conditions are formulated as:

- there is a linear-time algorithm A such that for all $F \in \mathbf{Samp}[g(n)]$ $\Pr_{x \leftarrow F_n}[A(x) = L(x)] \geq 1 - \alpha(n)$ for all n large enough;
- $(L, D) \notin \text{Heur}_{1-\beta(n)}\mathbf{R}$.

Lemma 3.1. For every time constructible functions $f(n)$, $h(n)$ and $g(n) \geq n$ if f and h satisfy the hierarchy property on sampling distributions with parameter $\lambda(n)$ and $g(n)\log g(n) = o(h(n))$ then f and g satisfy the strong hierarchy property on complexity of distributional problems with parameters $\alpha(n)$ and $\lambda(n)$ for $\alpha(n) = w(\lambda(n))$.

Proof. Let A_i be an enumeration of all randomized algorithms supplied with an alarm clock that interrupt their executions after $O(g(n))$ steps. We will think about A_i as algorithms that sample distributions; that is the output of $A_i(1^n)$ we interpret as a string from $\{0,1\}^n$ by some fixed way. Let F be an algorithm that samples a distribution as follows: on input 1^n with probability $\frac{1}{2}$ it executes $A_1(1^n)$ (and returns its result), with probability $\frac{1}{2^2}$ it executes $A_2(1^n)$, ..., with probability $\frac{1}{2^{n-1}}$ it executes $A_{n-1}(1^n)$ and with probability $\frac{1}{2^{n-1}}$ executes $A_n(1^n)$. Let F define an ensemble of distributions E. It is straightforward that $E \in \mathbf{Samp}[h(n)]$.

Since f and h satisfy the hierarchy property of sampling distributions, there exists an ensemble $D \in \mathbf{Samp}[f(n)]$ such that $\Delta(D_n, E_n) \geq 1 - \lambda(n)$ for infinitely many numbers n. We denote the set of all such n as $I = \{n_1, n_2, \dots\}$. For $n \in I$ there exists a set $S_n \subseteq \{0,1\}^n$ such that $D_n(S_n) - E_n(S_n) \geq 1 - \lambda(n)$, hence $E_n(S_n) \leq \lambda(n)$.

We will define a language L such that $L \subseteq \bigcup\limits_{n \in I} S_n$. Let T_i be an enumeration of all algorithms. We define L such that for every $x \in S_{n_k}$, $x \in L$ if and only if T_k does not stop on the input x or rejects it. By the construction $(L, D) \notin \mathrm{Heur}_{1-\lambda(n)}\mathbf{R}$.

We consider an algorithm that returns 0 on every input. If $R \in \mathbf{Samp}[g(n)]$, then there exists i such that A_i samples R. For $n \geq i$ for every set $S \subseteq \{0,1\}^n$ the following inequality holds: $E(S) \geq 2^{-i}R(S)$. Hence for every ensemble R from $\mathbf{Samp}[g(n)]$ this algorithm has error at most $c\lambda(n)$, where c is a constant that depends only on the ensemble R; $c\lambda(n) < \alpha(n)$ for n large enough. □

We also prove the opposite implication

Lemma 3.2. If f and g satisfy the hierarchy property of complexity of distributional problems with parameters $\alpha(n)$ and $\beta(n)$ then f and g satisfy the sampling hierarchy property with parameter $\alpha + \beta$.

Proof. For all $F \in \mathbf{Samp}[g(n)]$ there exists a polynomial time algorithm A that solves (L, F) in $\mathrm{Heur}_{\alpha(n)}\mathbf{P}$ and also $(L, D) \notin \mathrm{Heur}_{1-\beta(n)}\mathbf{P}$. Let S_n be set of all $x \in \{0,1\}^n$ such that $A(x) = L(x)$. We know that $F_n(S_n) \geq 1 - \alpha(n)$ for all n and $D_n(S_n) \leq \beta(n)$ for for infinitely many n. Hence $\Delta(D_n, F_n) \geq F_n(S_n) - D_n(S_n) \geq 1 - \alpha(n) - \beta(n)$ for infinitely many n. □

Lemma 3.1 and Lemma 3.2 implies that if f and g satisfy the hierarchy property of complexity of distributional problems with two infinitely small parameters then f and $g/\log^2 g$ satisfy the strong hierarchy property on complexity of distributional problems with two infinitely small parameters.

Watson recently proved the following theorem:

Theorem ([9])**.** For all $a > 0$, $\epsilon > 0$ and $k \in \mathbb{N}$ there exists an ensemble $D \in \mathbf{PSamp}$ such that:

- for all n the distribution D_n is concentrated on $\{1, 2, \dots, k\}$;

- for every ensemble of distributions $F \in \mathbf{Samp}[D]n^a$ there exist infinitely many n such that $\Delta(D_n, F_n) \geq 1 - \frac{1}{k} - \epsilon$.

In other words Watson prove that for every $a > 0$, $\epsilon > 0$ and every constant k there exists $b > 0$ such that n^a and n^b satisfy the hierarchy property on sampling distributions with parameter $\frac{1}{k} + \epsilon$. In fact Watson proved the stronger statement since ensemble D is concentrated on k inputs.

Watson conjectured that for every $a > 0$ there exists infinitely small function $\alpha(n)$ there exists $b > 0$ such that n^a and n^b satisfy the hierarchy property on sampling distributions with parameter $\alpha(n)$. This statement is still an open question.

We prove the following theorem:

Theorem 3.1. For every a, b, c such that $0 < a < b$ and $c > 0$ functions $f(n) = n^{\log^b n}$ and $g(n) = n^{\log^a n}$ satisfies the sampling hierarchy property with the parameter $\lambda(n) = \frac{1}{2^{(\log\log\log n)^c}}$.

Corollary 3.1. For every a, b, c such that $0 < a < b$ and $c > 0$ functions $f(n) = n^{\log^b n}$ and $g(n) = n^{\log^a n}$ satisfies the strong hierarchy property on complexity of distributional problems with parameters $\alpha(n) = \beta(n) = \frac{1}{2^{(\log\log\log n)^c}}$.

Proof. Follows from Lemma 3.1 and Theorem 3.1. □

Corollary 1.1. *For all $\epsilon > 0$ and $c > 0$ there exist a language L and a linear-time algorithm A such that for every polynomial-time samplable ensemble of distributions F and all n, $\Pr_{x \leftarrow F_n}[A(x) = L(x)] \geq 1 - \frac{1}{2^{(\log\log\log n)^c}}$ and there exists $D \in \mathbf{Samp}[n^{\log^\epsilon n}]$ such that $(L, D) \notin \mathrm{Heur}_{1-\frac{1}{2^{(\log\log\log n)^c}}}\mathbf{R}$.*

Before giving a formal proof of Theorem 3.1 we present an idea of the proof.

In the follows we assume that random variables and elements of ensembles of distributions take values from the set $\{0, 1, \ldots, 2^n - 1\}$ instead of $\{0, 1\}^n$.

Our proof like a proof of the Watson's theorem is based on the tree-like diagonalization. We construct a distribution D and diagonalize over all distributions samplable in $O(g(n))$ steps by the enumeration of their generators A_i. For the i-th distribution we will prove that the statistical distance between D and $A_i(1^n)$ is large for some n from $[n_i, n_i^*]$, where n_i^* is significantly more than n_i. For every i we construct a tree T_i with vertices uniquely marked with numbers from $[n_i, n_i^*]$. The root of T_i is marked by n_i^* and leaves of T_i are marked with numbers that are about n_i. The number of a parent is greater then the number of a child also the number of a parent is bounded by a quazipolynomial in numbers of its child. Let t be an element from $\{0, 1, \ldots, 2^{n_i} - 1\}$ such that in all leaves A_i-probability of t is less then $\lambda(m_i)$, where m_i is the maximum leaf. Such t exists since there are not too many leaves, the possible values of distributions is at least 2^{n_i} and for every distribution the number of elements with probability at least $\lambda(n)$ is at most $\frac{1}{\lambda(n)}$. The distribution $D_{n_i^*}$ is concentrated on t. We assume that for all $n \in [n_i; n_i^*]$ the statistical distance between $A_i(1^n)$ and D_n is less then $1 - \lambda(n)$. Our goal is to define D in such a way that in at least one leaf D is concentrated on t. This will contradict our assumption and the definition of t.

We will transmit information about t from a parent to at least one of its children. The distribution D on the children of p has the following property: if D_p is concentrated (with probability $1-\epsilon$) on some element, then D_n is concentrated on the same element for at least one child n of p. From the assumption about statistical distances we have that $\Pr[A_i(1^p)=t] \geq \lambda(p)-\epsilon$, hence there are at most $\frac{2}{\lambda}$ candidates on the role of t if we have an access to $A_i(1^p)$. We generate a list of all elements with $A_i(1^p)$-probability at least $\lambda(p)-\epsilon$. In the first child of p we make D concentrated on the first element of the list, on the second child on the second element and so on. There is a problem that there are possibly different lists will be generated in different children; we solve this problem by using several thresholds for frequencies. Formally we do it in the following lemma:

Lemma 3.3. There is an algorithm $C^{\bullet}(n,i,\delta,\lambda)$ that has an oracle access to some random variable γ taking values in $\{0,1,\ldots,2^n-1\}$ such that for all positive integer n and $\delta,\lambda \in (0,1]$ if $\Pr[\gamma=t] \geq \lambda$ for some t, then there is some integer $0 \leq i \leq \lceil 1+\frac{1}{\lambda}\rceil^2$ such that $\Pr[C^{\gamma}(n,i,\delta,\lambda)=t] \geq 1-\delta$ and $C^{\bullet}$ runs at most $\mathrm{poly}(n,\log\frac{1}{\delta},\frac{1}{\lambda})$ steps.

Proof. Consider the following algorithm $C^{\gamma}(n,i,\delta,\lambda)$:

1. Let $k=\lceil\frac{1}{\lambda}+1\rceil$ and $\epsilon=\frac{\lambda^3}{10k}$;
2. We interpret i as a pair (a,b), where $a,b\in[k]$;
3. Request the oracle for $N=\lceil\frac{2(n+1+\log\frac{1}{\delta})}{\epsilon^2}\rceil$ samples of γ;
4. Consider the list $y_1,\ldots,y_s$ of all elements with frequency at least $\lambda-\epsilon a$;
5. Return y_b if $b\leq s$ or 0 otherwise.

Note that for $\lambda\in(0,1]$

$$k(\lambda-\epsilon(2k)) \geq (\frac{1}{\lambda}+1)(\lambda-\lambda^3/5) = 1+\lambda-\lambda^2/5-\lambda^3/5 > 1. \quad (1)$$

Hence the number of elements x such that $\Pr[\gamma=x]>\lambda-\epsilon k$ is less than k; by the similar reasons $s<k$, where s the size of the list in the 4-th step of the algorithm C.

Consider intervals $I_j=[\lambda-\epsilon j-\epsilon/2;\lambda-\epsilon j+\epsilon/2]$. There is $a\in[k]$ such that $\Pr[\gamma=x]\notin I_a$ for all x since otherwise $1=\sum_x \Pr[\gamma=x]\geq k(\lambda-\epsilon k-\epsilon/2)$ that contradicts inequality (1). Hence there is $a\in[k]$ such that $|\Pr[\gamma=x]-\lambda-\epsilon a|>\epsilon/2$ for all x.

Let $x_1,\ldots,x_l$ be the list of all elements x such that $\Pr[\gamma=x]>\lambda-\epsilon a$. We know that if $\Pr[\gamma=x]>\lambda-\epsilon a$, then $\Pr[\gamma=x]>\lambda-\epsilon a+\epsilon/2$ and also if $\Pr[\gamma=x]\leq\lambda-\epsilon a$, then $\Pr[\gamma=x]<\lambda-\epsilon a-\epsilon/2$. For given a for every $j\in[l]$, x_j appears in the list from 4th step of algorithm C with probability at least $1-2e^{-\epsilon^2N/2}$. If $\Pr[\gamma=x]\leq\lambda-\epsilon a$ then by Chernoff bound x does not appear in the list from the 4th step of the algorithm C with probability at least $1-2e^{-\epsilon^2N/2}$. Since γ is concentrated on the set of size 2^n with probability at least $1-2^{n+1}e^{-\epsilon^2N/2}\geq 1-\delta$ the list generated on 4th step of algorithm C is precisely the list $x_1,\ldots,x_l$. Since $\Pr[D=t]>\lambda$, there is b such that $x_b=t$. Hence if $i=(a,b)$ then $\Pr[C^{\gamma}(n,i,\delta,\lambda)=t]\geq 1-\delta$. $\square$

Proof of Theorem 3.1. Our proof is based on the tree-like delayed diagonalization. We diagonalize against all randomized algorithms supplied with a $O(g(n))$-alarm clock, we interpret them as samplers of distributions. Let $A_1, A_2, \dots$ be an enumeration of all randomized algorithms supplied with a $O(g(n))$-alarm clock.

Let us consider an $\epsilon > 0$ such that $(1 + a)(1 + \epsilon) < (1 + b)$ and fix some c. We define integer sequences n_i and n_i^* such that $n_1 = 1$, $n_i^* = 2^{(\log n_i)^{(1+\epsilon)^{d_i}}}$, where $d_i = \lceil \log_{1+\epsilon} 2 \rceil \lceil (\log \log n_i)^2 \rceil$ and $n_{i+1} = n_i^* + 1$. For every i we define an ensemble of distributions D_n for $n \in \{n_i, n_i+1, \dots, n_i^*\}$ such that there exists $k \in \{n_i, n_i+1, \dots, n_i^*\}$ such that $\Delta(D_k, A_i(1^k)) \geq \lambda(k)$.

Lemma 3.4. For every $\epsilon > 0$ there exists a family of trees T_i such that

1. The set of vertices of T_i is a subset of $\{n_i, n_i + 1, \dots, n_i^*\}$.
2. n_i^* is the root of T_i.
3. All leaves of T_i have numbers at most $m_i = 2n_i$.
4. The depth of T_i is $d_i = \lceil \log_{1+\epsilon} 2 \rceil \lceil (\log \log n_i)^2 \rceil$.
5. If p is a parent of n then $p \leq 2^{\log^{1+\epsilon} n}$.
6. There is an algorithm that for any vertex n of T_i outputs the parent p of n and the number of children of p that are less than n in $poly(n)$ steps.
7. For every inner vertex v of T_i, v has $k = \lceil \frac{1}{\lambda(n_i^*)} + 1 \rceil^2$ children.

Proof. Let us denote $\delta = \lceil \log_{1+\epsilon} 2 \rceil$.

We define a tree T_i as a complete balanced tree with depth d_i. The number of leaves in the tree can be estimated as follows: $k^{d_i} \leq (2^{(\log\log\log n_i^*)^{3c}})^{\delta(\log\log n_i)^2} \leq (2^{(\log\log n_i)^{12c}})^{\delta(\log\log n_i)^2} = 2^{\delta(\log\log n_i)^{24c}} \leq n_i$.

The root n_i^* is the only vertex on the zero level. There are exactly k^s vertices on s-th level. Let $a_{i,j} = 2^{(\log n_i)^{(1+\epsilon)^j}}$, where $j \in \{0, 1, 2, \dots\}$. Vertices of T_i on level $(d_i - s)$ are $[a_{i,s}; a_{i,s} + k^{d_i - s} - 1]$.

Note that $a_{i,s+1} - a_{i,s} \geq a_{i,1} - a_{i,0} = 2^{(\log n_i)^{(1+\epsilon)}} - n_i \geq 2^{(\log n_i)^{(1+\epsilon)} - 1} > n_i \geq k^{d_i} \geq k^{d_i - s}$. Hence on all levels there is enough place for vertices.

The parent of j-th vertex on s-th level has number $\lfloor \frac{j}{k} \rfloor$. Let $h(n) = n^{\log^\epsilon n}$. Since $h(n + k) \geq h(n) + k$ we have $h(2^{\log^{(1+\epsilon)^s} n_i} + j) \geq h(2^{\log^{(1+\epsilon)^s} n_i}) + j \geq 2^{\log^{(1+\epsilon)^{s+1}} n_i} + j/k$, therefore the property 5 is satisfied. The verification of other properties is straightforward. $\square$

Now we describe an algorithm that samples D_n for $n \in \{n_i, \dots, n_i^*\}$ in $O(f(n))$ steps.

1. If $n = n_i^*$ then output the minimal $t_i \in \{0, 1, \dots, 2^{n_i} - 1\}$ such that for all $l \in [n_i; m_i]$ we have that $\Pr[A_i(1^l) = t_i] < \lambda(n_i)/2$. Such t_i indeed exists since for every l there are at most $\frac{2}{\lambda(n_i)}$ elements with $A_i(1^l)$-probability at least $\lambda(n_i)/2$ and $\frac{2}{\lambda(n_i)} m_i \leq 2^{n_i}$. Such t_i can be found in at most $m_i c_i g(m_i) 2^{c_i g(m_i)}$ steps by brute force search over all possible random bits, where c_i is a constant that depends on i.

$$m_i c_i g(m_i) 2^{c_i g(m_i)} \leq 2^{m_i g(m_i)} \leq 2^{2n_i g(2n_i)} \leq 2^{2n_i (2n_i)^2 \log^a n_i} <$$
$$2^{2^{4\log^{(1+a)} n_i}} \leq 2^{2^{2^{4(1+a)\log\log n_i}}} \leq 2^{2^{2^{(\log\log n_i)^2}}} < n_i^* = o(f(n_i^*))$$

2. If n is not a vertex of T_i then return 0.

3. Otherwise, let p be the parent of n and j is a number of n in the list of all children of p. By the property of T_i, $p \leq 2^{\log^{1+\epsilon} n}$ and such p can be found in $poly(n)$ steps. We return $C^{A_i(1^p)}(p, j, \lambda(n)/2, \lambda(p)/2)$, where C is the algorithm from Lemma 3.3. By Lemma 3.3 C runs at most poly(p) steps and on every step the simulation of $A_i(1^p)$ occupies at most $c_i g(p)$ steps. Note that $c_i g(p)\mathrm{poly}(p) < 2^{2\log^{a+1} p} < 2^{2\log^{1+a}(2^{\log^{1+\epsilon} n})} = 2^{2\log^{(1+a)(1+\epsilon)} n} < 2^{\log^{(1+b)} n} = f(n)$.

For the sake of contradiction we assume that for all $n \in \{n_i, \dots, n_i^*\}$, $\Delta(D_n, A_i(1^n)) < 1 - \lambda(n)$. By induction on the level s of T_i we prove that there is a vertex v of level s in T_i such that $D_v(t_i) \geq 1 - \lambda(v)/2$. If $D_v(t_i) \geq 1 - \lambda(v)$ for some leaf v then $\Pr[A_i(1^v) = t_i] \geq (1 - \lambda(v)/2) - (1 - \lambda(v)) = \lambda(v)/2$ but we define t_i such that $\Pr[A_i(1^v) = t_i] < \lambda(v)/2$. Hence we will get a contradiction in leaves.

The base of induction follows from the construction of $D_{n_i^*}$. Let us prove the inductive step from s to $s+1$. Let v be a vertex of level s such that $D_v(t_i) \geq 1-\lambda(v)/2$. If v is a leaf then we are done. Otherwise $\Pr[A_i(1^v) = t_i] > \lambda(v)/2$ since $\Delta(D_v, A_i(1^v)) < 1 - \lambda(v)$. Hence by Lemma 3.3 there is a child u with number j among the all children of v such that $\Pr[C^{A_i(1^v)}(v, j, \lambda(u)/2, \lambda(v)/2)] > 1 - \lambda(u)/2$. □

Our proof in contrast to Watson's proof does not use error correcting codes with list decoding. This is because we find one element that has small probability for all leaves of the tree. This trick was impossible in Watson's case since all distributions was concentrated on constant number of points. In Watson's proof there were a lot of information transmitted from the root to leaves, and parts of this information was stored in different vertices. Watson used list error decoding codes in order to prevent information distortion.

Now we show why this approach cannot be adapted to the case of $g(n) = n^a$ and polynomial $f(n)$. The problem is the following: for nonconstant $\lambda(n)$ the tree T_i should have nonconstant degree: every inner vertex has at least k_i children, where k_i goes to infinity. In the root of the tree we have to make exponential in any leaf number of steps; and the parent of every node n should be at most polynomial of every children. Thus for every leaf l the distance between root and l is at least $\Omega(\log \ell)$. Let m_i be the leaf with the maximal number; then the distance from the root to m_i is at least $L = \Omega(\log m_i)$. Let S be the set of vertices such that their numbers are less then m_i but the numbers of their parents are more then m_i. Note that all vertices on the distance L from the root must either be in S or have a descendent in S. Therefore the size of S should be at least k_i^L that is greater then m_i for large i, since k_i goes to infinity. But this is a contradiction since S is set of vertices with numbers less then m_i.

3.2 Weak hardness

In this section we consider statement of the problem with the weak notion of hardness and tight hierarchy ($f(n) = poly(g(n))$). We start from equivalent formulations:

Proposition 3.1. The following statements are equivalent:

1. There exists infinitely small functions $\beta(n)$ and $\alpha(n) = o(\beta(n))$ such that for all $a > 0$ there exists an ensemble of distributions $D \in \mathbf{PSamp}$ and a language L such that the following holds:

   - $(L, F) \in \mathrm{Heur}_{\alpha(n)}\mathbf{P}$ for all $F \in \mathbf{Samp}[n^a]$;
   - $(L, D) \notin \mathrm{Heur}_{\beta(n)}\mathbf{P}$.

2. There exists infinitely small functions $\beta(n)$ and $\alpha(n) = o(\beta(n))$ such that for all $a > 0$ and $b > 0$ there exist an ensemble of distributions $D \in \mathbf{PSamp}$, an increasing sequence of integers l_n and a sequence of sets $S_n \subseteq \{0,1\}^{l_n}$ such that the following holds:

   - $D(S_n) > \beta(l_n)$ for all n;
   - For all $F \in \mathbf{Samp}[n^a]$, $F(S_n) \leq \alpha(n)$ for infinitely many n.

3. There exists infinitely small functions $\beta(n)$ and $\alpha(n) = o(\beta(n))$ such that for all $a > 0$ there exists an ensemble of distributions $D \in \mathbf{PSamp}$ and a language L such that the following holds:

   - There exists linear-time algorithm A such that for all $F \in \mathbf{Samp}[n^a]$, $\Pr\limits_{x \leftarrow F_n}[L(x) = A(x)] \geq 1 - \alpha(n)$ for all n large enough;
   - $(L, D) \notin \mathrm{Heur}_{\beta(n)}\mathbf{R}$.

Proof. Note that if $\alpha(n) = o(\beta(n))$, then $\alpha(n) = o(\sqrt{\alpha(n)\beta(n)})$ and $\sqrt{\alpha(n)\beta(n)} = o(\beta(n))$.

$1 \to 2$. We apply statement 1 to $a' = 2a$. Let A_i be an enumeration of all algorithms with an alarm-clock $n^{1.5a}$. We will think about A_i as algorithms that samples distributions; that is the output of $A_i(1^n)$ we interpret as a string from $\{0,1\}^n$ by some fixed way. Let F be an algorithm that samples an ensemble of distributions as follows: on input 1^n with probability $\frac{1}{2}$ it executes $A_1(1^n)$ (and returns its result), with probability $\frac{1}{2^2}$ it executes $A_2(1^n)$, ..., and with probability $\frac{1}{2^{n-1}}$ it executes $A_{n-1}(1^n)$ and with probability $\frac{1}{2^{n-1}}$ executes $A_n(1^n)$. Let F define an ensemble of distributions E. It is straightforward that $E \in \mathbf{Samp}[n^{a'}]$. Statement 1 implies that $(L, E) \in \mathrm{Heur}_{\alpha(n)}\mathbf{P}$. Let an algorithm T decide (L, E) in the class $\mathrm{Heur}_{\alpha(n)}\mathbf{P}$. We denote $S_n = \{x \in \{0,1\}^n \mid T(x) \neq L(x)\}$. $E_n(S_n) \leq \alpha(n)$ for all n, hence for every ensemble of distributions $R \in \mathbf{Samp}[n^a]$ there exists a constant C such that $R_n(S_n) \leq C\alpha(n)$ that is less then $\sqrt{\alpha(n)\beta(n)}$ for n large enough. Since $(L, D) \notin \mathrm{Heur}_{\beta(n)}\mathbf{P}$, we have that $D(S_n) > \beta(n)$. Hence we find sequence of integers l_n and a sequence of sets $S_n \subseteq \{0,1\}^{l_n}$ such that the following holds:

- $D(S_n) > \beta(l_n)$ for all n ;
- For all $R \in \mathbf{Samp}[n^a]$, $R(S_n) \leq \sqrt{\alpha(n)\beta(n)}$ for n large enough.

$2 \to 3$. The proof is analogous to the end of the proof of Lemma 3.1. We apply statement 2 to $a' = a+1$. Let A_i be an enumeration of all algorithms with a $O(n^a)$-alarm-clock; we interpret them as samplers of ensembles of distributions. Let F be an algorithm that samples an ensemble of distribution as follows: on input 1^n with probability $\frac{1}{2}$ it executes $A_1(1^n)$ (and returns its result), with probability $\frac{1}{2^2}$ it executes $A_2(1^n)$, ..., with probability $\frac{1}{2^{n-1}}$ it executes $A_{n-1}(1^n)$ and with probability $\frac{1}{2^{n-1}}$ executes $A_n(1^n)$. Let F define an ensemble of distributions E. It is straightforward that $E \in \mathbf{Samp}[n^{a+1}]$. Let D and S_n be from statement 2 for $a' = a+1$. We define a set $I = \{n_1, n_2, \dots\}$ that consists of all n such that $E_n(S_n) \leq \alpha(n)$.

We will define a language L such that $L \subseteq \bigcup\limits_{n \in I} S_n$. Let T_i be an enumeration of all algorithms. We define L such that for every $x \in S_{n_k}$, $x \in L$ if and only if T_k does not stop on the input x or rejects it. By the construction $(L, D) \notin \mathrm{Heur}_{\beta(n)}\mathbf{R}$.

We consider an algorithm that returns 0 on every input. If $R \in \mathbf{Samp}[n^a]$, then there exists i such that A_i samples R. For $n \geq i$ for every set $S \subseteq \{0,1\}^n$ the following inequality holds: $E(S) \geq 2^{-i}R(S)$. Hence for every ensemble R from $\mathbf{Samp}[n^a]$ this algorithm has error at most $c\alpha(n)$, where c is a constant that depends only on the ensemble R; $c\alpha(n) < \sqrt{\alpha(n)\beta(n)}$ for n large enough.

$3 \to 1$. This implication is straightforward. $\square$

We prove the statement that is weaker then statement 2 from Proposition 3.1. Namely we prove it in the case $\alpha(n) = \beta(n) = \frac{1}{n^b}$. By the similar way it is possible to prove it for other infinitely small functions: $\frac{1}{2^n}$, $\frac{1}{\log n}$ etc.

Theorem 1.2. *For all integer $a > 0$ and $b > 0$ there exist an ensemble of distributions $D \in \mathbf{PSamp}$, a sequence of integers l_n and a sequence of sets $S_n \subseteq \{0,1\}^{l_n}$ such that the following holds:*

- *$D(S_n) > \frac{1}{l_n^b}$ for all n;*
- *For all $F \in \mathbf{Samp}[n^a]$, $F(S_n) \leq \frac{1}{l_n^b}$ for infinitely many n.*

Proof. Consider an enumeration of all algorithms F_i with alarm clock n^{a+1} such that every algorithm appears infinitely many times in this enumeration; we consider F_i as samplers of distributions. We define integer sequences n_i and n_i^* such that $n_1 = 1$, $n_i^* = 2^{n_i^a}$, and $n_{i+1} = 2n_i^*$.

Split all strings of length n on n^b nonempty sets; we call them intervals and denote by $T_{j,n}$ for $j \in \{1, 2, \dots, n^b\}$. For $n \in [n_i; n_i^*]$ we define a graph (it will be a forest) as follows:

- The set of vertexes of the graph is the set of all intervals $T_{j,n}$ for $n = 2^k$ and $n_i \leq n \leq n_i^*$;
- All elements of T_{j,n_i} are roots of trees of the forest;
- For $n \in \{n_i, 2n_i, 4n_i, \dots, n_i^*/2\}$, $T_{j,n}$ has 2^b children: $\{T_{j',2n} \mid 2^b(j-1) \leq j' \leq 2^b j - 1\}$.
- All elements of T_{j,n_i^*} are leaves of trees of the forest;

We define a sampler for D as follows. It gets on the input 1^n:

- if $n = n_i^*$ for some i, then find an interval T_{j,n_i} with the smallest probability according to $F_i(1^{n_i})$. If there are several such T_{j,n_i}, we take one with the minimal j. (Note that this can be done in $poly(n_i^*)$ time by bruteforce). Then chose random descendent of T_{j,n_i} on length n_i^* and return some string form this descendent. Note that $\Pr[F_i(1^{n_i}) \in T_{j,n_i}] \leq \frac{1}{n_i^b}$.

- if $n_i \le n < n_i^*$ for some i, then run $F_i(1^{2n})$ and if the result belongs to a descendent of $T_{j,n}$ for some j, then return random string from $T_{j,n}$.

Let us prove that for all i there exists j and $n \in [n_i; n_i^*]$ such that $\Pr[D(1^n) \in T_{j,n}] > \frac{1}{n^b}$ and $\Pr[F_i(1^n) \in T_{j,n}] \le \frac{1}{n^b}$. (This will conclude the proof of the theorem if we choose $S_i = T_{j,n}$.) Assume the opposite; that is for all j and $n \le n_i^*$ if $\Pr[F_i(1^n) \in T_{j,n}] \le \frac{1}{n^b}$, then $\Pr[D(1^n) \in T_{j,n}] < \frac{1}{n^b}$. Let T_{j,n_i} be an interval with the smallest probability according to $F_i(1^{n_i})$, hence $\Pr[F_i(1^{n_i}) \in T_{j,n_i}] \le \frac{1}{n_i^b}$. By induction on l we prove that for all $n = 2^l n_i$ (and $n \le n_i^*$) there exists k such that $T_{k,n}$ is a descendant of T_{j,n_i} and $\Pr[D(1^n) \in T_{k,n}] \le \frac{1}{n^b}$. The base case $l = 0$ is already proved. Let us prove the inductive step from l to $l+1$. Let $n = 2^l n_i$. Assume that $\Pr[D(1^n) \in T_{k,n}] \le \frac{1}{n^b}$ then by the pigeonhole principle and construction of D there is one of 2^b children of $T_{k',2n}$ such that $\Pr[F_i(1^{2n}) \in T_{k',2n}] \le \frac{1}{(2n)^b}$ and hence by assumption $\Pr[D(1^{2n}) \in T_{k',2n}] \le \frac{1}{(2n)^b}$. Therefore there exists k such that $\Pr[D(1^{n_i^*}) \in T_{k,n_i^*}] \le \frac{1}{(n_i^*)^b}$ and T_{k,n_i^*} is a descendant of T_{j,n_i}, but the construction of D implies that the D-probability of every descendant of T_{j,n_i} on length n_i^* is equal to $\frac{n_i^b}{(n_i^*)^b} > \frac{1}{(n_i^*)^b}$. □

Corollary 3.2. For all $a > 0$ and $b > 0$ there exists a ensemble of distributions $D \in$ **PSamp** and a language L such that the following holds:

- There exists a linear-time algorithm A such that $\Pr_{x \leftarrow F_n}[A(x) \ne L(x)] = O(\frac{1}{n^b})]$ for all $F \in \mathbf{Samp}[n^a]$;
- $(L, D) \notin \mathrm{Heur}_{\frac{1}{n^b}}\mathbf{R}$.

Proof. The theorem follows from Theorem 1.2 by the argument that is analogous to the proof of implication $2 \to 3$ of Proposition 3.1. □

4 Computable distributions

Ensemble of distributions D_n is computable in time $t(n)$ if for all n probabilities of all elements according to D_n are binary rational numbers and there exists an algorithm $A(x)$ that runs in $O(t(|x|))$ steps and computes the cumulative distribution function of D_n (i.e. $\sum_{y \le x} D_n(x)$, where $\le$ is lexicographical order). The set of all ensembles that are computable in time $t(n)$ we denote as $\mathbf{Comp}[t(n)]$. The set $\mathbf{PComp} = \bigcup_{c>0} \mathbf{Comp}[n^c]$ is the set of all ensembles computable in polynomial time.

Lemma 4.1. If an ensemble $D \in$ **PSamp** and for all n the distribution D_n is concentrated on one element, then $D \in$ **PComp**.

Proof. In order to compute the distribution function it is sufficient to find an element x_0 with D-probability 1 and compare the given input with x_0. If we execute the sampling algorithm using all zeros instead of random bits, then its result would be x_0. □

Now we prove the statement that is analogous to hierarchy property of n^a and n^b of sampling distributions but for computable distributions.

Proposition 4.1. For all $a > 0$ there exists an ensemble $D \in \mathbf{PComp}$ such that for all ensembles $F \in \mathbf{Comp}[n^a]$ there are infinitely many numbers n such that $\Delta(D_n, F_n) \geq 1 - 2^{-n}$.

Proof. Let A_i be enumeration of all deterministic algorithms supplied with alarm clock n^{a+1} such that every algorithm appears as A_i infinitely many times. We interpret A_i as algorithms that compute distribution functions, however among such algorithms there are possibly incorrect ones. If A_i corresponds to some distribution F, then we define D_i such that $\Delta(D_i, F_i) \geq 1-2^{-i}$. To do this we concentrate D_i on the input with probability at most 2^{-i} according F_i. If A_i is incorrect then D_i would be concentrated on some input. The result distribution would be polynomial time computable by Lemma 4.1.

Now we describe how to find an element with F_i-probability at most 2^{-i}. We use the binary search. If A_i does not correspond to any distribution, then we either understand this during the realization of the binary search, in this case we stop and return 0^i, or apply binary search and find an element $x \in \{0,1\}^i$ such that $A_i(x) - A_i(x') \leq 2^{-i}$, where x' is lexicographical predecessor of x (if $x = 0^i$, then assume that $A_i(x') = 0$). □

Now we prove a statement that is similar to hierarchy property of n^a and n^b on complexity of distributional problems but for computable distributions.

Theorem 1.1. *For every $a > 0$ there exists language L and ensemble of distributions $D \in \mathbf{PComp}$ such that*

- $(L, F) \in \mathrm{Heur}_{O(\frac{1}{2^n})}\mathbf{DTime}[n]$ *for all* $F \in \mathbf{Comp}[n^a]$*;*
- $(L, D) \notin \mathrm{Heur}_{1-\frac{1}{2^{n-1}}}\mathbf{R}$.

Proof. We cannot literally repeat the proof of Lemma 3.1 despite of we have even already proved Proposition 4.1. The reason is the following: not every algorithm computes the distributional function, it is not necessary that it computes even monotonic function. And it is not easy to verify that algorithms computes a distribution function.

Let A_i be enumeration of all algorithms supplied with alarm-clock Cn^a, where C is some constant. We interpret them as algorithms that computes distribution functions. However we remember that it is not necessary that all of them computes a correct distribution function. We interpret the result of $A_i(x)$ as binary real number between 0 and 1.

For every n we will show that it is possible in poly(n) time to find $x_n \in \{0,1\}^n$ such that if $i \in \{1, 2, \ldots, n\}$ and A_i is distributional function, then the A_i-probability of x_n is at most 2^{i-n}. The distribution D_n would be concentrated on x_n; the resulting ensemble is computable in polynomial time by Lemma 4.1. If for all n we find such x_n, then we may define L similarly to the proof of Lemma 3.1. Namely we will choose L such that $L \subseteq \bigcup\limits_n \{x_n\}$ and $x_n \in L$ if and only if n-th algorithm in the enumeration of all algorithms rejects x_n. For all $F \in \mathbf{Comp}[n^a]$ the algorithm that returns 0 on all inputs decides (L, F) in $\mathrm{Heur}_{2^{i-n}}\mathbf{DTime}[n]$, if F is computable by A_i in our enumeration. By the construction $(L, D) \notin \mathrm{Heur}_{1-\frac{1}{2^{n-1}}}\mathbf{P}$.

Now we describe the procedure of finding strings x_n. Initially $I = \{1, 2, \ldots, n\}$, we will delete element i from I if we discover that A_i is not a distribution function on $\{0,1\}^n$.

On each iteration we define $F(x) = \sum_{i \in I} \frac{1}{2^i} A_i(x)$. By binary search we try to find such element $x \in \{0,1\}^n$ that $F(x) - F(x') \leq 2^{-n}$, where x' is lexicographical predecessor of x and $F(x') = 0$ if $x = 0^n$. If binary search succeeds, then $x_n := x$. If binary search fails then it means that we discover nonmonotonicity of $F(x)$, using this we may find $i \in I$ such that A_i is nonmonotonic and exclude all such i from I and start new iteration. If $I = \emptyset$ then choose $x_n = 0^n$, in other cases for all $i \in I$ if A_i computes a correct distribution function then x_n has probability at most 2^{i-n}. □

References

[1] Laszlo Babai, Paul Erdos, and Stanley Selkow. Random graph isomorphism. *SIAM J. Comput.*, 9(3):628–635, 1980.

[2] Shai Ben-David, Benny Chor, Oded Goldreich, and Michael Luby. On the theory of average case complexity. *J. Comput. Syst. Sci.*, 44(2):193–219, 1992.

[3] Yuri Gurevich and Saharon Shelah. Expected computation time for hamiltonian path problem. *SIAM J. Comput.*, 16(3):486–502, 1987.

[4] Dan Gutfreund, Ronen Shaltiel, and Amnon Ta-Shma. If NP languages are hard on the worst-case, then it is easy to find their hard instances. *Computational Complexity*, 16(4):412–441, 2007.

[5] Dmitry Itsykson, Alexander Knop, and Dmitry Sokolov. Heuristic time hierarchies via hierarchies for sampling distributions. In *Proceedings of ISAAC-2015, to appear*, 2015.

[6] Ming Li and Paul M.B. Vitanyi. Average case complexity under the universal distribution equals worst case complexity. *Information Processing Letters*, 42:145–149, 1992.

[7] Konstantin Pervyshev. On heuristic time hierarchies. In *IEEE Conference on Computational Complexity*, pages 347–358, 2007.

[8] E.R. van Dama and M. Muzychuk. Some implications on amorphic association schemes. *Journal of Combinatorial Theory, Series A*, 117:111–127, 2010.

[9] Thomas Watson. Time hierarchies for sampling distributions. In *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 429–440, 2013.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de