Electronic Colloquium on Computational Complexity, Report No. 178 (2014)

Heuristic time hierarchies via hierarchies for sampling distributions *

Dmitry Itsykson † Alexander Knop† Dmitry Sokolov†

Abstract

We give a new simple proof of the time hierarchy theorem for heuristic **BPP** originally proved by Fortnow and Santhanam [FS04] and then simplified and improved by Pervyshev [Per07]. In the proof we use a hierarchy theorem for sampling distributions recently proved by Watson [Wat13]. As a byproduct we get that $\mathbf{P} \not\subseteq \mathbf{BPTime}[n^k]$ if one way functions exist. As far as we know this statement was not known before. We also show that our technique may be extended for time hierarchies in some other heuristic classes.

1 Introduction

The time hierarchy theorem for a computational model states that given more time it is possible to solve more computational problems. For deterministic Turing machines this theorem was proved by Hartmanis and Stearns [HS67] by using diagonalization. To show that there exists a language that is solvable in $O(n^3)$ steps but not solvable in $O(n^2)$ one may consider a language that contains a string x if Turing machine M_x rejects x in n^2 steps. Time hierarchy theorems are known for all syntactic computational models (a model is syntactic if it is possible to enumerate all correct machines of this model). Standard diagonalization does not work if it is impossible to negate the answer of a machine in polynomial time (for example this is true for nondeterministic algorithms); but delayed diagonalization [Zak83] works well for all syntactic models.

A computational model is semantic if it is impossible to enumerate correct machines. For example $\mathbf{BPTime}, \mathbf{RTime}, \mathbf{ZPTime}$ are semantic models; we can't enumerate correct machines since they have to satisfy promises. There are not known tight time hierarchy theorem for any semantic model. The best current result for time hierarchy for randomized computations with bounded error is superpolynomial: $\mathbf{BPTime}[n^{\log n}] \subsetneq \mathbf{BPTime}[2^{n^\epsilon}]$ [KV87]. However, we are not able to prove that $\mathbf{BPTime}[n] \subsetneq \mathbf{BPTime}[n^{100 \log n}]$.

The first advancement in that direction was a time hierarchy theorem for randomized classes with several bits of nonuniform advice [Bar02, FS04], the latest results include a time hierarchy for classes with only one bit of advice : $\mathbf{BPTime}/1$ [FS04],

*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

ZPTime/1, **MATime**/1, etc. [vMP07]. The idea of the proofs of time hierarchies with nonuniform advice from [FS04] (the similar idea was used in [Bar02]) is based on the existence of an optimal algorithm for some **PSPACE**-complete language. The proof from [vMP07] is based on a tricky delayed diagonalization.

Fortnow and Santhanam also proved the time hierarchy theorem for heuristic randomized algorithms with bounded error, such algorithms may give an incorrect answer (and also violate promise) on the small fraction of inputs. This proof was also based on an optimal algorithm for **PSPACE**-complete language. Pervyshev [Per07] simplifies the time hierarchy theorem for heuristic **BPTime**. Pervyshev used delayed diagonalization against all randomized Turing machines. A randomized Turing machine may accept with any probability of error, thus it can't be simulated with bounded error. Let M be a randomized Turing machine that may violate a promise. Suppose we have to simulate it on an input x. Pervyshev suggested a method to simulate it heuristically: for every input x we put into the correspondence a set of strings $\{y_1, y_2, \ldots, y_N\}$, where N is large enough. On every y_i we execute $M(x)$ many times and calculate the frequency of ones μ_i. We accept y_i if μ_i is greater than θ_{y_i}, where $\theta_{y_i} = \frac{2}{5} + \frac{i}{5N}$. Note that if $M(x)$ satisfies the promise of bounded error then the answer of our simulation is the same for all y_i. And if $M(x)$ violates the promise then our simulation may violate the promise only for a small fraction of y_i, namely for such y_i that θ_{y_i} is very close to $\Pr[M(x) = 1]$.

We give a new proof of time hierarchy theorem for heuristic **BPTime**. Our proof is slightly simpler than the Pervyshev's one, namely we don't use the multithreshold trick described above. Our proof is based on the hierarchy for polynomial-time samplable distributions recently proved by Watson [Wat13]. Watson proved that for any integer constant k, positive a and ϵ there exists a polynomial-time samplable ensemble of random variables γ_n that takes values from the set $\{1, 2, \ldots, k\}$ such that for every samplable in n^a steps ensemble of random variables α_n with values in $\{1, 2, \ldots, k\}$ the statistical distance between α_n and β_n is at least $1 - \frac{1}{k} - \epsilon$ for some n. We use this result only for $k = 2$; this particular case can be proved elementary by delayed diagonalization. We define a language L_γ that is based on the ensemble γ_n; we prove that L_γ is solvable in Heur_ϵ**BPP**. If L_γ is solvable in randomized heuristic time n^a by an algorithm A, then A may be used to generate in n^a steps an ensemble of random variables α_n that is close to γ_n; the latter contradicts the theorem of Watson.

This method can also be used to prove hierarchy theorems for other heuristic classes. Pervyshev proved the time hierarchy theorem for heuristic nondeterministic computations. This proof can also be formulated in our framework. For that we extend the notion of samplability of random variables. We define a class of random variables (taking values in $\{0, 1\}$) that can be sampled in nondeterministic polynomial time: the sampling algorithm applies function from **NP** to random bits. Watson's theorem also holds for nondeterministically samplable random variables. For proving time hierarchy for heuristic **NP** we need a more accurate version of Watson's theorem, namely we prove a hierarchy for a nondeterministic sampling for the case of a sampling algorithm that uses exactly n random bits, where n is the index of the random variable. In fact, this time hierarchy was explicitly proved in [Per07].

We also note that our method works for heuristic hierarchies for all classes $\mathfrak{C}$**Time** and **BP** $\cdot \mathfrak{C}$**Time**, where $\mathfrak{C}$ is a syntactic computational model that is closed under the application of the majority.

1.1 Conditional results

There are several known conditions that imply time hierarchy theorem for **BPTime**. The existence of a **BPP**-complete problem (under strong enough reductions) implies a time hierarchy theorem for **BPTime** (see for example [Bar02]). The paper of Fortnow and Santhanam [FS04] implies that if it is possible to approximate the running time of the optimal algorithm for the **PSPACE**-complete language in polynomial time, then there exists a time hierarchy for **BPTime**. The time hierarchy theorem for **BPTime** also holds in case of the existence of $\log n \to n$ pseudorandom generator (i.e. generator mapping a seed of size $C \log n$ to n pseudorandom bits), since in that case $\mathbf{BPTime}[n^k] \subseteq \mathbf{DTime}[n^{k+\epsilon}]$ and the hierarchy follows from the deterministic time hierarchy. Such pseudorandom generator exists if, for example, $\mathbf{E} \setminus \mathbf{Size}[2^{\epsilon n}] \neq \emptyset$ (see for example [Mil01]). But the existence of $n \to \mathrm{poly}(n)$ pseudorandom is not sufficient for full derandomization, and thus **BPTime** hierarchy is not completely trivial.

As a corollary of our proof of heuristic time hierarchy we get that $\mathbf{P} \not\subseteq \mathrm{Heur}_{1/2-\epsilon}\mathbf{BPTime}[n^k]$ under the existence of $n \to \mathrm{poly}(n)$ pseudorandom generator (that is equivalent to the existence of one-way functions). We also note that if $\mathbf{NP} \subseteq \mathbf{BPP}$ then $\mathbf{BPP} \not\subseteq \mathbf{BPTime}[n^k]$ for all k. In terms of Impagliazzo's worlds [Imp95] the **BPTime** hierarchy theorem holds in Algorithmica and Criptomania worlds.

2 Preliminaries

For two random variables χ_1, χ_2 with values from a set K the statistical distance between them is $\Delta(\chi_1, \chi_2) = \max\limits_{S \in [k]} |\Pr_{\chi_1}[S] - \Pr_{\chi_2}[S]|$.

Let U_n denote a uniform distribution over $\{0,1\}^n$.

We say that a language L is heuristically decidable in nondeterministic time $O(f(n))$ with an error $\delta(n)$ (we denote this as $L \in \mathrm{Heur}_{\delta(n)}\mathbf{NTime}[f(n)]$) iff there is a nondeterministic algorithm A that runs in at most $O(f(n))$ steps such that for any n we have that $\Pr_{x \leftarrow U_n}[A(x) = L(x)] \geq 1 - \delta(n)$. We also define $\mathrm{Heur}_{\delta(n)}\mathbf{NP} = \bigcup\limits_{k \geq 0} \mathrm{Heur}_{\delta(n)}\mathbf{NTime}[n^k]$.

The class $\mathrm{Heur}_{\delta(n)}\mathbf{BPTime}[f(n)]$ consists of languages L such that there exists a probabilistic algorithm A that runs in at most $O(f(n))$ steps and for every n the following holds: $\Pr_{x \leftarrow U_n}[\Pr[A(x) = L(x)] \geq \frac{3}{4}] \geq 1 - \delta(n)$, where the inner probability is over random bits of the algorithm A. We also denote $\mathrm{Heur}_{\delta(n)}\mathbf{BPP} = \bigcup\limits_{k \geq 0} \mathrm{Heur}_{\delta(n)}\mathbf{BPTime}[n^k]$.

In the proof of our results we use Boolean samplers from [Gol11].

Definition 2.1. A Boolean sampler is a randomized algorithm S, that takes on input an integer number n and rational numbers δ, ϵ. Algorithm S has an oracle access to a function $f\colon \{0,1\}^n \to \{0,1\}$; S makes several nonadaptive requests to the function f and outputs a number in the range $[0,1]$. Let us denote $\bar{f} = \frac{1}{2^n}\sum_{x \in \{0,1\}^n} f(x)$. For every function $f\colon \{0,1\}^n \to \{0,1\}$ the following inequality should be satisfied: $\Pr[|S^f(n,\epsilon,\delta) - \bar{f}| \geq \epsilon] < \delta$.

A Boolean sampler is called averaging if it outputs the average value of requested values.

Theorem 2.1 ([Gol11])**.** There is an averaging Boolean sampler S which uses n random bits, makes $q(n,\epsilon,\delta) = O(\frac{1}{\epsilon^2\delta})$ requests to the function, and runs in time polynomial in n, $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$.

Corollary 2.1. There exists an averaging Boolean sampler S that uses $n-1$ random bits, makes $O(\frac{1}{\epsilon^2\delta})$ requests to the function, and runs in time polynomial in n, $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$.

Proof. Let S be a sampler from Theorem 2.1. Define the algorithm S' as follows: on input n, ϵ and δ it returns $\frac{1}{2}S^{f_0}(n-1,\epsilon,\frac{\delta}{2}) + \frac{1}{2}S^{f_1}(n-1,\epsilon,\frac{\delta}{2})$ where $f_0, f_1 : \{0,1\}^{n-1} \to \{0,1\}$ and $f_0(x) = f(x0)$, $f_1(x) = f(x1)$. Note that $\frac{\bar{f}_0+\bar{f}_1}{2} = \bar{f}$. Hence

$$\Pr[|S'^f(n,\epsilon,\frac{\delta}{2}) - \bar{f}| \geq \epsilon] \leq$$
$$\Pr[|S^{f_0}(n,\epsilon,\frac{\delta}{2}) - \bar{f}_0| \geq \epsilon] + \Pr[|S^{f_1}(n,\epsilon,\delta) - \bar{f}_1| \geq \epsilon] < \frac{\delta}{2} + \frac{\delta}{2} = \delta.$$

□

3 Hierarchy for HeurBPP

Definition 3.1. An ensemble of random variables γ_n is samplable in time $O(f(n))$ iff there exist an integer constant k and a deterministic algorithm A that on the input $(1^n, r)$ runs in $O(f(n))$ steps and $A(1^n, r)$ is distributed according to γ_n, where r is distributed uniformly over $\{0,1\}^{n^k}$. We denote the set of all ensembles samplable in time $O(f(n))$ as $\mathbf{DSamp}[f(n)]$.

The following theorem is a particular case of a theorem from [Wat13]. However the proof of this particular case is much simpler than Watson's proof of his more general statement. We give it for the sake of completeness.

Theorem 3.1 ([Wat13])**.** For every $a > 0$ and $\epsilon > 0$, there exists $b > 0$ and an ensemble of random variables $\gamma_n \in \mathbf{DSamp}[n^b]$ that take values from $\{0,1\}$ such that for every ensemble $\alpha_n \in \mathbf{DSamp}[n^a]$ there exists n_0 such that the statistical distance between α_{n_0} and γ_{n_0} is at least $\frac{1}{2} - \epsilon$.

Proof. We use delayed diagonalization. Let (E_i, k_i) be an enumeration of all deterministic algorithms (we interpret them as generators of random variables) and integer constants k_i, we assume that E_i is supplied with an alarm clock that terminates its execution on an input $(1^n, r)$ after n^{a+1} steps. We define a sequence n_i as follows: $n_1 = 1$, $n_{i+1} = n_i^* + 1$ and $n_i^* = 2^{n_i^{a+1}}$. We define γ_n by the following algorithm $\Gamma(1^n, R)$, where R is a string of random bits of size at least Nn^k and $N = O(\frac{\log\epsilon}{\epsilon^2})$. For n such that $n_i \leq n \leq n_i^*$:

- if $n = n_i^*$ then γ_n is concentrated on the element from $\{0,1\}$ that has the minimal probability according to $E_i(1^{n_i}, r)$ where r is uniformly distributed over $\{0,1\}^{n_i^{a+1}}$. This can be done by a brute-force search in time $\text{poly}(n_i^*)$;
- if $n \in \{n_i, \ldots, n_i^* - 1\}$ generate N independent random strings $r_1, r_2, \ldots, r_N \leftarrow U_{(n+1)^k}$ (but formally we already have all random bits in the string R, i.e. $R = r_1 r_2 \ldots r_N$); execute $E_i(1^{n+1}, r_j)$ for every $j \in \{1, 2, \ldots, N\}$ and return the most frequent answer.

Let α_n be generated by an algorithm $A(1^n, r)$ with n^k random bits in time $O(n^a)$, and (A, k) be (E_i, k_i) in our enumeration. We prove by contradiction that there exists $n \in \{n_i, \ldots, n_i^*\}$ such that $\Delta(\gamma_n, \alpha_n) > \frac{1}{2} - \epsilon$, where Δ denotes the statistical distance. Assume that $\Delta(\gamma_n, \alpha_n) \leq \frac{1}{2} - \epsilon$ for all n. Let b denote the element that has probability 1 according to $\gamma_{n_i^*}$. By induction on m (for $m \leq n_i^* - n_i$) we prove that $\Pr[\gamma_{n_i^*-m} = b] > 1 - \frac{\epsilon}{2}$. Base $m = 0$ is trivial. Now we prove the induction step. By the induction hypothesis

$$\Pr[\alpha_{n_i^*-m} = b] \geq \Pr[\gamma_{n_i^*-m} = b] - \Delta(\gamma_{n_i^*-m}, \alpha_{n_i^*-m}) \geq$$
$$\Pr[\gamma_{n_i^*-m} = b] - \frac{1}{2} + \epsilon > 1 - \frac{\epsilon}{2} - \frac{1}{2} + \epsilon = \frac{1}{2} + \frac{\epsilon}{2}.$$

Hence by Chernoff bounds $\Pr[\gamma_{n_i^*-m-1} = b] \geq 1 - 2e^{-2\epsilon^2 N}$ that is more than $1 - \frac{\epsilon}{2}$ for $N = O(\frac{\log \epsilon}{\epsilon^2})$. Finally we get a contradiction with $\Pr[\alpha_{n_i} = b] \leq \frac{1}{2}$. □

Let γ_n be an ensemble of random variables that take values from $\{0, 1\}$. We denote $L_\gamma = \bigcup_n \{r \in \{0,1\}^n \mid \Pr[\gamma_n = 1] > 0.r\}$, where $0.r$ is a binary number.

Lemma 3.1. For every polynomial-time samplable ensemble of random variables γ_n that take values from $\{0, 1\}$ the language $L_\gamma \in \text{Heur}_\epsilon\mathbf{BPP}$ for every constant ϵ.

Proof. Consider the following algorithm A: sample N independent instances of the random variable γ_n, let q be a fraction of 1s. If $q \geq 0.r$ then return 0 otherwise 1. By Chernoff bounds if $|0.r - \Pr[\gamma_n = 1]| > \epsilon/4$ then $\Pr[A(r) \neq L(r)] < 2e^{-\frac{1}{8}\epsilon^2 N}$; it is less than $\frac{1}{4}$ for $N = O(\frac{1}{\epsilon^2})$. Note that $\Pr_r[|0.r - \Pr[\gamma_n = 1]| \leq \epsilon/4] \leq 2^{-n} + \epsilon/2$ that is less than ϵ for large enough n. □

Lemma 3.2. Let L be a language such that for all $n \mid \Pr_{x \leftarrow U_n}[x \in L] - \Pr[\gamma_n = 1]| < \delta$ and $L \in \text{Heur}_\epsilon\mathbf{BPTime}[n^k]$ for some $\epsilon, \delta \geq 0$. Then there exists an ensemble of random variables β_n such that $\beta_n \in \mathbf{DSamp}[n^{k+1}]$ and $\Delta(\beta_n, \gamma_n) \leq \epsilon + \delta + \frac{1}{2^n}$.

Proof. Let E be a randomized algorithm that solves L in $\text{Heur}_\epsilon\mathbf{BPTime}[n^k]$. Let $\hat{E}(x)$ execute $E(x)$ for $N = O(n)$ times and return the most frequent answer. Consider an ensemble of random variables α_n defined in the following way: sample a random element $x \in \{0,1\}^n$ and return $L(x)$. Then consider the following algorithm that samples β_n: sample a random element $x \in \{0,1\}^n$ and return $\hat{E}(x)$. Since $|\Pr_{x \leftarrow U_n}[x \in L] - \Pr[\gamma_n = 1]| < \delta$ we have that $\Delta(\alpha_n, \gamma_n) < \delta$. Let C be a set of all x such that $\Pr[E(x) = L(x)] \geq \frac{3}{4}$. Chernoff bounds imply that for $x \in C$ we have that $\Pr[\hat{E}(x) = L(x)] > 1 - \frac{1}{2^n}$. Note that

$$\Delta(\alpha_n, \beta_n) = |\Pr[\alpha_n = 1] - \Pr[\beta_n = 1]| = |\Pr[\alpha_n = 1, \beta_n = 0] - \Pr[\alpha_n = 0, \beta_n = 1]| \leq$$
$$\Pr[\alpha_n = 1 \wedge \beta_n = 0] + \Pr[\alpha_n = 0 \wedge \beta_n = 1] = \Pr[\alpha_n \neq \beta_n].$$

Hence

$$\Delta(\alpha_n, \beta_n) \leq \Pr[\alpha_n \neq \beta_n] =$$
$$\Pr_{x,r}[L(x) \neq \hat{E}_r(x) | x \in C] \Pr_{x,r}[x \in C] + \Pr_{x,r}[L(x) \neq \hat{E}_r(x) | x \notin C] \Pr_{x,r}[x \notin C] \leq$$
$$\frac{1}{2^n} \Pr_{x,r}[x \in C] + \Pr_{x,r}[L(x) \neq \hat{E}(x) | x \notin C]\epsilon \leq \frac{1}{2^n} + \epsilon$$

where r is a string of random bits for $\hat{E}$. Hence by the triangle inequality we have that $\Delta(\beta_n, \gamma_n) \leq \frac{1}{2^n} + \epsilon + \delta$. □

Theorem 3.2 ([Per07])**.** For every $b > 0$ and $\delta > 0$ there exists a language L such that $L \notin \text{Heur}_{\frac{1}{2}-\delta}\mathbf{BPTime}[n^b]$ and $L \in \text{Heur}_{\delta}\mathbf{BPP}$.

Proof. Let γ_n be an ensemble from Theorem 3.1 for $\epsilon = \delta/2$ and $a = b+1$. By Lemma 3.1 $L_\gamma \in \text{Heur}_{\delta}\mathbf{BPP}$. Assume that $L_\gamma \in \text{Heur}_{\frac{1}{2}-\delta}\mathbf{BPTime}[n^b]$. Note that by construction of L_γ we have that $|\Pr_{x \leftarrow U_n}[x \in L_\gamma] - \Pr[\gamma_n = 1]| < \frac{1}{2^n}$. Hence by Lemma 3.2 there exists $\beta_n \in \mathbf{DSamp}[n^a]$ and $\Delta(\beta_n, \gamma_n) \leq \frac{1}{2} - \delta + \frac{1}{2^n} + \frac{1}{2^n} < \frac{1}{2} - \frac{\delta}{2}$ for n large enough. The latter contradicts Theorem 3.1. □

4 Conditional hierarchy

Theorem 4.1. Assume that one-way functions exist. Then for every $\epsilon > 0$ and $a > 0$ there exists a language $L \in \mathbf{P}$ such $L \notin \text{Heur}_{\frac{1}{2}-\epsilon}\mathbf{BPTime}[n^a]$.

Proof. Consider the random variable γ_n from Theorem 3.1 and let S be a generator that generates γ_n. We assume that S gets random bits as the second input. Let S use $p(n)$ random bits. Let G be pseudorandom generator that maps n random bits to $p(n)$ pseudorandom ones. Consider the random variable $S(1^n, G(r))$, where $r \leftarrow U_n$. Since G is a pseudorandom generator we have that $\Delta(S(1^n, G(U_n)), \gamma_n) = \Delta(S(1^n, G(U_n)), S(1^n, U_{p(n)})) < \epsilon/4$ for all n large enough.

Consider the language $L = \bigcup_n \{r \in \{0,1\}^n \mid S(1^n, G(r)) = 1\}$. It is obvious that $L \in \mathbf{P}$. Lemma 3.2 and Theorem 3.1 implies $L \notin \text{Heur}_{\frac{1}{2}-\epsilon}\mathbf{BPTime}[n^a]$. □

It may be interesting to compare Theorem 4.1 with the following statement: if one-way functions exist, then $(\mathbf{NP}, U) \notin \text{Heur}\mathbf{BPP}$ [BT06].

We also show that the **BPTime** time hierarchy holds if all languages from **NP** are easy.

Theorem 4.2. If $\mathbf{NP} \subseteq \mathbf{BPP}$, then $\mathbf{BPTime}[n^k] \subsetneq \mathbf{BPP}$ for all $k > 0$.

Proof. Assume, for the sake of contradiction, that $\mathbf{BPP} \subseteq \mathbf{BPTime}[n^k]$. By the argument similar to Adleman's theorem we get $\mathbf{BPTime}[n^k] \subseteq \mathbf{Size}[n^{2k+2}]$. Results of [Zac88] implies that if $\mathbf{NP} \subseteq \mathbf{BPP}$, then $\mathbf{PH} \subseteq \mathbf{BPP}$. So if $\mathbf{BPP} = \mathbf{BPTime}[n^k]$, then $\mathbf{PH} \subseteq \mathbf{Size}[n^{2k+2}]$ that contradicts Kannan's theorem [Kan82]. □

5 Hierarchy for Heur**NP**

In this section we show that our technique can also be used to prove a hierarchy theorem for heuristic **NP**.

Definition 5.1. An ensemble of random variables γ_n is samplable in nondeterministic time $O(f(n))$ with n^k random bits iff there exists a nondeterministic algorithm A that on an input $(1^n, r)$ runs in $O(f(n))$ steps, and $A(1^n, r)$ is distributed according γ_n, where r is distributed uniformly over $\{0,1\}^{n^k}$. We denote the set of all ensembles samplable in time $O(f(n))$ with n^k random bits as $\mathbf{NSamp}_{n^k}[f(n)]$.

The following theorem is an analogue of Theorem 3.1 for distributions samplable by nondeterministic algorithms with fixed number of random bits. The proof is almost the same but we use a Boolean sampler in order to save random bits.

Theorem 5.1. For every $a > 0$ and $\epsilon > 0$ there exists $b > 0$ and an ensemble of random variables $\gamma_n \in \mathbf{NSamp}_n[n^b]$ that take values from $\{0,1\}$ such that for every ensemble $\alpha_n \in \mathbf{NSamp}_n[n^a]$ with values from $\{0,1\}$ there exists n such that the statistical distance between α_n and γ_n is at least $\frac{1}{2} - \epsilon$.

Proof. We use delayed diagonalization. Let E_i be an enumeration of all nondeterministic algorithms (we interpret them as generators of random variables that use n random bits); we assume that E_i is supplied with an alarm clock that terminates its execution on an input $(1^n, r)$ after n^{a+1} steps. Let S be a Boolean sampler from Corollary 2.1. We define a sequence n_i as follows $n_1 = 1$, $n_{i+1} = n_i^* + 1$ and $n_i^* = 2^{n_i^{a+1}}$. We define γ_n by the following algorithm $\Gamma(1^n, r)$, where $r \leftarrow U_n$ is the string of random bits. For n such that $n_i \le n \le n_i^*$:

- if $n = n_i^*$ then γ_n is concentrated on the element from $\{0,1\}$ that has the minimal probability according to $E_i(1^{n_i}, r)$, where r is uniformly distributed over $\{0,1\}^n$. This can be done by brute-force search in time $\mathrm{poly}(n_i^*)$;
- if $n_i \le n < n_i^* - 1$ we execute $S^f(1^{n+1}, \frac{\epsilon}{2}, \frac{1}{4})$ using r as a random string, where $f : \{0,1\}^{n+1} \to \{0,1\}$ and $f(z) = E_i(1^{n+1}, z)$. Return 1 iff the result of the sampler exceeds $\frac{1}{2}$.

Let α_n be generated by a nondeterministic algorithm $A(1^n, r)$ with n random bits in $O(n^a)$ steps, and A have number i in our enumeration. We prove by contradiction that there exists n ($n_i \le n \le n_i^*$) such that $\Delta(\gamma_n, \alpha_n) > \frac{1}{2} - \epsilon$, where Δ denotes the statistical distance. Assume that $\Delta(\gamma_n, \alpha_n) \le \frac{1}{2} - \epsilon$ for all n. Let b denote the element that has probability 1 according to $\gamma_{n_i^*}$ (by the construction $\Pr[E_i(1^{n_i}) = b] \le \frac{1}{2}$). We prove by induction on k (for $0 \le k \le n_i^* - n_i$) that $\Pr[\gamma_{n_i^*-k} = b] > 1 - \frac{\epsilon}{2}$. The base $k = 0$ is trivial. Now we prove the induction step. By the induction hypothesis

$$\Pr[\alpha_{n_i^*-k} = b] \ge \Pr[\gamma_{n_i^*-k} = b] - \frac{1}{2} + \epsilon > 1 - \frac{\epsilon}{2} - \frac{1}{2} + \epsilon = \frac{1}{2} + \frac{\epsilon}{2}.$$

Hence by definition of a Boolean sampler $\Pr[\gamma_{n_i^*-k-1} = b] \ge 1 - \frac{\epsilon}{2}$. Finally we get a contradiction with $\Pr[\alpha_{n_i} = b] \le \frac{1}{2}$. □

Theorem 5.2 ([Per07])**.** For every $b > 0$ and $\delta > 0$ there exists a language L such that $L \notin \mathrm{Heur}_{\frac{1}{2}-\delta}\mathbf{NTime}[n^b]$ and $L \in \mathbf{NP}$.

Proof. Let γ_n be an ensemble from Theorem 5.1 for $\epsilon = \delta/2$ and $a = b + 1$, and S be a generator for this random variable. Consider the language $L = \{x | S(1^{|x|}, x) = 1\}$. It is easy to see that this language is in $\mathbf{NP}$. Let us prove that $L \notin \mathrm{Heur}_{\frac{1}{2}-\delta}\mathbf{NTime}[n^b]$. Assume the contrary and let nondeterministic algorithm A decide L in time n^b with error less than $\frac{1}{2} - \delta$. In this case for the random variable α_n that is distributed according to $A(x)$ for $x \leftarrow A_n$ we have that $\Delta(\alpha_n, \gamma_n) < \frac{1}{2} - \delta$ for all n. The latter contradicts Theorem 5.1. □

6 Generalization

Let $\mathfrak{C}$ be some computational model; for every input a $\mathfrak{C}$-machine either accepts or rejects; we assume that the notion of the running time of a $\mathfrak{C}$-machine on a given input is well defined. We denote the set of languages that can be decided by $\mathfrak{C}$-machines in $O(t(n))$ steps by $\mathfrak{C}\mathbf{Time}[t(n)]$. We also require the model $\mathfrak{C}$ to be closed under the application of majority: let F be a deterministic oracle Turing machine that on every input makes several oracle requests that depend only on the input, calculates majority and returns an answer from $\{0,1\}$. Let on the input x the machine F make oracle requests $y_1, y_2, \ldots, y_{k(x)}$. Let language O be from the class $\mathfrak{C}\mathbf{Time}[h(n)]$, then the computation of F with oracle O on the input x may be simulated on a $\mathfrak{C}$-machine in $O(n + h(|y_1|) + \cdots + h(|y_{k(x)}|))$ steps. Note that nondeterministic Turing machines are closed under the application of majority.

Definition 6.1. Class $\mathfrak{C}\mathbf{Time}[n^c]$ is the set of languages that can be decided in $O(n^c)$ time on $\mathfrak{C}$-machines. $\mathfrak{C}\mathbf{P} = \bigcup\limits_{c>0} \mathfrak{C}\mathbf{Time}[n^c]$.

Definition 6.2. $\mathfrak{C}$ is a syntactic computational model iff there exists an algorithm A that takes a number k and enumerates $\mathfrak{C}$-machines that stops in n^{k+1} steps such that for all languages $L \in \mathfrak{C}\mathbf{Time}[n^k]$ there exists a machine that decides this language.

Definition 6.3. For a syntactic model $\mathfrak{C}$ we say that $L \in \mathrm{Heur}_{\delta(n)}\mathfrak{C}\mathbf{Time}[f(n)]$ iff there exists $\mathfrak{C}$-algorithm A that runs in $O(f(n))$ steps such that for any n we have that $\Pr\limits_x[A(x, 1^n) = L(x)] \geq 1 - \delta(n)$. We denote $\mathrm{Heur}_{\delta(n)}\mathfrak{C}\mathbf{P} = \bigcup\limits_{k\in\mathbb{N}} \mathrm{Heur}_{\delta(n)}\mathfrak{C}\mathbf{Time}[n^k]$.

Definition 6.4. An ensemble of random variables γ_n is $\mathfrak{C}$-samplable in time $O(f(n))$ with n^k random bits iff there exists a $\mathfrak{C}$-algorithm A that on an input $(1^n, r)$ runs in $O(f(n))$ steps and $A(1^n, r)$ is distributed according to γ_n, where r is distributed uniformly over $\{0,1\}^{n^k}$. We denote the set of all ensembles samplable in time $O(f(n))$ with n^k random bits as $\mathfrak{C}\mathbf{Samp}_{n^k}[f(n)]$. We also define $\mathfrak{C}\mathbf{Samp}[f(n)] = \cup_k \mathfrak{C}\mathbf{Samp}_{n^k}[f(n)]$.

The following theorem is completely analogous to Theorem 5.1:

Theorem 6.1. For every $a > 0$ and $\epsilon > 0$ there exists $b > 0$ and an ensemble of random variables $\gamma_n \in \mathfrak{C}\mathbf{Samp}_n[n^b]$ that take values from $\{0,1\}$ such that for every ensemble $\alpha_n \in \mathfrak{C}\mathbf{Samp}_n[n^a]$ with values from $\{0,1\}$ there exists n such that the statistical distance between α_n and γ_n is at least $\frac{1}{2} - \epsilon$.

The following theorem is completely analogous to Theorem 5.2:

Theorem 6.2. For every syntactic model $\mathfrak{C}$ for all $b > 0$ and $\delta > 0$ there is language L such that $L \notin \mathrm{Heur}_{\frac{1}{2}-\delta}\mathfrak{C}\mathbf{Time}[n^b]$ and $L \in \mathfrak{C}\mathbf{P}$.

Following [Sch89] we define class $\mathbf{BP} \cdot \mathfrak{C}\mathbf{Time}[f(n)]$ that consists of languages L such that there exists k and $\mathfrak{C}$-machine M such that

1. For all x the following holds: $\Pr_{r \leftarrow U_{n^k}}[M(x, r) = L(x)] \geq \frac{3}{4}$;
2. $M(x, r)$ runs in $O(f(n))$ steps.

Definition 6.5. For syntactic model $\mathfrak{C}$ we say that $L \in \mathrm{Heur}_{\delta(n)}\mathbf{BP} \cdot \mathfrak{C}\mathbf{Time}[f(n)]$ iff there exists $k \leq c$ and $\mathfrak{C}$-machine M such that

1. For all n the following holds: $\Pr_{x \leftarrow U_n}[\Pr_{r \leftarrow U_{n^k}}[M(x,r) = L(x)] \geq \frac{3}{4}] \geq 1 - \delta(n)$;
2. $M(x,r)$ runs in $O(f(n))$ steps.

We also define $\mathrm{Heur}_{\delta(n)}\mathbf{BP} \cdot \mathfrak{C}\mathbf{P} = \cup_{c>0}\mathrm{Heur}_{\delta(n)}\mathbf{BP} \cdot \mathfrak{C}\mathbf{Time}[n^c]$

The following theorem is completely analogous to Theorem 3.1:

Theorem 6.3. For every $a > 0$ and $\epsilon > 0$ there exists $b > 0$ and an ensemble of random variables $\gamma_n \in \mathfrak{C}\mathbf{Samp}[n^b]$ that take values from $\{0,1\}$ such that for every ensemble $\alpha_n \in \mathfrak{C}\mathbf{Samp}[n^a]$ there exists n such that the statistical distance between α_n and γ_n is at least $\frac{1}{2} - \epsilon$.

The following theorem is completely analogous to Theorem 3.2:

Theorem 6.4. For every $b > 0$ and $\delta > 0$ there is language L such that $L \notin \mathrm{Heur}_{\frac{1}{2}-\delta}\mathbf{BP} \cdot \mathfrak{C}\mathbf{Time}[n^b]$ and $L \in \mathrm{Heur}_{\delta}\mathbf{BP} \cdot \mathfrak{C}\mathbf{P}$.

Since $\mathbf{AM} = \mathbf{BP} \cdot \mathbf{NP}$ Theorem 6.4 implies heuristic time hierarchy for **AMTime**.

7 Further research

For deterministic computations the standard diagonalization implies that $\mathbf{P} \subsetneq \mathrm{Heur}_{1-\delta}\mathbf{DTime}[n^k]$ while the best known result for nondeterministic computations is $\mathbf{NP} \subsetneq \mathrm{Heur}_{\frac{1}{2}-\delta}\mathbf{NTime}[n^k]$. Is it possible to improve error from $\frac{1}{2} - \delta$ to $1 - \delta$ for nondeterministic or randomized algorithms with bounded error?

The second and third open questions are to prove **BPTime** hierarchy in Heuristica, where $\mathbf{NP} \not\subseteq \mathbf{BPP}$ but $(\mathbf{NP}, \mathbf{PSamp}) \subseteq \mathrm{Heur}\mathbf{BPP}$, and in Pessiland, where $(\mathbf{NP}, \mathbf{PSamp}) \not\subseteq \mathrm{Heur}\mathbf{BPP}$ but there are no one-way functions.

Acknowledgments The authors thank Edward A. Hirsch for useful comments.

References

[Bar02] Boaz Barak. A probabilistic-time hierarchy theorem for "slightly non-uniform" algorithms. In *RANDOM '02: Proceedings of the 6th International Workshop on Randomization and Approximation Techniques*, pages 194–208, London, UK, 2002. Springer-Verlag.

[BT06] Andrej Bogdanov and Luca Trevisan. Average-case complexity. *Foundation and Trends in Theoretical Computer Science*, 2(1):1–106, 2006.

[FS04] Lance Fortnow and Rahul Santhanam. Hierarchy theorems for probabilistic polynomial time. In *FOCS*, pages 316–324, 2004.

[Gol11] Oded Goldreich, editor. *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, volume 6650 of *Lecture Notes in Computer Science*. Springer, 2011.

[HS67] J. Hartmanis and R. E. Stearns. On the computational complexity of algorithms. *Journal of Symbolic Logic*, 32(1):120–121, 1967.

[Imp95] R. Impagliazzo. A personal view of average-case complexity. In *SCT '95: Proceedings of the 10th Annual Structure in Complexity Theory Conference (SCT'95)*, page 134, Washington, DC, USA, 1995. IEEE Computer Society.

[Kan82] Ravi Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55(1):40–56, 1982.

[KV87] Marek Karpinski and Rutger Verbeek. *Randomness, provability, and the separation of Monte Carlo time and space*, pages 189–207. Springer-Verlag, London, UK, 1987.

[Mil01] Peter Bro Miltersen. *Handbook on Randomization*, volume II, chapter 19. Derandomizing Complexity Classes. Kluwer Academic Publishers, July 2001.

[Per07] Konstantin Pervyshev. On heuristic time hierarchies. In *IEEE Conference on Computational Complexity*, pages 347–358, 2007.

[Sch89] Uwe Schöning. Probabilistic complexity classes and lowness. *Journal of Computer and System Sciences*, 39(1):84–100, 1989.

[vMP07] Dieter van Melkebeek and Konstantin Pervyshev. A generic time hierarchy with one bit of advice. *Computational Complexity*, 16(2):139–179, 2007.

[Wat13] 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.

[Zac88] Stathis Zachos. Probabilistic quantifiers and games. *J. Comput. Syst. Sci.*, 36(3):433–451, 1988.

[Zak83] Stanislav Zak. A turing machine time hierarchy. *Theoretical Computer Science*, 26(3):327–333, 1983.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de