Probabilistic Type-2 Operators and "Almost"-Classes*

Ronald V. Book† Heribert Vollmer‡ Klaus W. Wagner‡

Abstract

We define and examine several probabilistic operators ranging over sets (i.e., operators of type 2), among them the formerly studied ALMOST-operator. We compare their power and prove that they all coincide for a wide variety of classes. As a consequence, we characterize the ALMOST-operator which ranges over infinite objects (sets) by a bounded-error probabilistic operator which ranges over strings, i.e. finite objects. This leads to a number of consequences about complexity classes of current interest. As applications, we obtain (a) a criterion for measure 1 inclusions of complexity classes, (b) a criterion for inclusions of complexity classes relative to a random oracle, (c) a new upper time bound for ALMOST-PSPACE, and (d) a characterization of ALMOST-PSPACE in terms of checking stack automata. Finally, a connection between the power of ALMOST-PSPACE and that of probabilistic NC^1 circuits is given.

1 Introduction

In a fundamental paper, John Gill introduced probabilistic Turing machines and the complexity classes they define [18]. During the run of their computation these machines have the possibility to toss fair coins, and then continue their work depending on the outcome. In the polynomial time case this yields the well-known classes PP (for probabilistic polynomial time; with unbounded error probability) and BPP (for bounded error probabilistic polynomial time) which are regarded as natural probabilistic counterparts of the deterministic class P; and moreover BPP is felt to be the class of "tractable" problems (since the error bound can be made arbitrarily small).

*A preliminary abstract of this paper appeared in the proceedings of the 23rd International Colloquium on Automata, Languages, and Programming. Research supported by NSF Grant CCR-93-02057, DFG Grant Wa 847/1, and a Feodor-Lynen-Fellowship from the Alexander von Humboldt Foundation.

†Department of Mathematics, University of California at Santa Barbara, Santa Barbara, CA 93106.

‡Theoretische Informatik, Universität Würzburg, Am Exerzierplatz 3, D-97072 Würzburg, Germany. Research performed while visiting the Department of Mathematics, University of California, Santa Barbara.

But how do define probabilistic analogues for other (possibly not deterministic) classes? The "traditional" way is to consider *operators* in an abstract way as we will do in this paper. This kind of randomness can best be visualized as allowing Turing machines access to a random tape, or equivalently supplying them together with their regular input with an input sequence of random bits. Thus, here the random bits may be multiply accessed. (This should be contrasted with the machines with built-in probabilism described above: If those machines want to re-use their random bits later, they have to store them on their worktape—which might make a difference for space-bounded computations. Therefore the aforementioned built-in probabilism is also called *one-way access* to randomness, see [30].)

Well known examples for operators as just described are Wagner's counting operator $\mathrm{C^p}$ [44], and the corresponding bounded error operator $\mathrm{BP^p}$ (see e.g. [37]). It is relatively easy to see that $\mathrm{C^pP} = \mathrm{PP}$ and $\mathrm{BP^pP} = \mathrm{BPP}$, i.e. when applied to the class P these operators yield as results the classical probabilistic classes. But the operators can be applied in a general way to arbitrary classes $\mathcal{K}$, giving $\mathrm{C^p}\mathcal{K}$ and $\mathrm{BP^p}\mathcal{K}$. For example, the class $\mathrm{BP^pNP}$ has attracted some attention and has been shown to be equal to Babai's class AM (for "Arthur-Merlin," a class defined in terms of interactive proof systems, see [3]).

Yet another approach to define probabilistic computation is to consider complexity classes of the form ALMOST-$\mathcal{K}$, see e.g. [1, 6, 9, 10, 11, 28, 30]. ALMOST-$\mathcal{K}$ is defined to be the class of all sets which are in $\mathcal{K}^A$ for almost every oracle A. For example, $L \in$ ALMOST-P if and only if the set of all A such that $L \in \mathrm{P}^A$ has measure 1 (in the usual product measure on sets, for details see Section 2). Thus, the machines have here in this case in a sense some kind of access to a database (oracle), and they are required to work correctly for almost all such databases.

It has been observed that in a number of classes, the ALMOST- and the $\mathrm{BP^p}$-operator coincide, e.g. ALMOST-$\mathrm{P} = \mathrm{BPP}$ and ALMOST-$\mathrm{NP} =$ AM [6, 28]. However, the general relationship between the operators is open. Especially, no characterization of ALMOST-PSPACE is known.

In this paper we introduce a type-2 probabilistic quantifier, which we will denote by BP^2, and show that for a wide variety of classes, the ALMOST- and the BP^2 operators coincide, i.e., ALMOST-$\mathcal{K} = \mathrm{BP}^2\mathcal{K}$. "Type 2" means that the operator is based on a quantifier that does not range over words but over *sets* (databases, i.e., oracles). Thus ALMOST-classes are classes accepted with bounded-error probability by machines with access to a random database. Moreover, it is not too hard to see that the type-2 operator BP^2 can often be replaced equivalently by a "classical" operators ranging over finite (i.e., type 1) objects (words). The most important special case here is the case of so called *leaf language definable* classes $\mathcal{K}$ (a definition is given in Section 2). Here we see that the BP^2 operator coincides with a type-1 operator ranging over exponentially long strings (compared to the length of the input), in contrast to the usual quantifiers where polynomially long words are considered. We will denote this new operator by $\mathrm{BP}^{\exp}$. Combining this with the above, we get for such $\mathcal{K}$ that ALMOST-$\mathcal{K}$ = $\mathrm{BP}^2\mathcal{K}$ = $\mathrm{BP}^{\exp}\mathcal{K}$, for example ALMOST-$\mathrm{PSPACE}$ = $\mathrm{BP}^{\exp}\mathrm{PSPACE}$ or ALMOST-PP = $\mathrm{BP}^{\exp}\mathrm{PP}$. Thus, in this case it turns out that working in the "ALMOST-mode" is equivalent to working with random input sequences. We think this characterization is advantageous compared to the definition since here we only have to deal with finite objects (strings) in contrast to (infinite) oracles.

We give several applications of our characterization: Since for all classes $\mathcal{K}$, $\mathrm{BP}^{\mathrm{p}}\mathcal{K} \subseteq \mathrm{BP}^2\mathcal{K}$, we see that a relativizable inclusion $\mathcal{K}_1 \subseteq \mathrm{BP}^{\mathrm{p}}\mathcal{K}_2$ implies the measure 1 inclusion $\mathcal{K}_1 \subseteq \mathcal{K}_2$. As a consequence, we show that e.g. the set of all oracles relative to which the polynomial time hierarchy is strictly included in $\oplus\mathrm{P}$ has measure 1 (a result which has already been proved by Regan and Royer [35]). We improve the best known EXPSPACE upper bound for ALMOST-PSPACE to $\Sigma_2^{\exp} \cap \Pi_2^{\exp}$. We prove that ALMOST-$\mathrm{PSPACE}$ allows a machine characterization in terms of checking stack automata. These automata were introduced by Oscar Ibarra in [21], where it was also shown that when working nondeterministically these machines are strictly power powerful than when working deterministically. Our results imply that the nondeterministic mode is (under reasonable complexity theoretic assumptions) even more powerful than the bounded-error probabilistic mode. Finally, we draw a connection between the power of ALMOST-PSPACE and a problem from circuit complexity by showing that proving upper bounds for probabilistic NC^1 circuits better than the up to now known BPP-bound will result in better upper bounds for ALMOST-PSPACE.

All in all we see that our systematic comparison of several ways of introducing randomness into computation allows us to improve a number of results for complexity classes of current topical interest. Along the way, we get new insights into the relationship between statements holding for a measure 1 set of oracles and statements holding for an algorithmically random oracle in the sense of Martin-Löf (see [10]), thus improving results in [9, 22].

2 Preliminaries

We assume that the reader is familiar with basic complexity theory notions, classes and reducibilities, see e.g. [4, 32]. Let $\{0,1\}^*$ denote the set of finite binary words, whereas $\{0,1\}^\omega$ denotes the set of infinite binary words. Following common use, we identify a language, i.e. a subset of $\{0,1\}^*$, with its characteristic sequence, which is an element of $\{0,1\}^\omega$. For $w \in \{0,1\}^*$ we denote the i-th bit of w by $w(i)$. Similarly, for $A \in \{0,1\}^\omega$ we denote the i-th bit of A by $A(i)$. Using the lexicographic ordering of $\{0,1\}^*$, there is a natural bijection between $\{0,1\}^*$ and the set $\mathbb{N}$ of natural numbers. Thus, we will also write $A(w)$ for $w \in \{0,1\}^*$ and $A \in \{0,1\}^\omega$. We then mean the bit in A at "position w," i.e. $A(w) = 1$ if $w \in A$ and $A(w) = 0$ otherwise.

For concreteness, we fix the following pairing function: Let $u, v \in \{0,1\}^*$, $u = u_1 \cdots u_{|u|}$, $v = v_1 \cdots v_{|v|}$, where $u_1, \ldots, u_{|u|}, v_1, \ldots, v_{|v|} \in \{0,1\}$. Then we define $\langle u, v \rangle =_{\mathrm{def}} u_1 u_1 \cdots u_{|u|} u_{|u|} 01 v_1 v_1 \cdots v_{|v|} v_{|v|}$.

Let $\{M_i\}_{i \in \mathbb{N}}$ be a recursive enumeration of all oracle Turing machines. Let $M_i^A(x)$ be the result of M_i's work on input x and oracle A if this computation stops, and let $M_i^A(x)$ be undefined otherwise. Define $L(M_i) =_{\mathrm{def}} \{\, (A, x) \mid M_i^A(x) = 1 \,\}$ and $L(M_i^A) =_{\mathrm{def}} \{\, x \mid M_i^A(x) = 1 \,\}$.

A class $\mathcal{K}^{(\cdot)} \subseteq 2^{\{0,1\}^\omega \times \{0,1\}^*}$ is a *relativized class* if and only if there exists a recursive function f (the *enumeration function*) such that

- $M_{f(j)}^A(x)$ halts for every j, x, A with result 0 or 1.
- $\mathcal{K}^{(\cdot)} = \{\, L(M_{f(j)}) \mid j \in \mathbb{N} \,\}$.

Define $\mathcal{K}^A =_{\mathrm{def}} \{\, L(M_{f(j)}^A) \mid j \in \mathbb{N} \,\}$ and $\mathcal{K} =_{\mathrm{def}} \mathcal{K}^\emptyset$.

Proposition 2.1 *For every relativized class $\mathcal{K}^{(\cdot)}$, every set in $\mathcal{K}^A$ is recursive in A. Particularly, every set in $\mathcal{K}$ is recursive.*

Proof. By the requirement that the machines that form a recursive enumeration halt on all their inputs. ❑

We say that a relativized class $\mathcal{K}$ is *invariant under finite variations of the oracle,* if and only if $\mathcal{K}^A = \mathcal{K}^B$ for every $A, B \in \{0,1\}^\omega$ such that $A \triangle B$ is finite. A relativized class $\mathcal{K}$ with enumeration function f is *uniformly invariant under finite variations of the oracle,* if and only if for every $u \in \{0,1\}^*$ and every $i \in \mathbb{N}$ there exists a $j \in \mathbb{N}$ such that for all oracles A we have $L(M_{f(i)}^{u \cdot A}) = L(M_{f(j)}^A)$, where $u \cdot A$ is defined as $u(1)u(2)\cdots u(|u|)A(|u|+1)A(|u|+2)\cdots$. Note that the uniform invariance under finite variations of the oracle implies the (simple) invariance under finite variations of the oracle.[1]

A special type of relativized classes are those defined by *leaf languages* (see [12, 20] and the recent textbook [32]). Let $\{N_i\}_{i \in \mathbb{N}}$ be a recursive enumeration of all polynomial time nondeterministic oracle Turing machines such that for every $i \in \mathbb{N}$, every oracle A and every input x, every path of N_i on input x with oracle A is time bounded by $|x|^i + i$ and produces a symbol from some finite alphabet Σ_i. Let $\beta_{M_i}^A(x)$ be the string of the such produced symbols (based on the natural order of paths of the machine). For some

[1] Merkle [27] has pointed out that the converse can also be shown using the patching methods from [26].

$B \subseteq \Sigma^*$, the class $(B)\mathrm{P}^{(\cdot)}$ is the class of all languages L for which there is some $i \in \mathbb{N}$ such that $L = \{ (A, x) \mid \beta^A_{N_i}(x) \in B \}$. In this case, B is the so called *leaf language* defining class $(B)\mathrm{P}^{(\cdot)}$. As above, define $(B)\mathrm{P}^A$ for some oracle A and $(B)\mathrm{P} = (B)\mathrm{P}^{\emptyset}$. Note that every $(B)\mathrm{P}^{(\cdot)}$ is uniformly invariant under finite variations of the oracle.

Let $\mu\colon 2^{\{0,1\}^\omega} \to [0,1]$ be the product measure based on the measure $\mu_0\colon 2^{\{0,1\}} \to [0,1]$ which is defined by $\mu_0(\{0\}) = \mu_0(\{1\}) = \frac{1}{2}$. If $\Pi(\cdot)$ is some predicate with a free set variable, then we write also $\mu A(\Pi(A))$ instead of $\mu(\{ A \mid \Pi(A) \})$.

We will use the following well known fact:

Proposition 2.2 (Kolmogorov 0-1-law) *If $\mathcal{C} \subseteq \{0,1\}^\omega$ is measurable and closed under finite variations, then either $\mu(\mathcal{C}) = 0$ or $\mu(\mathcal{C}) = 1$.*

We will also make use of the following observation, which is an immediate consequence of the Lebesgue Density Theorem (see also [36, p. 272] or [28, Fact on p. 163]):

Proposition 2.3 *Let $\mathcal{K}^{(\cdot)}$ be a relativized class with enumeration function f, which is uniformly invariant under finite variations of the oracle. Let $\epsilon > 0$ and let L be a language. Then for every $i \in \mathbb{N}$ such that $\mu A(L = L(M^A_{f(i)})) > 0$ there exists a $j \in \mathbb{N}$ such that $\mu A(L = L(M^A_{f(j)})) > 1 - \epsilon$.*

3 Bounded-error probabilistic operators

In this subsection we will introduce several bounded-error probabilistic operators. We start with "classical" operators, i.e. operators based on quantifiers which range over (finite) words. Let $h : \{0,1\}^* \to \mathbb{N}$ be any recursive function. For relativized classes $\mathcal{K}^{(\cdot)}$ we define the operators BP_h, BP^{p}, and $\mathrm{BP}^{\exp}$ as follows:

- $L \in \mathrm{BP}_h\mathcal{K}$ iff there exists an $L' \in \mathcal{K}$ such that for every x,
$$\#\{ z \mid |z| = h(|x|) \wedge (x \in L \leftrightarrow (\emptyset, \langle x, z\rangle) \in L') \} \geq \frac{2}{3} \cdot 2^{h(|x|)}.$$
- $L \in \mathrm{BP}^{\mathrm{p}}\mathcal{K}$ iff there exists a polynomial q such that $L \in \mathrm{BP}_q\mathcal{K}$
- $L \in \mathrm{BP}^{\exp}\mathcal{K}$ iff there exists a polynomial q such that $L \in \mathrm{BP}_{2^q}\mathcal{K}$

A result by Nisan and Wigderson [28] states the coincidence of BP^{p} and $\mathrm{BP}^{\exp}$ when applied to some important complexity classes (Note that Σ^{p}_k-machines can have access to the bits of an exponentially long auxiliary input string via a special index tape):

Theorem 3.1 $\mathrm{BP}^{\exp}\Sigma^{\mathrm{p}}_k = \mathrm{BP}^{\mathrm{p}}\Sigma^{\mathrm{p}}_k$ *and* $\mathrm{BP}^{\exp}\Pi^{\mathrm{p}}_k = \mathrm{BP}^{\mathrm{p}}\Pi^{\mathrm{p}}_k$ *for every* $k \geq 1$

Type 2 operators are operators ranging over languages (oracles). More specifically, let $\mathcal{K}^{(\cdot)}$ be a relativized class. (If no confusion can arise, we will from now on omit the superscript $(\cdot)$.) Then we define type 2 operators ALMOST-, BP^2, $\widehat{\mathrm{BP}}^2$, and $\widetilde{\mathrm{BP}}^2$ as follows:

- $L \in$ ALMOST-$\mathcal{K}$ iff $\mu A(L \in \mathcal{K}^A) = 1$.
- $L \in \mathrm{BP}^2\mathcal{K}$ iff there exists an $L' \in \mathcal{K}$ such that for all x,
$$\mu A(x \in L \leftrightarrow (A, x) \in L') \geq \tfrac{2}{3}.$$
- $L \in \widehat{\mathrm{BP}}^2\mathcal{K}$ iff for every polynomial p there exists an $L' \in \mathcal{K}$ such that for all x,
$$\mu A(x \in L \leftrightarrow (A, x) \in L') \geq 1 - 2^{-p(|x|)}.$$
- $L \in \widetilde{\mathrm{BP}}^2\mathcal{K}$ iff there exists an $L' \in \mathcal{K}$ such that $\mu A(L = \{\, x \mid (A, x) \in L' \,\}) \geq \frac{2}{3}$.

In the definition of ALMOST- and $\widetilde{\mathrm{BP}}^2$ the condition on oracle A makes use of all instances of A. Thus the use of infinite objects in this definition seems to be unavoidable. The situation is different for the operators BP^2 and $\widehat{\mathrm{BP}}^2$ where the conditions on an oracle involve only single inputs x. Since the underlying machines halt for every input x with every oracle A, only a finite part of the oracle (whose length is computable) is really used. This suggests that for every relativized class $\mathcal{K}$ there exists a recursive function h such that $\mathrm{BP}^2\mathcal{K} = \mathrm{BP}_h\mathcal{K}$. However, there are technical difficulties to state such a general theorem since if we simulate machines with oracle queries by machines which instead consult their input bits in a straightforward way, then we end up with another relativized class. However, we are especially interested in the case of leaf language defined classes, and there we can prove the following:

Proposition 3.2 $\mathrm{BP}^2(B)\mathrm{P} = \mathrm{BP}^{\exp}(B)\mathrm{P}$ *for every recursive set B.*

Proof. We just noted that by simply replacing oracle queries by consuming input bits we might leave the relatived class under consideration. However, in the case of leaf language classes the robustness of the underlying class of machines allows the required simulation. To be more precise, for every polynomial time machine M with run time p we can construct a machine M' such that $(A, x) \in L(M)$ if and only if $(\emptyset, \langle x, z\rangle) \in L(M')$, where z is the length $2^{p(|x|)+1} - 1$ prefix of A.

Vice versa, we find for every polynomial time NTM M and every polynomial q an NTM M' such that for every x and every z of length $2^{q(|x|)}$ we have $(\emptyset, \langle x, z\rangle) \in L(M)$ if and only if $(A, x) \in L(M')$ for every oracle A with prefix z.

Thus, we see that though the accepted language changes when we go from M to M', the obtained classes under the operators BP^2 and $\mathrm{BP}^{\exp}$ are the same. ❑

Corollary 3.3 *1.* $\mathrm{BP}^2\mathcal{K} = \mathrm{BP}^{\mathrm{exp}}\mathcal{K}$ *for* $\mathcal{K} = \mathrm{NP}$, coNP, Σ_k^{p}, Π_k^{p}, Δ_k^{p}, Θ_k^p *(for* $k \geq 2$*),* PH, $\oplus\,\mathrm{P}$, PP, PSPACE*, and many others.*

2. $\mathrm{BP}^2\mathrm{L} = \mathrm{BP}^{\mathrm{p}}\mathrm{L}$.

Proof. Statement 1 follows immediately from Proposition 3.2. The proof for Statement 2 is very similar. ❑

Corollary 3.4 *If* $\mathcal{K} = (B)\mathrm{P}$ *for a recursive* B*, then every set in the class* $\mathrm{BP}^2\mathcal{K}$ *is recursive.*

We say that class $\mathcal{K}^{(\cdot)}$ has the *amplification property,* if $\mathrm{BP}^2\mathcal{K} = \widehat{\mathrm{BP}}^2\mathcal{K}$. We say that $\mathcal{K}^{(\cdot)}$ with enumeration function f has the *uniform amplification property,* if for every polynomial p and every $i \in \mathbb{N}$ there exists a $j \in \mathbb{N}$ such that for every oracle B and every input x, if $\mu A(x \in L(M_{f(i)}^{A\oplus B})) \geq \frac{2}{3}$ then $\mu A(x \in L(M_{f(j)}^{A\oplus B})) \geq 1 - 2^{-p(|x|)}$; and if $\mu A(x \notin L(M_{f(i)}^{A\oplus B})) \geq \frac{2}{3}$ then $\mu A(x \notin L(M_{f(j)}^{A\oplus B})) \geq 1 - 2^{-p(|x|)}$.

It is straightforward to verify that the following classes have the uniform amplification property (see [37]): NP, coNP, Σ_k^{p}, Π_k^{p}, Δ_k^{p}, Θ_k^p (for $k \geq 2$), PH, $\oplus\,\mathrm{P}$, PP, PSPACE, and many more.

4 Relationships between different operators

We start with the following inclusion chain between classes defined by the different type 2 operators:

Theorem 4.1 *If* $\mathcal{K}$ *is a relativizable class which is uniformly invariant under finite variations of the oracle, then*

$$\widehat{\mathrm{BP}}^2\mathcal{K} \subseteq \text{ALMOST-}\mathcal{K} = \widetilde{\mathrm{BP}}^2\mathcal{K} \subseteq \mathrm{BP}^2\mathcal{K}.$$

Proof. Let f be an enumeration function for $\mathcal{K}$.

$\widehat{\mathrm{BP}}^2\mathcal{K} \subseteq$ ALMOST-$\mathcal{K}$: Let $L \in \widehat{\mathrm{BP}}^2\mathcal{K}$. Then, for every $k \in \mathbb{N}$, there exists a machine $M_{f(i)}$ such that $\mu A(x \in L \iff x \in L(M_{f(i)}^A)) \geq 1 - 2^{-2|x|-k-2}$. Now, $\mu A(L \in \mathcal{K}^A) \geq \mu A(L = L(M_{f(i)}^A)) = 1 - \mu A(L \neq L(M_{f(i)}^A))$, and $\mu A(L \neq L(M_{f(i)}^A)) = \mu A(\exists x(x \in L \leftrightarrow x \notin L(M_{f(i)}^A))) \leq \sum_x \mu A(x \in L \leftrightarrow x \notin L(M_{f(i)}^A)) \leq \sum_x 2^{-2|x|-k-2} = 2^{-k}$. Hence, $\mu A(L \in \mathcal{K}^A) \geq 1 - 2^{-k}$ for every k; i.e. $L \in$ ALMOST-$\mathcal{K}$.

ALMOST-$\mathcal{K} \subseteq \widetilde{\mathrm{BP}}^2\mathcal{K}$: Let $\mu A(L \in \mathcal{K}^A) = 1$. Then there exists an $i \in \mathbb{N}$ such that $\mu A(L = L(M_{f(i)}^A)) > 0$. By Proposition 2.3 there exists a machine $M_{f(j)}$ such that $\mu A(L = L(M_{f(j)}^A)) \geq \frac{2}{3}$.

$\widetilde{\mathrm{BP}}^2\mathcal{K} \subseteq$ ALMOST-$\mathcal{K}$: Let $\mu A(L = L(M^A_{f(i)})) \geq \frac{2}{3}$. By Proposition 2.3, for every $\epsilon > 0$ there exists a $j \in \mathbb{N}$ such that $\mu A(L = L(M^A_{f(j)})) \geq 1 - \epsilon$. Hence, $\mu A(L \in \mathcal{K}^A) = 1$.

Finally, $\widetilde{\mathrm{BP}}^2\mathcal{K} \subseteq \mathrm{BP}^2\mathcal{K}$ is obvious. ❑

An immediate consequence is that we obtain the equivalence between all type 2 bounded-error probabilistic quantifiers for classes which have the amplification property.

Corollary 4.2 *If $\mathcal{K}$ is a relativizable class which is uniformly invariant under finite variations of the oracle and has the amplification property, then*

$$\mathrm{BP}^2\mathcal{K} = \text{ALMOST-}\mathcal{K} = \widetilde{\mathrm{BP}}^2\mathcal{K} = \widehat{\mathrm{BP}}^2\mathcal{K}.$$

The first equality of the just given corollary was proved independently, and in fact somewhat earlier, by Merkle and Wang in [26]. The special case ALMOST-$\mathrm{L} = \mathrm{BP}^2\mathrm{L}$ can already be found in [30].

In the light of Proposition 3.2 this result says that all our bounded-error probabilistic operators of type 2, which are defined by quantifiers over infinite sets, can also be defined by quantifiers ranging over finite words, when we deal with leaf language defined classes. This makes these operators easier to understand and to handle.

Corollary 4.3 *For any recursive B such that $(B)\mathrm{P}$ has the amplification property, we have* ALMOST-$(B)\mathrm{P} = \mathrm{BP}^{\exp}(B)\mathrm{P}$.

Thus, we get as consequences all the known results about ALMOST-classes mentioned in the introduction, i.e. (1) ALMOST-$\mathrm{P} = \mathrm{BPP}$, (2) ALMOST-$\mathrm{NP} = \mathrm{BP}^{\mathrm{p}}\mathrm{NP} = \mathrm{AM}$, (3) ALMOST-$\mathrm{PH} = \mathrm{BP}^{\mathrm{p}}\mathrm{PH} = \mathrm{PH}$; but as well characterizations of ALMOST-classes which are of current topical interest, where no coincidence with a "classical" class is known, e. g. (4) ALMOST-$\mathrm{PP} = \mathrm{BP}^{\exp}\mathrm{PP}$, (5) ALMOST-$\mathrm{PSPACE} = \mathrm{BP}^{\exp}\mathrm{PSPACE}$. Observe that for the equalities (2) and (3), we need Theorem 3.1, which builds on the pseudorandom numger generator construction from [28]. A similar newer construction of a pseudorandom generator for space-bounded computations is presented in [29]. One might first suspect that this newer generator leads to a positive settlement of the ALMOST-$\mathrm{PSPACE} \stackrel{?}{=} \mathrm{PSPACE}$ question, but this is not the case since this generator can only fool a machine with *one-way access* to its random bits. These questions are discussed in the appendix of [30].

The following result shows that for almost all oracles A, the class of recursive sets in $\mathcal{K}^A$ coincides with ALMOST-$\mathcal{K}$. Let REC denote the class of all recursive sets.

Theorem 4.4 *If $\mathcal{K} = (B)\mathrm{P}$ for a recursive set B and if $\mathcal{K}$ is uniformly invariant under finite variations of the oracle, then*

$$\mu A(\text{ALMOST-}\mathcal{K} = \mathcal{K}^A \cap \mathrm{REC}) = 1.$$

Proof. 1. Because of ALMOST-$\mathcal{K} \subseteq \mathrm{BP}^2\mathcal{K}$ and Proposition 3.4 every $L \in$ ALMOST-$\mathcal{K}$ is recursive. Hence, $\mu A(\text{ALMOST-}\mathcal{K} \not\subseteq \mathcal{K}^A \cap \mathrm{REC}) = \mu A(\text{ALMOST-}\mathcal{K} \not\subseteq \mathcal{K}^A) = \mu A(\exists L(L \in \text{ALMOST-}\mathcal{K} \wedge L \notin \mathcal{K}^A)) = \mu(\bigcup_{L\in \mathrm{ALMOST}-\mathcal{K}}\{ A \mid L \notin \mathcal{K}^A \}) = 0$, since this is a countable union of measure 0 sets.

2. We conclude as follows: $\mu A(\mathcal{K}^A \cap \mathrm{REC} \not\subseteq \text{ALMOST-}\mathcal{K}) = \mu A(\exists L(L \in \mathcal{K}^A \cap \mathrm{REC} \wedge L \notin \text{ALMOST-}\mathcal{K})) = \mu(\bigcup_{L\in\mathrm{REC}}\{ A \mid L \in \mathcal{K}^A \wedge \mu B(L \in \mathcal{K}^B) < 1 \}) = \mu(\bigcup_{L\in\mathrm{REC}}\{ A \mid L \} \in \mathcal{K}^A \wedge \mu B(L \in \mathcal{K}^B) = 0) = 0$, where the third equality is a consequence of Proposition 2.2. ❑

Bennett and Gill [6] showed that the class of all oracles relative to which $\mathrm{BPP} = \mathrm{P}$ has measure 1. Using the operator BP^2, we can generalize this result for a large variety of relativized classes instead of P.

Theorem 4.5 *If $\mathcal{K} = (B)\mathrm{P}$ for some recursive set B and if $\mathcal{K}$ has the uniform amplification property, then*

$$\mu A(\mathrm{BP}^2\mathcal{K}^A = \mathcal{K}^A) = 1.$$

Proof. The proof follows the one given by Bennett and Gill for their just mentioned result.

Recall that $N_1, N_2, \ldots$ denotes an effective enumeration of all polynomial time machines, where for every i, N_i is time-bounded by the polynomial $n^i + i$. Define a function f as follows: For every $i \in \mathbb{N}$, let $f(i)$ be the index (in our enumeration of all oracle Turing machines, see Section 2) of the machine which on input x simulates all paths of N_i and then accepts iff $\beta^A_{N_i}(x) \in B$. Obviously $(B)\mathrm{P}^A = \{ L(M^A_{f(i)}) \mid i \in \mathbb{N} \}$.

Fix a number $k \geq 0$. By the uniform amplification property there exists a function r such that for every A, if $\mu C(x \in L(M^{A,C}_{f(i)})) \geq \frac{2}{3}$ then $\mu C(x \in L(M^{A,C}_{f(r(i))})) \geq 1 - 2^{-2\cdot|x|-i-k-2}$, and if $\mu C(x \in L(M^{A,C}_{f(i)})) \leq \frac{1}{3}$ then $\mu C(x \in L(M^{A,C}_{f(r(i))})) \leq 2^{-2\cdot|x|-i-k-2}$.

Now, it will be our goal to replace the two oracles in the above by queries to just one oracle. For this, define a function s by the following construction: Machine $M^A_{f(s(i))}$ simulates the computation of $M^{A,C}_{f(r(i))}$ but handles oracle queries differently: When $M^{A,C}_{f(r(i))}$ asks query z to oracle A, then $M^A_{f(s(i))}$ asks z to its oracle; and when $M^{A,C}_{f(r(i))}$ asks query z to oracle C, then $M^A_{f(s(i))}$ asks query $2^{|x|^{r(i)}+r(i)} + z$ to its oracle (in order for the "+"

to make sense, we of course use the bijection between $\{0,1\}^*$ and $\mathbb{N}$ mentioned in Section 2). Hence, $M^{u\cdot C,C}_{f(r(i))} = M^{u\cdot C}_{f(s(i))}$, if $|u| = e(x,i) =_{\text{def}} 2^{|x|^{r(i)}+r(i)}$.

Now we argue as follows: If the implications $\mu C(x \in L(M^{A,C}_{f(i)})) \geq \frac{2}{3} \implies x \in L(M^{A}_{f(s(i))})$ and $\mu C(x \in L(M^{A,C}_{f(i)})) \leq \frac{1}{3} \implies x \notin L(M^{A}_{f(s(i))})$ both hold for every $i \in \mathbb{N}$ and every input x, then certainly $\mathrm{BP}^2(B)\mathrm{P}^A = (B)\mathrm{P}^A$. This allows us to show that the set of all A such that $\mathrm{BP}^2(B)\mathrm{P}^A \neq (B)\mathrm{P}^A$ has measure zero by the following calculation, which we can make for every numer $k \in \mathbb{N}$:

$$
\begin{aligned}
&\mu A(\mathrm{BP}^2(B)\mathrm{P}^A \neq (B)\mathrm{P}^A) \leq \\
&\quad\leq \mu A((\exists i \in \mathbb{N})(\exists x \in \{0,1\}^*)((\mu C(x \in L(M^{A,C}_{f(i)})) \geq \tfrac{2}{3} \wedge x \notin L(M^{A}_{f(s(i))})) \\
&\qquad\qquad \vee(\mu C(x \in L(M^{A,C}_{f(i)})) \leq \tfrac{1}{3} \wedge x \in L(M^{A}_{f(s(i))})))) \\
&\quad\leq \sum_{i\in\mathbb{N}} \sum_{x\in\{0,1\}^*} (\mu A(\mu C(x \in L(M^{A,C}_{f(i)})) \geq \tfrac{2}{3} \wedge x \notin L(M^{A}_{f(s(i))})) \\
&\qquad\qquad + \mu A(\mu C(x \in L(M^{A,C}_{f(i)})) \leq \tfrac{1}{3} \wedge x \in L(M^{A}_{f(s(i))}))) \\
&\quad\leq \sum_{i\in\mathbb{N}} \sum_{x\in\{0,1\}^*} \Big(\sum_{\substack{u \text{ s.t. } |u|=e(x,i) \\ C \text{ s.t. } \mu C(x\in L(M^{A,C}_{f(i)}))\geq\frac{2}{3}}} \tfrac{1}{2^{e(x,i)}} \cdot \mu A(A = u\cdot C \wedge x \notin L(M^{A}_{f(s(i))})) \\
&\qquad\qquad + \sum_{\substack{u \text{ s.t. } |u|=e(x,i) \\ C \text{ s.t. } \mu C(x\in L(M^{A,C}_{f(i)}))\geq\frac{1}{3}}} \tfrac{1}{2^{e(x,i)}} \cdot \mu A(A = u\cdot C \wedge x \in L(M^{A}_{f(s(i))}))\Big) \\
&\quad\leq \sum_{i\in\mathbb{N}} \sum_{x\in\{0,1\}^*} \sum_{u \text{ s.t. } |u|=e(x,i)} \tfrac{1}{2^{e(x,i)}} \cdot 2^{-2|x|-i-k-2} \\
&\quad= \sum_{i\in\mathbb{N}} \sum_{x\in\{0,1\}^*} 2^{-2|x|-i-k-2} = \tfrac{1}{2^k}\Big(\sum_{i=0}^{\infty} \tfrac{1}{2^{i+1}}\Big)\Big(\sum_{n=0}^{\infty} 2^n \cdot \tfrac{1}{2^{2n+1}}\Big) = \tfrac{1}{2^k}
\end{aligned}
$$

❑

5 Measure 1 inclusions between complexity classes

Inclusions between classes that hold relative to oracles with probability 1 have been an important topic in complexity theory, see e.g. [6, 13, 14, 35] and many more. From Theorem 4.1, we obtain the following general result:

Theorem 5.1 *Let $\mathcal{K}_1, \mathcal{K}_2$ be relativizable classes, where $\mathcal{K}_2$ is uniformly invariant under finite variations of the oracle and has the amplification property. Then the following holds:*

$$\mathcal{K}_1 \subseteq \mathrm{BP}^2\mathcal{K}_2 \iff \mu A(\mathcal{K}_1 \subseteq \mathcal{K}_2^A) = 1.$$

Proof. " $\Longrightarrow$ ": If $\mathcal{K}_1 \subseteq \mathrm{BP}^2\mathcal{K}_2$, then by Corollary 4.2, $\mathcal{K}_1 \subseteq \text{ALMOST-}\mathcal{K}_2$. Thus, for every $L \in \mathcal{K}_1$, $\mu A(L \in \mathcal{K}_2^A) = 1$. Then we conclude $\mu A(K_1 \not\subseteq \mathcal{K}_2^A) \leq \sum_{L \in \mathcal{K}_1} \mu A(L \notin \mathcal{K}_2^A) = 0$.

"$\Longleftarrow$": If $\mu A(\mathcal{K}_1 \subseteq \mathcal{K}_2^A) = 1$, then for every $L \in \mathcal{K}_1$, we have $\mu A(L \in \mathcal{K}_2^A) = 1$; thus $L \in \text{ALMOST-}\mathcal{K}_2^A$, which implies by Theorem 4.1 that $L \in \mathrm{BP}^2\mathcal{K}_2$. ❑

For classes defined via leaf languages, we find the following "lifting" for measure 1 inclusions:

Theorem 5.2 *Let $\mathcal{K}_1, \mathcal{K}_2$ be relativizable classes, where $\mathcal{K}_2 = (B)\mathrm{P}$ for some recursive leaf language B, and $\mathcal{K}_2$ has the amplification property. Then the following holds:*

$$\mu A(\mathcal{K}_1^A \subseteq \mathcal{K}_2^A) = \mu A(\mathcal{K}_1^A \subseteq \mathrm{BP}^2\mathcal{K}_2^A).$$

In particular,

$$\mu A(\mathcal{K}_1^A \subseteq \mathcal{K}_2^A) = 1 \quad \Longleftrightarrow \quad \mu A(\mathcal{K}_1^A \subseteq \mathrm{BP}^2\mathcal{K}_2^A) = 1$$

$$\mu A(\mathcal{K}_1^A \not\subseteq \mathcal{K}_2^A) = 1 \quad \Longleftrightarrow \quad \mu A(\mathcal{K}_1^A \not\subseteq \mathrm{BP}^2\mathcal{K}_2^A) = 1$$

Proof. Follows from Theorem 4.5. ❑

From this, we conclude immediatly the following easily applicable criterion to get measure 1 inclusions:

Corollary 5.3 *Let $\mathcal{K}_1, \mathcal{K}_2$ be relativizable classes, where $\mathcal{K}_2 = (B)\mathrm{P}$ for some recursive leaf language B, and $\mathcal{K}_2$ has the amplification property. Then the following holds:*

If $\mathcal{K}_1 \subseteq \mathrm{BP}^2\mathcal{K}_2$ is relativizable, then $\mu A(\mathcal{K}_1^A \subseteq \mathcal{K}_2^A) = 1$.

This gives us a number of applications:

Let PH denotes the union of all classes of the polynomial time hierarchy and $\oplus\mathrm{P}$ denotes Papadimitrou and Zachos's "modest counting class" [34].

Corollary 5.4 $\mu A(\mathrm{PH}^A \subsetneq \oplus\mathrm{P}^A) = 1$.

Proof. Toda [40] showed that $\mathrm{PH} \subseteq \mathrm{BP}^{\mathrm{p}} \oplus \mathrm{P}$. It is easy to observe that his result in fact holds relativizably. For every leaf language definable classes $\mathcal{K}$, we have obviously $\mathrm{BP}^{\mathrm{p}}\mathcal{K} \subseteq \mathrm{BP}^2\mathcal{K}$. Now the inclusion follows immediately from Corollary 5.3. The strictness follows from the proof given by Cai in [13], where he not only separates PH from PSPACE, but in fact from $\oplus\mathrm{P}$. ❑

Corollary 5.4 has already been shown in [35].

Corollary 5.5 $\mu A(\mathrm{coNP}^A \not\subseteq \mathrm{AM}^A) = 1$

Proof. Follows from the fact $\mu A(\mathrm{coNP}^A \not\subseteq \mathrm{NP}^A) = 1$ [6], from Corollary 5.2, and from $\mathbf{BP}^{\mathrm{p}}\mathrm{NP} \subseteq \mathbf{BP}^2\mathrm{NP}$. ❑

Corollary 5.6 $\mu A(\mathrm{PH}^A \subsetneq \mathrm{C}_{=}\mathrm{P}^A \subsetneq \mathrm{PP}^A \subsetneq \mathrm{PSPACE}^A) = 1$.

Proof. $\mathrm{PH}^A \subseteq \mathbf{BP}^{\mathrm{p}}\,\mathrm{C}_{=}\mathrm{P}^A$ for all oracles A follows by observing that the proof given for $\mathrm{PH} \subseteq \mathbf{BP}^{\mathrm{p}}\,\mathrm{C}_{=}\mathrm{P}$ in [41] relativizes. Thus $\mu A(\mathrm{PH}^A \subseteq \mathrm{C}_{=}\mathrm{P}^A) = 1$ follows from Corollary 5.3. Green's result that $\mathrm{C}_{=}\mathrm{P} \neq \mathrm{co}\,\mathrm{C}_{=}\mathrm{P}$ for all random oracles [19] now shows that $\mu A(\mathrm{PH}^A \subsetneq \mathrm{C}_{=}\mathrm{P}^A) = 1$. $\mathrm{C}_{=}\mathrm{P}^A \subseteq \mathrm{PP}^A$ holds for all oracles A. $\mu A(\mathrm{C}_{=}\mathrm{P}^A \subsetneq \mathrm{PP}^A) = 1$ follows again from Green's result. The strict inclusion of PP in PSPACE relative to any random oracle was shown in [2]. This immediately yields $\mu A(\mathrm{PP}^A \subsetneq \mathrm{PSPACE}^A) = 1$. ❑

Let US denote the class of all sets A for which there is a nondeterministic polynomial time Turing machine M such that for all x, $x \in A$ if and only if M on input x has exactly one accepting path [7]. Valiant and Vazirani [42] show that NP randomly reduces to US. However, the error probability in their reduction is not small enough to get $\mathrm{NP} \subseteq \mathbf{BP}^2\mathrm{US}$ (which would then immediately allow us to apply Corollary 5.3), and as argued in [16] there is no obvious way to amplify the reduction.

However, the situation becomes simpler if we consider the disjunctive truth-table closure of US (which we denote by $\mathcal{R}^{\mathrm{p}}_{\mathrm{dtt}}(\mathrm{US})$). Let Θ_2^{p} denote the restriction of P^{NP}, where the deterministic base machines are allowed to ask only $O(\log n)$ oracle queries on inputs of length n [45]. Now we can show:

Corollary 5.7 $\mu A(\Theta_2^{\mathrm{p}A} = \mathcal{R}^{\mathrm{p}}_{\mathrm{dtt}}(\mathrm{US})^A) = 1$.

Proof. In [16, Fact 1] it is argued that a random reduction from $\mathcal{K}$ to $\mathcal{K}'$ in the sense of Valiant/Vazirani can be amplified if the class $\mathcal{K}'$ is closed under disjunctive truth-table reducibility. Moreover, in [16, Lemma 1] it is shown that Θ_2^{p} randomly reduces to US in the sense of Valiant/Vazirani. Both results together yield $\Theta_2^{\mathrm{p}} \subseteq \mathbf{BP}^2\mathcal{R}^{\mathrm{p}}_{\mathrm{dtt}}(\mathrm{US})$. It can easily be checked that this inclusion even holds relativizably. Thus, the statement follows from Corollary 5.3. ❑

6 Random oracles

In the preceding sections, we obtained results for a class of oracles with measure 1. In this section, we want to contrast these results with results for *one single random oracle.* We denote by RAND the class of all random oracles in the sense of Martin-Löf, see [10].

The relationship between statements holding for a measure 1 set of oracles vs. those holding for a single random oracle vs. those holding for all random oracles has been examined in several papers [9, 10, 22]. Our Theorem 6.1 extends these results.

We recall that some $\mathcal{C} \subseteq \{0,1\}^\omega$ is *recursively open,* if $\mathcal{C} = W{\cdot}\{0,1\}^\omega$ for some recursively enumerable set $W \subseteq \{0,1\}^*$. A set $\mathcal{C}$ is *recursively* G_δ*,* if $\mathcal{C} = \bigcap_{i=1}^\infty \mathcal{C}_i$ where the $\mathcal{C}_1, \mathcal{C}_2, \ldots$ are recursively open. A set $\mathcal{C}$ is *recursively* F_σ*,* if $\mathcal{C}$ is the complement of a set which is recursively G_δ. The *σ-algebra over a class $\mathcal{K} \subseteq 2^{\{0,1\}^\omega}$* is the smallest class containing $\mathcal{K}$ closed under complementation and countable intersection. Observe that if a set $\mathcal{C}$ is in the σ-algebra over the class of all recursivly G_δ sets which are closed under finite variation, then $\mathcal{C}$ itself is closed under finite variation.

We need the following easy consequence of a result from Kautz [22, 23]:

Lemma 6.1 *If $\mathcal{C}$ is in the σ-algebra over the class of all recursivly G_δ sets which are closed under finite variation, then the following are equivalent:*

(1) $\mu(\mathcal{C}) > 0$.
(2) $\mu(\mathcal{C}) = 1$.
(3) $\mathrm{RAND} \cap \mathcal{C} \neq \emptyset$.
(4) $\mathrm{RAND} \subseteq \mathcal{C}$.

Proof. (Sketch) The result for sets which are recursively G_δ or recursively F_σ can be found in [22, 23]. An induction shows that if the result holds for any class, then it also holds for both its closure under complementation and its closure under countable intersection. ❑

Now we obtain immediatly the following improvement of a result from [10, 9], where additional assumptions on $\mathcal{K}_1, \mathcal{K}_2$ were made:

Theorem 6.2 *Let $\mathcal{K}_1, \mathcal{K}_2$ be relativizable classes which are closed under finite variations of the oracle. Then the following are equivalent:*

1. $\mu A(\mathcal{K}_1^A \subseteq \mathcal{K}_2^A) > 0$.
2. $\mu A(\mathcal{K}_1^A \subseteq \mathcal{K}_2^A) = 1$.
3. $\mathcal{K}_1^A \subseteq \mathcal{K}_2^A$ *for some random oracle A.*

4. $\mathcal{K}_1^A \subseteq \mathcal{K}_2^A$ *for all random oracles* A.

Proof. It was shown in [10], that if $\mathcal{K}$ is a relativized class which is invariant under finite variations of the oracle, then for any $i \in \mathbb{N}$ the set $\{\, A \mid L(M_i^A) \in \mathcal{K}^A \,\}$ is recursively G_δ and closed under finite variation. The theorem now is an application of Lemma 6.1. ❑

Thus, we have the following results:

Corollary 6.3 *1.* $\mathrm{PH}^A \subsetneq \oplus\mathrm{P}^A$ *for all random oracles A [35].*

2. $\mathrm{coNP}^A \not\subseteq \mathrm{AM}^A$ *for all random oracles* A.
3. $\mathrm{PH}^A \subsetneq \mathrm{C_{=}P}^A \subsetneq \mathrm{PP}^A \subsetneq \mathrm{PSPACE}^A$ *for all random oracles* A.
4. $\Theta_2^{\mathrm{p}\,A} = \mathcal{R}_{\mathrm{dtt}}^{\mathrm{p}}(\mathrm{US})^A$ *for all random oracles* A.

Proof. Using Theorem 6.2, all results follow immediately from Corollaries 5.4 to 5.7 from Section 5. ❑

7 Type 2 operators vs. polynomially bounded operators

In this section, we want to compare type 2 operators with the familiar operators ranging over polynomially length bounded strings [48, 37]. To this end we define type 2 existential and universal operators. Let $\mathcal{K}$ be a relativized class with k oracles with enumeration function f (a machine $M_{f(i)}$ with the k oracles $A_1, \ldots, A_k$ uses in fact the one oracle $\bigcup_{i=1}^{k}\{1^i0x | x \in A_i\}$).

- $L \in \exists^2\mathcal{K}$ iff there exists an $i \in \mathbb{N}$ such that for all x,
$$(x, A_1, \cdots, A_{k-1}) \in L \iff \exists A_k (M_{f(i)}^{A_1,\cdots,A_{k-1},A_k}(x) = 1)$$
- $L \in \forall^2\mathcal{K}$ iff there exists an $i \in \mathbb{N}$ such that for all x,
$$(x, A_1, \cdots, A_{k-1}) \in L \iff \forall A_k (M_{f(i)}^{A_1,\cdots,A_{k-1},A_k}(x) = 1)$$

Clearly, if $\mathcal{K}$ is a relativized class then so are $\exists^2\mathcal{K}$ and $\forall^2\mathcal{K}$.

Define $\Sigma_k^{\exp}$ to be the set of all languages A accepted by Σ_k machines (i.e. alternating Turing machines with $k-1$ alternations, starting in an existential state), which on inputs of length n run in time bounded by $2^{p(n)}$ for some polynomial p. Now the classes of the $\exists^2$-$\forall^2$-hierarchy (restricted to "ordinary" languages of words) can be characterized as follows [38, 31, 46]:

Theorem 7.1 *1.* $\exists^2\forall^2\exists^2\cdots Q_k^2\overline{Q}_k^2 \cdot \mathrm{P} = \exists^2\forall^2\exists^2\cdots Q_k^2\overline{Q}_k^{\mathrm{p}} \cdot \mathrm{P} = \exists^2\forall^2\exists^2\cdots Q_k^2 \cdot \mathrm{PSPACE} = \Sigma_k^{\exp}$

2. $\forall^2\exists^2\forall^2\cdots\overline{Q}_k^2 Q_k^2\cdot \mathrm{P} = \forall^2\exists^2\forall^2\cdots\overline{Q}_k^2 Q_k^{\mathrm{p}}\cdot \mathrm{P} = \forall^2\exists^2\forall^2\cdots\overline{Q}_k^2\cdot \mathrm{PSPACE} = \Pi_k^{\mathrm{exp}}$

where $Q_k = \exists$ and $\overline{Q}_k = \forall$ if k is odd, and $Q_k = \forall$ and $\overline{Q}_k = \exists$ if k is even. The operators $\exists^{\mathrm{p}}$ and $\forall^{\mathrm{p}}$ are the classical operators defined by the $\exists$ and $\forall$ quantifier, resp., ranging over polynomially length bounded words.

Let L be the class of logspace-decidable sets. Let NC^k be the class of sets decidable by uniform circuit families of polynomial size and $O(\log^k n)$ depth [17]. (Without going into details, we remark that we adopt the uniformity condition from [5].) Let BPNC^k denote the bounded error probabilistic analogue of NC^k (see [17]), i.e., BPNC^k circuits have regular input gates plus gates for probabilistic bits. The probability is then taken over all possible inputs to the latter gates, where we assume (as usual) uniform distribution. We remark that $\mathrm{BPNC}^k = \mathbf{BP}^{\mathrm{p}}\mathrm{NC}^k$.

To compare type 2 operators with the "usual" operators, we use *translational methods,* which have a long history in complexity theory, see e.g. [8]. In all these arguments, *padding* plays a crucial role—in the just mentioned paper, tally versions of languages were used. We here introduce the following form of padding: For a language A and some integer m, define

$$A_m =_{\mathrm{def}} \{\, x10^{2^{|x|^m}-|x|-1} \mid x \in A \,\}.$$

Then, the following lemma is easy to see:

Lemma 7.2 *1. $A \in \Sigma_k^{\mathrm{exp}}$ iff there exists some $m \in \mathbb{N}$ such that $A_m \in \Sigma_k^{\mathrm{p}}$.*

2. *$A \in \mathrm{PSPACE}$ iff there exists some $m \in \mathbb{N}$ such that $A_m \in \mathrm{L}$, iff there exists some $m \in \mathbb{N}$ such that $A_m \in \mathrm{NC}^1$,*
3. *$A \in \mathbf{BP}^2\mathrm{PSPACE}$ iff there exists some $m \in \mathbb{N}$ such that $A_m \in \mathbf{BP}^{\mathrm{p}}\mathrm{L}$, iff there exists some $m \in \mathbb{N}$ such that $A_m \in \mathrm{BPNC}^1$.*
4. *$A \in \mathbf{BPTIME}(2^{\mathrm{Pol}})$ iff there exists some $m \in \mathbb{N}$ such that $A_m \in \mathrm{BPP}$.*

Proof. By standard translational arguments. For the only non-trivial claims (2 and 3) we remark that PSPACE can be characterized by polynomial time alternating Turing machines [15], whereas NC^1 can be characterized by logarithmic time alternating Turing machines [5]. ❑

Theorem 7.3 $\mathbf{BP}^2\mathrm{PSPACE} \subseteq \mathbf{BPTIME}(2^{\mathrm{Pol}}) \subseteq \Sigma_2^{\mathrm{exp}} \cap \Pi_2^{\mathrm{exp}}$.

Proof. The result is obtained by applying the translational results from Lemma 7.2 to Sipser's and Lautemann's result that BPP is included in the polynomial time hierarchy [39, 24]. ❑

The up to now best known upper bound for ALMOST-PSPACE is ALMOST-$\mathrm{PSPACE} \subseteq$ EXPSPACE [25]. We obtain the following improvement:

Corollary 7.4 ALMOST-$\mathrm{PSPACE} \subseteq \Sigma_2^{\exp} \cap \Pi_2^{\exp}$.

In the theory of efficient algorithms, if no good parallel algorithm for a given problem is within reach, one tries to design efficient probabilistic parallel algorithms, i.e., to prove that the problem under consideration is in BPNC^k for some k. Therefore, it is of great importance to have tight upper bounds for those classes. Unfortunately, essentially only $\mathrm{BPNC}^k \subseteq \mathrm{BPP}$ is known. It turns out that this problem is related to that of giving upper bounds for ALMOST-PSPACE: Any upper bound for BPNC^1 better than BPP will give us an upper bound for ALMOST-PSPACE better than the one given in Corollary 7.4; for example:

Corollary 7.5 *If* $\mathrm{BPNC}^1 \subseteq \mathrm{P}$, *then* ALMOST-$\mathrm{PSPACE} \subseteq \mathrm{EXPTIME}$.

8 A characterization of the class $\mathrm{BP}^2\mathrm{DSPACE}(s)$

In the Section 4, we saw that ALMOST-PSPACE $=$ BP^2PSPACE, and we gave new upper time bounds for that class in Section 7. However, the question of whether BP^2$\mathrm{PSPACE} = \mathrm{PSPACE}$ remains unresolved. In this section, we give a machine characterization of BP^2PSPACE which makes this equality seem unlikely to us.

A *checking stack* [21] is a stack which can be used only in the following way in two phases. In the first phase, the *writing phase,* the head of the checking stack can only write new symbols on top of the stack (in a one-way manner); it cannot erase symbols or visit some inner part of the stack. In the second phase, the *checking phase,* the head of the checking stack can only read the contents of the stack (in a two-way manner), but without changing the stack, that is, without erasing symbols or pushing new symbols on top.

A CS-DTM (CS-NTM, CS-PTM, CS-BPTM) is a deterministic (nondeterministic, probabilistic, bounded-error probabilistic) Turing machine with a two-way input tape, a constant number of working tapes, and a checking stack. For $s\colon \mathbb{N} \to \mathbb{N}$, we define $L \in$ CS-$\mathcal{X}$SPACE(s) if there exists a CS-$\mathcal{X}$TM (for $\mathcal{X}$ either D, N, P, or BP) such that every computation path of M on input x halts and is space-bounded by $s(|x|)$, where the workspace used in the checking stack is not taken into account. For these definitions as well as general background and results, see [47, pp. 252ff].

Now we see that ALMOST-PSPACE is exactly the class of all languages accepted by probabilistic checking stack automata working in polynomial space; more generally:

Theorem 8.1 *For every fully space-constructible function* $s\colon \mathbb{N} \to \mathbb{N}$ *such that* $s(n) \geq \log n$ *for all* n,

$$\mathrm{BP^2DSPACE}(s) = \mathrm{CS\text{-}BPSPACE}(s).$$

Proof. "$\subseteq$": Let $L \in \mathrm{BP^2DSPACE}(s)$; let M be a deterministic oracle Turing machine such that for every input x and every oracle A the machine halts using workspace no more than $s(|x|)$, and

$$x \in \mathrm{L} \implies \mu A(M^A(x) = 1) \geq \tfrac{2}{3}$$
$$x \notin \mathrm{L} \implies \mu A(M^A(x) = 1) \leq \tfrac{1}{3}$$

Notice that because every oracle query of M while working on input x is length bounded by $s(|x|)$, only an initial segment of length at most $2^{s(|x|)+1}$ of any oracle A can influence the value of $M^A(x)$.

Now construct a CS-BPTM M' (without oracle) simulating M as follows: On input x, the M' first computes (in binary) $2^{s(|x|)+1}$ and then creates in a nondeterministic manner for every string z of length $2^{s(|x|)+1}$ exactly one computation path while writing z into the checking stack. Then on the path corresponding to some string z M' simulates the computation of M on input x with oracle $A \in z \cdot \{0,1\}^\omega$, where an oracle query of M to A is replaced by looking up the corresponding bit of z in the checking stack.

For all z such that $|z| = 2^{s(|x|)+1}$, M' in this simulation accepts a string x on path z if and only if M with any oracle $A \in z \cdot \{0,1\}^\omega$ accepts x. Thus we obtain that $\mu A(M^A(x) = 1)$ is exactly the probability with which M' accepts x, which proves the inclusion from left to right.

"$\supseteq$": Let M be a CS-BPTM accepting a language L such that every computation path of M on any input x halts and is space-bounded by $s(|x|)$ (recall that the workspace used in the checking stack is not taken into account). In the writing phase of any computation, no configuration (that is a tuple consisting of worktape contents, top symbol of the checking stack, worktape and input tape head positions, and internal state of the machine) can appear twice, since then the computation were not halting. Hence, on every computation path of M on x, the writing phase is time-bounded by $c^{s(|x|)}$ for some $c > 0$. Thus, the contents of the checking stack is length bounded by the same function.

Now a deterministic s-space-bounded oracle machine M' can simulate M as follows: M' works as M but

(a) if M branches nondeterministically during its computation for the i-th time, then M' queries the i-th bit of the oracle instead. Hence the oracle of M' describes the nondeterministic path of M.

(b) M' does not store the contents of the checking stack (since it is too long to be written down) but the position of the checking stack head in binary. Note that this takes no more than $c \cdot s(|x|)$ bits. If during the simulation of the checking phase, M' needs the j-th symbold of the stack, it starts a re-simulation of the writing-phase up to the moment where the j-th symbol is printed. Using the help of the oracle, it is ensured that the correct computation path of M is taken in the re-simulation.

Now we have that M accepts x on some path $z \in \{0,1\}^\omega$ if and only if M' accepts x with oracle z. Thus, the probability that M accepts an input x is exactly the measure of the set of all oracles A such that M' accepts x using oracle A, which finishes the proof. ❑

Corollary 8.2 *1.* BP$^{\mathrm{exp}}\mathrm{PSPACE}$ = CS-BPSPACE(Pol).

2. BP$^{\mathrm{p}}\mathrm{L}$ = CS-BPSPACE$(\log)$.

Proof. Follows from Theorem 8.1, recalling that BP$^2\mathrm{PSPACE}$ = BP$^{\mathrm{exp}}\mathrm{PSPACE}$ and BP$^2\mathrm{L}$ = BP$^{\mathrm{p}}\mathrm{L}$ by Proposition 3.2. ❑

So far it is not known whether CS-BPSPACE(s) coincides with one of the well-studied complexity classes. However, in this context the following results should be mentioned which can be found in [21]: For arbitrary $s \geq log$, the equations CS-DSPACE(s) = DSPACE(s) and CS-NSPACE(s) = NSPACE$(2^{O(s)})$ hold. That is: Checking stack automata working nondeterministically are more powerful than those working deterministically.

Our Theorem 8.1 now gives the following extension: For polynomial space plus checking stack, the nondeterministic computation mode is strictly more powerful than the bounded-error probabilistic mode (unless Σ_2^{exp} = EXPSPACE).

Acknowledgments. Some of the ideas presented here evolved during a seminar on Randomness and Computation, held in the Spring 95 Quarter at the University of California at Santa Barbara. Thanks are due to Zhe Dang, Todd Ebert, and Sarah Hough for helpful discussions. We thank Noam Nisan for pointing out some of the subtleties of [28, 29, 30]. We benefitted very much from detailed comments by Wolfang Merkle on an earlier version of our paper. We also acknowledge helpful comments from Eric Allender, Jack Lutz, Elvira Mayordomo, and Ken Regan.

References

[1] K. AMBOS-SPIES, Randomness, relativizations, and polynomial reducibilities, *Proceedings of the 1st Structure in Complexity Theory Conference* (1986), Springer Lecture Notes in Computer Science Vol. 223, pp. 200–207.

[2] J. ASPNES, R. BEIGEL, M. FURST, S. RUDICH, The expressive power of voting polynomials, *Proceedings of the 23rd Symposium on Theory of Computing* (1991), pp. 402–409.

[3] L. BABAI, Trading group theory for randomness; *Proceedings of the 17th Symposium on Foundations of Computer Science* (1975), pp. 421–429.

[4] J. L. BALCÁZAR, J. DÍ AZ, J. GABARRÓ, *Structural Complexity I* (Springer Verlag, Berlin – Heidelberg – New York, ^{2}1995).

[5] D. BARRINGTON, N. IMMERMAN, H. STRAUBING, On uniformity within NC^1; *Journal of Computer and System Sciences* **41** (1990), pp. 274–306.

[6] C. BENNETT, J. GILL, Relative to a random oracle $\mathrm{P}^A \neq \mathrm{NP}^A \neq \mathrm{coNP}^A$ with probability 1; *SIAM J. Comput.* **10** (1981), pp. 96–113.

[7] A. BLASS, Y. GUREVICH, On the unique satisfiability problem; *Information and Control* **55** (1982), pp. 80–88.

[8] R. V. BOOK, Tally languages and complexity classes; *Information and Control* **26** (1974), pp. 186–193.

[9] R. V. BOOK, On languages reducible to algorithmically random languages; *SIAM J. Comput.* **23** (1994), pp. 1275–1282.

[10] R. V. BOOK, J. H. LUTZ, K. W. WAGNER, An observation on probability versus randomness with applications to complexity classes; *Mathematical Systems Theory* **27** (1994), pp. 201–209.

[11] R. V. BOOK, S. TANG, Polynomial-time reducibilities and almost all oracle sets; *Theoretical Computer Science* **81** (1991), pp. 201–209.

[12] D. P. BOVET, P. CRESCENZI, R. SILVESTRI, A uniform approach to define complexity classes; *Theoretical Computer Science* **104** (1992), pp. 263–283.

[13] J. Y. CAI, With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy; *Journal of Computer and System Sciences* **38** (1989), pp. 68–85.

[14] J. Y. CAI, Probability one separation of the boolean hierarchy; *Proceedings of the 4th Symposium on Theoretical Aspects of Computer Science* (1987), Springer Lecture Notes in Computer Science Vol. 38, pp. 148–158.

[15] A. CHANDRA, D. KOZEN, L. STOCKMAYER, Alternation; *Journal of the ACM* **28** (1981), pp. 114–133.

[16] R. CHANG, J. KADIN, P. ROHATGI, Connections between the complexity of unique

satisfiability and the threshold behaviour of randomized reductions; *Proceedings of the 6th Structure in Complexity Theory Conference* (1991), pp. 255–269.

[17] S. A. COOK, A taxonomy of problems with fast parallel algorithms; *Information and Control* **64** (1985), pp. 2–22.

[18] J. GILL, Computational complexity of probabilistic complexity classes; *SIAM Journal on Computing* **6** (1977), pp. 675–695.

[19] F. GREEN, On the power of deterministic reductions to $\mathrm{C_{=}P}$; *Mathematical Systems Theory* **26** (1993), pp. 215–234.

[20] U. HERTRAMPF, C. LAUTEMANN, T. SCHWENTICK, H. VOLLMER, K.W. WAGNER, On the power of polynomial time bit-reductions; *Proceedings of the 8th Structure in Complexity Theory Conference* (1993), pp. 200–207.

[21] O. H. IBARRA, Characterizations of some tape and time complexity classes of Turing machines in terms of multihead and auxiliary stack automata; *Journal of Computer and System Sciences* **5** (1971), pp. 88–117.

[22] S. KAUTZ, Degrees of random sets; Ph. D. dissertation, Cornell University, 1991.

[23] S. KAUTZ, An improved zero-one law for algorithmically random sequences; draft (1994).

[24] C. LAUTEMANN, BPP and the polynomial hierarchy; *Information Processing Letters* **117** (1983), pp. 215–217.

[25] J. LUTZ, personal communication, 1995.

[26] W. MERKLE, Y. WANG, Separations by random oracles and "Almost" classes for generalized reducibilities; 1996. An extended abstract appeared in the *Proceedings of the 20th International Symposium on Mathematical Foundations of Computer Science* (1995), Springer Lecture Notes in Computer Science Vol. 969, pp. 179–190.

[27] W. MERKLE, personal communication, 1996.

[28] N. NISAN, A. WIGDERSON, Hardness vs. Randomness; *Journal of Computer and System Sciences* **49** (1994), pp. 149–167.

[29] N. NISAN, Pseudorandom generators for space-bounded computation; *Journal of Combinatorica* **12** (1992), pp. 449–461.

[30] N. NISAN, On read-once vs. multiple access to randomness in logspace; *Theoretical Computer Sience* **107** (1993), pp. 135–144.

[31] P. ORPONEN, Complexity classes of alternating machines with oracles; *Proceedings of the 10th International Colloquium on Automata, Languages and Programming* (1983), Springer Lecture Notes in Computer Science Vol. 154, pp. 573–584.

[32] C. H. PAPADIMITRIOU, *Computational Complexity* (Addison-Wesley, Reading, Mass., 1994).

[33] C. H. PAPADIMITRIOU, M. YANNAKAKIS, The complexity of facets (and some facets of complexity); *Journal of Computer and System Sciences* **28** (1984), pp. 244–259.

[34] C. H. PAPADIMITRIOU, S. K. ZACHOS, Two remarks on the power of counting; *Proceedings of the 6th GI-Conference on Theoretical Computer Science* (1983), Springer Lecture Notes in Computer Science Vol. 145, pp. 269–275.

[35] K. W. REGAN, J. S. ROYER, On closure properties of bounded two-sided error complexity classes; *Mathematical Systems Theory* **28** (1995), pp. 229–243.

[36] H. ROGERS, *Theory of Recursive Functions and Effective Computability* (McGraw-Hill, New York, NY, 1967).

[37] U. SCHÖNING, Probabilistic complexity classes and lowness; *Journal of Computer and System Sciences* **39** (1989), pp. 84–100.

[38] J. SIMON, On Some Central Problems in Computational Complexity; Dissertation, Cornell University (1975).

[39] M. SIPSER, A complexity theoretic approach to randomness; *Proceedings of the 15th Symposium on Theory of Computing* (1983), pp. 330–335.

[40] S. TODA, PP is as hard as the polynomial time hierarchy; *SIAM Journal on Computing* **20** (1991), pp. 865–877.

[41] T. TODA, M. OGIWARA, Counting classes are at least as hard as the polynomial time hierarchy; *SIAM Journal on Computing* **21** (1992), pp. 315–328.

[42] L. G. VALIANT, V. V. VAZIRANI, NP is as easy as detecting unique solutions, *Theoretical Computer Science* **47** (1986), pp. 85–93.

[43] K. W. WAGNER, The complexity of combinatorial problems with succinct input representation; *Acta Informatica* **23** (1986), pp. 325—356.

[44] K. W. WAGNER, Some observations on the connection between counting and recursion; *Theoretical Computer Science* **47** (1986), pp. 131–147.

[45] K. W. WAGNER, Bounded query classes; *SIAM J. Comput.* **19** (1990), pp. 833–846.

[46] K. W. WAGNER, High-order operators in complexity theory; manuscript.

[47] K. W. WAGNER, G. WECHSUNG, *Computational Complexity* (Deutscher Verlag der Wissenschaften, Berlin, 1986).

[48] C. WRATHALL, Complete sets and the polynomial-time hierarchy; *Theoretical Computer Science* **3** (1977), pp. 23–33.