Small Bounded-Error Computations and Completeness

Elmar Böhler Christian Glaßer Daniel Meister

Theoretische Informatik
Bayerische Julius-Maximilians-Universität Würzburg
97074 Würzburg, Germany

{boehler, glasser, meister}@informatik.uni-wuerzburg.de

13th August 2003

Abstract

SBP is a probabilistic promise class located between MA and $\mathrm{AM} \cap \mathrm{BPP}_{\mathrm{path}}$. The first part of the paper studies the question of whether SBP has many-one complete sets. We relate this question to the existence of uniform enumerations. We construct an oracle relative to which SBP and AM do not have many-one complete sets. In the second part we introduce the operator SB$\cdot$. We prove that, for any class $\mathcal{C}$ with certain properties, $\mathrm{BP}\cdot\exists\cdot\mathcal{C}$ contains every class defined by applying an operator sequence over $\{\mathrm{U}\cdot, \exists\cdot, \mathrm{BP}\cdot, \mathrm{SB}\cdot\}$ to $\mathcal{C}$.

1 Introduction

Probabilistic Computations. In the 1970's, Rabin [Rab76] and Solovay and Strassen [SS77] developed fast probabilistic algorithms for problems like primality test. These algorithms find the correct answer (e.g., "the input is prime" or "the input is not prime") with high probability. Even though at that time no deterministic polynomial-time algorithm for primality test was known, probabilistic algorithms provided a feasable way to perform primality tests in practice. This was new: problems that are not known to be solvable in deterministic polynomial time could be handled in practice.

Probabilistic Turing machines introduced by Gill [Gil72, Gil77] formalize probabilistic algorithms. A Turing machine is called probabilistic if each step depends on the outcome of an unbiased coin toss. Such machines accept an input if and only if the probability of acceptance is greater than $\frac{1}{2}$. The restriction to polynomial-time computations leads to PP, the class of languages recognizable by *polynomial-time probabilistic* Turing machines. However, PP is too powerful: It even covers computations where the probabilities of acceptance and rejection are very close. The probabilistic algorithms for primality test do not need such a fine distinction. A suitable restriction of PP that covers these probabilistic algorithms is called BPP [Gil72, Gil77]. This is the class of languages recognizable by *bounded-error probabilistic polynomial-time* Turing machines. For such machines one additionally demands a probability gap. This means that the acceptance probability must never belong to some interval around $\frac{1}{2}$ (e.g., $[\frac{1}{4}, \frac{3}{4}]$). Recently, Agrawal, Kayal, and Saxena showed that primality can be tested deterministically in polynomial time [AKS02], but BPP is still considered to be an important complexity class.

PP is most likely not contained in the polynomial-time hierarchy [Tod91]. In contrast, BPP belongs to Σ_2^{P} [Lau83, Sip83] and by its closure under complement also to Π_2^{P}. Moreover, the class BPP allows *probability amplification* [Sip83]: the size of the probability gap (i.e., size of the interval) can be increased to any fixed value arbitrarily close to 1. Thus, a probabilistic computation in this sense almost always results in the correct answer.

The Class SBP. SBP is a probabilistic complexity class that is located between BPP and PP [BGM03]. This class generalizes BPP in the following way. The probability limit of BPP is $\frac{1}{2}$. This means that an input is accepted if and only if the acceptance probability is at least $\frac{1}{2}$. Additionally, BPP computations respect a probability gap. In the definition of SBP we still demand a probability gap, but now we allow probability limits that are exponentially small. So, a small acceptance probability already suffices to accept the input.

Babai [Bab85] introduced the Arthur-Merlin classes MA and AM. Languages in these classes can be decided by a game between the two players Arthur and Merlin. MA is a subset of AM. SBP is located exactly between MA and AM [BGM03]. With respect to oracles, these three classes are different. Han, Hemaspaandra, and Thierauf [HHT97] define $\mathrm{BPP}_{\mathrm{path}}$ to be the class of languages accepted by polynomial-time threshold machines with ratio gap at $\frac{1}{2}$. They show $\mathrm{BPP} \subseteq \mathrm{BPP}_{\mathrm{path}} \subseteq \mathrm{PP}$. SBP is located between BPP and $\mathrm{BPP}_{\mathrm{path}}$ [BGM03]. SBP can be defined as the class of sets A for which there exist $f \in \mathrm{FP}$, $g \in \#\mathrm{P}$, and $\varepsilon > 0$ such that for all x,

$$\begin{aligned} x \in A \quad &\longrightarrow \quad g(x) > (1+\varepsilon) \cdot f(x) \\ x \notin A \quad &\longrightarrow \quad g(x) < (1-\varepsilon) \cdot f(x). \end{aligned}$$

If we allow f to be a $\#\mathrm{P}$-function, we capture exactly the languages in $\mathrm{BPP}_{\mathrm{path}}$ [BGM03].

Promise Classes and Completeness. Classes like BPP and SBP share an important property stressing their difference to classes like P and NP. BPP and SBP are promise classes. Usual (nonpromise) complexity classes are defined via machines. However, for BPP and SBP we additionally assume that all computations respect the probability gap. So, we make assumptions about the computation process.

Machines with certain resources can be enumerated recursively. This enumeration gives a way to construct complete problems. In contrast, because of the additional assumptions about the computation process, we do not know recursive enumerations for most of the known promise classes. As a consequence, we cannot easily construct complete sets. So, for promise classes it is always a challenging question whether complete problems exist. $\mathrm{NP} \cap \mathrm{SPARSE}$ is one of the rare examples of a promise class that has complete sets although only with respect to Turing reducibility [HY84]. However, for most of the promise classes we do not expect complete problems to exist.

Paper Outline. In this paper we study the question of whether SBP has many-one complete sets. This relates to the question of whether SBP is recursively enumerable. We show that SBP allows an enumeration in a weak sense. This may be considered the best possible case since we can show that SBP is enumerable in a stronger sense if and only if SBP has many-one complete sets, what we do not expect. For the weak enumeration we utilize a method by Buhrman, Fenner, Fortnow, and van Melkebeek [BFFvM00]. For the result regarding the stronger enumeration we make use of a method introduced by Hartmanis and Hemachandra [HH88].

The main result in this first part of the paper gives evidence against the existence of many-one complete sets for SBP. BPP is contained in several subclasses of AM. In this sense BPP is a restriction of AM. We construct an oracle relative to which AM does not have a set that is many-one hard for BPP. So, any single set in AM does not seem to be powerful enough to solve arbitrary problems in BPP. As a consequence, relative to this oracle, SBP does not have many-one complete sets and is therefore not uniformly enumerable (in the stronger sense).

In the second part of this paper we investigate a new operator. Coming from BPP, Schöning defined the operator $\mathrm{BP}\cdot$ [Sch89]. Similarly, we start from SBP, capture the main ingredients of its definition, and define the operator $\mathrm{SB}\cdot$. We show closure properties of classes defined with this and other operators and study their inclusion structure. Our main result in this part shows that $\mathrm{BP}\cdot\exists\cdot\mathcal{C}$ contains all complexity classes defined by arbitrary application of the operators $\mathrm{U}\cdot$, $\exists\cdot$, $\mathrm{BP}\cdot$, and $\mathrm{SB}\cdot$ in any order and number to a complexity class $\mathcal{C}$, if $\mathcal{C}$ fulfills some basic properties.

2 Preliminaries

For basics we refer to textbooks such as [WW86] or [Pap94]. We fix the alphabet $\Sigma = \{0, 1\}$; each input is a word over Σ, and each set (or *language*) is a subset of Σ^*. The characteristic function of a set $A \subseteq \Sigma^*$ is denoted by c_A. For two words $x, y \in \Sigma^*$, xy and $x \cdot y$ denote the concatenation of x and y. The injective function $\langle \cdot, \cdot \rangle$ maps two words to one word in the following way. For two words $x, y \in \Sigma^*$, $x = x_1 \dots x_k$, $y = y_1 \dots y_\ell$, $k, \ell \geq 0$, let $\langle x, y \rangle \stackrel{df}{=} 0x_1 \cdot \ldots \cdot 0x_k 1y_1 \cdot \ldots \cdot 1y_\ell$. Thus, the length of $\langle x, y \rangle$ is twice that of xy. Similarly, we extend this pairing function to higher arties such that $|\langle x_1, \dots, x_k \rangle| = k \cdot |x_1 \cdots x_k|$. For every $n \in \Sigma^*$, let $\mathrm{id}(n) \stackrel{df}{=} n$. We require every complexity class $\mathcal{C}$ to contain one non-empty set that is a proper subset of Σ^*. We call such classes *non-trivial*. With a non-deterministic or probabilistic computation, we associate a computation tree that represents all possible computation paths. Let B be some computable set, let $x \in \Sigma^*$ and q be some polynomial. We define

$$\mathrm{count}_B^q(x) = \left|\left\{ y \,:\, |y| = q(|x|) \;\wedge\; \langle x, y \rangle \in B \right\}\right|.$$

For all polynomials p that we use in this report, we assume $p(\mathbb{N}) \subseteq \mathbb{N}$.

2.1 Reducibilities

All reducibilities in this paper are polynomial-time computable. A is polynomial-time many-one reducible to B ($A \leq_m^{\mathrm{P}} B$) if there is a function $f \in \mathrm{FP}$ such that for all $x \in \Sigma^*$,

$$x \in A \;\longleftrightarrow\; f(x) \in B.$$

Ladner, Lynch, and Selman [LLS75] introduced several other polynomial-time bounded reducibilities. For any two reducibilities $\leq_a$ and $\leq_b$, they defined $\leq_a$ to be *stronger* than $\leq_b$ if for each two computable sets A and B, $A \leq_a B$ implies $A \leq_b B$. A complexity class $\mathcal{C}$ is closed under reducibility $\leq_a$ if for every set A with $A \leq_a B$ where $B \in \mathcal{C}$, A is contained in $\mathcal{C}$. If we denote the set of all sets that are $\leq_a$-reducible to some set in $\mathcal{C}$ by $\mathcal{R}_a(\mathcal{C})$, then $\mathcal{C}$ is closed under $\leq_a$ if and only if $\mathcal{R}_a(\mathcal{C}) = \mathcal{C}$.

Definition 2.1 *Let A and B be two sets. A is* conjunctive reducible *to B, $A \leq_c^{\mathrm{P}} B$, if and only if there is some function $f \in \mathrm{FP}$ such that for all $x \in \Sigma^*$ there is a positive integer k such that*

$$x \in A \longleftrightarrow f(x) = \langle x_1, \ldots, x_k \rangle \text{ and } c_B(x_1) \wedge \ldots \wedge c_B(x_k) = 1.$$

A is disjunctive reducible *to B, $A \leq_d^{\mathrm{P}} B$, if and only if there is some function $f \in \mathrm{FP}$ such that for all $x \in \Sigma^*$ there is a positive integer k such that*

$$x \in A \longleftrightarrow f(x) = \langle x_1, \ldots, x_k \rangle \text{ and } c_B(x_1) \vee \ldots \vee c_B(x_k) = 1.$$

A is majority reducible *to B, $A \leq_{maj}^{\mathrm{P}} B$, if and only if there is some function $f \in \mathrm{FP}$ such that for all $x \in \Sigma^*$ there is a natural k such that*

$$x \in A \longleftrightarrow f(x) = \langle x_1, \ldots, x_k \rangle \text{ and } c_B(x_1) + \ldots + c_B(x_k) > \frac{k}{2}.$$

For example, P is closed under all three of the reducibilities defined above. Without loss of generality, we can always assume that the number of questions computed by f is given by some polynomial s, i.e., $f(x) = \langle x_1, \ldots, x_k \rangle$ and $k = s(|x|)$ for every $x \in \Sigma^*$. Moreover, we may assume that the questions are of equal length, even $|x_1| = \ldots = |x_k| = r(|x|)$ for some polynomial r. Finally, if A is majority reducible to B via some reducing function $f \in \mathrm{FP}$, there is always $g \in \mathrm{FP}$ that majority reduces A to B with an odd number of questions: We can fix a word $w \notin B$ and define $g(x) \stackrel{df}{=} \langle x_1, \ldots, x_k, w \rangle$ where $f(x) = \langle x_1, \ldots, x_k \rangle$ for $x \in \Sigma^*$ and k even. Therefore, we will henceforth assume that all functions used in majority reductions calculate an odd number of values. We observe that $\leq_m^{\mathrm{P}}$ is stronger than $\leq_c^{\mathrm{P}}$, $\leq_d^{\mathrm{P}}$ [LLS75], and $\leq_{maj}^{\mathrm{P}}$, and $\leq_c^{\mathrm{P}}$ itself is stronger than $\leq_{maj}^{\mathrm{P}}$ which can be seen by a construction that duplicates question w often enough.

We can define bounded variants of the above defined reducibilities: We say that A is k-*conjunctive* reducible to B, denoted by $A \leq_{kc}^{\mathrm{P}} B$, if the number of questions computed by the reducing function f is bounded by k. As we have seen in previous discussions, this is equivalent to the notion where we require f to compute exactly k questions. We will show that, for two natural numbers k_1 and k_2 greater than 1, a complexity class $\mathcal{C}$ is closed under $\leq_{k_1 c}^{\mathrm{P}}$ if and only if $\mathcal{C}$ is closed under $\leq_{k_2 c}^{\mathrm{P}}$. We say that $\mathcal{C}$ is closed under *bounded conjunctive* reducibility, $\leq_{bc}^{\mathrm{P}}$, if for every reducing function f there is k such that f is a k-conjunctive reducting function. To prove that $\mathcal{C}$ is closed under $\leq_{bc}^{\mathrm{P}}$, it is sufficient to show that $\mathcal{C}$ is closed under $\leq_{2c}^{\mathrm{P}}$, which implies the claim.

Lemma 2.2 *A complexity class $\mathcal{C}$ is closed under $\leq_{bc}^{\mathrm{P}}$ if and only if $\mathcal{C}$ is closed under $\leq_{2c}^{\mathrm{P}}$.*

Proof: Let A be some set and $B \in \mathcal{C}$, let $A \leq_{bc}^{\mathrm{P}} B$ via function $f \in \mathrm{FP}$. There is a natural k such that $A \leq_{kc}^{\mathrm{P}} B$ via f. We assume that k is some power of 2 and $k > 2$. We show that $A \leq_{k'c}^{\mathrm{P}} B'$ for $k' = \frac{k}{2}$ and some set $B' \in \mathcal{C}$. Define B' as

$$B' \stackrel{df}{=} \{x_1 x_2 : x_1 \in B \wedge x_2 \in B \wedge |x_1| = |x_2|\}.$$

B' 2-conjunctive reduces to some set in $\mathcal{C}$, hence $B' \in \mathcal{C}$. Let $f(x) = \langle x_1, \ldots, x_k \rangle$, then

$$f'(x) = \langle x_1 x_2, \ldots, x_{k-1} x_k \rangle$$

for every $x \in \Sigma^*$. Therefore, A is bounded conjunctive reducible to B' via f'. Repeated application of this construction shows that A is 2-conjunctive reducible to some set in $\mathcal{C}$. Since $\mathcal{C}$ is closed under 2-conjunctive reducibility, $A \in \mathcal{C}$, and $\mathcal{C}$ is closed under $\leq_{bc}^{\mathrm{P}}$. □

2.2 Operator Classes and the Arthur-Merlin Hierarchy

We repeat some results about operators and start with the operator $\exists\cdot$. In connection with the operator $\forall\cdot$, both operators applied alternately on P yield a characterization of the classes $\Sigma_{\mathrm{k}}^{\mathrm{P}}$ and $\Pi_{\mathrm{k}}^{\mathrm{P}}$ of the polynomial-time hierarchy [Sto77, Wra77].

Definition 2.3 *Let $\mathcal{C}$ be a complexity class and let A be some set. $A \in \exists\cdot\mathcal{C}$ if and only if there are $B \in \mathcal{C}$ and a polynomial q such that for every $x \in \Sigma^*$,*

$$x \in A \longleftrightarrow \operatorname{count}_B^q(x) \geq 1.$$

Observe that $\mathrm{NP} = \exists\cdot\mathrm{P}$. Furthermore, $\exists\cdot\exists\cdot\mathrm{P} = \exists\cdot\mathrm{P} = \mathrm{NP}$, which can be generalized as we will see later. A slight modification of the definition of $\exists\cdot$ leads to the related operator $\mathrm{U}\cdot$.

Definition 2.4 *Let $\mathcal{C}$ be a complexity class. $A \in \mathrm{U}\cdot\mathcal{C}$ if and only if there are $B \in \mathcal{C}$ and a polynomial q such that for all $x \in \Sigma^*$,*

$$x \in A \longleftrightarrow \operatorname{count}_B^q(x) = 1 \quad \text{and} \quad \operatorname{count}_B^q(x) \leq 1.$$

Clearly, $\mathrm{U}\cdot\mathrm{P} = \mathrm{UP}$.

Lemma 2.5 *If $\mathcal{C}$ is non-trivial and closed under $\leq_m^{\mathrm{P}}$, then $\mathcal{C} \subseteq \mathrm{U}\cdot\mathcal{C} \subseteq \exists\cdot\mathcal{C}$.*

Proof: Let $A \in \mathcal{C}$. We define $B \stackrel{df}{=} \{\langle x, x\rangle : x \in A\}$. If $A \subset \Sigma^*$, then $B \leq_m^{\mathrm{P}}$-reduces to A. If $A = \Sigma^*$, then $B \leq_m^{\mathrm{P}}$-reduces to any non-trivial set in $\mathcal{C}$. In both cases, it holds that $B \in \mathcal{C}$. For every $x \in \Sigma^*$, we obtain $\operatorname{count}_B^{\mathrm{id}}(x) \leq 1$. Therefore, $A \in \mathrm{U}\cdot\mathcal{C}$. The second inclusion is by definition. □

This proof is the only one where we explicitly distinguish the cases $A \neq \Sigma^*$ and $A = \Sigma^*$. It illustrates the need of a class to be non-trivial. For trivial classes, we cannot conclude $B \in \mathcal{C}$ for every $A \in \mathcal{C}$. It follows that if $\mathcal{C}$ is non-trivial and is closed under $\leq_m^{\mathrm{P}}$, then $\mathrm{U}\cdot\mathcal{C}$ and $\exists\cdot\mathcal{C}$ are also non-trivial, which is true for all opertators in this paper.

In [HH88], the authors give evidence that not all sets in NP belong to UP. In fact, it is very unlikely that any NP-complete set is in UP. We cannot decide whether a non-deterministic polynomial-time Turing machine behaves in the sense of UP. We can only trust the promise that a given machine behaves in the right way. Therefore, classes like UP are called *promise classes*. $\mathrm{U}\cdot\mathcal{C}$ is a promise class, and the promise is the limited number of accepting computation paths.

Another kind of complexity classes are probabilistic classes. The main idea is to equip each probabilistic computation with two probability values. If an input should be accepted, the probability that it is actually accepted by the computation is bounded below by one of the values. If it is to be rejected, a small number of computation paths can err, which means that the probability to find a path with the wrong result is bounded above by the second probability value. For further discussions on this concept, we refer to [Gil77]. Gill introduced several probabilistic complexity classes such as PP or BPP. Schöning [Sch89] derived the following operator from BPP.

Definition 2.6 *Let $\mathcal{C}$ be a complexity class and let A be some set. $A \in \mathrm{BP}{\cdot}\mathcal{C}$ if and only if there are $B \in \mathcal{C}$, a polynomial q, and $\varepsilon \in \left(0, \frac{1}{2}\right)$ such that for every $x \in \Sigma^*$,*

$$
\begin{aligned}
x \in A \quad &\longrightarrow \quad \mathrm{count}_B^q(x) > \left(\frac{1}{2} + \varepsilon\right) \cdot 2^{q(|x|)} \quad \textit{and} \\
x \notin A \quad &\longrightarrow \quad \mathrm{count}_B^q(x) < \left(\frac{1}{2} - \varepsilon\right) \cdot 2^{q(|x|)}.
\end{aligned}
$$

For convenience, we gave a definition of $\mathrm{BP}{\cdot}$ in terms of the number of accepting paths. It holds that $\mathrm{BP}{\cdot}\mathrm{P} = \mathrm{BPP}$.

Lemma 2.7 ([Sch89]) *If $\mathcal{C}$ is closed under $\leq_m^{\mathrm{P}}$, then $\mathcal{C} \subseteq \mathrm{BP}{\cdot}\mathcal{C}$.*

Proof: Let $A \in \mathcal{C}$ and define $B \stackrel{\mathrm{df}}{=} \{\langle x, z\rangle : x \in A \wedge z \in \Sigma^*\}$. It is easy to see that $B \leq_m^{\mathrm{P}}$-reduces to A, hence $B \in \mathcal{C}$. Now, if $x \notin A$, then $\mathrm{count}_B^{\mathrm{id}}(x) = 0$. If $x \in A$, then $\mathrm{count}_B^{\mathrm{id}}(x) = 2^{|x|}$. Therefore, $A \in \mathrm{BP}{\cdot}\mathcal{C}$ via B, id, and any $\varepsilon \in \left(0, \frac{1}{2}\right)$. □

All three operators, $\exists{\cdot}$, $\mathrm{U}{\cdot}$, and $\mathrm{BP}{\cdot}$, are monotonic with respect to inclusion. This means that for an operator O and complexity classes $\mathcal{C}$ and $\mathcal{D}$, $\mathcal{C} \subseteq \mathcal{D}$ implies $O\mathcal{C} \subseteq O\mathcal{D}$. If $\mathcal{C}$ is a complexity class closed under $\leq_m^{\mathrm{P}}$, then $\mathrm{U}{\cdot}\mathcal{C}$, $\exists{\cdot}\mathcal{C}$, and $\mathrm{BP}{\cdot}\mathcal{C}$ are closed under $\leq_m^{\mathrm{P}}$. We draw the following observation from Lemmata 2.5 and 2.7 and the monotonicity of the operators.

Lemma 2.8 *If $\mathcal{C}$ is closed under $\leq_m^{\mathrm{P}}$, then $\exists{\cdot}\mathcal{C} \subseteq \exists{\cdot}\mathrm{BP}{\cdot}\mathcal{C}$ and $\mathrm{BP}{\cdot}\mathcal{C} \subseteq \exists{\cdot}\mathrm{BP}{\cdot}\mathcal{C}$.*

Babai [Bab85] introduced Arthur-Merlin games and corresponding complexity classes MA and AM.

Definition 2.9 *Let A be some set. $A \in \mathrm{MA}$ if and only if there are $B \in \mathrm{P}$, polynomials q_1 and q_2, and $\varepsilon \in \left(0, \frac{1}{2}\right)$ such that for every $x \in \Sigma^*$,*

$$
\begin{aligned}
x \in A \quad &\longrightarrow \quad \bigvee_{y \in \Sigma^*} \left(|y| = q_1(|x|) \wedge \mathrm{count}_B^{q_2}(\langle x, y\rangle) > \left(\frac{1}{2} + \varepsilon\right) \cdot 2^{q_2(|\langle x,y\rangle|)}\right) \quad \textit{and} \\
x \notin A \quad &\longrightarrow \quad \bigwedge_{y \in \Sigma^*} \left(|y| = q_1(|x|) \longrightarrow \mathrm{count}_B^{q_2}(\langle x, y\rangle) < \left(\frac{1}{2} - \varepsilon\right) \cdot 2^{q_2(|\langle x,y\rangle|)}\right).
\end{aligned}
$$

Note that $\exists\cdot$BPP is contained in MA, but both classes do not seem to be equal by an oracle construction [FFKL93]. The class AM can be defined as $\mathrm{AM} \stackrel{df}{=} \mathrm{BP}\cdot\mathrm{NP} = \mathrm{BP}\cdot\exists\cdot\mathrm{P}$. It is known that $\mathrm{MA} \subseteq \mathrm{AM}$ [Bab85, Sch89]. One can continue the alternating application of BP$\cdot$ and $\exists\cdot$ to obtain classes in the style $\ldots\exists\cdot\mathrm{BP}\cdot\exists\cdot\ldots\mathrm{P}$ which build the Arthur-Merlin hierarchy. However, Babai showed that this hierarchy collapses to its second level, i.e., to AM.

In connection with operators, we are interested in the question whether closure of some complexity class with respect to a specified reducibility entails the same closure after application of an operator.

Lemma 2.10 *If $\mathcal{C}$ is closed under $\leq_c^{\mathrm{P}}$, then $\mathrm{U}\cdot\mathcal{C}$ is closed under $\leq_c^{\mathrm{P}}$.*

Proof: Let A be some set, let $B \in \mathrm{U}\cdot\mathcal{C}$ such that $A \leq_c^{\mathrm{P}} B$ via some function $f \in \mathrm{FP}$. As discussed above let, $f(x) = \langle x_1, \ldots, x_k\rangle$, $k = s(|x|)$, and $|x_1| = \ldots = |x_k| = r(|x|)$ for all $x \in \Sigma^*$ and polynomials r and s. Let $B \in \mathrm{U}\cdot\mathcal{C}$ via some set $C \in \mathcal{C}$ and a polynomial q. We define a new set C' as

$$C' \stackrel{df}{=} \left\{ \langle x, y_1 \cdot \ldots \cdot y_k\rangle : f(x) = \langle x_1, \ldots, x_k\rangle \ \wedge \bigwedge_{1\leq i\leq k} \big(\langle x_i, y_i\rangle \in C \ \wedge \ |y_i| = q(|x_i|)\big) \right\}.$$

C' is conjunctive reducible to some set in $\mathcal{C}$, therefore $C' \in \mathcal{C}$. Let $q' = s \cdot q(r)$. There is at most one word y of length $q'(|x|)$ for every $x \in \Sigma^*$ such that $\langle x, y\rangle \in C'$, and exactly one, if $x \in A$. Therefore, $A \in \mathrm{U}\cdot\mathcal{C}$. □

It is an open question whether UP is closed under $\leq_{maj}^{\mathrm{P}}$ if $\mathcal{C}$ is closed under $\leq_{maj}^{\mathrm{P}}$. Equivalently, which closure properties of a class $\mathcal{C}$ are required such that $\mathrm{U}\cdot\mathcal{C}$ is closed under $\leq_{maj}^{\mathrm{P}}$?

Lemma 2.11 *Let $\mathcal{C}$ be a complexity class. If $\mathcal{C}$ is closed under $\leq_m^{\mathrm{P}}$, then $\mathrm{U}\cdot\mathrm{U}\cdot\mathcal{C} = \mathrm{U}\cdot\mathcal{C}$ and $\exists\cdot\exists\cdot\mathcal{C} = \exists\cdot\mathcal{C}$. If $\mathcal{C}$ is closed under $\leq_{maj}^{\mathrm{P}}$, then $\mathrm{BP}\cdot\mathrm{BP}\cdot\mathcal{C} = \mathrm{BP}\cdot\mathcal{C}$.*

Proof: The proofs of the cases U$\cdot$ and $\exists\cdot$ follow the same scheme. Let $A \in \mathrm{U}\cdot\mathrm{U}\cdot\mathcal{C}$ via some set $B \in \mathcal{C}$ and polynomials p_1 and p_2. For every $x \in \Sigma^*$,

$$x \in A \ \longleftrightarrow \ \left|\left\{y : |y| = p_1(|x|) \ \wedge \ \mathrm{count}_B^{p_2}(\langle x, y\rangle) \geq 1\right\}\right| \geq 1.$$

We define B' as

$$B' = \{\langle x, y_1 y_2\rangle : \langle\langle x, y_1\rangle, y_2\rangle \in B \ \wedge \ |y_1| = p_1(|x|) \ \wedge \ |y_2| = p_2(|\langle x, y_1\rangle|)\}.$$

By assumption, $B' \in \mathcal{C}$. Let $q = p_1 + p_2(2\cdot(id + p_1))$. If $x \notin A$, then $\mathrm{count}_{B'}^q(x) = 0$; otherwise, $\mathrm{count}_{B'}^q(x) \geq 1$. To show $\exists\cdot\exists\cdot\mathcal{C} = \exists\cdot\mathcal{C}$, it suffices to replace U$\cdot$ by $\exists\cdot$.

The statement for BP$\cdot$ follows by amplification [Sch89]. □

As remarked, UP does not seem to be closed under $\leq_{maj}^{\mathrm{P}}$. It is an interesting question whether $\mathrm{BP}\cdot\mathrm{BP}\cdot\mathrm{UP} \subseteq \mathrm{BP}\cdot\mathrm{UP}$. In other words, does the successive application of BP$\cdot$ to UP produce an infinite hierarchy? At least, these classes are all contained in AM.

3 Completeness

Usual complexity classes are defined via machines. When considering promise classes, one additionally makes assumptions about the computation process of these machines. For example for UP, we use the resources of a non-deterministic polynomial-time Turing machine, and additionally we assume that for all inputs there is at most one accepting path. Machines having certain resources can be enumerated recursively. This enumeration gives a way to construct complete problems. For example,

$$\{0^i 10^t 1x \;:\; \text{the } i\text{-th NP machine accepts } x \text{ within } t \text{ steps}\}$$

is a many-one complete set for NP. In contrast, because of the additional assumption about the computation process, most of the known promise classes do not admit a recursive enumeration. As a consequence, we cannot construct complete sets in this way.

This section studies the question of whether SBP has many-one complete sets. This question is related to whether SBP is recursively enumerable. We show that SBP allows an enumeration in a weak sense (i.e., the enumeration does not tell us its probability gap). In contrast, we show that SBP is enumerable in a stronger sense if and only if SBP has many-one complete sets. The section ends with the construction of an oracle relative to which SBP does not have many-one complete sets. Even more, it does not contain a set that is many-one hard for BPP.

We fix enumerations $\{f_i\}_{i\geq 0}$ of all FP-functions and $\{g_j\}_{j\geq 0}$ of all $\#\mathrm{P}$-functions. Let F_i be a deterministic polynomial-time Turing machine that computes f_i in time $n^i + i$ and let G_j be a non-deterministic polynomial-time Turing machine that computes g_j in time $n^j + j$.

3.1 Uniform Enumerations

We want to consider enumerations of SBP. First, we state precisely what an SBP-machine is. The definition is based on SBP's characterization via FP- and $\#\mathrm{P}$-functions [BGM03, Proposition 2].

Definition 3.1 *An* SBP*-machine is a triple of natural numbers* (i, j, n) *where* $n \geq 2$ *such that for all words* w*, either*

$$\begin{array}{rcl} g_j(w) & > & \left(1 + \frac{1}{n}\right) \cdot f_i(w) \quad \textit{or} \\ g_j(w) & < & \left(1 - \frac{1}{n}\right) \cdot f_i(w). \end{array}$$

Definition 3.2 *The language accepted by the* SBP*-machine* (i, j, n) *is defined as*

$$L_{\mathrm{SBP}}(i, j, n) \stackrel{df}{=} \{w \;:\; g_j(w) > f_i(w)\}.$$

Proposition 3.3 *A language belongs to* SBP *if and only if it is accepted by an* SBP*-machine.*

Proof: Follows from SBP's characterization via FP- and $\#\mathrm{P}$-functions [BGM03, Proposition 2]. □

For promise classes like UP it is clear how to define the notion of uniform enumeration. Since the promise is "*one* accepting path", we only need an enumeration of machines. However, for classes like BPP, NP $\cap$ SPARSE, and SBP, there is some freedom in the definition. Here the promise (i.e., census function for NP $\cap$ SPARSE and probability gap for BPP and SBP) varies. Should the enumeration tell us just machines or both, machines and promises? We consider both notions and start with the stronger one.

Definition 3.4 *We call* SBP uniformly enumerable *if there is a recursive function* $h : \mathbb{N} \to \mathbb{N}^3$ *such that*

1. *every* $(i, j, n) \in \mathrm{range}(h)$ *is an* SBP*-machine, and*
2. *for all sets* $A \in \mathrm{SBP}$ *there exists an* SBP*-machine* $(i, j, n) \in \mathrm{range}(h)$ *that accepts* A*.*

Function h *uniformly enumerates* SBP*.*

The first statement demands $L_{\mathrm{SBP}}(h) \subseteq \mathrm{SBP}$ and the second statement demands $\mathrm{SBP} \subseteq L_{\mathrm{SBP}}(h)$ for a function h uniformly enumerating SBP.

Lemma 3.5 SBP *is uniformly enumerable via* h *such that* $\mathrm{range}(h) \subseteq \mathbb{N}^2 \times \{2\}$*.*

Proof: This follows by amplification [BGM03, Proposition 3]: We start with an enumeration h' and define an enumeration h that simulates h'. If h' outputs (i, j, n), then h amplifies this machine and outputs the amplified machine $(i', j', 2)$. □

Hartmanis and Hemachandra [HH88] investigated uniform enumerations for UP and BPP. The following proposition shows that we defined "uniform enumeration for SBP" in their sense.

Proposition 3.6 SBP *is uniformly enumerable if and only if for every* $0 < \varepsilon < \frac{1}{2}$ *there is a recursive function* $h_\varepsilon : \mathbb{N} \to \mathbb{N}^2$ *such that the following holds.*

1. *For every* $(i, j) \in \mathrm{range}(h_\varepsilon)$ *and every* $x \in \Sigma^*$*,*
$$\begin{aligned} g_j(x) &> (1+\varepsilon) \cdot f_i(x) \quad \textit{or} \\ g_j(x) &< (1-\varepsilon) \cdot f_i(x). \end{aligned}$$

2. *For every* $A \in \mathrm{SBP}$ *there exists* $(i, j) \in \mathrm{range}(h_\varepsilon)$ *such that*
$$x \in A \longleftrightarrow g_j(x) > f_i(x).$$

Proof: "$\Longleftarrow$" Let h be the enumeration for $\varepsilon = \frac{1}{3}$ and define $h'(m) \stackrel{df}{=} (i, j, 3)$ where $h(m) = (i, j)$. We show that SBP is uniformly enumerable via h'. If $(i, j, n) \in \mathrm{range}(h')$, then $n = 3$ and $(i, j) \in \mathrm{range}(h)$. Hence for $x \in \Sigma^*$,
$$\begin{aligned} g_j(x) &> (1+\tfrac{1}{3}) \cdot f_i(x) \quad \text{or} \\ g_j(x) &< (1-\tfrac{1}{3}) \cdot f_i(x). \end{aligned}$$

Hence (i, j, n) is an SBP-machine. This shows item 1 in Definition 3.4.

Let $A \in$ SBP. There exist $(i, j) \in \mathrm{range}(h)$ such that

$$\begin{array}{lcl} x \in A & \longrightarrow & g_j(x) > (1+\frac{1}{3}) \cdot f_i(x) \quad \text{and} \\ x \notin A & \longrightarrow & g_j(x) < (1-\frac{1}{3}) \cdot f_i(x). \end{array}$$

Hence $(i, j, 3) \in \mathrm{range}(h')$ is an SBP-machine that accepts A. This shows item 2 in Definition 3.4.

"$\Longrightarrow$" Assume SBP is uniformly enumerable via h. By Lemma 3.5, we may assume that for all $(i, j, n) \in \mathrm{range}(h)$, $n = 2$. Let $0 < \varepsilon < 1/2$ and define h' to be the projection of h that neglects the last component of h.

If $A \in$ SBP, then there exists an SBP-machine $(i, j, 2) \in \mathrm{range}(h)$ that accepts A. Therefore, for all words x,

$$\begin{array}{lcl} x \in A & \longrightarrow & g_j(x) > (1+\frac{1}{2}) \cdot f_i(x) \quad \longrightarrow \quad g_j(x) > (1+\varepsilon) \cdot f_i(x) \quad \text{and} \\ x \notin A & \longrightarrow & g_j(x) < (1-\frac{1}{2}) \cdot f_i(x) \quad \longrightarrow \quad g_j(x) < (1-\varepsilon) \cdot f_i(x). \end{array}$$

Since $(i, j) \in \mathrm{range}(h')$, this shows 3.6.2.

Let $(i, j) \in \mathrm{range}(h')$ and $x \in \Sigma^*$. Hence $(i, j, 2) \in \mathrm{range}(h)$ and therefore,

$$\begin{array}{lcl} g_j(x) & > & (1+\frac{1}{2}) \cdot f_i(x) \quad \text{or} \\ g_j(x) & < & (1-\frac{1}{2}) \cdot f_i(x). \end{array}$$

We obtain 3.6.1. □

Hartmanis and Hemachandra [HH88] showed that UP is uniformly enumerable if and only if UP has many-one complete sets. The same technique shows similar results for NP $\cap$ coNP, BPP, and BQP [BFFvM00]. We apply this technique to SBP. Interestingly enough, this technique does not show a similar result for NP $\cap$ SPARSE [BFFvM00]. Intuitively, NP $\cap$ SPARSE is quite close to complexity classes that have no promise, since its promise aims at the accepted language and not at the computation process. As shown by Hartmanis and Yesha [HY84], NP $\cap$ SPARSE has a Turing-complete set.

Theorem 3.7 SBP *is uniformly enumerable if and only if it has many-one complete sets.*

Proof: "$\Longrightarrow$": Let H be a deterministic Turing machine that computes the enumeration h. By Lemma 3.5, we can assume that for all $(i, j, n) \in \mathrm{range}(h)$, $n = 2$. We define the following set that we will prove to be $\leq_m^{\mathrm{P}}$-complete for SBP.

$$L \stackrel{df}{=} \{\langle x, 0^n, i, j, w\rangle \;:\; |w|^{i+j} + i + j \leq n, H(x) \text{ outputs } (i, j, 2) \text{ within } n \text{ steps}, g_j(w) \geq f_i(w)\}$$

Containment in SBP. Define the following functions.

$$f(z) \stackrel{df}{=} \begin{cases} f_i(w) & : \text{ if } z = \langle x, 0^n, i, j, w\rangle \text{ such that } |w|^{i+j} + i + j \leq n \\ & \quad\; \text{and } H(x) \text{ outputs } (i, j, 2) \text{ within } n \text{ steps} \\ 1 & : \text{ otherwise} \end{cases}$$

$$g(z) \stackrel{df}{=} \begin{cases} g_j(w) & : \text{ if } z = \langle x, 0^n, i, j, w\rangle \text{ such that } |w|^{i+j} + i + j \leq n \\ & \quad\; \text{and } H(x) \text{ outputs } (i, j, 2) \text{ within } n \text{ steps} \\ 0 & : \text{ otherwise} \end{cases}$$

Note that the conditions in these definitions can be tested in time polynomial in $|z|$. Assume we are given some $z = \langle x, 0^n, i, j, w\rangle$ that satisfies these conditions. The value $f_i(w)$ is computed by F_i in time at most $|w|^i + i \leq n \leq |z|$. Hence $f \in \mathrm{FP}$. Likewise, $g_j(w)$ is the number of accepting paths of $G_j(w)$. The running time of $G_j(w)$ is at most $|w|^j + j \leq n \leq |z|$. This shows $g \in \#\mathrm{P}$.

Observe that for all $z \in \Sigma^*$,

$$z \in L \longleftrightarrow g(z) \geq f(z). \tag{1}$$

Assume that there exists z such that

$$(1 - \tfrac{1}{2}) \cdot f(z) \leq g(z) \leq (1 + \tfrac{1}{2}) \cdot f(z). \tag{2}$$

From the definition of f and g it follows that $z = \langle x, 0^n, i, j, w\rangle$ such that $|w|^{i+j} + i + j \leq n$ and $H(x)$ outputs $(i, j, 2)$ within n steps. Therefore, $f(z) = f_i(w)$ and $g(z) = g_j(w)$.

$$(1 - \tfrac{1}{2}) \cdot f_i(w) \leq g_j(w) \leq (1 + \tfrac{1}{2}) \cdot f_i(w). \tag{3}$$

Hence, $h(x) = (i, j, 2)$ is not an SBP-machine. This contradicts our assumption. Together with equation (1) this implies for all $z \in \Sigma^*$,

$$\begin{array}{lcll} z \in L & \longrightarrow & g(z) > \left(1 + \frac{1}{2}\right) \cdot f(z) & \text{and} \\ z \notin L & \longrightarrow & g(z) < \left(1 - \frac{1}{2}\right) \cdot f(z). & \end{array}$$

This shows $L \in \mathrm{SBP}$.

Hardness for SBP. Let $L' \in \mathrm{SBP}$. There exists an x such that $h(x) = (i, j, 2)$ and the SBP-machine $(i, j, 2)$ accepts L'. Let t be the computation time of $H(x)$ and let

$$s(w) \stackrel{\mathit{df}}{=} \langle x, 0^{\max\{t, |w|^{i+j}+i+j\}}, i, j, w\rangle.$$

Since in this definition x, t, i, and j appear as constants, $s \in \mathrm{FP}$. From the definition of L it follows that $L' \leq_m^{\mathrm{P}} L$ via reduction function s.

"$\Longleftarrow$": Let L be an SBP-complete set accepted by SBP-machine $(i, j, 2)$. For $k \geq 0$, let

$$h(k) \stackrel{\mathit{df}}{=} (i', j', 2)$$

where i' is the index of $f_i \circ f_k \in \mathrm{FP}$ and j' is the index of $g_j \circ f_k \in \#\mathrm{P}$. Since these indices can be determined effectively, h is a recursive function.

If $(i', j', 2) \in \mathrm{range}(h)$ and $(i', j', 2)$ is not an SBP-machine, then $(i, j, 2)$ is not an SBP-machine, either. Hence, every $(i', j', 2) \in \mathrm{range}(h)$ is an SBP-machine.

If $A \in \mathrm{SBP}$, then $A \leq_m^{\mathrm{P}} L$ via some polynomial-time reduction function f_k. We obtain:

$$w \in A \longrightarrow f_k(w) \in L \longrightarrow g_j(f_k(w)) > (1 + \tfrac{1}{2}) \cdot f_i(f_k(w)) \longrightarrow g_{j'}(w) > (1 + \tfrac{1}{2}) \cdot f_{i'}(w)$$

$$w \notin A \longrightarrow f_k(w) \notin L \longrightarrow g_j(f_k(w)) < (1 - \tfrac{1}{2}) \cdot f_i(f_k(w)) \longrightarrow g_{j'}(w) < (1 - \tfrac{1}{2}) \cdot f_{i'}(w)$$

Hence there is an SBP-machine $(i', j', 2) \in \mathrm{range}(h)$ that accepts A. □

In Theorem 3.12 below we will construct an oracle relative to which SBP does not have many-one complete sets. Relative to this oracle, SBP is not uniformly enumerable. Hence we do not expect SBP to be uniformly enumerable. However, SBP is uniformly enumerable in the following weaker sense. Here we do not demand that the enumeration function outputs the size of the probability gap (i.e., parameter n in Definition 3.4). Buhrman, Fenner, Fortnow, and van Melkebeek consider a similar weak enumeration for NP $\cap$ SPARSE [BFFvM00].

Definition 3.8 SBP *is* uniformly enumerable without gap *if there is a recursive function* $h : \mathbb{N} \to \mathbb{N}^2$ *such that*

1. *for every* $(i, j) \in \mathrm{range}(h)$ *there exists an* $n \geq 2$ *such that* (i, j, n) *is an* SBP*-machine, and*

2. *for all* $A \in$ SBP *there exists an* SBP*-machine* (i, j, n) *that accepts* A *and* $(i, j) \in \mathrm{range}(h)$.

Theorem 3.9 SBP *is uniformly enumerable without gap.*

The enumeration is based on a trick (also called *cheat*) which was used by Buhrman, Fenner, Fortnow, and van Melkebeek [BFFvM00] to obtain a weak enumeration for NP $\cap$ SPARSE. However, for SBP an additional hurdle appears. The machine model for NP $\cap$ SPARSE has the following property: Every machine that accepts a finite language, is a valid machine (since we can choose the bounding polynomials to be arbitrarily large). However, for SBP this does not hold. If there exists some x such that $f_i(x) = g_j(x)$, then for all n, (i, j, n) is not a valid SBP-machine. Our proof prevents that this happens.

Proof: We start from an enumeration of all pairs (i', j') of natural numbers. For every (i', j') define the following functions.

$$f(x) \stackrel{df}{=} \begin{cases} 2 \cdot f_{i'}(x) + 1 & : \text{ if } f_{i'}(x) > 0 \\ 1 & : \text{ otherwise} \end{cases}$$

$$g(x) \stackrel{df}{=} \begin{cases} 2 \cdot g_{j'}(x) & : \text{ if for all } y, |y| \leq \log\log|x|, \\ & \quad \text{either } g_{j'}(y) < \frac{1}{2} \cdot f_{i'}(y) \text{ or } g_{j'}(y) > \frac{3}{2} \cdot f_{i'}(y) \\ 0 & : \text{ otherwise} \end{cases}$$

Note that $f \in$ FP and $g \in \#$P (the condition in g's definition can be tested deterministically in time polynomial in $|x|$). This definition ensures that for all x, $f(x) \neq g(x)$, yet the change in the ratio of f and g is insignificant. Determine i and j such that $f = f_i$ and $g = g_j$. Output the pair (i, j). So the new enumeration consists of all pairs (i, j).

If $A \in$ SBP, then there exists an SBP-machine $(i', j', 2)$ that accepts A. We may assume $f_{i'} > 0$. The pair (i, j) appears in the enumeration. If $x \in A$, then $g_{j'}(x) - 1 \geq \frac{3}{2} \cdot f_{i'}(x)$ and therefore,

$$g_j(x) = 2 \cdot g_{j'}(x) \geq \frac{3}{2} \cdot 2 \cdot f_{i'}(x) + 2 > \frac{3}{2} \cdot f_i(x).$$

If $x \notin A$, then $g_{j'}(x) < \frac{1}{2} \cdot f_{i'}(x)$ and therefore,

$$g_j(x) = 2 \cdot g_{j'}(x) < \tfrac{1}{2} \cdot 2 \cdot f_{i'}(x) < \tfrac{1}{2} \cdot f_i(x).$$

It follows that $(i, j, 2)$ is an SBP-machine that accepts A and (i, j) appears in the enumeration. This shows statement 2 of Definition 3.8.

Suppose that (i, j) appears in the enumeration. If $(i, j, 2)$ is an SBP-machine, then we are done. Otherwise, by definition of g_j, for at most finitely many x, $g_j(x) \neq 0$.

$$n \stackrel{df}{=} 2 + \max(\{g_j(x) \,:\, x \in \Sigma^*\} \cup \{f_i(x) \,:\, x \in \Sigma^* \wedge g_j(x) > 0\})$$

We know that for all x, $f_i(x) \neq g_j(x)$. If $g_j(x) > f_i(x)$, then

$$g_j(x) \geq f_i(x) + 1 > (1 + \tfrac{1}{n}) \cdot f_i(x).$$

If $g_j(x) < f_i(x)$, then

$$g_j(x) \leq f_i(x) - 1 < (1 - \tfrac{1}{n}) \cdot f_i(x).$$

Therefore, (i, j, n) is an SBP-machine. This shows statement 1 of Definition 3.8. We conclude that SBP is effectively enumerable without gap. □

3.2 Oracle Construction

This subsection gives evidence that SBP does not have many-one complete sets. Similar results are known for other promise classes: Sipser [Sip82] proved that R and $\mathrm{NP} \cap \mathrm{coNP}$ have no many-one complete sets relative to oracles. By Gurevich [Gur83], it follows that $\mathrm{NP} \cap \mathrm{coNP}$ has no Turing-complete sets in a relativized world. Hartmanis and Hemachandra [HH88] show that relative to oracles, UP and BPP do not have many-one complete sets. By Ambos-Spies [AS86], it follows that there is an oracle relative to which BPP does not have Turing-complete sets. Hemaspaandra, Jain, and Vereshchagin [HJV93] improve these results and show that ZPP, R, UP and FewP do not have Turing complete sets relative to oracles.

We construct an oracle relative to which AM does not contain any many-one hard set for BPP. Hence SBP does not have many-one complete sets relative to this oracle. Since $\mathrm{BPP} \subseteq \mathrm{SBP} \subseteq \mathrm{AM} \subseteq \Pi_2^{\mathrm{P}}$, our oracle is optimal in the sense that Π_2^{P} contains complete sets that are many-one hard for BPP.

Using amplification, AM can be characterized in the following way.

Proposition 3.10 *A set A is in* AM *if there are $B \in \mathrm{NP}$ and a polynomial r such that for all $x \in \Sigma^*$,*

$$\begin{array}{lcl} x \in A & \longrightarrow & \mathrm{count}_B^r(x) > \frac{3}{4} \cdot 2^{r(|x|)} \quad \textit{and} \\ x \notin A & \longrightarrow & \mathrm{count}_B^r(x) < \frac{1}{4} \cdot 2^{r(|x|)}. \end{array}$$

We fix an enumeration of pairs of natural numbers (i, j) where i stands for the i-th polynomial q_i and j for the j-th non-deterministic polynomial-time Turing-machine M_j. For every set $A \in \mathrm{AM}$

there is a tuple in this enumeration such that A is characterized by $B = L(M_j)$ and $r = q_i$ (Proposition 3.10). We call such a pair (i, j) an AM-calculation. Let

$$L_{\mathrm{AM}}(i,j) \stackrel{df}{=} \{x : \mathrm{count}^{q_i}_{L(M_j)}(x) > \tfrac{1}{2} \cdot 2^{q_i(|x|)}\}$$

be the language accepted by (i, j).

We start the construction with a lemma providing the main argument. The idea is typical for oracle constructions dealing with promise classes: Either the machine does not accept a given language (items (i) and (ii)), or we can destroy the promise of the machine (item (iii)).

Lemma 3.11 *Let M be a non-deterministic polynomial-time oracle Turing machine with running time p and let q be a polynomial. Let $n \geq 1$, $O \subseteq \Sigma^{<n}$, and $F \subseteq \Sigma^n$ such that $\|F\| \leq 2^{n-3}$. For every $x \in \Sigma^*$ where $p(|x| + q(|x|)) \leq 2^{n-4}$, there exists an $X \subseteq \Sigma^n - F$ such that one of the following holds:*

(i) $\|X\| \leq \frac{1}{4} \cdot 2^n$ *and the number of $y \in \Sigma^{q(|x|)}$ such that $M^{O \cup X}(x, y)$ accepts is greater than* $\frac{3}{4}2^{q(|x|)}$.

(ii) $\|X\| \geq \frac{3}{4} \cdot 2^n$ *and the number of $y \in \Sigma^{q(|x|)}$ such that $M^{O \cup X}(x, y)$ accepts is less than* $\frac{1}{4}2^{q(|x|)}$.

(iii) The number of $y \in \Sigma^{q(|x|)}$ such that $M^{O \cup X}(x, y)$ accepts belongs to the interval $[\frac{1}{4}2^{q(|x|)}, \frac{3}{4}2^{q(|x|)}]$.

Proof: Assume there exists x such that for all $X \subseteq \Sigma^n - F$, the statements (i), (ii), and (iii) do not hold. Choose an $X \subseteq \Sigma^n - F$ of minimal cardinality such that there are more than $\frac{3}{4}2^{q(|x|)}$ words $y \in \Sigma^{q(|x|)}$ such that $M^{O \cup X}$ accepts (x, y). Such a set X exists, since (i), (ii), and (iii) do not hold. $\|X\| > \frac{1}{4} \cdot 2^n$, since otherwise (i) holds. Choose pairwise different elements $x_0, x_1, \ldots, x_{2p(|x|+q(|x|))} \in X$. For every i, $0 \leq i \leq 2p(|x| + q(|x|))$, we argue as follows: Since X is minimal and since (iii) does not hold, there are less than $\frac{1}{4}2^{q(|x|)}$ words $y \in \Sigma^{q(|x|)}$ such that $M^{O \cup X - \{x_i\}}$ accepts (x, y). Hence, for at least one half of all $y \in \Sigma^{q(|x|)}$ it holds that $M^{O \cup X}(x, y)$ accepts and x_i is queried on all accepting paths. By pigeon hole principle, there is a $y \in \Sigma^{q(|x|)}$ such that $M^{O \cup X}(x, y)$ accepts and every accepting path asks more than $p(|x| + |y|)$ queries. This contradicts the running time of M. $\square$

Theorem 3.12 *There exists an oracle relative to which* AM *does not contain a set that is $\leq^{\mathrm{P}}_m$-hard for* BPP.

Proof: Fix an enumeration of all triples (i, j, k) of natural numbers where (i, j) represents a possible AM-calculation and k stands for the k-th FP-function f_k. For every oracle Z and every i and j, let

$$W^Z_{i,j} \stackrel{df}{=} \{0^n : n \text{ is a power of the } \langle i,j \rangle\text{-th prime number and } \|Z \cap \Sigma^n\| \geq \tfrac{1}{2} \cdot 2^n\}$$

be our witness languages. We construct the oracle O and consider f_k as a possible many-one reduction function that reduces $W^O_{i,j}$ to the language accepted by the AM-calculation (i, j). The construction ensures one of the following:

- Either $W^O_{i,j}$ is in BPP but not many-one reducible to $L_{\mathrm{AM}}(i,j)$
- or (i,j) is not an AM-calculation.

We construct the oracle in stages such that in each stage we diagonalize against one triple (i,j,k). Let (i,j,k) be the next triple on our list, and let O be the oracle constructed so far. Let $q = q_i$ and $M = M_j$, and let p be the running time of M. Let n be a power of the $\langle i,j \rangle$-th prime number. We choose n large enough such that $O \subseteq \Sigma^{<n}$ and adding words of length n to the oracle does not effect diagonalizations made in previous steps.

Let $x \stackrel{\mathit{df}}{=} f^O_k(0^n)$. F denotes the set of all queries of length n that are asked during computation $f^O_k(0^n)$. We may assume that n was chosen large enough such that $\|F\| \leq 2^{n-3}$ and $p(|x| + q(|x|)) \leq 2^{n-4}$. We apply Lemma 3.11 and obtain an $X \subseteq \Sigma^n - F$ such that (i), (ii), or (iii) holds. Let $O := O \cup X$. This finishes the diagonalization against (i,j,k) and we can proceed with the next triple on our list.

It remains to show that relative to O, AM does not have a set that is $\leq^{\mathrm{P}}_m$-hard for BPP. Assume that there exists a set A in AM^O such that A is $\leq^{\mathrm{P}}_m$-hard for BPP^O. There must be an AM^O-calculation (i,j) that accepts A. Let $q = q_i$ and $M = M_j$, and let p be the running time of M.

Case 1: There exists $k' \geq 0$ such that in the diagonalization against the triple (i,j,k'), statement (iii) of Lemma 3.11 holds. We obtain

$$\#\{y \in \Sigma^{q(|x|)} \,:\, M \text{ accepts } (x,y)\} \in \left[\tfrac{1}{4} \cdot 2^{q(|x|)}, \tfrac{3}{4} \cdot 2^{q(|x|)}\right].$$

This contradicts our assumption that (i,j) is an AM^O-calculation. Hence Case 1 is not possible.

Case 2: For all $k' \geq 0$, during the diagonalization against the triple (i,j,k'), either statement (i) or statement (ii) of Lemma 3.11 holds. It follows that for all n, if n is a power of the $\langle i,j \rangle$-th prime number, then either $\|O \cap \Sigma^n\| \leq \frac{1}{4} \cdot 2^n$ or $\|O \cap \Sigma^n\| \geq \frac{3}{4} \cdot 2^n$. Hence $W^O_{i,j} \in \mathrm{BPP}^O$. Since A is $\leq^{\mathrm{P}}_m$-hard for BPP^O, there exists k such that $W^O_{i,j} \leq^{\mathrm{P}}_m A$ via f^O_k. Consider the stage in the construction where we treated the triple (i,j,k). By our assumption, either statement (i) or statement (ii) of Lemma 3.11 holds. Therefore, $0^n \in W^O_{i,j} \longleftrightarrow x = f^O_k(0^n) \notin A$. This contradicts our assumption and shows that relative to O, AM does not have a set that is $\leq^{\mathrm{P}}_m$-hard for BPP. $\square$

Corollary 3.13 *There exists an oracle relative to which neither* SBP *nor* AM *have many-one complete sets.*

4 The Operator SB$\cdot$

The complexity class PP is the largest class of languages acceptable by polynomial-time probabilistic Turing machines [Gil77]. An input is accepted by a probabilistic Turing machine if and only if an accepting computation appears with probability more than $\frac{1}{2}$. A threshold machine looks only on the rate of accepting paths with respect to the total number of computation paths and accepts an input if and only if more than half of the paths are accepting. Hence, the difference to probabilistic machines is, that we do not require balanced computation trees. Nevertheless, in case of polynomial-time Turing machines, the notions of threshold and probabilistic machines coincide [Sim75]. If we modify both definitions by demanding a probability gap at $\frac{1}{2}$ this results, for probabilism, in BPP and, in case of threshold machines, in $\mathrm{BPP_{path}}$. By loosening the definition of BPP, we obtain SBP, which is located between BPP and $\mathrm{BPP_{path}}$ [BGM03]. To understand SBP as a probabilistic class, we modify the definition of a probabilistic Turing machine so that an input x is accepted if the probability of an accepting computation is more than some polynomial power of $\frac{1}{2}$, i.e., $2^{-p(|x|)}$ for some polynomial p; and x is not accepted if the probability of an accepting path is at most $2^{-p(|x|)}$. It is not hard to see that the class of languages acceptable in the sense of this more general model is still equal to PP. Applying the same modification, i.e., requiring a probability gap, leads to the definition of SBP. In this section we introduce and investigate the operator $\mathrm{SB}\cdot$ which is derived from SBP in the same manner as $\mathrm{BP}\cdot$ is derived from BPP.

Definition 4.1 *Let $\mathcal{C}$ be a complexity class. For every set A, $A \in \mathrm{SB}{\cdot}\mathcal{C}$ if and only if there exist $B \in \mathcal{C}$, polynomials p and q, and $\varepsilon \in (0,1)$ such that for every $x \in \Sigma^*$,*

$$\begin{array}{lcl} x \notin A & \longrightarrow & \mathrm{count}_B^q(x) < (1-\varepsilon)\cdot 2^{p(|x|)} \quad \textit{and} \\ x \in A & \longrightarrow & \mathrm{count}_B^q(x) > (1+\varepsilon)\cdot 2^{p(|x|)}. \end{array}$$

If $A \in \mathrm{SB}{\cdot}\mathcal{C}$, then we also say that A is in $\mathrm{SB}{\cdot}\mathcal{C}$ *via some set B, polynomials p and q, and ε*. The first polynomial always refers to the exponent of 2 and the second to that being involved in function count.

The property of each language in $\mathrm{SB}{\cdot}\mathcal{C}$ is that there is a computation such that a small interval of possible numbers of accepting paths of a computation is forbidden, and this gap separates the numbers that entail rejection and acceptance. Note that the gap varies in size and position relative to the number of computation paths and the size of the input in contrast to the fixed relative size and position of the gap in case of $\mathrm{BP}\cdot$. As we will see later, for inputs of growing size, the gap of size $2\varepsilon \cdot 2^{p(|x|)}$ becomes smaller relative to the number of computation paths if $q - p$ is not a constant (we will show that we can assume $p < q$). We verify $\mathrm{SB}{\cdot}\mathrm{P} = \mathrm{SBP}$. In [BGM03] it is shown that SBP can be characterized equivalently in different ways. We want to mention that each of these characterizations could serve as the foundation of the definition of the operator $\mathrm{SB}\cdot$, but we would obtain operators of different power. The reason is that amplification was used to prove these equivalences—a technique that is not applicable in general.

We observe that $\mathrm{SB}\cdot$ is monotonic with respect to inclusion. Before we look closer at the power that $\mathrm{SB}\cdot$ provides, we show basics about of the choice of both the polynomials p and q.

Lemma 4.2 *Let $\mathcal{C}$ be a complexity class, and let $A \in \mathrm{SB}{\cdot}\mathcal{C}$ via some $B \in \mathcal{C}$ and polynomials p and q in the sense of Definition 4.1.*

1. *$\mathrm{BP}{\cdot}\mathcal{C} \subseteq \mathrm{SB}{\cdot}\mathcal{C}$, and if $\mathcal{C}$ is closed under $\leq_m^{\mathrm{P}}$, then $\exists{\cdot}\mathcal{C} \subseteq \mathrm{SB}{\cdot}\mathcal{C}$.*
2. *If $\mathcal{C}$ is closed under $\leq_{bc}^{\mathrm{P}}$, p is constant, and q is unbounded, then $A \in \exists{\cdot}\mathcal{C}$.*
3. *If there is a natural n_0 such that for all $n \geq n_0$, $q(n) \leq p(n)$, then A is finite.*
4. *If $\mathcal{C}$ is closed under $\leq_{maj}^{\mathrm{P}}$, $p < q$, and q is constant, then $A \in \mathcal{C}$.*

5. *If $\mathcal{C}$ is closed under $\leq_m^p$, then there are $B' \in \mathcal{C}$ and polynomials p', q' where $p'(n) > 0$ for all $n \in \mathbb{N}$, such that $A \in \mathrm{SB}\cdot\mathcal{C}$ via B', p', and q'.*

Proof:

1. Let $A \in \mathrm{BP}\cdot\mathcal{C}$ via some set $B \in \mathcal{C}$, some $\varepsilon \in (0,1)$, and polynomial p. Hence $A \in \mathrm{SB}\cdot\mathcal{C}$ via set B, the polynomials $p-1$ and p, and 2ε. Let $A \in \exists\cdot\mathcal{C}$ via some set $B \in \mathcal{C}$ and a polynomial p. Define a new set B' as
$$B' \stackrel{df}{=} 0B \cup 1B \cup B''$$
where $B'' = \{e, 0, 1\}$ if B is non-empty and $B'' = \emptyset$ otherwise (e denotes the empty word). B' many-one reduces to B and is therefore contained in $\mathcal{C}$. Hence $A \in \mathrm{SB}\cdot\mathcal{C}$ in the sense of Definition 4.1 via B', the polynomials 0 and $p+1$, and any $\varepsilon \in (0,1)$.

2. We define a new set B'. Fix some $x \in \Sigma^*$ and let $k = 2^{p(|x|)}$ and
$$B' \stackrel{df}{=} \left\{ \langle x, y_1 \cdot \ldots \cdot y_k \rangle : \bigwedge_{i=1}^{k} \Big(\langle x, y_i \rangle \in B \ \wedge\ |y_i| = q(|x|) \ \wedge \bigwedge_{1 \leq j < i} y_i \neq y_j \Big) \right\}.$$
Obviously, $B' \leq_{bc}^{\mathrm{P}} B$, and therefore $B' \in \mathcal{C}$. For any $x \in \Sigma^*$ that is not from A, there are less than k words $z \in \Sigma^{q(|x|)}$ such that $\langle x, z \rangle \in B$. Hence, there is no $y \in \Sigma^{k \cdot q(|x|)}$ such that $\langle x, y \rangle \in B'$. Otherwise, if $x \in A$, there are at least k different $z \in \Sigma^{q(|x|)}$ with $\langle x, z \rangle \in B$, which means that there is at least one $y \in \Sigma^{k \cdot q(|x|)}$ such that $\langle x, y \rangle \in B'$. Hence, $A \in \exists\cdot\mathcal{C}$.

3. Since $2^{q(|x|)} < (1+\varepsilon) \cdot 2^{p(|x|)}$ for every $\varepsilon \in (0,1)$ and every $x \in \Sigma^*$ with length at least n_0, A does not contain any word of length at least n_0. Hence it is finite.

4. Observe that p must also be constant. We show that $A \leq_{maj}^{\mathrm{P}} B$, which entails $A \in \mathcal{C}$. Let $w \in \Sigma^*$ such that $w \in B$ (we can assume that $B \neq \emptyset$). Let $x \in \Sigma^*$; let $p = p(|x|)$, $q = q(|x|)$. Our reducing function f computes the following sequence of questions. Let $k = 2^q - 2^{p+1} - 1$, or $k = 1$ in case of $q = p+1$.
$$f(x) \stackrel{df}{=} \langle \underbrace{\langle x, 0^q \rangle, \ldots, \langle x, 1^q \rangle}_{2^q}, \underbrace{w, \ldots, w}_{k} \rangle$$
If $q = p+1$ and $x \in A$, then more than half of the computed queries $\langle x, z \rangle$, $z \in \Sigma^q$, are contained in B, and the number of queries that are accepted is at least 2 plus the number of queries that are not accepted. Hence more than half of the queries computed by f on input x are accepted. If $x \notin A$, then more than half of the queries of $f(x)$ are rejected, so that we obtain $A \leq_{maj}^{\mathrm{P}} B$ in this case. Similar arguments hold in case of $q > p+1$.

5. Let
$$B' \stackrel{df}{=} \{ \langle x, ay \rangle : a \in \{0,1\}, \langle x, y \rangle \in B, |y| = q(|x|) \},$$
let $q'(n) \stackrel{df}{=} q(n) + 1$, and $p'(n) = p(n) + 1$. If $x \notin A$, then $\mathrm{count}_{B'}^{q'}(x) < (1-\varepsilon) \cdot 2^{p'(|x|)}$, and if $x \in A$, then $\mathrm{count}_{B'}^{q'}(x) > (1+\varepsilon) \cdot 2^{p'(|x|)}$. Since $B' \leq_m^{\mathrm{P}} B$ and $\mathcal{C}$ is closed under $\leq_m^{\mathrm{P}}$ it follows that $B' \in \mathcal{C}$.

$\square$

Lemma 4.2 gives reason enough to assume $p < q$ henceforth. Besides, we observe in this case that the gap between the allowed numbers of accepting computations for rejection and acceptance gets closer to 0 relative to the number of computation paths with growing computation length.

One of the major properties of complexity classes defined by application of $\mathrm{SB}\cdot$ is the ability to reduce errors, which means reducing the ratio between the number of accepting paths and the number of all paths for inputs that are to be rejected, while maintaining the ratio of accepting paths and all paths for inputs that are to be accepted. More formally, the gap of width $2\varepsilon \cdot 2^{p(|x|)}$ for the number of accepting paths in the rejecting and accepting case can be extended to every desired value. In the case of BPP this is well-known as probability amplification.

Lemma 4.3 (Amplification 1) *Let $\mathcal{C}$ be a complexity class closed under $\leq_c^{\mathrm{P}}$, and let $A \in \mathrm{SB}\cdot\mathcal{C}$. For every polynomial r, there exist some set $B' \in \mathcal{C}$ and polynomials p' and q' such that for every $x \in \Sigma^*$,*

$$\begin{aligned} x \in A \quad &\longrightarrow \quad \mathrm{count}_{B'}^{q'}(x) \geq 2^{r(|x|)} \cdot 2^{p'(|x|)} \quad \textit{and} \\ x \notin A \quad &\longrightarrow \quad \mathrm{count}_{B'}^{q'}(x) \leq 2^{p'(|x|)}. \end{aligned}$$

Proof: Let $A \in \mathrm{SB}\cdot\mathcal{C}$ via some set $B \in \mathcal{C}$, polynomials p and q, and $\varepsilon > 0$ in the sense of Definition 4.1. Observe that there is a natural constant a such that both of the following properties hold for an appropriate natural number b:

$$(1+\varepsilon)^a > 4 \cdot (1-\varepsilon)^a \text{ and } \frac{1}{2^b} > (1-\varepsilon)^a \geq \frac{1}{2^{b+1}}$$

For convenience, we assume a to be as small as possible even though there is no need to any limitation of the size of a. We define a new set B' as

$$B' \stackrel{df}{=} \left\{ \langle x, y_1 \cdot \ldots \cdot y_k \rangle : k = a \cdot r(|x|) \ \wedge \bigwedge_{1 \leq i \leq k} \big(\langle x, y_i \rangle \in B \ \wedge \ |y_i| = q(|x|) \big) \right\}.$$

Obviously, $B' \leq_c^{\mathrm{P}} B$. Let $q' = a \cdot r \cdot q$ and $p' = ap - b$. We conclude the proof with the following argumentation, where $n \stackrel{df}{=} |x|$.

$$\begin{aligned} x \in A \quad \longrightarrow \quad \mathrm{count}_B^q(x) \ &> (1+\varepsilon) \cdot 2^{p(n)} \\ \longrightarrow \quad \mathrm{count}_{B'}^{q'}(x) &= (\mathrm{count}_B^q(x))^{ar(n)} \\ &> (4 \cdot (1-\varepsilon)^a)^{r(n)} \cdot 2^{ap(n)r(n)} \\ &\geq \frac{2^{r(n)}}{2^{br(n)}} \cdot 2^{ap(n)r(n)} \\ &= 2^{r(n)} \cdot 2^{p'(n)} \\ x \notin A \quad \longrightarrow \quad \mathrm{count}_B^q(x) \ &< (1-\varepsilon) \cdot 2^{p(n)} \\ \longrightarrow \quad \mathrm{count}_{B'}^{q'}(x) &= (\mathrm{count}_B^q(x))^{ar(n)} \\ &< \frac{1}{2^{br(n)}} \cdot 2^{ap(n)r(n)} \\ &= 2^{p'(n)}. \end{aligned}$$

$\square$

The main idea of the proof is to concatenate computation paths. Every such path is accepting if and only if all partial computations along this path are accepting. Since we cannot assume a fixed machine model, we express the concatenation in terms of reducibility. The number of accepting paths in the accepting and rejecting case are bounded above and below by powers of the original bounds.

In some cases it suffices to amplify the acceptance probability by only a constant factor. In these cases we can formulate a similar amplification lemma for classes that are closed under the stronger bounded conjunctive reduction.

Corollary 4.4 (Amplification 2) *Let $\mathcal{C}$ be a complexity class closed under $\leq_{bc}^{\mathrm{P}}$, and let $A \in \mathrm{SB}\cdot\mathcal{C}$. For every natural number a, there exist some set $B \in \mathcal{C}$ and polynomials p and q such that for every $x \in \Sigma^*$,*

$$\begin{array}{lcl} x \in A & \longrightarrow & \mathrm{count}_B^q(x) > a \cdot 2^{p(|x|)} \quad \textit{and} \\ x \notin A & \longrightarrow & \mathrm{count}_B^q(x) < 2^{p(|x|)}. \end{array}$$

5 Closure Properties

In this section we investigate the closure properties of classes that are derived from a basic class $\mathcal{C}$ by application of operators.

Lemma 5.1 *If $\mathcal{C}$ is closed under $\leq_m^{\mathrm{P}}$, then $\mathrm{SB}\cdot\mathcal{C}$ is closed under $\leq_m^{\mathrm{P}}$.*

Proof: Let A be some set, $B \in \mathrm{SB}\cdot\mathcal{C}$ such that $A \leq_m^{\mathrm{P}} B$ via function $f \in \mathrm{FP}$. Let r be a polynomial such that $|f(x)| = r(|x|)$ for every $x \in \Sigma^*$. We have to show that $A \in \mathrm{SB}\cdot\mathcal{C}$. Let $B \in \mathrm{SB}\cdot\mathcal{C}$ via some set $C \in \mathcal{C}$, polynomials p and q, and $\varepsilon > 0$ in the sense of Definition 4.1. Define a new set C' as

$$C' \stackrel{df}{=} \big\{\langle x, y\rangle : \langle f(x), y\rangle \in C \ \wedge\ |y| = q(r(|x|))\big\}.$$

$C' \leq_m^{\mathrm{P}} C$, hence $C' \in \mathcal{C}$. Let $q' = q(r)$. Since $\mathrm{count}_{C'}^{q'}(x) = \mathrm{count}_C^q(f(x))$, for all $x \in \Sigma^*$, $A \in \mathrm{SB}\cdot\mathcal{C}$. $\square$

A result similar to Lemma 5.1 holds for $\mathrm{U}\cdot$, $\exists\cdot$, and $\mathrm{BP}\cdot$. This means that for a complexity class $\mathcal{C}$ closed under $\leq_m^{\mathrm{P}}$, $\mathrm{U}\cdot\mathcal{C}$, $\exists\cdot\mathcal{C}$, and $\mathrm{BP}\cdot\mathcal{C}$ are all closed under $\leq_m^{\mathrm{P}}$. In fact, the proof of Lemma 5.1 states the closure under $\leq_m^{\mathrm{P}}$ for all complexity classes whose acceptance behavior depends only on the number of accepting paths of a computation. Unfortunately, it is not clear whether there are other reducibilities such that the closure of a complexity class $\mathcal{C}$ under such a reducibility entails the same closure for $\mathrm{SB}\cdot\mathcal{C}$. In particular we do not know whether SBP is closed under $\cap$, which impedes its closure under $\leq_c^{\mathrm{P}}$. For $\mathrm{U}\cdot$, $\exists\cdot$, and $\mathrm{BP}\cdot$ again, the statement of Lemma 5.1 is even true if we replace $\leq_m^{\mathrm{P}}$ by $\leq_c^{\mathrm{P}}$.

SBP is closed under $\cup$ [BGM03]. A related result can be shown for complexity classes that are definable by application of $\mathrm{SB}\cdot$.

Lemma 5.2 *If $\mathcal{C}$ is closed under $\leq_c^{\mathrm{P}}$ and $\leq_d^{\mathrm{P}}$, then $\mathrm{SB}\cdot\mathcal{C}$ is closed under $\leq_d^{\mathrm{P}}$.*

Proof: Let A be a set, $B \in \mathrm{SB}\cdot\mathcal{C}$ such that $A \leq_d^{\mathrm{P}} B$ via function $f \in \mathrm{FP}$. There are polynomials r and s such that for every $x \in \Sigma^*$, $f(x) = \langle x_1, \ldots, x_k \rangle$, $k = s(|x|)$, and $|x_1| = \ldots = |x_k| = r(|x|)$. We assume s to be non-decreasing. If A is finite, then A conjunctive reduces to some set in $\mathcal{C}$, and therefore is in $\mathcal{C}$ and $\mathrm{SB}\cdot\mathcal{C}$. Now, let A be infinite. If r is constant, f generates queries from a finite set of words. A conjunctive reduces to some set in $\mathcal{C}$ and therefore is in $\mathrm{SB}\cdot\mathcal{C}$. Let r be unbounded. Applying Lemma 4.3, let $B \in \mathrm{SB}\cdot\mathcal{C}$ via some set $C \in \mathcal{C}$ and polynomials p and q such that for every $x \in \Sigma^*$,

$$\begin{aligned} x \in B &\longrightarrow \mathrm{count}_C^q(x) \geq 2^{s(|x|)+2} \cdot 2^{p(|x|)} \quad \text{and} \\ x \notin B &\longrightarrow \mathrm{count}_C^q(x) \leq 2^{p(|x|)}. \end{aligned}$$

By Lemma 4.2, there is a natural number n_0 such that for all $n > n_0$, $p(n) < q(n)$. We define a new set C' as

$$\begin{aligned} C' \overset{df}{=} \Bigg\{ &\langle x, y_1 \cdot \ldots \cdot y_k \rangle : f(x) = \langle x_1, \ldots, x_k \rangle \;\wedge \\ &\bigvee_{i \in \{1,\ldots,k\}} \langle x_i, y_i \rangle \in C \;\wedge\; y_1, \ldots, y_k \in \Sigma^{q(r(|x|))} \Bigg\} \setminus \\ &\Bigg\{ \langle x, y \rangle : r(|x|) \leq n_0 \;\wedge\; x \notin A \;\wedge\; y \in \Sigma^{s(|x|)\cdot q(r(|x|))} \Bigg\}. \end{aligned}$$

If $x \in \Sigma^*$ is not contained in A and $r(|x|) \leq n_0$, then there is no y such that $\langle x, y \rangle \in C'$. Since the set of words $x \in \Sigma^*$ such that $x \notin A$ and $r(|x|) \leq n_0$ is finite, C' disjunctive reduces to some set in $\mathcal{C}$ and therefore is in $\mathcal{C}$. Let $q' = s \cdot q(r)$. For $x \in \Sigma^*$, we have to count the number of words y of length $q'(|x|)$ such that $\langle x, y \rangle \in C'$. Let $k \overset{df}{=} s(|x|)$ and $\ell \overset{df}{=} q(r(|x|))$. If $x \in A$, then there are at least $2^{s(r(|x|))+2} \cdot 2^{p(r(|x|))} \cdot 2^{(k-1)\cdot\ell}$ such words. If $x \notin A$, the number of these words is at most

$$2^{k\cdot\ell} - \left(2^{\ell} - 2^{p(r(|x|))}\right)^k = \sum_{i=1}^{k} -(-1)^i \cdot \binom{k}{i} \cdot 2^{i\cdot p(r(|x|))} \cdot 2^{(k-i)\cdot\ell}.$$

For every $x \in \Sigma^*$, $r(|x|) > n_0$, and every $i \in \{1, \ldots, k\}$ it holds that

$$2^{i\cdot p(r(|x|))} \cdot 2^{(k-i)\cdot\ell} \leq 2^{p(r(|x|))} \cdot 2^{(k-1)\cdot\ell}.$$

Replacing the term $-(-1)^i$ by 1 this shows that the sum is bounded above by $2^k \cdot 2^{p(r(|x|))} \cdot 2^{(k-1)\cdot\ell}$. Let $p' = s + 1 + p(r) + (s-1) \cdot q(r)$. Hence $A \in \mathrm{SB}\cdot\mathcal{C}$ via C', p', q', and some $\varepsilon \in (0, 1)$. $\square$

Let us reconsider the proof of Lemma 5.2. Instead of demanding closure of $\mathcal{C}$ under $\leq_c^{\mathrm{P}}$ and $\leq_d^{\mathrm{P}}$, we can loosen the requirements to closure of $\mathcal{C}$ under $\leq_{bc}^{\mathrm{P}}$ and $\leq_{bd}^{\mathrm{P}}$. Applying lemma 4.4, the rest of the proof would show that $\mathrm{SB}\cdot\mathcal{C}$ is closed under $\leq_{bd}^{\mathrm{P}}$, too. The class $\mathrm{BH}(\mathrm{NP})$, the Boolean closure of NP, is closed under bounded truth-table reducibility [KSW87]. So we can conclude that $\mathrm{SB}\cdot\mathrm{BH}(\mathrm{NP})$ is closed under $\leq_{bd}^{\mathrm{P}}$. Furthermore, Θ_2^{P}, the truth-table closure of NP, is closed under truth-table reducibility [Wag90]. Hence $\mathrm{SB}\cdot\Theta_2^{\mathrm{P}}$ is closed under $\leq_d^{\mathrm{P}}$.

Lemma 5.3 *If $\mathcal{C}$ is closed under $\leq_c^{\mathrm{P}}$, then $\exists\cdot\mathcal{C}$ is closed under $\leq_{maj}^{\mathrm{P}}$.*

Proof: Let A be a set that majority reduces to some set $B \in \exists\cdot\mathcal{C}$ via reduction function $f \in \mathrm{FP}$. There are polynomials s and r such that for every $x \in \Sigma^*$,

$$f(x) = \langle y_1, \ldots, y_k \rangle,\ k = s(|x|) \text{ and } |y_1| = \ldots = |y_k| = r(|x|).$$

Remember that for every n, $s(n)$ is odd. Let $B \in \exists\cdot\mathcal{C}$ via some set $C \in \mathcal{C}$ and polynomial p. Define a new set C' as

$$\begin{aligned} C' \stackrel{df}{=} \Big\{ \langle x, \varphi \cdot z_1 \cdot \ldots \cdot z_\ell \rangle \quad : \quad & f(x) = \langle y_1, \ldots, y_k \rangle \ \wedge\ k = s(|x|) \ \wedge\ \ell = \frac{k+1}{2} \ \wedge \\ & \bigwedge_{1 \leq i \leq \ell} \Big(\langle y_{\varphi(i)}, z_i \rangle \in C \ \wedge\ |z_i| = p(r(|x|)) \Big) \ \wedge \\ & \varphi : \{1, \ldots, \ell\} \to \{1, \ldots, k\} \text{ injective} \Big\}. \end{aligned}$$

For simplicity, we say that the representation of φ is of length $\lceil \log k \rceil \cdot \ell$. Obviously, C' conjunctive reduces to some set in $\mathcal{C}$. So it is contained in $\mathcal{C}$. Let $p' = \frac{1}{2} \cdot (s+1) \cdot \big(\lceil \log s \rceil + p(r) \big)$. We obtain $A \in \exists\cdot\mathcal{C}$ via C' and p'. $\Box$

Schöning proved the following amplification lemma.

Lemma 5.4 ([Sch89]) *Let $\mathcal{C}$ be a complexity class closed under $\leq_{maj}^{\mathrm{P}}$. For any set $A \in \mathrm{BP}\cdot\mathcal{C}$ and any polynomial r there is some set $B \in \mathcal{C}$ and a polynomial q such that for every $x \in \Sigma^*$,*

$$\left| \left\{ y \in \Sigma^{q(|x|)} : \langle x, y \rangle \in B \ \longleftrightarrow\ x \in A \right\} \right| > \left(1 - \frac{1}{2^{r(|x|)}} \right) \cdot 2^{q(|x|)}.$$

The following lemma is well-known. For the sake of completeness we include the proof.

Lemma 5.5 *If $\mathcal{C}$ is non-trivial and closed under $\leq_{maj}^{\mathrm{P}}$, then $\mathrm{BP}\cdot\mathcal{C}$ is closed under $\leq_{maj}^{\mathrm{P}}$.*

Proof: Let $B \in \mathrm{BP}\cdot\mathcal{C}$ and $A \leq_{maj}^{\mathrm{P}} B$. There are a function $f \in \mathrm{FP}$ and polynomials p and s such that for all $x \in \Sigma^*$

$$f(x) = \langle y_1, \ldots, y_{2p(|x|)+1} \rangle$$

and $|y_i| = s(|x|)$ for $i \in \{1, \ldots, 2p(|x|)+1\}$ and $x \in A$ if and only if

$$\#\{i : 1 \leq i \leq 2p(|x|) + 1 \text{ and } y_i \in B\} \geq p(|x|) + 1.$$

Let

$$F_x \stackrel{df}{=} \{y_i : 1 \leq i \leq 2p(|x|) + 1 \text{ and } f(x) = \langle y_1, \ldots, y_{2p(|x|)+1} \rangle\}.$$

By lemma 5.4, there is a set $C \in \mathcal{C}$ and a polynomial q such that the following holds for a polynomial r, $r \geq 3 + \log p(|x|)$, and all $x \in \Sigma^*$:

$$\begin{aligned} x \in B \quad &\longrightarrow \quad \mathrm{count}_C^q(x) > \left(1 - \frac{1}{2^{r(|x|)}} \right) \cdot 2^{q(|x|)} \\ x \notin B \quad &\longrightarrow \quad \mathrm{count}_C^q(x) < \frac{1}{2^{r(|x|)}} \cdot 2^{q(|x|)}. \end{aligned}$$

Now define a set C' as

$$C' \stackrel{df}{=} \{\langle x, z\rangle : z \in \Sigma^{q(s(|x|))} \text{ and } \#\{y : y \in F_x \text{ and } \langle y, z\rangle \in C\} \geq p(|x|) + 1\}.$$

C' majority reduces to C. Note that in the case that $yinB$ the probability for $\langle y, z\rangle \in C$ is very high and therefore the probability that for $y_{i1}, \ldots, y_{ik} \in B$ there is a single z with $\langle y_{ij}, z\rangle \in C$ for all $1 \leq j \leq k$ is still very high. This leads to the following argumentation, for which we fix $x \in \Sigma^*$, and let $n \stackrel{df}{=} |x|$. If $x \in A$, then $F_x \cap B \geq p(n) + 1$ and for each $y \in F_x \cap B$ it holds that $\mathrm{count}_C^q(y) > \left(1 - \frac{1}{2^{r(|y|))}}\right) \cdot 2^{q(|y|)}$. Therefore, we have

$$\begin{aligned} x \in A \longrightarrow \mathrm{count}_{C'}^{q(s)}(x) &> 2^{q(s(n))} - \frac{p(n)+1}{2^{r(s(n))}} 2^{q(s(n))} \\ &\geq \left(\frac{1}{2} + \frac{1}{4}\right) 2^{q(s(n))}. \end{aligned}$$

On the other hand, if $x \notin A$, then $F_x \cap B \leq p(|x|)$ and therefore,

$$x \notin A \longrightarrow \mathrm{count}_{C'}^{q(s)}(x) < \frac{p(n)+1}{2^{r(s(n))}} 2^{q(s(n))} \leq \left(\frac{1}{2} - \frac{1}{4}\right) \cdot 2^{q(s(n))}.$$

Hence, A is in $\mathrm{BP}{\cdot}\mathcal{C}$. □

6 Inclusion Properties of Classes with $\mathrm{SB}\cdot$

In this section, we want to investigate the power of $\mathrm{SB}\cdot$ in connection with the operators $\exists\cdot$, $\mathrm{BP}\cdot$, $\mathrm{U}\cdot$, and $\mathrm{SB}\cdot$ itself. We show that not all combinations of operators will lead to new complexity classes. In fact, $\mathrm{SB}\cdot$ is powerful enough to assimilate some operators to its left or right side. Before we start, we note that if a complexity class $\mathcal{C}$ is closed under some reducibility $\leq_a$, then C is also closed under each reducibility which is stronger than $\leq_a$.

We obtain our final results by a sequence of inclusion results. The following table gives an overview of these results.

$\mathcal{C}$ is closed under	result	obtained in
$\leq_c^{\mathrm{P}}$	$\mathcal{C} \subseteq \mathrm{BP}{\cdot}\mathcal{C} \subseteq \mathrm{BP}{\cdot}\mathrm{U}{\cdot}\mathcal{C} \subseteq \mathrm{SB}{\cdot}\mathcal{C} \subseteq \mathrm{BP}{\cdot}\exists{\cdot}\mathcal{C}$	6.2
$\leq_c^{\mathrm{P}}$	$\mathrm{SB}{\cdot}\exists{\cdot}\mathcal{C} = \mathrm{BP}{\cdot}\exists{\cdot}\mathcal{C}$	6.3
$\leq_c^{\mathrm{P}}$	$\exists{\cdot}\mathrm{SB}{\cdot}\mathcal{C} = \mathrm{SB}{\cdot}\mathcal{C}$	6.4
$\leq_c^{\mathrm{P}}$	$\exists{\cdot}\mathrm{BP}{\cdot}\mathcal{C} \subseteq \mathrm{SB}{\cdot}\mathcal{C}$	6.5
$\leq_c^{\mathrm{P}}$	$\mathrm{U}{\cdot}\mathrm{SB}{\cdot}\mathcal{C} = \mathrm{SB}{\cdot}\mathrm{U}{\cdot}\mathcal{C} = \mathrm{SB}{\cdot}\mathcal{C}$	6.6
$\leq_{maj}^{\mathrm{P}}$	$\mathrm{SB}{\cdot}\mathrm{BP}{\cdot}\mathcal{C} = \mathrm{SB}{\cdot}\mathcal{C}$	6.7
$\leq_c^{\mathrm{P}}$	$\mathrm{SB}{\cdot}\mathrm{SB}{\cdot}\ldots\mathrm{SB}{\cdot}\mathcal{C} = \mathrm{SB}{\cdot}\mathrm{SB}{\cdot}\mathcal{C} = \mathrm{BP}{\cdot}\exists{\cdot}\mathcal{C}$	6.8
$\leq_c^{\mathrm{P}}$	$\mathrm{BP}{\cdot}\mathrm{SB}{\cdot}\mathcal{C} = \mathrm{BP}{\cdot}\exists{\cdot}\mathcal{C}$	6.9

Before combining $\mathrm{SB}\cdot$ with other operators we first locate a complexity class $\mathrm{SB}{\cdot}\mathcal{C}$ with regard to other operators applied to $\mathcal{C}$. The special case of $\mathrm{SB}{\cdot}\mathcal{C} = \mathrm{SBP}$ is already known to lie between

BP·UP and AM [BGM03]. This result can be generalized. To prove this, we have to provide some additional definitions and two results from [Sip83]. A *linear hash function* $h : \Sigma^m \to \Sigma^k$ is given by a Boolean $k \times m$-matrix M. A word $x = x_1 \cdot \ldots \cdot x_m$ is mapped to $y = y_1 \cdot \ldots \cdot y_k$ if and only if $y = M \cdot x^T$ (the inner product modulo 2). For $X \subseteq \Sigma^m$ and a family H of k hash functions $h_1, \ldots, h_k$, the predicate $\mathrm{Collision}(X, H)$ is true if and only if

$$\bigvee_{x,y_1,\ldots,y_k \in X} \; \bigwedge_{i \in \{1,\ldots,k\}} \big(x \neq y_i \;\wedge\; h_i(x) = h_i(y_i)\big).$$

We say that X *has a collision with respect to* H. The set of all families of ℓ hash functions from Σ^m to Σ^k is denoted by $\mathcal{H}(\ell, m, k)$.

Theorem 6.1 ([Sip83]) *(i) Let $X \subseteq \Sigma^m$ be a set of at most 2^{k-1} elements. If we choose a hash family H uniformly at random from $\mathcal{H}(k, m, k)$, then the probability of a collision of X with respect to H is at most $\frac{1}{2}$.*

(ii) For any hash family $H \in \mathcal{H}(k, m, k)$ and any set $X \subseteq \Sigma^m$ of cardinality at least $k \cdot 2^k$, X must have a collision with respect to H.

Proposition 6.2 *If $\mathcal{C}$ is closed under $\leq_c^{\mathrm{P}}$, then $\mathcal{C} \subseteq \mathrm{BP}\cdot\mathcal{C} \subseteq \mathrm{BP}\cdot\mathrm{U}\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\mathcal{C} \subseteq \mathrm{BP}\cdot\exists\cdot\mathcal{C}$.*

Proof: The first inclusion is due to Lemma 2.7, the second inclusion is immediate by the monotony of BP· and Lemma 2.5. Let us look at the third one. Let $A \in \mathrm{BP}\cdot\mathrm{U}\cdot\mathcal{C}$ via some set $B \in \mathrm{U}\cdot\mathcal{C}$, a polynomial q, and $\varepsilon \in \big(0, \frac{1}{2}\big)$, and let $B \in \mathrm{U}\cdot\mathcal{C}$ via some set $C \in \mathcal{C}$ and a polynomial p. We define a new set C' as

$$C' \stackrel{\mathit{df}}{=} \big\{\langle x, y \cdot z\rangle : \langle\langle x, y\rangle, z\rangle \in C\big\}.$$

C' many-one reduces to C and therefore is in $\mathcal{C}$. $A \in \mathrm{SB}\cdot\mathcal{C}$ via C', $q - 1$, $p + q$, and ε.

Finally we consider the fourth inclusion. Let $A \in \mathrm{SB}\cdot\mathcal{C}$. By Lemma 4.3, there are $B \in \mathcal{C}$ and polynomials p and q such that for every $x \in \Sigma^*$,

$$\begin{array}{lcl} x \in A & \longrightarrow & \mathrm{count}_B^q(x) \geq 2^{|x|+1} \cdot 2^{p(|x|)} \quad \text{and} \\ x \notin A & \longrightarrow & \mathrm{count}_B^q(x) \leq 2^{p(|x|)}. \end{array}$$

Let $X_x \stackrel{\mathit{df}}{=} \big\{y \;:\; y = q(|x|) \;\wedge\; \langle x, y\rangle \in B\big\}$ for every $x \in \Sigma^*$. $|X_x| = \mathrm{count}_B^q(x)$. Define a new set D as

$$D \stackrel{\mathit{df}}{=} \big\{\langle x, H\rangle \;:\; x \in \Sigma^* \;\wedge\; k = p(|x|) + 1 \;\wedge\; H \in \mathcal{H}(k, q(|x|), k) \;\wedge\; \mathrm{Collision}(X_x, H)\big\}.$$

Note that $D \in \exists\cdot\mathcal{C}$ by the closure of $\mathcal{C}$ under $\leq_c^{\mathrm{P}}$. Let $x \in \Sigma^*$ such that $2^{|x|} \geq k$ and $k = p(|x|)+1$. If $x \notin A$, then $|X_x| \leq 2^{k-1}$. If $x \in A$, then $|X_x| \geq 2^{|x|+k}$. So we can apply Theorem 6.1 and obtain the following for a suitable polynomial r and all $x \in \Sigma^*$:

$$\begin{array}{lcl} x \in A & \longrightarrow & \mathrm{count}_D^r(x) \geq 2^{r(|x|)} \\ x \notin A & \longrightarrow & \mathrm{count}_D^r(x) \leq \dfrac{1}{2} \cdot 2^{r(|x|)}. \end{array}$$

To achieve the error-bound requirements of BP·, we use D' instead of D, where D' is defined as

$$D' = \big\{\langle x, H \cdot H'\rangle : \langle x, H\rangle, \langle x, H'\rangle \in D\big\}.$$

Since D' conjunctive reduces to D, $D' \in \mathcal{C}$. Let $r' = 2 \cdot r$. We obtain for every $x \in \Sigma^*$,

$$\begin{array}{lcl} x \in A & \longrightarrow & \mathrm{count}_{D'}^{r'}(x) \geq 2^{r'(|x|)} \quad \text{and} \\ x \notin A & \longrightarrow & \mathrm{count}_{D'}^{r'}(x) \leq \frac{1}{4} \cdot 2^{r'(|x|)}. \end{array}$$

It follows that $A \in \mathrm{BP}\cdot\exists\cdot\mathcal{C}$. □

If we let $\mathcal{C} = \mathrm{P}$, which is obviously closed under $\leq_c^{\mathrm{P}}$, we obtain inclusions for the class SBP that have already been shown in [BGM03]. This example illustrates, that we can always replace $\mathcal{C}$ by P to get a feeling for the statements. The following corollary, for example, states in case of $\mathcal{C} = \mathrm{P}$ that $\mathrm{SB}\cdot\mathrm{NP} = \mathrm{AM}$.

Corollary 6.3 *If $\mathcal{C}$ is non-trivial and closed under $\leq_c^{\mathrm{P}}$, then* $\mathrm{SB}\cdot\exists\cdot\mathcal{C} = \mathrm{BP}\cdot\exists\cdot\mathcal{C}$.

Proof: By Lemma 5.3, $\exists\cdot\mathcal{C}$ is closed under $\leq_c^{\mathrm{P}}$. Applying Proposition 6.2 to $\exists\cdot\mathcal{C}$, we obtain $\mathrm{BP}\cdot\exists\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\exists\cdot\mathcal{C} \subseteq \mathrm{BP}\cdot\exists\cdot\exists\cdot\mathcal{C}$. By Lemma 2.11, $\mathrm{BP}\cdot\exists\cdot\exists\cdot\mathcal{C} = \mathrm{BP}\cdot\exists\cdot\mathcal{C}$. □

This corollary is the starting point of an investigation dealing with the operators $\exists\cdot$, $\mathrm{BP}\cdot$, and $\mathrm{SB}\cdot$. Let Q be a word of length k over the alphabet $\{\exists\cdot, \mathrm{BP}\cdot, \mathrm{SB}\cdot\}$. The i-th letter of Q is denoted by Q_i. Let $\mathcal{C}$ be a complexity class. We define

$$Q\mathcal{C} = Q_1\big(\ldots Q_{k-1}(Q_k\mathcal{C})\ldots\big).$$

Can we determine, only by looking at the quantifier prefix Q, the shortest prefix Q' such that $Q\mathcal{C} = Q'\mathcal{C}$? A similar result is known in the context of Arthur-Merlin games. In fact, if $\mathcal{C} = \mathrm{P}$ and Q only contains the operators $\exists\cdot$ and $\mathrm{BP}\cdot$, $Q\mathrm{P}$ is an Arthur-Merlin class and is always contained in AM.

Proposition 6.4 *If $\mathcal{C}$ is non-trivial and closed under $\leq_c^{\mathrm{P}}$, then* $\exists\cdot\mathrm{SB}\cdot\mathcal{C} = \mathrm{SB}\cdot\mathcal{C}$.

Proof: By Lemmata 2.5 and 5.1, $\mathrm{SB}\cdot\mathcal{C} \subseteq \exists\cdot\mathrm{SB}\cdot\mathcal{C}$. Let $A \in \exists\cdot\mathrm{SB}\cdot\mathcal{C}$ via some set $B \in \mathrm{SB}\cdot\mathcal{C}$ and polynomial q_1. We apply Lemma 4.3 to B. There are some set $C \in \mathcal{C}$, polynomials p and q_2 such that for every $x \in \Sigma^*$,

$$\begin{array}{lcl} x \in A & \longrightarrow & \bigvee\limits_{\substack{z \\ |z|=q_1(|x|)}} \mathrm{count}_C^{q_2}(\langle x, z\rangle) \geq 2^{q_1(|x|)+2} \cdot 2^{p(|x|)} \quad \text{and} \\ x \notin A & \longrightarrow & \bigwedge\limits_{\substack{z \\ |z|=q_1(|x|)}} \mathrm{count}_C^{q_2}(\langle x, z\rangle) \leq 2^{p(|x|)}. \end{array}$$

Define a new set C' as

$$C' = \big\{\langle x, zy\rangle : \langle\langle x, z\rangle, y\rangle \in C \ \wedge\ |z| = q_1(|x|) \ \wedge\ |y| = q_2(|\langle x, z\rangle|)\big\}.$$

Obviously, C' many-one reduces to some set in $\mathcal{C}$. Let $r = q_1 + q_2\big(2\cdot(\mathrm{id} + q_1)\big)$. For every $x \notin A$, it holds that

$$\mathrm{count}_{C'}^{r}(x) \leq 2^{q_1(|x|)} \cdot 2^{p(|x|)},$$

since there are only $2^{q_1(|x|)}$ possible words of length $q_1(|x|)$. On the other hand, if $x \in A$, then

$$\mathrm{count}^{r}_{C'}(x) \geq 2^{q_1(|x|)+2} \cdot 2^{p(|x|)}.$$

Letting $p' = q_1 + p + 1$, we conclude that $A \in \mathrm{SB}\cdot\mathcal{C}$ via the set C', the polynomials p' and r, and any $\varepsilon \in (0,1)$. $\square$

The proof of Proposition 6.4 shows a slightly stronger result than stated. If $x \in A$, then we demand the existence of only one $y \in \Sigma^*$ of defined length such that the condition is true. For all the other words $y' \neq y$ of same length, we do not need to restrict the number of accepting paths of the computation deciding whether $\langle x, y'\rangle \in B$. The proof states that even this putatively slightly more powerful model is contained in $\mathrm{SB}\cdot\mathcal{C}$. In case of $\mathcal{C} = \mathrm{P}$, we would obtain a new class $\exists\cdot\mathrm{SBP}^*$ that may be more powerful than $\exists\cdot\mathrm{SBP}$, similar to the relation of $\exists\cdot\mathrm{BPP}$ to MA. Fenner *et al.* [FFKL93] showed that there is an oracle that separates $\exists\cdot\mathrm{BPP}$ and MA. The proof of Proposition 6.4 states that $\exists\cdot\mathrm{SBP}^* \subseteq \mathrm{SBP}$.

Corollary 6.5 *If $\mathcal{C}$ is non-trivial and closed under $\leq^{\mathrm{P}}_{c}$, then* $\exists\cdot\mathrm{BP}\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\mathcal{C}$.

Proof: By Lemma 4.2, monotony of $\exists\cdot$, and Proposition 6.4, it holds that $\exists\cdot\mathrm{BP}\cdot\mathcal{C} \subseteq \exists\cdot\mathrm{SB}\cdot\mathcal{C} = \mathrm{SB}\cdot\mathcal{C}$. $\square$

As a byproduct, we obtain the well-known inclusion $\exists\cdot\mathrm{BP}\cdot\mathcal{C} \subseteq \mathrm{BP}\cdot\exists\cdot\mathcal{C}$ for any non-trivial complexity class $\mathcal{C}$ that is closed under $\leq^{\mathrm{P}}_{c}$.

For the operator $\mathrm{U}\cdot$, we show that it cannot bring new power to $\mathrm{SB}\cdot$ at all, neither to its left nor its right hand side. The proof uses the fact that the number of accepting paths of a computation is bounded for sets in $\mathrm{U}\cdot\mathcal{C}$. Note that, if $\mathcal{C}$ is closed under $\leq^{\mathrm{P}}_{c}$, then $\mathrm{U}\cdot\mathcal{C}$ is closed under $\leq^{\mathrm{P}}_{c}$, too.

Proposition 6.6 *If $\mathcal{C}$ is closed under $\leq^{\mathrm{P}}_{c}$, then* $\mathrm{U}\cdot\mathrm{SB}\cdot\mathcal{C} = \mathrm{SB}\cdot\mathrm{U}\cdot\mathcal{C} = \mathrm{SB}\cdot\mathcal{C}$.

Proof: By Lemmata 5.1, 2.5, and Proposition 6.4, we obtain $\mathrm{U}\cdot\mathrm{SB}\cdot\mathcal{C} = \mathrm{SB}\cdot\mathcal{C}$. By Lemma 2.5 and the monotony of $\mathrm{SB}\cdot$, it holds that $\mathrm{SB}\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\mathrm{U}\cdot\mathcal{C}$.

Let $A \in \mathrm{SB}\cdot\mathrm{U}\cdot\mathcal{C}$; we show $A \in \mathrm{SB}\cdot\mathcal{C}$. By amplification, we obtain a set $B \in \mathrm{U}\cdot\mathcal{C}$ and polynomials p and q_1 such that for every $x \in \Sigma^*$,

$$\begin{aligned} x \in A &\quad\longrightarrow\quad \mathrm{count}^{q_1}_{B}(x) \geq 4 \cdot 2^{p(|x|)} \quad \text{and} \\ x \notin A &\quad\longrightarrow\quad \mathrm{count}^{q_1}_{B}(x) \leq 2^{p(|x|)}. \end{aligned}$$

Let $B \in \mathrm{U}\cdot\mathcal{C}$ via some set $C \in \mathcal{C}$ and a polynomial q_2. Define a new set C' as

$$C' \stackrel{df}{=} \big\{\langle x, y_1 \cdot y_2\rangle : |y_1| = q_1(|x|) \ \wedge\ |y_2| = q_2(|\langle x, y_1\rangle|) \ \wedge\ \langle\langle x, y_1\rangle, y_2\rangle \in C\big\}.$$

Let $q = q_1 + q_2\big(2 \cdot (\mathrm{id} + q_1)\big)$. We obtain

$$\begin{aligned} x \in A &\quad\longrightarrow\quad \mathrm{count}^{q}_{C'}(x) \geq 4 \cdot 2^{p(|x|)} \quad \text{and} \\ x \notin A &\quad\longrightarrow\quad \mathrm{count}^{q}_{C'}(x) \leq 2^{p(|x|)}. \end{aligned}$$

This shows $A \in \mathrm{SB}\cdot\mathcal{C}$ via C', $p+1$, q, and any $\varepsilon \in (0,1)$. $\square$

Proposition 6.7 *If $\mathcal{C}$ is closed under $\leq^{\mathrm{P}}_{maj}$, then* $\mathrm{SB}\cdot\mathrm{BP}\cdot\mathcal{C} = \mathrm{SB}\cdot\mathcal{C}$.

Proof: Obviously, $\mathrm{SB}\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\mathrm{BP}\cdot\mathcal{C}$. Let $A \in \mathrm{SB}\cdot\mathrm{BP}\cdot\mathcal{C}$. $\mathrm{BP}\cdot\mathcal{C}$ is closed under $\leq^{\mathrm{P}}_{maj}$ by Lemma 5.5, and we can apply Lemma 4.4 on A. So there are some set $B \in \mathrm{BP}\cdot\mathcal{C}$ and polynomials p and q such that for all $x \in \Sigma^*$,

$$\begin{aligned} x \in A \quad &\longrightarrow \quad \mathrm{count}^q_B(x) \geq 16 \cdot 2^{p(|x|)} \quad \text{and} \\ x \notin A \quad &\longrightarrow \quad \mathrm{count}^q_B(x) \leq 2^{p(|x|)}. \end{aligned}$$

By Lemma 4.2, we can assume $p(n) > 0$ for all $n \in \mathbb{N}$. Let t be some monotonic polynomial such that for every $n \in \mathbb{N}$, $t(n) > q(n)$ and $t(n) > p(n)$. By Lemma 5.4, there are some set $C \in \mathcal{C}$ and a polynomial q' such that for all $x, y \in \Sigma^*$, $|y| = q(|x|)$,

$$\begin{aligned} \langle x,y\rangle \in B \quad &\longrightarrow \quad \mathrm{count}^{q'}_C(\langle x,y\rangle) \geq \left(1 - \frac{1}{2^{t(|\langle x,y\rangle|)}}\right) \cdot 2^{q'(|\langle x,y\rangle|)} \quad \text{and} \\ \langle x,y\rangle \notin B \quad &\longrightarrow \quad \mathrm{count}^{q'}_C(\langle x,y\rangle) \leq \frac{1}{2^{t(|\langle x,y\rangle|)}} \cdot 2^{q'(|\langle x,y\rangle|)}. \end{aligned}$$

Define a new set C' as

$$C' \stackrel{df}{=} \big\{\langle x, y\cdot z\rangle : \langle\langle x,y\rangle, z\rangle \in C \ \wedge\ |y| = q(|x|) \ \wedge\ |z| = q'(|\langle x,y\rangle|)\big\}.$$

We have to count the number of accepting paths at the end of the computation for some input $x \in \Sigma^*$. Let $r \stackrel{df}{=} q'\big(2\cdot(\mathrm{id}+q)\big)$ and $s \stackrel{df}{=} q + r$ and $|x| \stackrel{df}{=} n$. We remark that in the following equations, both bounds are less tight than they would be if we used the precise values (we use $t(n)$ instead of $t(2\cdot(n+q(n)))$).

$$\begin{aligned} x \in A \quad \longrightarrow \quad \mathrm{count}^s_{C'}(x) &\geq 16 \cdot 2^{p(n)} \cdot \left(1 - \frac{1}{2^{t(n)}}\right) \cdot 2^{r(n)} \\ &> 16 \cdot \big(2^{p(n)} \cdot 2^{r(n)} - 2^{r(n)}\big) \\ &\geq 16 \cdot 2^{p(n)-1} \cdot 2^{r(n)} \\ &= 8 \cdot 2^{p(n)} \cdot 2^{r(n)} \\ &> 6 \cdot 2^{p(n)} \cdot 2^{r(n)} \\ &= \left(1 + \frac{1}{2}\right) \cdot 2^{p(n)+r(n)+2} \\ x \notin A \quad \longrightarrow \quad \mathrm{count}^s_{C'}(x) &\leq 2^{p(n)} \cdot 2^{r(n)} + \big(2^{q(n)} - 2^{p(n)}\big) \cdot \frac{1}{2^{t(n)}} \cdot 2^{r(n)} \\ &\leq 2^{p(n)} \cdot 2^{r(n)} + \frac{2^{q(n)+r(n)}}{2^{t(n)}} \\ &< 2^{p(n)} \cdot 2^{r(n)} + 2^{r(n)} \\ &\leq 2 \cdot 2^{p(n)} \cdot 2^{r(n)} \\ &= \left(1 - \frac{1}{2}\right) \cdot 2^{p(n)+r(n)+2} \end{aligned}$$

Hence, A is in $\mathrm{SB}\cdot\mathcal{C}$. $\square$

Corollary 6.8 *If $\mathcal{C}$ is non-trivial and closed under $\leq_c^{\mathrm{P}}$, then* $\mathrm{SB}\cdot \ldots \mathrm{SB}\cdot\mathcal{C} = \mathrm{SB}\cdot\mathrm{SB}\cdot\mathcal{C} = \mathrm{BP}\cdot\exists\cdot\mathcal{C}$.

Proof: We first prove the easy case $\mathrm{SB}\cdot\mathrm{SB}\cdot\mathcal{C} = \mathrm{BP}\cdot\exists\cdot\mathcal{C}$. We know the inclusions $\exists\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\mathcal{C} \subseteq \mathrm{BP}\cdot\exists\cdot\mathcal{C}$. We apply $\mathrm{SB}\cdot$ on every class, which preserves the inclusion structure by monotony: $\mathrm{SB}\cdot\exists\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\mathrm{SB}\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\mathrm{BP}\cdot\exists\cdot\mathcal{C}$. Remember that $\exists\cdot\mathcal{C}$ is closed under $\leq_{maj}^{\mathrm{P}}$ by Lemma 5.3. We observe

$$\mathrm{BP}\cdot\exists\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\exists\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\mathrm{SB}\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\mathrm{BP}\cdot\exists\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\exists\cdot\mathcal{C} \subseteq \mathrm{BP}\cdot\exists\cdot\mathcal{C}.$$

Now, let us look at $\mathrm{SB}\cdot\mathrm{SB}\cdot\mathrm{SB}\cdot\mathcal{C}$. We conclude $\mathrm{SB}\cdot\mathrm{SB}\cdot\mathrm{SB}\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\mathrm{BP}\cdot\exists\cdot\mathcal{C} \subseteq \mathrm{BP}\cdot\exists\cdot\mathcal{C}$. The proposition follows by induction. □

Proposition 6.9 *If $\mathcal{C}$ is closed under $\leq_c^{\mathrm{P}}$, then* $\mathrm{BP}\cdot\mathrm{SB}\cdot\mathcal{C} = \mathrm{BP}\cdot\exists\cdot\mathcal{C}$.

Proof: Since $\exists\cdot\mathcal{C}$ is closed under $\leq_{maj}^{\mathrm{P}}$ by Lemma 5.3, we observe $\mathrm{BP}\cdot\mathrm{BP}\cdot\exists\cdot\mathcal{C} = \mathrm{BP}\cdot\exists\cdot\mathcal{C}$ by Lemma 2.11. By Lemma 4.2 and Proposition 6.2, we know $\exists\cdot\mathcal{C} \subseteq \mathrm{SB}\cdot\mathcal{C} \subseteq \mathrm{BP}\cdot\exists\cdot\mathcal{C}$. Applying the $\mathrm{BP}\cdot$-operator on each class yields our result: $\mathrm{BP}\cdot\exists\cdot\mathcal{C} \subseteq \mathrm{BP}\cdot\mathrm{SB}\cdot\mathcal{C} \subseteq \mathrm{BP}\cdot\mathrm{BP}\cdot\exists\cdot\mathcal{C} = \mathrm{BP}\cdot\exists\cdot\mathcal{C}$. □

7 Collapse of a Hierarchy

In this final section we summarize the computational power of complexity classes that are defined by means of the four operators that have been used in this paper. We see that the computational power of any class defined by application of $\mathrm{U}\cdot$, $\exists\cdot$, $\mathrm{BP}\cdot$, and $\mathrm{SB}\cdot$ in any order and number does not exceed the simple combination $\mathrm{BP}\cdot\exists\cdot$.

Theorem 7.1 *Let $\mathcal{C}$ be a complexity class that is closed under $\leq_c^{\mathrm{P}}$. Let Q be a word over the alphabet $\{\exists\cdot, \mathrm{BP}\cdot, \mathrm{SB}\cdot\}$. If Q contains one of the four words* $\mathrm{BP}\cdot\exists\cdot$, $\mathrm{SB}\cdot\exists\cdot$, $\mathrm{BP}\cdot\mathrm{SB}\cdot$, *or* $\mathrm{SB}\cdot\mathrm{SB}\cdot$ *as a factor, then $Q\mathcal{C} = \mathrm{BP}\cdot\exists\cdot\mathcal{C}$.*

Proof: By results of the previous sections, it holds that $\mathrm{BP}\cdot\exists\cdot\mathcal{C} \subseteq Q\mathcal{C}$. If $\mathcal{C}$ is not closed under $\leq_{maj}^{\mathrm{P}}$, replace $\mathcal{C}$ by $\exists\cdot\mathcal{C}$. From right to left we replace every occurrence of $\mathrm{SB}\cdot$ by the subword $\mathrm{BP}\cdot\exists\cdot$, and each such replacement yields a superclass of the previous one. By Propositions 6.2, 6.4 and Lemma 4.2, we replace $\exists\cdot\mathrm{BP}\cdot$ by $\mathrm{BP}\cdot\exists\cdot$ at the rightmost occurrence and always obtain superclasses. Now, only $\mathrm{BP}\cdot$ and $\exists\cdot$ appear, and every $\exists\cdot$ appears to the right of every $\mathrm{BP}\cdot$. The rightmost letter is $\exists\cdot$. By Lemmata 5.3 and 2.11, we obtain that $Q\mathcal{C} \subseteq \mathrm{BP}\cdot\exists\cdot\mathcal{C}$, which concludes the proof. □

Corollary 7.2 *Let $\mathcal{C}$ be non-trivial and closed under $\leq_c^{\mathrm{P}}$. If Q is a word over the alphabet $\{\mathrm{U}\cdot, \exists\cdot, \mathrm{BP}\cdot, \mathrm{SB}\cdot\}$, then $Q\mathcal{C} \subseteq \mathrm{BP}\cdot\exists\cdot\mathcal{C}$.*

Proof: From right to left replace every occurrence of $\mathrm{U}\cdot$ by $\exists\cdot$, and we obtain a superclass $Q'\mathcal{C}$ of $Q\mathcal{C}$. Now, $\mathrm{BP}\cdot\exists\cdot Q'\mathcal{C}$ is a superclass of $Q'\mathcal{C}$ and contained in $\mathrm{BP}\cdot\exists\cdot\mathcal{C}$ by Theorem 7.1. □

References

[AKS02] M. Agrawal, N. Kayal, and N. Saxena. Primes is in P. Manuscript, 2002.

[AS86] K. Ambos-Spies. A note on complete problems for complexity classes. *Information Processing Letters*, 23:227–230, 1986.

[Bab85] L. Babai. Trading group theory for randomness. In *Proceedings 17th Symposium on Theory of Computing*, pages 421–429. ACM Press, 1985.

[BFFvM00] H. Buhrman, S. Fenner, L. Fortnow, and D. van Melkebeek. Optimal proof systems and sparse sets. In *Proceedings 17th Symposium on Theoretical Aspects of Computer Science*, volume 1770 of *Lecture Notes in Computer Science*, pages 407–418. Springer Verlag, 2000.

[BGM03] E. Böhler, C. Glaßer, and D. Meister. Error-bounded probabilistic computation between MA and AM. In *Proceedings 28th Mathematical Foundations of Computer Science*, 2003. To appear.

[FFKL93] S. Fenner, L. Fortnow, S. Kurtz, and L. Li. An oracle builder's toolkit. In *Proceedings 8th Structure in Complexity Theory*, pages 120–131, 1993.

[Gil72] J. Gill. *Probabilistic Turing Machines and Complexity of Computations*. PhD thesis, University of California Berkeley, 1972.

[Gil77] J. Gill. Computational complexity of probabilistic turing machines. *SIAM Journal on Computing*, 6:675–695, 1977.

[Gur83] Y. Gurevich. Algebras of feasible functions. In *Proceedings of the 24th Annual Symposium on Foundations of Computer Science*, pages 210–214. IEEE Computer Society Press, 1983.

[HH88] J. Hartmanis and L. A. Hemachandra. Complexity classes without machines: On complete languages for UP. *Theoretical Computer Science*, 58:129–142, 1988.

[HHT97] Y. Han, L. A. Hemaspaandra, and T. Thierauf. Threshold computation and cryptographic security. *SIAM Journal on Computing*, 26(1):59–78, 1997.

[HJV93] L. Hemaspaandra, S. Jain, and N. Vereshchagin. Banishing robust Turing completeness. *International Journal of Foundations of Computer Science*, 3-4:245–265, 1993.

[HY84] J. Hartmanis and Y. Yesha. Computation times of NP sets of different densities. *Theoretical Computer Science*, 34:17–32, 1984.

[KSW87] J. Köbler, U. Schöning, and K. W. Wagner. The difference and the truth-table hierarchies for NP. *RAIRO Inform. Théor.*, 21:419–435, 1987.

[Lau83] C. Lautemann. BPP and the polynomial hierarchy. *Information Processing Letters*, 17:215–217, 1983.

[LLS75] R. E. Ladner, N. A. Lynch, and A. L. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1:103–123, 1975.

[Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.

[Rab76] M. O. Rabin. Probabilistic algorithms. In J. Traub, editor, *Algorithms and Complexity: New Directions and Results*, pages 21–39. Academic Press, London, 1976.

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

[Sim75] J. Simon. *On Some Central Problems in Computational Complexity*. PhD thesis, Cornell University, 1975.

[Sip82] M. Sipser. On relativization and the existence of complete sets. In *Proceedings 9th ICALP*, volume 140 of *Lecture Notes in Computer Science*, pages 523–531. Springer Verlag, 1982.

[Sip83] M. Sipser. A complexity theoretic approach to randomness. In *Proceedings of the 15th Symposium on Theory of Computing*, pages 330–335, 1983.

[SS77] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. *SIAM Journal on Computing*, 6:84–85, 1977.

[Sto77] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.

[Tod91] S. Toda. PP is as hard as the polynomial time hierarchy. *SIAM Journal on Computing*, 20:865–877, 1991.

[Wag90] K. W. Wagner. Bounded query classes. *SIAM Journal on Computing*, 19:833–846, 1990.

[Wra77] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3:23–33, 1977.

[WW86] K. W. Wagner and G. Wechsung. *Computational Complexity*. VEB Verlag der Wissenschaften, Berlin, 1986.