Amer. Math. Soc. Transl.
(2) Vol. **00**, XXXX

Relativizability in Complexity Theory

Nikolai K. Vereshchagin

Abstract. Starting with the paper of Baker, Gill and Solovay [**5**] in complexity theory, many results have been proved which separate certain relativized complexity classes or show that they have no complete language. All results of this kind were, in fact, based on lower bounds for Boolean decision trees or circuits of a certain type, or for machines with polylogarithmic restrictions on time. The following question arises: Are these methods of proving "relativized" results universal? We propose a general framework in which assertions of universality of this kind may be formulated and proved as convenient criteria.

Using these criteria we obtain some new "relativized" results and new proofs of some known results. For example, for many of the complexity classes studied in the literature all relativizable inclusions between the classes are found.

Contents

Some notation.

$\mathbb{B}$	the set $\{0,1\}$.
$\mathbb{B}^*$	the set of all words over alphabet $\mathbb{B}$ (=binary words).
$\mathbb{B}^n$	the set of all binary words of length n.
$\mathbb{B}^{\leq n}$	the set of all binary words of length at most n.

1991 *Mathematics Subject Classification.* Primary 68Q15.

$\lvert x\rvert$	the length of the word x.
xy	the concatenation of words x and y.
$B \oplus C$	$= \{0u \mid u \in B\} \cup \{1v \mid v \in C\}$, where B, C are languages.
$\lvert M\rvert$	the number of elements in a set M.
$\mathbb{N}$	the set of natural numbers.
Ω	the set of functions form $\mathbb{B}^*$ to $\mathbb{B}$.
$\mathrm{Dom}\, f$	the domain of function f.
$f\lvert M$	the restriction of function f to set M.
$\mathrm{Prob}[M]$	the probability of event M.
$\leq_m^p$	polynomial-time m-reducibility (Karp reducibility).
$\leq_T^p$	polynomial-time Turing reducibility (Cook reducibility).

1. Introduction

Most theorems in recursion theory are known to be relativizable. This means that for any language A, a theorem remains true if we take machines supplied with oracle A as the model of computation. This is not true in complexity theory. In 1975 in the paper [**5**], oracles A and B were constructed such that $\mathrm{P}^A \neq \mathrm{NP}^A$ and $\mathrm{P}^B = \mathrm{NP}^B$. This means that although we do not know which of the two assertions P = NP and P ≠ NP is true, neither of them is relativizable. After [**5**], many theorems of the following kind have been proved (for pairs of specific complexity classes K_1, K_2): there are oracles A and B such that $K_1^A \neq K_2^A$ and $K_1^B = K_2^B$. Since many interesting complexity classes lie between P and PSPACE, for such classes one can always take the oracle B constructed in [**5**] as the second oracle because in fact $\mathrm{P}^B = \mathrm{PSPACE}^B$ for that oracle. In 1989 the first non-relativizable theorem in complexity theory appeared. In [**36**] it was shown that PH ⊆ IP. Earlier, in [**18**], it was proved that $\exists A$ co-$\mathrm{NP}^A \not\subseteq \mathrm{IP}^A$. However only a few non-relativizable results in complexity theory are known.

All known proofs of results of the form $\exists A\ K_1^A \neq K_2^A$ (that is, $\exists A\ K_1^A \not\subseteq K_2^A$ or the converse) consist of two parts: the "diagonal" part (constructing an oracle step by step), which is the same in all proofs, and the specific "combinatorial" part, in which it is proved that every step can be made. Our first result is a formalization of this statement. The proof of Theorem 1 in Section 3.1 is a general formulation of the diagonal part of such proofs. Corollary 3 shows what combinatorial assertion is to be proved in every specific case.

Theorems of the following form have also appeared in the literature: There exists an oracle A for which the class K^A has no Karp complete (or Cook complete) language. (For the definition of Karp and Cook reductions see Sections 3.2 and 3.3.) For example, in [**47**], it is proved that there is an oracle A for which the class $\mathrm{NP}^A \cap$ co-NP^A has no Karp complete language (more precisely, no language complete under polynomial many-one reductions relative to A), and there is an oracle A for which the class R^A has no Karp complete language.

All we have said about proofs of theorems of the form $\exists A\ K_1^A \not\subseteq K_2^A$ is true for proofs of non-existence of complete languages in complexity classes. Theorem 4 in Section 3.2 provides the diagonal part of such proofs in a general form.

Both Theorem 1 and Theorem 4 give the criteria. Theorem 1 is a criterion of whether

$$\forall A\ K_1^A \subseteq K_2^A, \tag{1}$$

while Theorem 4 is a criterion of whether

$$\forall A\ (K_2^A \text{ has a Karp complete problem for the class } K_1^A). \tag{2}$$

Roughly speaking, the criteria are as follows. Let K be a complexity class. Let us replace all polynomial restrictions in the definition of the class K by polylogarithmic ones and replace decision problems (i.e. languages) by separation problems. Let KLOGS denote the resulting "counterpart" of the class K. Then the assertion (1) is equivalent to the absolute inclusion $K_1\mathrm{LOGS} \subseteq K_2\mathrm{LOGS}$, and the assertion (2) is true iff the class K_2LOGS has a language complete for the class K_1LOGS. Analysis of proofs of relativizable assertions of the form (1) (for example, BPP $\subseteq \Sigma_2 \cap \Pi_2$ from [**48**]) shows that the more natural formulations of such assertions have the form $K_1\mathrm{LOGS} \subseteq K_2\mathrm{LOGS}$.

To formulate these criteria in a rigorous form we present in Section 2 a uniform way to define complexity classes. The same way was proposed independently in [**10**].

Similar criteria exist also for theorems of the following two forms:

(3) $\forall A$ (the class K_2^A has a Cook complete language for the class K_1^A)

and

(4) $$\forall A\ (\forall L_1 \in K_1^A\ \exists L_2 \in K_2^A :\ L_1 \text{ is Cook reducible to } L_2),$$

that is, "K_1^A is Cook reducible to K_2^A." These criteria are formulated in Sections 3.3 and 3.4. The obtained criteria make easier to solve problems of the forms (1)–(4) both psychologically and technically. In Sections 4, 5 and 6 we ascertain, for several known classes K_1, K_2 between P and PSPACE, which of the two assertions—(1) or the negation of (1)—is true or is unknown. We do the same thing also for assertions of the form (2), (3) and (4). Some new positive and negative results of this type are proved. To obtain one of them, namely, to build an oracle under which AM is not included in PP, we prove a new lower bound for perceptron complexity (Section 7), which is interesting in its own right. Some problems of this kind remain open.

In Section 8 we present more involved oracle constructions, namely we build oracles under which some classes coincide and some other classes do not coincide (say an oracle under which $\mathrm{P} = \mathrm{R} \neq \mathrm{BPP}$). The hardest of the results obtained is the existence of an oracle A such that $\mathrm{P}^A \neq \mathrm{NP}^A$ but both co-NP^A-sets and NP^A-sets are P^A-separable and in addition $\mathrm{P}^A = \mathrm{BPP}^A$. All those constructions are done using the same method. We formalize the method and exhibit two theorems that cannot be proven by this method.

As the relationships between relativized complexity classes depend on the oracle, it is natural to ask what happens for a "typical" oracle. A possible refinement of the notion of typicalness is randomness with respect to the uniform measure. In Section 9, we study the relationships between the classes NP^A, co-NP^A and P^A for a random oracle A. More precisely, we say that assertion $S(A)$ holds for a random A, or for almost all A, if the uniform measure of the set $\{A \mid S(A)\}$ is equal to 1. We consider only properties $S(A)$ satisfying the following two conditions: the set $\{A \mid S(A)\}$ is measurable and $S(A)$ is stable with respect to any changing the values of A on a finite number of arguments. By the 0-1-law of A. N. Kolmogorov, either $S(A)$ holds for a random A, or $\neg S(A)$ holds for a random A in this case.

2. A uniform way to define complexity classes

Consider the definitions of two popular complexity classes, NP and BPP, in a convenient form.

Definition 1. $L \in \mathrm{NP}$ if there exists a polynomial-time function $s : \mathbb{B}^* \to \mathbb{N}$ and a polynomial-time predicate $P(x, i)$ such that $x \in L \Leftrightarrow \exists i \leq s(x)\, P(x, i)$,

Definition 2. $L \in \mathrm{BPP}$ if there exists a polynomial-time function $s : \mathbb{B}^* \to \mathbb{N}$ and a polynomial-time predicate $P(x, i)$ such that for any $x \in L$ the ratio $\frac{\left|\{i \in \mathbb{N} | 1 \leq i \leq (x), P(x,i)\}\right|}{s(x)}$ is greater than $2/3$ and for any $x \notin L$ this ratio is less than $1/3$.

Let $f(x)$ denote in both definitions the sequence of values of the predicate $P(x, i)$ for $i \leq s(x)$. Then the membership of x in L is defined in terms of the word

$f(x)$. Any bit of the word $f(x)$ can be computed in time polynomial in $|x|$ given its number. Now we come to the following definition.

DEFINITION 3. A function f is bit-computable in time t if

1. the function $x \mapsto |f(x)|$ is computable in time $t(|x|)$, [1]
2. the partial binary predicate $P(x,i) = (i\text{th bit of the word } f(x))$ can be computed by a machine M that works in time $t(|x|)$ on all $x \in \mathbb{B}^*$ and all $i \le |f(x)|$.

Functions that are bit-computable in time $\mathrm{poly}(n)$ (where $\mathrm{poly}(n)$ is a polynomial) are called *polynomial-time bit-computable*. For example, the function $f(x) = 0^{2^{|x|}}$ is polynomial-time bit-computable.

A *separation problem* is a function from the set $\mathbb{B}^*$ into the set $\{0,1,*\}$. The meaning of this definition is that we have to separate the set $\{x \mid F(x) = 1\}$ from the set $\{x \mid F(x) = 0\}$. We identify a language $L \subseteq \mathbb{B}^*$ with its characteristic function, denoted by the same letter:

$$L(x) = \begin{cases} 1 & \text{if } x \in L; \\ 0 & \text{if } x \notin L. \end{cases}$$

Thus any language can be considered as a separation problem. Let us define a partial ordering on the set $\{0,1,*\}$ assuming that $* < 0$, $* < 1$. A separation problem P_1 is *easier* than a separation problem P_2 ($P_1 \le P_2$ in symbols) if $P_1(x) \le P_2(x)$ for all $x \in \mathbb{B}^*$. In other words, $P_2(x) = P_1(x)$ for all x such that $P_1(x) \ne *$. If L is a language, P a separation problem, and $P \le L$ then we say that L is a *solution to* P.

Both Definitions 1 and 2 have the following form. For a fixed separation problem F we declare that a language L is in the class if there exists a polynomial-time bit-computable function f such that $L(x) = F(f(x))$ for all $x \in \mathbb{B}^*$. Let $\mathrm{POLY}(F)$ denote the class defined in this way by means of a separation problem F. We say that a class K is *represented by a separation problem* F if $K = \mathrm{POLY}(F)$. For example, the class NP is represented by the following separation problem:

$$F_{\mathrm{NP}}(\alpha) = \begin{cases} 1 & \text{if } \exists i \le |\alpha|\ \alpha(i) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

To represent the class BPP we can take as F the separation problem

$$F_{\mathrm{BPP}}(\alpha) = \begin{cases} 1 & \text{if } \#_1(\alpha) > \frac{2}{3}|\alpha|, \\ 0 & \text{if } \#_1(\alpha) < \frac{1}{3}|\alpha|, \\ * & \text{otherwise,} \end{cases}$$

where $\#_1(x)$ denotes the number of 1's in the binary word x. In Section 4, we shall give the definitions in the form $\mathrm{POLY}(F)$ of the following classes: R, UP, FewP, Few, Σ_k, $\oplus$P, PP, PSPACE, MA, AM, IP. Let us now give such a definition for the simplest class, P. Recall that a language L belongs to the class P if there is a polynomial-time Turing machine M that recognizes L. Let $F_{\mathrm{P}}(\alpha) = (\text{the first bit of } \alpha)$. It is easy to see that $\mathrm{P} = \mathrm{POLY}(F_{\mathrm{P}})$.

[1] Convention: we assume that natural numbers are represented in binary. Moreover, we identify natural numbers and binary words: a natural number n is identified with the binary notation of the number $n+1$ without the leading 1.

The definition of the class $\mathrm{POLY}(F)$ easily relativizes as follows. An *oracle* is any language. An *oracle machine* is a Turing machine having an extra tape called *oracle tape*; this tape has a read/write head. That head can write only zeros and ones. To run an oracle machine on an input we must supply it with an oracle. Let A be an oracle. Then the machine works as a usual two-tape Turing machine with one exception. If the oracle machine gets into a certain state, then the word u written on the oracle tape (starting from the first cell up to the cell where the head is now) is considered as a question to the oracle. In this case oracle provides its answer $A(u)$ in the cell viewed by the head. The time needed for the oracle to provide its answer is assumed to be 1. Let M be an oracle machine and A an oracle. Then $M^A(x)$ denotes the output produced by M with oracle A on input x, and $t_{M^A}(x)$ the running time necessary to provide this output. Call an oracle machine M *polynomial* [*exponential*] if there exists a polynomial $q(n)$ [a constant c] such that $t_{M^A}(x) \le q(|x|)$ [$t_{M^A}(x) \le 2^{c|x|+c}$] for all $x \in \mathbb{B}^*$ and all $A \subseteq \mathbb{B}^*$. A function f is called polynomial [exponential] *relative to* A, if there exists a polynomial [exponential] oracle machine M such that $f(x) = M^A(x)$ for all x (that is, M^A computes f).

We want now to define the notion of bit-computability relative to an oracle. To this end we just allow machine M in the definition of bit-computability to query oracle A and we allow the function $|f(x)|$ to be computable in time $t(|x|)$ by a machine with oracle A.

Definition 4. $\mathrm{POLY}^A(F)$ is the class of languages L such that $L(x) = F(f(x))$ for all $x \in \mathbb{B}^*$ for some function f that is polynomial-time bit-computable relative to A.

Let $\log n$ denote $\lceil \log_2(n+1) \rceil$. Functions of the form $p(\log n)$, where p is a polynomial, will be called polylogarithms. The expression $\mathrm{polylog}(n)$ will denote a polylogarithm. We shall study Turing machines whose running time is bounded by a polylogarithm in the length of the input. An ordinary Turing machine in polylog-time can read only a prefix of the input word having polylogarithmic length. We shall use, therefore, a model of Turing machines which is commonly used when time restrictions are so stringent. The input word is given as an oracle in this model. More specifically, besides the work tape, the machine has an additional tape called the input tape, on which at the beginning of a computation the length of the input word x is written. The machine may at any moment of a computation ask a question of the form '$x(i) =?$', that is, it can write down on the input tape the number $i \le |x|$ and then receive the ith symbol of x, $x(i)$, written on the input tape. The time to write down i is added to the total time, but then the "oracle" supplies immediately $x(i)$. (We could consider another model in which the machine does not obtain the length of the input word, and when it asks '$x(i) =?$' with $i > |x|$ it receives the answer "undefined"; evidently, every machine working in time $t(|x|)$ can by simulated by a machine of this new type in time $t(|x|) + (\log(|x|))^{O(1)}$.) If time restrictions are polynomial, then our model is equivalent to ordinary Turing machines.

Functions that are bit-computable in time $\mathrm{polylog}(n)$ [$2^{O(n)}$] are called *polylog-time bit-computable* [*exponential-time bit-computable*, respectively]. For example, the function $f(x) = x$ is polylog-time bit-computable.

Note that if both f, g are polylog-time bit-computable then so is their superposition $f(g(x))$. If f is polylog-time bit-computable and g is polynomial-time

bit-computable then $f(g(x))$ is polynomial-time bit-computable. Similarly, if f is polylog-time bit-computable and g is exponential-time bit-computable then $f(g(x))$ is exponential-time bit-computable.

DEFINITION 5. A separation problem G polylog-time bit-reduces to a separation problem F, $G \preceq^l_m F$, if $G(\alpha) \leq F(f(\alpha))$ for some polylog-time bit-computable function f and for all α. $\mathrm{LOGS}(F)$ is the class of all separation problems G that polylog-time bit-reduce to G. $\mathrm{LOG}(F)$ is the class of all the languages in $\mathrm{LOGS}(F)$.

It is easy to see that the relation $\preceq^l_m$ is reflexive and transitive.

The class $\mathrm{LOGS}(F)$ is called the *polylog counterpart* of the class $\mathrm{POLY}(F)$. More precisely, the separation problem F defines a pair—the class $\mathrm{POLY}(F)$ and its polylog counterpart $\mathrm{LOGS}(F)$ (as we shall see later, the class $\mathrm{LOGS}(F)$ is not uniquely determined by the class $\mathrm{POLY}(F)$). If K denotes a complexity class, then $K\mathrm{LOGS}$ will denote the polylog counterpart of this class, for example, $\mathrm{PLOGS} =_{\mathrm{def}} \mathrm{LOGS}(F_{\mathrm{P}})$, $\mathrm{NPLOGS} =_{\mathrm{def}} \mathrm{LOGS}(F_{\mathrm{NP}})$ and $\mathrm{BPPLOGS} =_{\mathrm{def}} \mathrm{LOGS}(F_{\mathrm{BPP}})$.

All the three classes PLOGS, NPLOGS and BPPLOGS) can defined in a standard vein using polylog-time machines. Let $\mathrm{D}(F)$ denote the set $\{x \in \mathbb{B}^* \mid F(x) \neq *\}$. It is easy to verify that the following assertions are true.

$F \in \mathrm{PLOGS}$ iff there is a deterministic polylog-time Turing machine M such that $M(\alpha) = F(\alpha)$ for all $\alpha \in \mathrm{D}(F)$.

$F \in \mathrm{NPLOGS}$ iff there is a polylog-time nondeterministic machine M such that if $F(\alpha) = 1$, then M accepts α, and if $F(\alpha) = 0$, then M rejects α. By a polylog-time nondeterministic machine we mean a nondeterministic Turing machine all of whose computations on input α have no more than $\mathrm{polylog}(|\alpha|)$ steps.

$F \in \mathrm{BPPLOGS}$ iff there is a polylog-time probabilistic machine M such that if $F(\alpha) = 1$ then $\mathrm{Prob}[M(\alpha) = 1] > 2/3$, and if $F(\alpha) = 0$ then $\mathrm{Prob}[M(\alpha) = 1] < 1/3$ (if $F(\alpha) = *$, then this probability can be arbitrary). By a probabilistic polylog-time machine we mean a probabilistic Turing machine M whose computation time on input α is bounded by $\mathrm{polylog}(|\alpha|)$ (for all outcomes of coin tossing).

3. General criteria

3.1. A criterion of relativizable inclusion. We prove in this section that a complexity class K_1^A is included in a complexity class K_2^A for all oracles A if and only if the (absolute: no oracles) inclusion between their polylog-counterparts holds. This is true for all classes of the form $\mathrm{POLY}(F)$ whenever the separation problem F is nondegenerate in the following sense:

(5) there exists a polynomial-time bit-computable function $f : \mathbb{N} \to \mathbb{B}^*$ such that $|f(n)| = n$ and $F(f(n)) \neq *$ for all $n \in \mathbb{N}$;

(6) there are two words (denote them $zero_F$ and one_F) such that $F(zero_F) = 0$, $F(one_F) = 1$.

All the problems defining the complexity classes mentioned above are nondegenerate.

THEOREM 1 ([**10, 51**]). *Assume that a separation problem F satisfies the condition* (5) *and a separation problem G satisfies the condition* (6). *Then the following*

conditions are equivalent:

(7) $\mathrm{LOGS}(F) \subseteq \mathrm{LOGS}(G)$,

(8) $F \in \mathrm{LOGS}(G)$, *and*

(9) $\mathrm{POLY}^A(F) \subseteq \mathrm{POLY}^A(G)$ *for all* A.

If F is a language (that is, $\mathrm{D}(F) = \mathbb{B}^*$), then all these conditions are equivalent to the following condition:

(10) $\mathrm{LOG}(F) \subseteq \mathrm{LOG}(G)$.

Proof. Obviously, (7) implies (8). Let us prove that (8) implies (7). Assume that F belongs to the class $\mathrm{LOGS}(G)$ and let g be a polylog-time bit-computable function such that $F(\alpha) \le G(g(\alpha))$. Let us prove that $\mathrm{LOGS}(F) \subseteq \mathrm{LOGS}(G)$. Assume that H is in $\mathrm{LOGS}(F)$ and that f is a polylog-time bit-computable function such that $H(\alpha) \le F(f(\alpha))$. Then $H(\alpha) \le G(g(f(\alpha)))$ for all $\alpha \in \mathbb{B}^*$. Since $g(f(\alpha))$ is polylog-time bit-computable, it follows that H belongs to $\mathrm{LOGS}(G)$.

Obviously, (7) implies (10), and if F is a language, then (10) implies (8).

Let us prove that (8) implies (9). Assume that f is a polylog-time bit-computable function such that $F(\alpha) \le G(f(\alpha))$. Assume that $L \in \mathrm{POLY}^A(F)$, that is, there is a function g that is polynomial-time bit-computable relative to A such that $L(x) = F(g(x))$. Consequently, $L(x) = G(f(g(x)))$. As the function $f(g(x))$ is polynomial-time bit-computable relative to A it follows, that $L \in \mathrm{POLY}^A(G)$.

Let us prove that if (8) is not true, then neither is (9). Assume that F is not in $\mathrm{LOGS}(G)$. This means that for any separation problem $H \in \mathrm{LOGS}(G)$, there is an $\alpha \in \mathbb{B}^*$ such that $F(\alpha) \not\le H(\alpha)$. We claim that in this case for any separation problem $H \in \mathrm{LOGS}(G)$, there are infinitely many $\alpha \in \mathbb{B}^*$ such that $F(\alpha) \not\le H(\alpha)$. Indeed, assume that it is not true, that is, there are a number n and a polylog-time bit-computable function f such that $F(\alpha) \le G(f(\alpha))$ for all $\alpha \in \mathbb{B}^*$ of length greater than n. Then the function

$$f_1(\alpha) = \begin{cases} f(\alpha) & \text{if } |\alpha| > n, \\ \mathit{zero}_G & \text{if } |\alpha| \le n,\ F(\alpha) = 0, \\ \mathit{one}_G & \text{otherwise.} \end{cases}$$

is polylog-time bit-computable and $F(\alpha) \le G(f_1(\alpha))$ for all $\alpha \in \mathbb{B}^*$.

Let us choose a function encoding pairs of words by words. Assume that x is in $\mathbb{B}^*$. Let us double all the bits of x and add the word "01" to the end of the resulting word. Let $\bar{x}$ stand for the resulting word (for example, $\overline{001} = 00001101$). The word $\bar{x}y$ will be considered as the code of the pair $\langle x, y\rangle$. Obviously, for a given $\bar{x}y$ we can find x and y in polynomial time and for a given word u we can decide in polynomial time whether u has the form $\bar{x}y$. For an oracle A and for an $n \in \mathbb{N}$, let $[A]_n$ denote the word of length n, whose ith bit is equal to $A(\bar{n}i)$.[2]

We shall construct an oracle A such that the language $L^A = \{n \mid F([A]_n) = 1\}$ belongs to the set $\mathrm{POLY}^A(F) \setminus \mathrm{POLY}^A(G)$. The assertion $L^A \in \mathrm{POLY}^A(F)$ will follow from the following global assertion:

(11) $\forall n \in \mathbb{N}\ F([A]_n) \ne *$.

[2]Recall that we identify natural numbers with binary words.

If (11) is true then $L^A(n) = F([A]_n)$ for all n. Since the function $h(n) = [A]_n$ is polynomial-time bit-computable relative to A, the assertion (11) implies that the language L^A is in $\mathrm{POLY}^A(F)$.

Let us enumerate all functions that are polynomial-time bit-computable relative to oracles. This means that we enumerate pairs of oracle machines involved in the definition of polynomial bit-computability relative to an oracle. Let $f_i^A(x)$ denote the ith function (A is considered as the second argument of the function). Choose a polynomial-time decidable language E such that $F(E_n) \neq *$ for all $n \in \mathbb{N}$. Such a language exists because F satisfies the condition (5). We start with $A = E$ to satisfy condition (11). Then we make a countable number of steps. On the ith step we change the value of A on a finite number of words to satisfy the following local condition

$$\exists n \in \mathbb{N}\ F([A]_n) \neq G(f_i^A(n)), \tag{12}$$

taking care that condition (11) remains full force. Then we *freeze* all the values of A needed to ensure the truth of the assertion (12) and also all the values of A that were changed. This is to be understood as follows. There is a finite set U of words such that (12) is true for A' whenever A' has the same values as A on all the elements of U. We find such a U and "label" all its elements and all the elements on which the values A were changed. The values of A on labeled words are called "frozen" and cannot be changed later. After an infinite number of steps, we shall obtain an oracle A such that both (11) and (12) are true for all $i \in \mathbb{N}$. This implies that $L^A \in \mathrm{POLY}^A(F) \setminus \mathrm{POLY}^A(G)$.

Now we describe the ith step. Let A be the oracle constructed on $(i-1)$st step (with some frozen values). For an $\alpha \in \mathbb{B}^n$, let $A[\alpha]$ stand for the oracle A with $[A]_n$ replaced by α, that is,

$$A[\alpha](u) = \begin{cases} A(u) & \text{if } u \text{ has not the form } \bar{n}i,\ i \le n, \\ \alpha(i) & \text{if } u = \bar{n}i, \text{ where } i \le n. \end{cases}$$

Let $H(\alpha) = G(f_i^{A[\alpha]}(|\alpha|))$.

Since A is polynomial-time decidable (A is obtained from E by finite number of changes), the function $\alpha \mapsto f_i^{A[\alpha]}(|\alpha|)$ is polylog-time bit-computable. It follows that $H \in \mathrm{LOGS}(G)$. Consequently, there are infinitely many $\alpha \in \mathbb{B}^*$ such that $F(\alpha) \not\leq H(\alpha)$. We conclude that there is an $\alpha \in \mathbb{B}^*$ such that $F(\alpha) \not\leq H(\alpha)$ and no value of A on a word of the form $\overline{|\alpha|}i$, $i \le |\alpha|$, is frozen. Choose such an α and replace A by $A[\alpha]$. Now the assertion (12) is true for $n = |\alpha|$ because $F([A]_n) = F(\alpha) \not\leq H(\alpha) = G(f_i^A(n))$.

Freeze a finite number of A's values ensuring the truth of condition (12). Note that the assertion (11) is not violated because $F([A]_n) = F(\alpha) \neq *$ (since $F(\alpha) \not\leq H(\alpha)$ and $*$ is the least element in the set $\{0, 1, *\}$). The implication (9)⇒(8) is proved. □

Remark 1. All complexity classes studied in the literature are represented by separation problems with the following property. When we add in the definition of the class $\mathrm{POLY}(F)$ the requirement $|f(x)| = 2^{\mathrm{poly}(|x|)}$ (the definition of polynomial bit-computability implies only that $|f(x)| \le 2^{\mathrm{poly}(|x|)}$), the class $\mathrm{POLY}(F)$ does not change. Moreover, the separation problems representing the known complexity

classes have the following property:

$$F \in \mathrm{LOGS}(\bar{F}), \tag{13}$$

where

$$\bar{F}(\alpha) = \begin{cases} F(\alpha) & \text{if } |\alpha| \text{ has the form } 2^k,\ k \in \mathbb{N}, \\ 0 & \text{otherwise.} \end{cases}$$

Note that (13) implies that $\mathrm{POLY}^A(\bar{F}) = \mathrm{POLY}^A(F)$ for all A (by Theorem 1).

If a separation problem F has property (13), then conditions (7), (8), and (9) are equivalent to the condition

$$\mathrm{EXP}^A(F) \subseteq \mathrm{EXP}^A(G) \text{ for all } A, \tag{14}$$

where $\mathrm{EXP}^A(H)$ is the class containing all the languages L such that $L(x) = H(g(x))$ for some function g that is exponential-time bit-computable relative to A.

Indeed, the implication (8)⇒(14) is true because if $f(\alpha)$ is polylog-time bit-computable and $g(x)$ is a exponential-time bit-computable relative to A, then the function $f(g(x))$ is exponential-time bit-computable relative to A (because $\mathrm{polylog}(2^{2^{O(n)}}) = \mathrm{poly}(2^{O(n)}) = 2^{O(n)}$). Conversely, let us prove the implication (14)⇒(8). Assume that F has the property (13) and assume that (8) is false. Then we see that $\bar{F} \notin \mathrm{LOGS}(G)$. Applying the same arguments as those in the proof of implication ¬(8)⇒ ¬(9), we can construct an oracle A such that the language $L^A = \{n \mid F([A]_{2^n}) = 1\}$ is in $\mathrm{EXP}^A(F) \setminus \mathrm{EXP}^A(G)$. □

For a family $\mathcal{F}$ of separation problems, let $\mathrm{POLY}^A(\mathcal{F}) = \bigcup_{F\in\mathcal{F}} \mathrm{POLY}^A(F)$. It is easy to see that for countable families $\mathcal{F}$, Theorem 1 generalizes to classes of the form $\mathrm{POLY}^A(\mathcal{F})$.

THEOREM 2 ([**10, 51**]). *Assume that all the elements of a countable family F of separation problems have the property* (5) *all the elements of a countable family G of separation problems have the property* (6). *Then the following assertions are equivalent:*

$$\mathrm{LOGS}(\mathcal{F}) \subseteq \mathrm{LOGS}(\mathcal{G}) \tag{15}$$

$$\mathrm{POLY}^A(\mathcal{F}) \subseteq \mathrm{POLY}^A(\mathcal{G}) \text{ for all } A. \tag{16}$$

Any mapping from the set of all oracles into the set of families of languages is called a *manifold*. A manifold is called *representable* [$\aleph_0$-*representable*], if it has the form $A \mapsto \mathrm{POLY}^A(F)$ for some nondegenerate F [$A \mapsto \mathrm{POLY}^A(\mathcal{F})$ for some countable family $\mathcal{F}$ containing nondegenerate separation problems].

Theorem 2 implies that an $\aleph_0$-representable manifold determines the defining family $\mathcal{F}$ uniquely up to polylog equivalence, that is,

$$(\forall A\ \mathrm{POLY}^A(\mathcal{F}) = \mathrm{POLY}^A(\mathcal{G})) \iff \mathrm{LOGS}(\mathcal{F}) = \mathrm{LOGS}(\mathcal{G}).$$

This is not true for absolute classes: there are separation problems F_1 and F_2 such that $\mathrm{POLY}(F_1) = \mathrm{POLY}(F_2)$ and $\mathrm{LOGS}(F_1) \neq \mathrm{LOGS}(F_2)$. In other words, there exists a non-relativizable assertion of the form $\mathrm{POLY}(F_1) = \mathrm{POLY}(F_2)$, namely the equality IP = PSPACE proven by Shamir in [**45**]. Both classes IP and PSPACE can be defined in our framework as shown in Section 4.

Consider the following application of Theorem 1. We want to prove the theorem from [**5**] stating that there is an oracle A such that $\mathrm{P}^A \neq \mathrm{NP}^A$. By Theorem 1, it suffices to prove that F_{NP} is not in PLOG. In other words, we have to prove

that no machine can recognize in polylog-time whether a 1 occurs in a given word. Assume that a polylog-time machine M does this job. Run M on a sufficiently long input word containing only 0's (the length of the input should be greater than the running time of M on it; such an n does exist because $n - \mathrm{polylog}(n) \to +\infty$). The output of M should be 0. But since M has not queried at least one bit of α, we can fool it by changing that bit of α to 1.

We have used in this proof only that the number of bits queried by M in a run on an input α is bounded by a polylogarithm of $|\alpha|$, and the running time can be arbitrary. This is true for all the known proofs of the theorems of the form $\exists A\ K_1^A \not\subseteq K_2^A$. More precisely, replace in the Definition 3 the restrictions for time by the restrictions for the number of queried bits of x. The resulting notion is called bit-computability *in $t(n)$ queries*. Let n.u.LOGS(G) stand for the class of separation problems F such that $F(\alpha) \le G(f(\alpha))$ for some function f bit-computable in polylog(n) queries. Then to prove that $\exists A\ \mathrm{POLY}^A(F) \not\subseteq \mathrm{POLY}^A(G)$ it is sufficient to prove that F is not in n.u.LOGS(G) because LOGS(G) $\subseteq$ n.u.LOGS(G). Assertions concerned with the number of queries can usually be proved by counting arguments.

A formal definition of a function bit-computable in $t(n)$ queries can be given using decision trees. Let $x_1, \ldots, x_n$ be Boolean variables and M a set. An $(M, x_1, \ldots, x_n)$-tree is a finite binary rooted tree whose leaves are labeled by elements of M and internal vertices by variables from the set $\{x_1, \ldots, x_n\}$. An $(M, x_1, \ldots, x_n)$-tree T computes the function $f : \mathbb{B}^n \to M$ defined as follows. Let $b_1 \ldots b_n$ be an assignment of Boolean values to $x_1, \ldots, x_n$. Let $v_1, v_2, \ldots, v_k$ be the path in T such that (i) v_1 is the root of T, (ii) for every $i < k$, v_i is an interval vertex and v_{i+1} is the left son of v_i if the value the variable labeling v_i is 0 and the right son of v_i otherwise, and (iii) v_k is a leaf. The value $f(b_1 \ldots b_n)$ is defined as the label of v_k. Let $T(x_1 \ldots x_n)$ denote the function computed by the tree T. The complexity of a tree is measured by its height.

A partial function $f : \mathbb{B}^n \to M$ *is computable in t queries* if there exists an $(M, x_1, \ldots, x_n)$-tree T of height at most t such that the function $T(x_1, \ldots, x_n)$ extends the function $f(x_1 \ldots x_n)$. Replace in Definition 3 the notion of computability in time $t(|x|)$ by the notion of computability in $t(|x|)$ queries. The resulting notion is called the *nonuniform bit-computability in time $t(n)$*, or *bit-computability in $t(n)$ queries*.

DEFINITION 6. n.u.LOGS(G) is the class of all the separation problems F such that $F(\alpha) \le G(f(\alpha))$ for some nonuniformly polylog-time bit-computable function f and for all $\alpha \in \mathbb{B}^*$. n.u.LOG(G) is the class of all languages from n.u.LOGS(G).

Obviously, LOGS(G) $\subseteq$ n.u.LOGS(F), and we obtain an easy corollary from Theorem 1.

COROLLARY 3. *If*

$$F \notin \text{n.u.LOGS}(G), \tag{17}$$

then the negation of (9) *is true.*

It is the assertion (17) that is proved by counting arguments in all the known proofs of theorems of the form

$$\exists A\ \mathrm{POLY}^A(F) \not\subseteq \mathrm{POLY}^A(G).$$

3.2. A criterion of relativizable existence of an m-complete language in a complexity class.

DEFINITION 7. A language L_1 is polynomial-time many-one reducible (=Karp reducible) to a language L_2, $L_1 \leq_m^p L_2$, if there exists a polynomial-time computable function f such that $x \in L_1 \Leftrightarrow f(x) \in L_2$. If we allow the function f to be computable by a polynomial-time machine with an oracle A, then the resulting reducibility is denoted by $\leq_m^{p,A}$.

Let $\leq$ stand for a reducibility on separation problems. We say that a separation problem H is $\leq$-*hard* for a class K of separation problems if every separation problem in K is $\leq$-reducible to H. If H is $\leq$-hard for K and H is in K, then we say that H is $\leq$-*complete* in K. We call a class K_1 of separation problems $\leq$-*hard* for a class K_2 of separation problems, if K_1 has a problem that is $\leq$-hard for K_2.

The following theorem gives a criterion of whether the class $\mathrm{POLY}^A(G)$ is $\leq_m^{p,A}$-hard for the class $\mathrm{POLY}^A(F)$ for all oracles A.

THEOREM 4 ([**10, 51**]). *Assume that a separation problem F satisfies the condition* (5) *and a separation problem G satisfies the condition* (6). *Then the following conditions are equivalent:*

(18) $\mathrm{LOG}(G)$ *is* $\preceq_m^l$*-hard for* $\mathrm{LOGS}(F)$,

(19) F *has a solution in* $\mathrm{LOG}(G)$,

(20) *the class* $\mathrm{POLY}^A(G)$ *is* $\leq_m^{p,A}$*-hard for the class* $\mathrm{POLY}^A(F)$ *for any oracle* A.

If F is a language, then all these assertions are equivalent to the assertion:

(21) *the class* $\mathrm{LOG}(G)$ *is* $\preceq_m^l$*-hard for the class* $\mathrm{LOG}(F)$.

PROOF. Let us prove the implication (18)⇒(19). Assume that (18) is true, that is, there is a language $H \in \mathrm{LOG}(G)$ such that every separation problem in the class $\mathrm{LOGS}(F)$ is $\preceq_m^l$-reducible to H. Then $F \preceq_m^l H$. Let $g : \mathbb{B}^* \to \mathbb{B}^*$ be a polylog-time bit-computable function reducing F to H. Then the language $H(g(\alpha))$ is a solution to F and belongs to $\mathrm{LOG}(G)$.

Let us prove the implication (19)⇒(18). Assume that a language $H \in \mathrm{LOG}(G)$ solves F. Then H is $\preceq_m^l$-hard for the class $\mathrm{LOGS}(F)$ because the problem F is $\preceq_m^l$-complete in $\mathrm{LOGS}(F)$.

Obviously, (18) implies (21). The implication (21)⇒(19) in the case when F is a language can be proven in precisely the same way as the implication (18)⇒(19) because $F \in \mathrm{LOG}(F)$ in this case.

Let us prove the implication (19)⇒(20). Assume that F has a solution $H \in \mathrm{LOG}(G)$. Theorem 1 implies that $\mathrm{POLY}^A(F) \subseteq \mathrm{POLY}^A(H) \subseteq \mathrm{POLY}^A(G)$ (note that in the proof of the implication (8)⇒(9) we have not used conditions (5) and (6)). It suffices to prove, therefore, that the class $\mathrm{POLY}^A(H)$ is $\leq_m^{p,A}$-hard for the class $\mathrm{POLY}^A(F)$. Actually, we shall prove that the class $\mathrm{POLY}^A(H)$ has an $\leq_m^p$-complete language. Let g_0^A, g_1^A, g_2^A, ... be an enumeration of all the functions polynomial-time bit-computable relative to A. Set $L_i^A(x) = H(g_i^A(x))$. By definition, $\mathrm{POLY}^A(H) = \{L_i^A \mid i \in \mathbb{N}\}$.

Let $p_i(|x|)$ be a polynomial upper bound for the time of bit-computation of the function $g_i^A(x)$ for a given $\bar{i}\bar{x}$. We shall prove that there is a function f^A polynomial-time bit-computable relative to A such that $f^A(\bar{i}\bar{x}0^{p_i(|x|)}) = g_i^A(x)$ for all $i \in \mathbb{N}$ and for all $x \in \mathbb{B}^*$. Suppose that we have already proved the existence of such a function f^A. Then let $L^A(u) = H(f^A(u))$. We obtain that $L^A \in \mathrm{POLY}^A(H)$. On the other hand, L^A is $\leq_m^p$-complete in the class $\mathrm{POLY}^A(H)$ because the function $x \mapsto \bar{i}\bar{x}0^{p_i(|x|)}$ is polynomial-time computable and reduces L_i^A to L^A for any $i \in \mathbb{N}$.

It remains to prove the existence of f^A. Let M^A be a machine that computes in time $p_i(|x|)$ the length of the word $g_i^A(x)$ for any given $\bar{i}\bar{x}$, and let N^A be a machine that computes the jth bit of the word $g_i^A(x)$ in time $p_i(|x|)$ for any given $\bar{i}\bar{x}j$. Then the length of the word $f^A(w)$ can be computed by the following machine $\bar{M}^A$: for a given word w check first whether w has the form $\bar{i}\bar{x}0^t$, and if not, output 0. Otherwise find i, x, and t and run M^A on $\bar{i}\bar{x}$. If the machine M^A produces a result within time t, then output that result, otherwise output 0. The following machine $\bar{N}^A$ outputs the jth bit of the word $f^A(w)$ for any given $\langle w, j\rangle$: run first $\bar{M}^A$ on w, let n stand for the result produced by $\bar{M}^A$. If $n = 0$, then output 0. Otherwise find i, x, and t such that $w = \bar{i}\bar{x}0^t$ and run N^A on $\bar{i}\bar{x}j$. If the machine N^A produces a result within time t, then output that result. Otherwise output 0.

Let us prove that if (19) is false, then (20) is false. Assume that F has no solutions in the class $\mathrm{LOG}(G)$. Let us construct an oracle A such that the class $\mathrm{POLY}^A(G)$ has no $\leq_m^{p,A}$-hard language for the class $\mathrm{POLY}^A(F)$. Let f_0^A, f_1^A, ..., f_i^A, ... be an enumeration of all the functions that are polynomial-time bit-computable relative to an oracle A and m_0^A, m_1^A, ..., m_j^A, ... an enumeration of all the $\leq_m^{p,A}$-reducing functions (that is, all the functions of the type $\mathbb{B}^* \to \mathbb{B}^*$ that are polynomial-time computable relative to A). The oracle A is considered as the second argument of both functions $f_i^A(x)$, $m_j^A(x)$. Without loss of generality we may assume that the polynomials bounding the computation times of $f_i^A(x)$, $m_j^A(x)$ does not depend on A.

Assume that $A \subseteq \mathbb{B}^*$. The language $A^i =_{\mathrm{def}} \{x \mid \bar{i}x \in A\}$ is called the *i*th *component of* A and $L_i(A)$ will denote the language $\{n \mid F([A^i]_n) = 1\}$. Recall that for $C \subseteq \mathbb{B}^*$, $[C]_n$ stands for the word of length n whose jth bit is equal to $C(\bar{n}j)$. It is clear that it suffices to construct an oracle A such that for all $i \in \mathbb{N}$, at least one of the following two assertions is true:

$$G(f_i^A(y)) = * \quad \text{for some } y \in \mathbb{B}^*; \tag{22}$$

and

$$\text{the language } L_i(A) \text{ is in } \mathrm{POLY}^A(F) \text{ and is not } \leq_m^{p,A}\text{-reducible to the separation problem } G(f_i^A(y)). \tag{23}$$

To make the condition (23) true it suffices to satisfy one global condition

$$F([A^i]_n) \neq * \text{ for all } n \in \mathbb{N} \tag{24}$$

and the following countable family of local conditions

$$\exists n \in \mathbb{N}\ F([A^i]_n) \neq G(f_i^A(m_j^A(n))),\ j \in \mathbb{N}. \tag{25}$$

Thus it suffices to construct an oracle A such that for all pairs $(i, j) \in \mathbb{N}^2$ at least one of the two assertions (22) and (24)&(25) is true.

Let us start with the oracle A being a polynomial-time decidable language such that for all i the assertion (24) is true. Fix an enumeration of the set $\mathbb{N}^2$. We make countable number of steps indexed by pairs (i,j). During the step (i,j) we redefine the ith component of A on a finite number of words to make the assertion (22) or the assertion (25) true. Evidently, if for some i there exists j such that we have satisfied the condition (22) on the step (i,j), then we can skip the remaining steps (i,j'). On each step we will freeze the value of A on some words.

Let us explain what is done on step number (i,j). Let A be the oracle we have after the previous step (with a finite set of frozen values). Consider two cases:

1st case: it is possible to change nonfrozen values of the ith component of A to make (22) true. Evidently, in this case it is enough to redefine only a finite number of nonfrozen values of A^i. Make those changes of A^i and freeze a finite number of values of A to guarantee the truth of (22). Since $A^{i'}$ is not changed for all $i' \neq i$, the assertions $(\mathrm{G}_{i'})$ for all $i' \neq i$ remain true.

2nd case: for any changes of nonfrozen values of A^i the assertion (22) remains false. Assume that $\alpha \in \mathbb{B}^*$. Let $A[\alpha, i]$ stand for the oracle B such that $B^{i'} = A^{i'}$ for all $i' \neq i$ and $B^i = (A^i)[\alpha]$ (we recall that the notation $C[\alpha]$ is defined in the proof Theorem 1). Consider the language

$$H = \{\alpha \in \mathbb{B}^* \mid G(f_i^{A[\alpha,i]}(m_j^{A[\alpha,i]}(|\alpha|))) = 1\}.$$

Let us prove that $H \in \mathrm{LOG}(G)$. Call $\alpha \in \mathbb{B}^*$ *free* if no value of A on any word of the form $\overline{|\alpha|}i$, $i \leq |\alpha|$, is frozen (that is, we can replace A with $A[\alpha, i]$ without changing frozen values). Note that the set of nonfree values is finite. For all the free α we have $G(f_i^{A[\alpha,i]}(y)) \neq *$ for all $y \in \mathbb{B}^*$. In particular, $G(f_i^{A[\alpha,i]}(m_j^{A[\alpha,i]}(|\alpha|))) \neq *$ for any free α. The function $\alpha \mapsto f_i^{A[\alpha,i]}(m_j^{A[\alpha,i]}(|\alpha|))$ is polylog-time bit-computable (because A is obtained from a polynomial-time decidable language by changing finite number of values). Therefore the function

$$g(\alpha) = \begin{cases} f_i^{A[\alpha,i]}(m_j^{A[\alpha,i]}(|\alpha|)) & \text{if } \alpha \text{ is free,} \\ one_G & \text{if } \alpha \text{ is not free and } \alpha \in H, \\ zero_G & \text{if } \alpha \text{ is not free and } \alpha \notin H, \end{cases}$$

is polylog-time bit-computable, and $H(\alpha) = G(g(\alpha))$ for all $\alpha \in \mathbb{B}^*$. Hence $H \in \mathrm{LOG}(G)$.

Thus, there are infinitely many α such that $F(\alpha) \not\leq H(\alpha)$. Pick a free α such that $F(\alpha) \not\leq H(\alpha)$. Then for $n = |\alpha|$ we have

$$F([A[\alpha,i]^i]_n) = F(\alpha) \not\leq H(\alpha) = G(f_i^{A[\alpha,i]}(m_j^{A[\alpha,i]}(n))).$$

Replace A with $A[\alpha, i]$ and freeze all the values of A on which the value of $f_i^A(m_j^A(n))$ depends as well as the values of A on all the words of the form $\bar{i}\bar{n}j$, $j \leq n$. Thus we have made the assertion (25) true. And the assertion (24) was not affected because $F(\alpha) \neq *$. Since we have redefined only the ith component of A all conditions (24) for $i' \neq i$ were not affected either. The implication (20)⇒(19) is proved. □

Corollary 5. *If F is a language, then the class $\mathrm{POLY}^A(F)$ has a $\leq_m^p$-complete language.*

Remark 2. It is clear from the proof of Theorem 4 that in the condition (20), we can replace the $\leq_m^{p,A}$-reducibility by the $\leq_m^p$-reducibility.

REMARK 3. It is clear from the proof of Theorem 4 that for any sequence $\{\langle F_i, G_i\rangle\}$, $i = 0, 1, 2, \ldots$ of pairs of separation problems such that F_i has no solution in $\mathrm{LOG}(G_i)$, we can construct an oracle A such that the class $\mathrm{POLY}^A(G_i)$ is not $\leq_m^{p,A}$-hard for the class $\mathrm{POLY}^A(F_i)$ for all i. To do so we have to consider for all i a countable number of components $A^{i,j} = \{x \in \mathbb{B}^* \mid \overline{ij}x \in A\}$, $j \in \mathbb{N}$. The same is true for Theorem 1 and for Theorems 7 and 8 below. We can also construct an oracle relative to which negative assertions of different types are true simultaneously. For example, if for all i there exists an oracle A_i such that $\mathrm{POLY}^{A_i}(F_i) \not\subseteq \mathrm{POLY}^{A_i}(G_i)$ and for all j there exists an oracle B_j such that the class $\mathrm{POLY}^{B_j}(H_j)$ is not $\leq_m^{p,B_j}$-hard for the class $\mathrm{POLY}^{B_j}(J_j)$, then there exists a single oracle A relative to which all these assertions are true.

COROLLARY 6. *If for nondegenerate separation problems F and G the assertion*

$$F \text{ has no solution in the class } \mathrm{n.u.LOG}(G), \tag{26}$$

is true, then there exists an oracle A such that the class $\mathrm{POLY}^A(G)$ has no $\leq_m^{p,A}$-hard language for the class $\mathrm{POLY}^A(F)$.

The assertion (26) is the assertion usually proven by counting arguments when one proves that there exists A such that the class $\mathrm{POLY}^A(G)$ is not $\leq_m^{p,A}$-hard for the class $\mathrm{POLY}^A(F)$.

EXAMPLE. In [**40**], it was proved that n.u.BPPLOG = n.u.PLOG. Obviously, the separation problem F_{BPP} defining the class BPP has no solution in the class n.u.PLOG. Consequently, there exists an oracle A such that the class BPP^A has no $\leq_m^{p,A}$-hard language for the class R^A.

REMARK 4. If we replace in the statement of Theorem 4 the separation problems F and G by countable classes $\mathcal{F}$ and $\mathcal{G}$ of separation problems then the implication (20)⇒(19) remains true. To keep the implication (19)⇒(20) true, we have to strengthen the condition (19) as follows: There exist a language H in $\mathrm{LOG}(\mathcal{G})$ and a computable function $f(i, \alpha)$ such that for any fixed i, the function $\alpha \mapsto f(i, \alpha)$ is polylog-time bit-computable and reduces the ith separation problem in $\mathcal{F}$ to H.

3.3. A criterion of whether a complexity class is Turing reducible to another complexity class.

DEFINITION 8. A language L_1 is polynomial-time Turing reducible (=Cook reducible) to a language L_2, $L_1 \leq_T^p L_2$, if there is a polynomial-time Turing machine M with oracle L_2 recognizing L_1. A language L_1 is polynomial-time Turing reducible to a language L_2 relative to A, $L_1 \leq_T^{p,A} L_2$, if there is a polynomial-time Turing machine M with two oracles A and L_2 recognizing L_1.

Let $\leq$ stand for $\leq_T^p$ or $\leq_T^{p,A}$. A class K_1 is $\leq$-reducible to a class K_2 (notation: $K_1 \leq K_2$) if $\forall L_1 \in K_1\ \exists L_2 \in K_2 \quad L_1 \leq L_2$.

To formulate a theorem giving a criterion of whether $K_1 \leq_T^{p,A} K_2$ for all A we define a polylog-time version of polynomial-time Turing reducibility, that is more flexible compared with the polylog-time many-one reducibility.

A separation problem F is *polylog-time* T-*reducible* to a separation problem G ($F \preceq_T^l G$ in symbols) if there are a polynomial-time Turing oracle machine M and a function $f : \mathbb{B}^* \times \mathbb{B}^* \to \mathbb{B}^*$ such that 1) the value $f(y, \alpha)$ can be bit-computed in

time $\mathrm{poly}(|y| + \log|\alpha|)$ for given y and α and 2) for all $\alpha \in \mathrm{D}(F)$ the following two assertions are true:

$$G(f(y,\alpha)) \neq * \text{ for all } y \in \mathbb{B}^*, \tag{27}$$

$$F(\alpha) = M^{G(f(\cdot,\alpha))}(|\alpha|), \tag{28}$$

where $G(f(\cdot,\alpha))$ stands for the language $\{y \in \mathbb{B}^* \mid G(f(y,\alpha)) = 1\}$.

We call $\langle M, f\rangle$ a pair *reducing* F to H. Note that in this definition it suffices to require that (27) and (28) are true for all but finitely many $\alpha \in \mathrm{D}(F)$. Let $\langle M, f\rangle^G(\alpha)$ denote the output of M on input $|\alpha|$ with oracle $G(f(\cdot,\alpha))$.

Obviously, the binary relation $\preceq^l_T$ is reflexive and transitive. It is clear that $F \preceq^l_m G \Rightarrow F \preceq^l_T G$.

THEOREM 7 ([**51**]). *If a separation problem F satisfies the condition* (5) *and a separation problem G satisfies the condition* (6)*, then the following assertions are equivalent:*

$$\mathrm{LOGS}(F) \preceq^l_T \mathrm{LOGS}(G), \tag{29}$$

$$F \preceq^l_T G, \tag{30}$$

$$\mathrm{POLY}^A(F) \leq^{p,A}_T \mathrm{POLY}^A(G) \text{ for all oracles } A. \tag{31}$$

If F is a language, then all three assertions are equivalent to the assertion

$$\mathrm{LOG}(F) \preceq^l_T \mathrm{LOG}(G). \tag{32}$$

PROOF. Evidently, (29) and (30) are equivalent.

Assume that F is a language. Then the implication (32)⇒(30) is true. On the other hand, assume that (30) is true. Let $\langle M, f\rangle$ be a pair reducing F to G. Let $l(n)$ be a polylogarithmic upper bound for the length of queries to oracle made by M on the input $n \in \mathbb{N}$. Consider the language $H = \{\bar{x}\alpha \mid |x| \leq l(|\alpha|), G(f(x,\alpha)) = 1\}$. Let us prove that H belongs to $\mathrm{LOG}(G)$. Since $\mathrm{D}(F) = \mathbb{B}^*$, we have $G(f(x,\alpha)) \neq *$ for all $x, \alpha \in \mathbb{B}^*$. Therefore, we have $H(\beta) = G(h(\beta))$, where

$$h(\beta) = \begin{cases} f(x,\alpha) & \text{if } \beta = \bar{x}\alpha, x \leq l(|\alpha|); \\ zero_G & \text{if } \beta \text{ is not of the form } \bar{x}\alpha, \text{ where } x \leq l(|\alpha|). \end{cases}$$

For a given β we can decide in time $\mathrm{polylog}(|\beta|)$ whether β has the form $\bar{x}\alpha$, $|x| \leq l(|\alpha|)$. Consequently, h is a polylog-time bit-computable function, hence, we have $H \in \mathrm{LOG}(G)$.

Set $g(x,\alpha) = \bar{x}\alpha$. Obviously, $g(x,\alpha)$ can be bit-computed in time $\mathrm{poly}(|x| + \log|\alpha|)$. The pair $\langle M, g\rangle$ reduces F to H, therefore $\{F\} \preceq^l_T \mathrm{LOG}(G)$. As F is $\preceq^l_m$-complete in $\mathrm{LOG}(F)$, we obtain $\mathrm{LOG}(F) \preceq^l_T \mathrm{LOG}(G)$.

Let us prove that (30) implies (31). Assume that $F \preceq^l_T G$. Let $\langle M, f\rangle$ be a pair reducing F to G, A an oracle, and L a language in the class $\mathrm{POLY}^A(F)$. Let g be a function polynomial bit-computable relative to A such that $L(x) = F(g(x))$. Then $L(x) = M^{G(f(\cdot,g(x)))}(|g(x)|)$ for all $x \in \mathbb{B}^*$. Since the function $|g(x)|$ is polynomial-time computable relative to A, the language L is $\leq^{p,A}_T$-reducible to the language $\{\bar{y}x \mid G(f(y,g(x))) = 1\}$. The latter language is in $\mathrm{POLY}^A(G)$ because $G(f(y,g(x))) \neq *$ for all $x, y \in \mathbb{B}^*$ and the function $\bar{y}x \mapsto f(y,g(x))$ is polynomial-time bit-computable relative to A.

Let us prove the implication $\neg(30) \Rightarrow \neg(31)$. Assume that $F \not\preceq_T^l G$. Let us prove that (31) is false. Note that in (31) the $\leq_T^{p,A}$-reducibility can be replaced by the $\leq_T^p$-reducibility. Indeed, if a language L_1 is $\leq_T^{p,A}$-reducible to a language L in $\mathrm{POLY}^A(G)$, then L_1 is $\leq_T^p$-reducible to the language $L \oplus A = \{0x \mid x \in L\} \cup \{1x \mid x \in A\}$, which is in $\mathrm{POLY}^A(G)$ (because $A \in \mathrm{POLY}^A(G)$, provided G satisfies (6) and the class $\mathrm{POLY}^A(G)$ is closed under the operation $\oplus$ for any A and G).

It suffices to construct an oracle A such that the following two conditions hold:

(33) $[A]_n \in \mathrm{D}(F)$ for all n,

and

(34) the language $\{n \mid F([A]_n) = 1\}$ is not $\leq_T^p$-reducible to any language in $\mathrm{POLY}^A(G)$.

Let M_1^B, M_2^B, ..., M_j^B, ... be an enumeration of all the polynomial-time oracle Turing machines. Let $f_1^A(x)$, $f_2^A(x)$, ..., $f_i^A(x)$, ... be an enumeration of all the functions that are polynomial bit-computable relative to A (A is considered as the second argument). We want to construct an oracle A such that

$$\exists n \in \mathbb{N}\ F([A]_n) \neq M_j^{G(f_i^A(\cdot))}(n) \quad \text{or} \quad \exists y\ G(f_i^A(y)) = * \tag{35}$$

for all $i, j \in \mathbb{N}$. At first, let A be equal to a polynomial-time decidable language satisfying the condition (33). We make a countable number of steps indexed by pairs $(i, j) \in \mathbb{N}^2$.

Step (i, j). Let A be the oracle (frozen values included) we have after the previous step. Call $\alpha \in \mathbb{B}^*$ *free* if no value of A on a word of the form $\overline{|\alpha|}k$, $k \leq |\alpha|$, is frozen. Consider two cases.

1st case: there are a free $\alpha \in \mathrm{D}(F)$ and a $y \in \mathbb{B}^*$ such that $G(f_i^{A[\alpha]}(y)) = *$. Then replace A by $A[\alpha]$ and freeze finitely many values of A to guarantee the validity of the assertion (35). Note that the condition (33) has not been affected.

2nd case: $G(f_i^{A[\alpha]}(y)) \neq *$ for all $y \in \mathbb{B}^*$ for all free $\alpha \in \mathrm{D}(F)$. We claim that there is a free $\alpha \in \mathrm{D}(F)$ such that $F(\alpha) \neq M_j^{G(f_i^{A[\alpha]}(\cdot))}(|\alpha|)$. Indeed, otherwise $F(\alpha) = M_j^{G(f_i^{A[\alpha]}(\cdot))}(|\alpha|)$ for all $\alpha \in \mathrm{D}(F)$. Then the function $g(y, \alpha) = f_i^{A[\alpha]}(y)$ is bit-computable in time $\mathrm{poly}(|y| + \log|\alpha|)$ and for the pair $\langle M, g\rangle$ the conditions (27) and (28) are fulfilled for all free $\alpha \in \mathrm{D}(F)$. Therefore, $F \preceq_T^l G$ and we get a contradiction. After that the proof goes similar to the proof of Theorem 1. □

3.4. A criterion of whether a complexity class has a Turing hard language for another complexity class.

THEOREM 8 ([**10, 51**]). *If a separation problem F satisfies the condition* (5) *and a separation problem G satisfies the condition* (6)*, then the following assertions are equivalent:*

(36) *the class* $\mathrm{LOG}(G)$ *is* $\preceq_T^l$*-hard for the class* $\mathrm{LOGS}(F)$*,*

(37) $\{F\} \preceq_T^l \mathrm{LOG}(G)$

(38) *the class* $\mathrm{POLY}^A(G)$ *is* $\leq_T^{p,A}$*-hard for the class* $\mathrm{POLY}^A(F)$ *for all* A*.*

If F is a language, then all the three assertions are equivalent to the assertion

(39) $\mathrm{LOG}(G)$ *is* $\preceq^l_T$*-hard for* $\mathrm{LOG}(F)$.

PROOF. Evidently, (36) and (37) are equivalent, and if F is a language, then they both are equivalent to (39).

Let us prove the implication (37)⇒(38). Assume that $F \preceq^l_T H \in \mathrm{LOG}(G)$. If H does not satisfy the condition (6), then $F \in \mathrm{PLOG}$ and therefore (38) is true. Otherwise, Theorem 4 implies that $\mathrm{POLY}^A(H)$ has a $\leq^p_m$-complete language. By Theorem 7 we have $\mathrm{POLY}^A(F) \leq^{p,A}_T \mathrm{POLY}^A(H)$, consequently, the class $\mathrm{POLY}^A(G)$ is $\leq^{p,A}_T$-hard for the class $\mathrm{POLY}^A(F)$.

Let us prove that (38) implies (37). Similarly to Theorem 8, we may replace the $\leq^{p,A}_T$-reducibility by the $\leq^p_T$-reducibility in (38).

Assume that (37) is false, that is, F is $\preceq^l_T$-reducible to no language in the class $\mathrm{LOG}(G)$. We have to construct an oracle A such that the class $\mathrm{POLY}^A(G)$ has no language that is $\leq^p_T$-hard for the class $\mathrm{POLY}^A(F)$. Let $f_0^A(y)$, $f_1^A(y)$, ..., $f_i^A(y)$, ... be an enumeration of all the functions that are polynomial-time bit-computable relative to A. Split A into components $A^i = \{x \mid \bar{i}x \in A\}$. It suffices to find an A such that for any $i \in \mathbb{N}$ at least one of the following two assertions holds:

(40) $G(f_i^A(y)) = *$ for some $y \in \mathbb{B}^*$,

(41) the language $L_i(A) = \{n \mid F([A^i]_n) = 1\}$ is in the class $\mathrm{POLY}^A(F)$ and is not $\leq^p_T$-reducible to the separation problem $G(f_i^A(y))$.

Let M_0^L, M_1^L, ..., M_j^L, ... be an enumeration of all the polynomial-time oracle Turing machines.

To make the assertion (41) true it suffices to satisfy the following requirements:

(42) $F([A^i]_n) \neq *$ for all n,

(43) $\exists n \in \mathbb{N}\ F([A^i]_n) \neq M_j^{G(f_i^A(\cdot))}(n)$

for all $j \in \mathbb{N}$. To construct an oracle A satisfying (40) or (42)&(43) for all pairs (i,j) we can follow the proof of Theorem 4. The only difference appears in the second case when the step (i,j) is described. Recall that in the second case $G(f_i^A(y)) \neq *$ for all $y \in \mathbb{B}^*$ and for all variations of nonfrozen values of A^i. We call a word $\alpha \in \mathbb{B}^*$ free if no value of A^i on a word of the form $\overline{|\alpha|}j$, $j \leq |\alpha|$, is frozen. We have to prove that there is a free $\alpha \in \mathrm{D}(F)$ such that $F(\alpha) \neq M_j^{G(f_i^{A[\alpha,i]}(\cdot))}(|\alpha|)$. Assume that there is no such α. Let $l(n)$ denote a polylogarithmic upper bound for the length of queries made by the machine M on input n. Consider the language

$$H = \{\bar{y}\alpha : |y| \leq l(|\alpha|), G(f_i^{A[\alpha,i]}(y)) = 1\}$$

and the function $g(y,\alpha) = \bar{y}\alpha$. Since $G(f_i^{A[\alpha,i]}(y)) \neq *$ for all free α and for all $y \in \mathbb{B}^*$, the language H is in $\mathrm{LOG}(G)$. Then for the pair $\langle M_j, g\rangle$, the assertions (27) and (28) are true for all free $\alpha \in \mathbb{B}^*$. Therefore, $F \preceq^l_T H$. This contradiction finishes the proof. □

COROLLARY 9. *If* $F \not\preceq^l_T \mathrm{n.u.LOG}(G)$, *then there exists an oracle* A *such that the class* $\mathrm{POLY}^A(G)$ *is not* $\leq^{p,A}_T$*-hard for the class* $\mathrm{POLY}^A(F)$.

REMARK 5. Let K_1, K_2 be classes of languages and A an oracle. In [**3**] it is noted that if the class K_2 is downward closed under $\leq_T^{p,A}$-reductions, then the class K_2 is $\leq_T^{p,A}$-hard for a class K_1 if and only if K_2 is $\leq_m^p$-hard for K_1. Indeed, suppose that L is a language in K_2 to which all the languages from K_1 are $\leq_T^{p,A}$-reducible. Then consider the language

$$L_1 = \{\bar{i}\bar{x}0^t \mid M_i^{A,L} \text{ on input } x \text{ outputs 1 in} \leq t \text{ steps}\},$$

where $M_0, M_1, \ldots$ is a numeration of polynomial-time Turing machines having two oracles. All the languages in the class K_1 are $\leq_m^p$-reducible to L_1. On the other hand, $L_1 \leq_T^{p,A} L$, hence, $L_1 \in K_2$.

4. Relativizable inclusions between particular complexity classes

In this section we consider many of the representable classes lying between P and PSPACE. As mentioned in Remark 1, all the particular complexity classes studied in the literature can be represented by means of separation problems that are non-zero only on the words of length 2^n, $n \in \mathbb{N}$. To simplify the notation, we consider in the sequel only separation problems satisfying this requirement. Let $\mathbb{F}_n$ denote $\mathbb{B}^{2^n}$ and let $\mathbb{F}$ denote $\bigcup_{n=0}^{\infty} \mathbb{F}_n$. We enumerate the bits of a word $\alpha \in \mathbb{F}_n$ either by binary words of length n, or by numbers from 0 to $2^n - 1$. For a word α in $\mathbb{F}$, the *norm* of α, $\|\alpha\|$, is defined as $\log_2 |\alpha|$. While defining particular separation problems we keep the following agreement: if the problem under consideration is defined only on a set $M \subseteq \mathbb{B}^*$, then its value on all the words from $\mathbb{B}^* \setminus M$ is equal to 0 (that is, the default value is 0).

We consider the following relativized complexity classes: P^A, UP^A, co-UP^A, $\mathrm{UP}^A \cap$ co-UP^A, FewP^A, co-FewP^A, $\mathrm{FewP}^A \cap$ co-FewP^A, Few^A, $\oplus\mathrm{P}^A$, R^A, co-R^A, $\mathrm{R}^A \cap$co-R^A, NP^A, co-NP^A, $\mathrm{NP}^A \cap$co-NP^A, BPP^A, MA^A, co-MA^A, $\mathrm{MA}^A \cap$co-MA^A, AM^A, co-AM^A, $\mathrm{AM}^A \cap$ co-AM^A, PP^A, Σ_k^A, Π_k^A, $\Pi_k^A \cap \Sigma_k^A$ $(k \geq 2)$, IP^A, co-IP^A, $\mathrm{IP}^A \cap$ co-IP^A, PH^A, PSPACE^A.

Below we remind the definitions of complexity classes from this list and give some comments.

1. $\mathrm{R}^A =_{\mathrm{def}} \mathrm{POLY}^A(F)$ where

$$F_{\mathrm{R}}(\alpha) = \begin{cases} 1 & \text{if } \#_1(\alpha) > 2/3, \\ 0 & \text{if } \#_1(\alpha) = 0, \\ * & \text{otherwise.} \end{cases}$$

2. $\mathrm{UP}^A =_{\mathrm{def}} \mathrm{POLY}^A(F)$ where

$$F_{\mathrm{UP}}(\alpha) = \begin{cases} 1 & \text{if } \#_1(\alpha) = 1, \\ 0 & \text{if } \#_1(\alpha) = 0, \\ * & \text{otherwise.} \end{cases}$$

3. The definition of the class FewP^A is as follows. A language L is in FewP^A if there are a polynomial q and a function f polynomial-time bit-computable relative to A such that (i) $\#_1(f(x)) < q(|x|)$ and (ii) $x \in L \iff \#_1(f(x)) > 0$ for all x.

It is easy to verify that $\mathrm{FewP}^A = \mathrm{POLY}^A(F)$ where

$$F(\alpha) = \begin{cases} 1 & \text{if } 0 < \#_1(\alpha) < \|\alpha\|, \\ 0 & \text{if } \#_1(\alpha) = 0, \\ * & \text{otherwise.} \end{cases}$$

4. Few^A is the class defined in the paper [**11**] as follows: a language L is in Few^A if there exist a function f polynomial-time bit-computable relative to A, a polynomial q and a predicate R defined on the set $\mathbb{B}^* \times \mathbb{N}$ and polynomial-time computable relative to A such that (i) $L(x) = R(x, \#_1 f(x))$ and (ii) $\#_1(f(x)) < q(|x|)$ for all $x \in \mathbb{B}^*$.

The representation of Few^A in the form $\mathrm{POLY}^A(F)$ is not so natural as for other classes. Assume that $|\alpha| = 2^{n+1}$ and $\alpha = \beta\gamma$ where $|\beta| = |\gamma| = 2^n$. Let

$$F(\alpha) = \begin{cases} * & \text{if } \#_1(\beta) \geq n, \\ \gamma(\#_1(\beta)) & \text{otherwise.} \end{cases}$$

It is easy to verify that $\mathrm{Few}^A = \mathrm{POLY}^A(F)$.

5. $\oplus\mathrm{P}^A =_{\mathrm{def}} \mathrm{POLY}^A(\mathrm{PARITY})$, where

$$\mathrm{PARITY}(\alpha) = \begin{cases} 0 & \text{if } \#_1(\alpha) \text{ is even,} \\ 1 & \text{otherwise.} \end{cases}$$

6. $\mathrm{PP}^A =_{\mathrm{def}} \mathrm{POLY}^A(\mathrm{MAJORITY})$, where

$$\mathrm{MAJORITY}(\alpha) = \begin{cases} 0 & \text{if } \#_1(\alpha) < \frac{1}{2}|\alpha|, \\ 1 & \text{otherwise.} \end{cases}$$

7. AM^A is the abbreviation for the class $\mathrm{AM}[2]^A$. The class AM^A is represented by the following separation problem F_{AM}. Let the notation $\mathrm{M}_d\, x \in M\ P(x)$ mean that $|\{x \in M : P(x)\}| > d \cdot |M|$. Then for $\alpha \in \mathbb{F}_{2n}$,

$$F_{\mathrm{AM}}(\alpha) = \begin{cases} 1 & \text{if } \mathrm{M}_{2/3}\, u \in \mathbb{B}^n\ \exists v \in \mathbb{B}^n\ \alpha(uv) = 1, \\ 0 & \text{if } \mathrm{M}_{2/3}\, u \in \mathbb{B}^n\ \forall v \in \mathbb{B}^n\ \alpha(uv) = 0, \\ * & \text{otherwise,} \end{cases}$$

where uv stands for the concatenation of words u and v.

8. MA^A is the class represented by the separation problem

$$F_{\mathrm{AM}}(\alpha) = \begin{cases} 1 & \text{if } \exists u \in \mathbb{B}^n\ \ \mathrm{M}_{2/3}\, v \in \mathbb{B}^n\ \alpha(uv) = 1, \\ 0 & \text{if } \forall u \in \mathbb{B}^n\ \ \mathrm{M}_{2/3}\, v \in \mathbb{B}^n\ \alpha(uv) = 0, \\ * & \text{otherwise,} \end{cases}$$

where $\alpha \in \mathbb{F}_{2n}$.

9. $\Sigma_k^A =_{\mathrm{def}} \mathrm{POLY}^A(F_k)$ where

$$F_k(\alpha) = \begin{cases} 1 & \exists y_1 \forall y_2 \ldots Q y_k\, \alpha(y_1 y_2 \ldots y_k) = 1,\ \|\alpha\| \text{ is a multiple of } k; \\ 0 & \text{otherwise,} \end{cases}$$

where Q stands for $\exists$ if k is odd and for $\forall$ otherwise, and all $y_1, \ldots, y_k$ range over $\mathbb{B}^{\|\alpha\|/k}$.

Note that $\Sigma_1^A = \mathrm{NP}^A$.

10. $\Pi_k^A =_{\text{def}} \text{co-}\Sigma_k^A$, $\text{PH}^A = \bigcup_k \Sigma_k^A$. The manifold PH^A, as observed by Silvestri [**46**], is an example of $\aleph_0$-representable manifold that is not representable. Indeed, assume $\text{PH}^A = \text{POLY}^A(F)$ for some F and for all A. On the other hand, we have $\text{PH}^A = \text{POLY}^A(\{F_k \mid k = 1, 2, \ldots\})$, where F_k is defined in the previous item. Theorem 2 implies that $F \in \text{LOGS}(\{F_k \mid k = 1, 2, \ldots\})$, hence $F \in \text{LOGS}(F_k)$ for some particular k. This implies that $\text{PH}^A = \Sigma_k^A$ for all A. However for any k there is an oracle separating PH from Σ_k [**25**] (see item 12 below).

11. PSPACE^A is the class of languages recognized by polynomial-space Turing machines with oracle A. Let us prove that the manifold PSPACE^A has the form $\text{POLY}^A(F)$ for some F. It is well known that any language L in PSPACE^A can be represented as follows:

$$L = \{x \mid \exists y_1 \in \mathbb{B}^n \ \forall y_2 \in \mathbb{B}^n \ldots Q y_n \in \mathbb{B}^n P^A(x, y_1 y_2 \ldots y_n), \text{ where } n = p(|x|)\},$$

where $P^A(x, u)$ is a predicate that is polynomial-time computable relative to A and $p(m)$ is a polynomial. The converse is true, too. Therefore, we can take the separation problem

$$F_{\text{PSPACE}}(\alpha) = \begin{cases} 1 & \text{if there is an } n \in \mathbb{N} \text{ such that } \|\alpha\| = n^2 \text{ and} \\ & \exists y_1 \in \mathbb{B}^n \ \forall y_2 \in \mathbb{B}^n \ldots Q y_n \in \mathbb{B}^n \ \alpha(y_1 y_2 \ldots y_n) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

It is clear that $\text{POLY}^A(F_{\text{PSPACE}}) = \text{PSPACE}^A$ and $\text{LOG}(F_{\text{PSPACE}})$ is the class of languages that can be recognized within polylogarithmic space.

12. To define the class IP^A take the following separation problem F_{IP}. On words $\alpha \in \mathbb{F}_{2^{n^2}}$, it is defined as follows

$$F_{\text{IP}}(\alpha) = \begin{cases} 1 & \text{if } \exists P : \mathbb{B}^* \to \mathbb{B}^n \\ & \text{Prob}[\alpha(r_1 r_2 \ldots r_n P(r_1) P(r_1 r_2) \ldots P(r_1 r_2 \ldots r_n)) = 1] > \frac{2}{3}, \\ 0 & \text{if } \forall P : \mathbb{B}^* \to \mathbb{B}^n \\ & \text{Prob}[\alpha(r_1 r_2 \ldots r_n P(r_1) P(r_1 r_2) \ldots P(r_1 r_2 \ldots r_n)) = 1] < \frac{1}{3}, \\ * & \text{otherwise,} \end{cases}$$

where the probability is considered with respect to the uniform distribution in $r_1 \ldots r_n$. Then $\text{POLY}^A(F_{\text{IP}}) =_{\text{def}} \text{IP}^A$.

To explain the intuitive meaning of the definition of F_{IP}, we recall the definition of the class IP^A according to [**4**] and convert it to a convenient form. Consider a game between two players called *Verifier, V,* and *Prover, P.* P tries to convince V that an input string is in a language L. The Verifier is bounded in time by a polynomial of the length of the input, he has access to oracle A and can toss a fair coin. P has unbounded computational resources. The convincing procedure is as follows. For a given input x, V tosses the coin several times and then asks P a question depending on x and the outcome of the tossing. Then P answers the question. Again V tosses the coin several times and then asks P another question depending on x, both outcomes of tossing and the previous answer of V. This repeats a polynomial number of times. Then P either accepts or rejects. The outcomes of tossing are known also to P. The action of V is governed by a polynomial-time Turing machine M. We say that M is *good* for L if (i) for any x in L, there is a strategy for P such that V will accepts with probability greater than 2/3 and (ii) for any x outside L and for all strategies of P, Verifier will reject with probability greater than 2/3. There are no restrictions for strategy of P, his strategy may be

even uncomputable. However it is easy to see that it is possible to restrict P's strategies with those computable in exponential time (and even within polynomial space). It is easy also to see that to give answers to V's questions P does not need the questions itself, as they can be computed from x and outcomes of tossing.

In a formal setting, a Verifier is a pair $V = (q, Q)$, where Q is a polynomial-time computable predicate on $\mathbb{B}^* \times \mathbb{B}^* \times \mathbb{B}^*$ and $q : \mathbb{N} \to \mathbb{N}$ is a polynomial. Any function $P : \mathbb{B}^* \to \mathbb{B}^*$ is called a Prover's strategy, or briefly a Prover. Assume that $x \in \mathbb{B}^*$, $|x| = m$. Assume that $r_1, \ldots, r_{q(m)}$ is a sequence of $q(m)$ binary words of length $q(m)$ (the outcomes of tossing). For all $i \le q(m)$, let

$$p_i = P(r_1 \ldots r_i) \text{ (the } i\text{th answer of the Prover).}$$

We say that the *result of* (P, V) *on input* x *and random inputs* $r_1, \ldots, r_{q(m)}$ is equal to 1 if all the words p_i have length $q(m)$ and $Q(x, r_1 \ldots r_{q(m)}, p_1 \ldots p_{q(m)}) = 1$; otherwise the result is equal to 0. Let $(P, V)(x)_{r_1 \ldots r_{q(m)}}$ denote the result of (P, V) on input x and random inputs $r_1, \ldots, r_{q(m)}$. We say that a language L belongs to IP if there is a Verifier V such that the following two assertions are true:

$$\forall x \in L \; \exists P \;\; \mathrm{Prob}[(P, V)(x)_{r_1 \ldots r_{q(|x|)}} = 1] > 2/3, \tag{44}$$

$$\forall x \notin L \; \forall P \;\; \mathrm{Prob}[(P, V)(x)_{r_1 \ldots r_{q(|x|)}} = 0] > 2/3. \tag{45}$$

If we allow Verifier to query the oracle A, then the resulting class is denoted by IP^A.

The alternative definition of the class IP according to [**21**] (when the Prover does not see the outcomes of tossing) also fits into our framework though in a tricky way. As proven in [**22**], these two definitions are equivalent and the proof of the equivalence relativizes.

It is easy to see that a language L is in $\mathrm{LOG}(F_{\mathrm{IP}})$ if there is a polylog-time Verifier (in the above formal definition, Q is polylog-time computable and q is a polylogarithm) for which (44) and (45) hold. Let IPLOG denote $\mathrm{LOG}(F_{\mathrm{IP}})$.

13. For any class K^A, the class co-K^A is defined as $\{L \mid \mathbb{B}^* \setminus L \in K^A\}$. Note that if the manifold K^A is representable [$\aleph_0$-representable], then the manifold co-$K^A = \{\mathbb{B}^* \setminus L \mid L \in K^A\}$ is representable [$\aleph_0$-representable]. If K_1^A, K_2^A are representable, say $K_i^A = \mathrm{POLY}^A(F_i)$, $i = 1, 2$, then the manifold $K_1^A \cap K_2^A$ is also representable. Indeed, take the following separation problem F:

$$F(\alpha) = \begin{cases} 1 & \text{if } \alpha = \overline{|\alpha_1|}\alpha_1\alpha_2, \text{where } F_1(\alpha_1) = F_2(\alpha_2) = 1, \\ 0 & \text{if } \alpha = \overline{|\alpha_1|}\alpha_1\alpha_2, \text{where } F_1(\alpha_1) = F_2(\alpha_2) = 0, \\ * & \text{otherwise.} \end{cases}$$

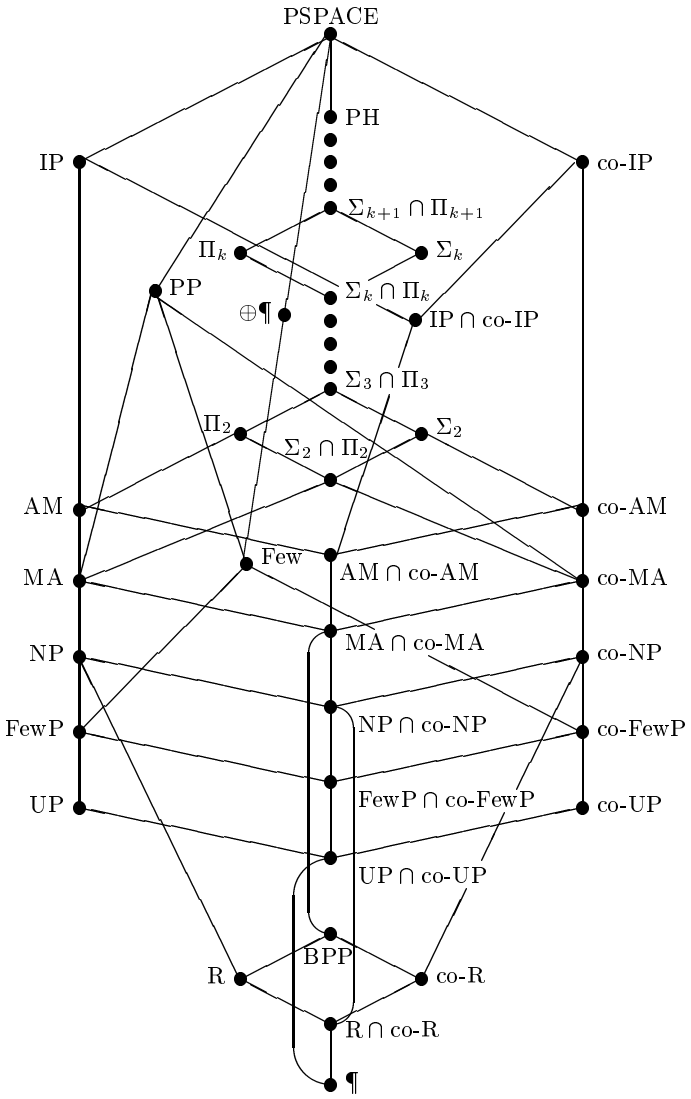
Obviously, this separation problem F also satisfies the following equations: $\mathrm{LOG}(F) = \mathrm{LOG}(F_1) \cap \mathrm{LOG}(F_2)$, $\mathrm{LOGS}(F) = \mathrm{LOGS}(F_1) \cap \mathrm{LOGS}(F_2)$, $\mathrm{EXP}^A(F) = \mathrm{EXP}^A(F_1) \cap \mathrm{EXP}^A(F_2)$.

All the known relativizable inclusions between the classes under consideration are shown in Figure 1. A line segment connects a class K_1^A with a class K_2^A if K_1^A is included in K_2^A, and K_2^A is positioned higher than K_1^A.

4.1. Historical references. The nontrivial inclusions shown in Figure 1 were proved by the following authors.

1. The assertion $\mathrm{MA}^A \subseteq \Sigma_2^A \cap \Pi_2^A$ follows from Gács' result (published in [**48**]) stating that $\mathrm{BPP}^A \subseteq \Sigma_2^A \cap \Pi_2^A$. Namely, in [**48**] a separation problem G is

FIGURE 1. Relativizable inclusions between complexity classes.



constructed such that $G(\alpha)$ is a solution of F_{BPP} and

$$G(\alpha) = 1 \iff \forall y \in \mathbb{B}^{p(\|\alpha\|)}\ \exists z \in \mathbb{B}^{p(\|\alpha\|)} Q(\alpha, y, z) \tag{46}$$

where p is a polynomial and Q is a polylog-time predicate (that is, $G \in \Pi_2\mathrm{LOG}$).

2. The assertion $\mathrm{AM}^A \subseteq \Pi_2^A$ follows from the cited Gács' result. However, for this assertion, it is important that in (46) the predicate $Q(\alpha, y, z)$ is monotone in α (that is, if α' can by obtained from α be replacing some 0's by 1's, then $Q(\alpha, y, z) \Rightarrow Q(\alpha', y, z)$).

3. The assertion $\mathrm{MA}^A \subseteq \mathrm{AM}^A$ was proved in [**4**].

4. $\mathrm{Few}^A \subseteq \oplus\mathrm{P}^A$ was proved in [**11**].

5. The assertion $\mathrm{MA}^A \subseteq \mathrm{PP}^A$ is proved as follows. By Theorem 1 it suffices to prove that MALOGS $\subseteq$ PPLOGS.

Assume that $F \in$ MALOGS. Then there are a polynomial p and a polylog-time predicate Q such that

$$F(\alpha) = 1 \Rightarrow \exists u \in \mathbb{B}^{p(l)} \ \ \mathrm{M}_{2/3}\, v \in \mathbb{B}^{p(l)} \ \ Q(\alpha, u, v) = 1,$$
$$F(\alpha) = 0 \Rightarrow \forall u \in \mathbb{B}^{p(l)} \ \ \mathrm{M}_{2/3}\, v \in \mathbb{B}^{p(l)} \ \ Q(\alpha, u, v) = 0,$$

where l stands for $\log|\alpha|$.

Using amplification we can construct a polynomial p_1 and a polylog-time predicate Q_1 such that

$$\begin{aligned} F(\alpha) = 1 &\Rightarrow \exists u \in \mathbb{B}^{p(l)} \ \ \mathrm{Prob}[Q_1(\alpha, u, r) = 1] > 1 - 4^{-p(l)} \\ F(\alpha) = 0 &\Rightarrow \forall u \in \mathbb{B}^{p(l)} \ \ \mathrm{Prob}[Q_1(\alpha, u, r) = 1] < 4^{-p(l)}, \end{aligned} \tag{47}$$

where probability is considered with respect to the uniform distribution in $r \in \mathbb{B}^{p_1(l)}$. Indeed, let $p_1(l) = C \cdot p(l)^2$ where C is a constant to be specified later. We view an $r \in \mathbb{B}^{p_1(l)}$ as the concatenation of $Cp(l)$ strings $v_1, \ldots, v_{Cp(l)}$ of length $p(l)$. Let $Q_1(\alpha, u, r) = \mathrm{MAJORITY}(Q(\alpha, u, v_1), \ldots, Q(\alpha, u, v_{Cp(l)}))$. Let us make use of the Chernoff's bound [**13**].

THEOREM 10. *Let $\xi_1, \ldots, \xi_n$ be independent random variables in the set $\{0, 1\}$ such that $\mathrm{Prob}[\xi_i = 1] = p$ for all i. Then for any $\delta \in (0; p(1-p)]$,*

$$\mathrm{Prob}\left[\left|\frac{1}{n}\sum_{i=1}^{n} \xi_i - p\right| \geq \delta\right] \leq 2e^{-\frac{\delta^2 n}{2p(1-p)}}.$$

By Chernoff's bound, for some positive constant c, $\mathrm{Prob}[Q_1(\alpha, r, s)] > 1 - 2^{-c \cdot C \cdot p(l)}$ whenever $\mathrm{M}_{2/3}\, v \in \mathbb{B}^{p(l)} \ Q(\alpha, u, v) = 1$ and $\mathrm{Prob}[Q_1(\alpha, r, s)] < 2^{-c \cdot C \cdot p(l)}$ whenever $\mathrm{M}_{2/3}\, v \in \mathbb{B}^{p(l)} \ Q(\alpha, u, v) = 0$. Let $C = \lceil 2/c \rceil$.

From (47) we conclude that

$$F(\alpha) = 1 \Rightarrow \mathrm{Prob}[Q_1(\alpha, u, r) = 1] > 2^{-p(l)}(1 - 4^{-p(l)}) > 4^{-p(l)}$$
$$F(\alpha) = 0 \Rightarrow \mathrm{Prob}[Q_1(\alpha, u, r) = 1] < 4^{-p(l)}$$

with respect to the uniform distribution in pairs $\langle u, r\rangle \in \mathbb{B}^{p(l)+p_1(l)}$. We shall now define a function $f : \mathbb{B}^n \to \mathbb{B}^k$ where $n = 2^l = |\alpha|$ and $k = 2^{p_1(l)+p(l)+1} - 2^{p_1(l)-p(l)+1}$. We index the first $2^{p_1(l)+p(l)}$ bits of $f(\alpha)$ by pairs $\langle u, r\rangle \in \mathbb{B}^{p_1(l)+p(l)}$. The bit of $f(\alpha)$ number $\langle u, r\rangle$ is equal to $Q_1(\alpha, u, r)$. The remaining $2^{p_1(l)+p(l)} - 2^{p_1(l)-p(l)+1}$ bits of $f(\alpha)$ are 1's. Obviously $f(\alpha)$ is polylog-time bit-computable. We claim that f reduces F to MAJORITY. If $F(\alpha) = 1$, then more than

$$2^{p_1(l)+p(l)}2^{-2p(l)} + 2^{p_1(l)+p(l)} - 2^{p_1(l)-p(l)+1} = |f(\alpha)|/2$$

bits of $f(\alpha)$ are 1's. If $F(\alpha) = 0$, then less than $|f(\alpha)|/2$ bits of $f(\alpha)$ are ones. Hence $F \in$ PPLOGS. □

6. The assertion $\mathrm{Few}^A \subseteq \Sigma_2^A \cap \Pi_2^A$ follows from $\mathrm{Few}^A \leq_T^p \mathrm{NP}^A$; the latter assertion is easy and well known. Later we will need the following stronger assertion.

LEMMA 1. $\mathrm{Few}^A \leq_T^p \mathrm{FewP}^A$ *for all A.*

PROOF. Fix $A \subseteq \mathbb{B}^*$. Assume that $L \in \mathrm{Few}^A$ and that L is defined by the polynomials p, q and polynomial-time predicates R^A, Q^A, that is,

$$L(x) = R^A(x, \big|\{y \in \mathbb{B}^{p(|x|)} \mid Q^A(x,y)\}\big|),$$
$$\big|\{y \in \mathbb{B}^{p(|x|)} \mid Q^A(x,y)\}\big| \le q(|x|).$$

Let

$$K = A \oplus \{\bar{x}z \mid \exists y \in \mathbb{B}^{p(|x|)} \ (z \text{ is a prefix of } y \text{ and } Q^A(x,y)\}.$$

Obviously, K is in FewP^A. Using the binary search and querying oracle K, we find in polynomial time for any given x all $y \in \mathbb{B}^{p(|x|)}$ such that $Q^A(x,y)$. Then we compute $L(x) = R^A(x, \big|\{y \in \mathbb{B}^{p(|x|)} \mid Q^A(x,y)\}\big|)$. □

7. The assertion $\mathrm{Few}^A \subseteq \mathrm{PP}^A$ was proved in [**31**].

4.2. Proving the completeness of Figure 1. We claim that all true relativizable inclusions are shown in Figure 1. It follows from the twelve assertions listed below. Namely, all the assertions $\exists A\ K_1^A \not\subseteq K_2^A$ such that

$$K_1 \not\le K_2 \text{ and } \forall K_1'(K_1' < K_1 \Rightarrow K_1' \le K_2),\ \forall K_2'\ (K_2 < K_2' \Rightarrow K_1 \le K_2')$$

are listed, where $K_1 < K_2$ means that there exists a directed path from the class K_1 to the class K_2 in the directed graph shown in Figure 1. Here is the list:

1. $\exists A\ \mathrm{UP}^A \cap \text{co-}\mathrm{UP}^A \not\subseteq \mathrm{BPP}^A$
2. $\exists A\ \mathrm{R}^A \cap \text{co-}\mathrm{R}^A \not\subseteq \oplus\mathrm{P}^A$
3. $\exists A\ \text{co-}\mathrm{UP}^A \not\subseteq \oplus\mathrm{IP}^A$
4. $\exists A\ \mathrm{FewP}^A \cap \text{co-}\mathrm{FewP}^A \not\subseteq \mathrm{UP}^A$
5. $\exists A\ \text{co-}\mathrm{R}^A \not\subseteq \mathrm{NP}^A$
6. $\exists A\ \mathrm{IP}^A \cap \text{co-}\mathrm{IP}^A \not\subseteq \mathrm{PH}^A$
7. $\exists A\ \mathrm{AM}^A \cap \text{co-}\mathrm{AM}^A \not\subseteq \mathrm{PP}^A$
8. $\exists A\ \mathrm{AM}^A \not\subseteq \Sigma_2^A$
9. $\exists A\ \mathrm{PP}^A \not\subseteq \mathrm{PH}^A$
10. $\exists A\ \oplus\mathrm{P}^A \not\subseteq \mathrm{PH}^A$
11. $\exists A\ \oplus\mathrm{P}^A \not\subseteq \mathrm{PP}^A$
12. $\exists A\ \Pi_k^A \not\subseteq \Sigma_k^A$ for $k \ge 3$

We give the proofs of all the assertions in the above list whose proofs do not require much space and give references for all other assertions.

1. $\exists A\ \mathrm{UP}^A \cap \text{co-}\mathrm{UP}^A \not\subseteq \mathrm{BPP}^A$

THEOREM 11 ([**51**]). $\exists A\ \mathrm{UP}^A \cap \text{co-}\mathrm{UP}^A \not\subseteq \mathrm{BPP}^A$

PROOF. Let us fix a convenient terminology (also used in other proofs). All the specific separation problems G used in the sequel satisfy the following property:

> for all $F \in \mathrm{LOGS}(G)$ there exists a polylog-time bit-computable function f such that $F(\alpha) \le G(f(\alpha))$ and for all α the norm of $f(\alpha)$ depends only on $\|\alpha\|$ and is equal to a polynomial of $\|\alpha\|$.

Assume that $F \in \mathrm{LOGS}(G)$ and let f be a polylog-time bit-computable function such that $F(\alpha) = G(f(\alpha))$ and $\|f(\alpha)\| = p(\|\alpha\|)$ for all $\alpha \in \mathrm{D}(F)$, where p is a polynomial. Then all the words r contained in the set $\mathbb{B}^{p(\|\alpha\|)}$ are called *experts* (for f and $\|\alpha\|$), and the rth bit of $f(\alpha)$ is called the *opinion of r about α*. Fix a polylog-time machine M that computes the rth bit of the word $f(\alpha)$ for a given α and $r \in \mathbb{B}^{p(\|\alpha\|)}$. We say that the *expert r queries $\alpha(u)$* (where $u \in \mathbb{B}^{\|\alpha\|}$), if M queries the uth bit of α during the run on the input $\langle \alpha, r\rangle$. It is clear that for all α and all $r \in \mathbb{B}^{p(\|\alpha\|)}$ there are at most $\mathrm{poly}(\|\alpha\|)$ different $u \in \mathbb{B}^{\|\alpha\|}$ such that r queries $\alpha(u)$. Call the fraction

$$\frac{\big|\{r \in \mathbb{B}^{p(\|\alpha\|)} \mid r \text{ queries } \alpha(u)\}\big|}{2^{p(\|\alpha\|)}}$$

the *weight of u relative to α*, in symbols: $w_\alpha(u)$. If M and p are not determined by the context we say "the weight of u relative to α for M, p". It is easy to prove the following general fact: $\sum_{u\in\mathbb{B}^{\|\alpha\|}} w_\alpha(u) \le q(\|\alpha\|)$, where q is the polynomial bounding the number of queries of every expert $r \in \mathbb{B}^{p(\|\alpha\|)}$.

Now let us start with the proof of Theorem 11. By Theorem 1, it suffices to prove that the separation problem

$$F(\alpha) = \begin{cases} 1 & \text{if } \alpha = \beta\gamma,\ \|\beta\| = \|\gamma\|,\ \#_1(\beta) = 1,\ \#_1(\gamma) = 0, \\ 0 & \text{if } \alpha = \beta\gamma,\ \|\beta\| = \|\gamma\|,\ \#_1(\beta) = 0,\ \#_1(\gamma) = 1, \\ * & \text{otherwise,} \end{cases}$$

does not belong to BPPLOGS (evidently, $\mathrm{POLY}^A(F) = \mathrm{UP}^A \cap \text{co-}\mathrm{UP}^A$).

Assume the contrary: suppose there are a polynomial p and a polylog-time predicate P such that

$$\#_1(\beta) = 1,\ \#_1(\gamma) = 0 \Rightarrow \mathrm{M}_{2/3}\, r \in \mathbb{B}^{p(n)}\, P(\beta\gamma, r) = 1,$$
$$\#_1(\beta) = 0,\ \#_1(\gamma) = 1 \Rightarrow \mathrm{M}_{2/3}\, r \in \mathbb{B}^{p(n)}\, P(\beta\gamma, r) = 0,$$

for all n and all $\beta, \gamma \in \mathbb{F}_n$. Fix n. Let $\beta_0 \in \mathbb{F}_n$, $\gamma_0 \in \mathbb{F}_n$ be the words containing only zeros. Without loss of generality we may assume that the fraction $\big|\{r \in \mathbb{B}^{p(n)} \mid P(\beta_0\gamma_0, r) = 1\}\big|/2^{p(n)}$ is greater than or equal to $1/2$. We shall enumerate bits in the first half β of the word $\beta\gamma$ (where $\beta, \gamma \in \mathbb{F}_n$) by the words of the form $0u$, $u \in \mathbb{B}^n$, and bits of the second half γ by the words of the form $1u$. (We follow this rule in the sequel, too.)

Let the number of queries of experts to $\beta_0\gamma_0$ be bounded by $k = \mathrm{poly}(n)$. Then $\sum_{u\in\mathbb{B}^n} w_{\beta_0\gamma_0}(1u) \le k$; therefore, there is a $u_0 \in \mathbb{B}^n$ such that $w_{\beta_0\gamma_0}(1u_0) \le \frac{k}{2^n} < \frac{1}{6}$ (if n is large enough). Let γ_1 denote the word whose u_0th bit is 1 and other bits are equal to 0. Replace the word $\beta_0\gamma_0$ by the word $\beta_0\gamma_1$. After this replacement at most $1/6$ experts change their opinions, hence, $\big|\{r \in \mathbb{B}^{p(n)} \mid P(\beta_0\gamma_1, r) = 1\}\big|/2^{p(n)}$ is greater than $1/3$. As $F(\beta_0\gamma_1) = 0$, we get a contradiction. □

2. $\exists A\ \mathrm{R}^A \cap \text{co-}\mathrm{R}^A \not\subseteq \oplus\mathrm{P}^A$.

THEOREM 12 ([**51**]). $\exists A\ \mathrm{R}^A \cap \text{co-}\mathrm{R}^A \not\subseteq \oplus\mathrm{P}^A$

PROOF. Evidently, the manifold $\mathrm{R}^A \cap \text{co-}\mathrm{R}^A$ can be represented by the following separation problem F. If $\gamma \in \mathbb{F}_1$, then $F(\gamma) = 0$. For $\gamma \in \mathbb{F}_{n+1}$, let α stand the first half of γ and β for the second half. Then

$$F(\gamma) = \begin{cases} 0 & \text{if } \#_1(\alpha) = 0,\ \#_1(\beta) > \frac{2}{3}|\beta|, \\ 1 & \text{if } \#_1(\alpha) > \frac{2}{3}|\alpha|,\ \#_1(\beta) = 0, \\ * & \text{otherwise.} \end{cases}$$

By Theorem 1, it suffices to prove that $F \not\preceq^l_m \mathrm{PARITY}$. Assume the contrary: suppose there exist a polynomial p and polylog-time predicate P such that

$$\forall n\ \forall \gamma \in \mathbb{F}_{n+1}\ F(\gamma) \le \sum_{r\in\mathbb{B}^{p(n)}} P(\gamma, r) = 1.$$

The signs $\sum$ and $+$ in this proof denote the addition modulo 2.

Fix a polylog-time machine M computing the predicate P and a large n. Let the number of queries to the word γ made by M on inputs of the form $\langle\gamma, r\rangle$, $r \in \mathbb{B}^{p(n)}$, be bounded by $k = \mathrm{poly}(n)$. Let us prove that for any fixed $r \in \mathbb{B}^{p(n)}$

the function $P(\gamma, r)$ is a polynomial of degree $\leq k$ (in the field of residues modulo 2) of variables $\gamma(v)$, $v \in \mathbb{B}^{n+1}$. Indeed,

$$P(\gamma, r) = \sum \prod_{i=1}^{k} \Big(\gamma(v(b_1 \dots b_{i-1}, r)) + b_i + 1\Big),$$

where the sum ranges over all the tuples $\langle b_1, \dots, b_k \rangle \in \mathbb{B}^k$ such that M outputs 1 if it receives the answers $b_1, \dots, b_k$ to the queries made to γ, and where $v(b_1 \dots b_i, r) \in \mathbb{B}^{n+1}$ is the index of the bit in γ queried by M if it receives the answers $b_1, \dots, b_i$ to the previous queries to γ.

Therefore, the function $\sum_{r \in \mathbb{B}^{p(n)}} P(\gamma, r)$ is a polynomial, Q, of degree at most k in the variables $\gamma(v)$. Divide the variables $\gamma(v)$, $v \in \mathbb{B}^{n+1}$ into two groups $\alpha(u)$, $u \in \mathbb{B}^n$, and $\beta(u)$, $u \in \mathbb{B}^n$, where $\alpha(u) = \gamma(0u)$ and $\beta(u) = \gamma(1u)$.

Consider two cases.

1st case: the constant term in Q is equal to zero. Set $\beta(u) = 0$ for all $u \in \mathbb{B}^n$. Let R denote the resulting polynomial of degree at most $k = \mathrm{poly}(n)$. The polynomial R has 2^n variables, has zero constant term and is equal to 1 if more than $\frac{2}{3}2^n$ variables are equal to 1. Let us derive a contradiction from the existence of such a polynomial. Let d be the degree of R. Obviosly $d > 0$. Pick a monomial f in R of degree d. Set all the variables outside f to one. The resulting polynomial is not a constant, hence it has a root. Thus there is an assignment having at least $2^n - d > \frac{2}{3}2^n$ ones on which R is zero.

The second case (the constant term in Q is equal to 1) can be reduced to the first case by adding 1 to Q. □

3. $\exists A$ co-UP$^A \not\subseteq$ IP^A. This assertion was in fact proved in [**18**] (technically speaking, a slightly weaker assertion $\exists A$ co-NP$^A \not\subseteq$ IP^A was proved in that paper). As the proof is very simple, we present it.

THEOREM 13 ([**18**]). $\exists A$ co-UP$^A \not\subseteq$ IP^A.

PROOF. By Theorem 1, it suffices to prove that the separation problem

$$F_{\text{co-UP}}(\alpha) = \begin{cases} 1 & \text{if } \#_1(\alpha) = 0, \\ 0 & \text{if } \#_1(\alpha) = 1, \\ * & \text{otherwise.} \end{cases}$$

is not in IPLOG.

Assume the contrary: suppose there exists a polylog-time Verifier V such that

$$\begin{aligned} \#_1(\alpha) = 0 \quad &\Rightarrow \exists P \ \ \mathrm{Prob}[(P,V)(\alpha) = 1] > 2/3, \\ \#_1(\alpha) = 1 \quad &\Rightarrow \forall P \ \ \mathrm{Prob}[(P,V)(\alpha) = 1] < 1/3, \end{aligned}$$

where $(P,V)(\alpha)$ stands the result output by V after the dialogue with P on input α.

Take a large n and set $\alpha_0 = 0^{2^n}$. Then there exists a P such that

$$\mathrm{Prob}[(P,V)(\alpha_0) = 1] > 2/3.$$

Consider the dialogue of P and V on input α_0. This dialogue depends on the outcome of coin tossing made by Verifier. We call different outcomes of coin tossing *experts* and the queries to α_0 made by Verifier during the dialogue with P on input

α_0 and outcome r *the queries of the expert* r *to* α. For a given $u \in \mathbb{B}^n$ call the fraction

$$\frac{|\{r \in \mathbb{B}^{p(n)} \mid \text{ makes the query } `\alpha_0(u) =?'\}|}{2^{p(n)}}$$

the *weight* of u. Obviously, if n is large enough, then there exists u having weight less than $1/3$. Change the uth bit in α_0; let α_1 denote the resulting word. Since $\mathrm{Prob}[(P,V)(\alpha_0) = 1] > 2/3$, we obtain $\mathrm{Prob}[(P,V)(\alpha_1) = 1] > 2/3 - 1/3 = 1/3$. On the other hand, this probability should be less than $1/3$. The contradiction finishes the proof. □

4. $\exists A\ \mathrm{FewP}^A \cap \text{co-FewP}^A \not\subseteq \mathrm{UP}^A$. We will prove in the next section the following stronger statement: $\exists A\ \mathrm{FewP}^A \cap \text{co-FewP}^A \not\leq_T^{p,A} \mathrm{UP}^A$.

5. $\exists A\ \text{co-R}^A \not\subseteq \mathrm{NP}^A$.

THEOREM 14 ([**51**]). $\exists A\ \text{co-R}^A \not\subseteq \mathrm{NP}^A$.

PROOF. Assume the contrary: suppose there exist a polynomial p and a polylog-time predicate $P(\alpha, r)$ such that

$$\begin{aligned} \#_1(\alpha) = 0 \quad &\Rightarrow \exists r \in \mathbb{B}^{p(\|\alpha\|)}\, P(\alpha, r) = 1 \\ \#_1(\alpha) > 2/3|\alpha| \quad &\Rightarrow \forall r \in \mathbb{B}^{p(\|\alpha\|)}\, P(\alpha, r) = 0 \end{aligned}$$

for all $\alpha \in \mathbb{F}$. We shall find an α such that $\#_1(\alpha) > (2/3)|\alpha|$ and such that $P(\alpha, r) = 1$ for some $r \in \mathbb{B}^{p(\|\alpha\|)}$. Take $\alpha_0 = 0^{2^n}$, where n is large. Then there is an $r_0 \in \mathbb{B}^{p(n)}$ such that $P(\alpha_0, r_0)$. Change the value of α_0 on all u such that the polylog-time machine computing $P(\alpha_0, r_0)$ does not query '$\alpha_0(u) =?$'. The resulting word α satisfies the desired conditions. □

6. $\exists A\ \mathrm{IP}^A \cap \text{co-IP}^A \not\subseteq \mathrm{PH}^A$. In [**1**] it was proved that $\exists A \quad \mathrm{IP}^A \not\subseteq \mathrm{PH}^A$. Some minor changes in that proof allows to prove that there exists an oracle A such that $\mathrm{IP}^A \cap \text{co-IP}^A \not\subseteq \mathrm{PH}^A$.

7. $\exists A\ \mathrm{AM}^A \cap \text{co-AM}^A \not\subseteq \mathrm{PP}^A$. This assertion will be proven in Section 7.

8. $\exists A\ \mathrm{AM}^A \not\subseteq \Sigma_2^A$. This assertion is proved in [**44**].

9. $\exists A\ \mathrm{PP}^A \not\subseteq \mathrm{PH}^A$. This follows from the fact that there is no $k \in \mathbb{N}$ such that the function $\mathrm{MAJORITY}(x_1, \ldots, x_n)$ can be represented in the form

$$\bigvee_{i_1=1}^{2^{\mathrm{polylog}(n)}} \bigwedge_{i_2=1}^{2^{\mathrm{polylog}(n)}} \cdots \bigvee_{i_{2k-1}=1}^{2^{\mathrm{polylog}(n)}} \bigwedge_{i_{2k}=1}^{2^{\mathrm{polylog}(n)}} f_{i_1 \ldots i_{2k}}(x_1, \ldots, x_n),$$

where each $f_{i_1 \ldots i_{2k}}(x_1, \ldots, x_n)$ is a variable or the negation of a variable ([**20**], [**2**], [**55**], [**25**]).

10. $\exists A\ \oplus \mathrm{P}^A \not\subseteq \mathrm{PH}^A$. This assertion is proved in [**20**], [**2**], [**55**], [**25**].

11. $\exists A\ \oplus \mathrm{P}^A \not\subseteq \mathrm{PP}^A$. This assertion is proved in [**8**]. In fact, this theorem easily follows from the assertion $\mathrm{PARITY} \not\leq_m^l \mathrm{MAJORITY}$ proved in [**37**].

12. $\forall k \geq 3\ \exists A\ \Pi_k^A \not\subseteq \Sigma_k^A$. Superpolynomial lower bounds for the size Σ_k-circuits necessary for the computation of Π_k-functions where first obtained by M. Sipser. We need a lower bound $2^{f(n)}$, where f grows faster than any polylogarithm. Such a bound is obtained in [**25**].

5. Turing reducibility between particular complexity classes

In this section we shall present all the known relativizable assertions of the form $K_1 \leq_T^p K_2$. Obviously, if $K_1 \subseteq K_2$, then $K_1 \leq_T^p K_2$, therefore all the inclusions in Figure 1 yield the assertions on Turing reducibility. Let us list all other known relativizable theorems of the form $K_1 \leq_T^p K_2$.

(1) The class K is $\leq_T^p$-reducible to the class co-K for any K, and vice versa.
(2) $\oplus\mathrm{P}^A \leq_T^p \mathrm{PP}^A$.
(3) $\mathrm{Few}^A \leq_T^p \mathrm{FewP}^A$.
(4) $\mathrm{PH}^A \leq_T^p \mathrm{PP}^A$.

The assertion (1) is evident. Both assertions (2) and (3) are simple; (3) was proved in the previous section (Lemma 1), and (2) will be proven right now. The assertion (4) was proved in [**49**].

THEOREM 15 ([**51**]). $\oplus\mathrm{P}^A \leq_T^p \mathrm{PP}^A$ *for any oracle* A.

PROOF. By Theorem 7 it suffices to prove that the language PARITY is $\preceq_T^l$-reducible to the language MAJORITY. When we prove that a problem F is $\preceq_T^l$-reducible or is not $\preceq_T^l$-reducible to a problem G it is convenient to think that the reducing pair $\langle M, f\rangle$ is a machine that works on the input α just as the machine M works on $|\alpha|$ and queries the oracle G instead of the oracle $G(f(\cdot,\alpha))$ (when M queries the value of the oracle $G(f(\cdot,\alpha))$ on a word y, we think that the new machine queries the value of G on the word $f(y,\alpha)$). Let us define the pair $\langle M, f\rangle$ reducing the function PARITY to the function MAJORITY in terms of the work of this new machine.

Having MAJORITY as oracle, we can find $\#_1(\alpha)$ in time polylog$(|\alpha|)$ as follows. Assume that $|\alpha| = 2^k$. Ask the oracle MAJORITY whether $\#_1(\alpha) \geq \frac{1}{2}|\alpha|$ is true. Assume that the answer is "yes". Then check whether $\#_1(\alpha) \geq \frac{3}{4}|\alpha|$. For that purpose take a word β consisting of $\frac{1}{2}|\alpha|$ zeros and query the oracle whether $\#_1(\alpha\beta) \geq \frac{1}{2}|\alpha\beta|$. It is easy to verify that this inequality is equivalent to the inequality $\#_1(\alpha) \geq \frac{3}{4}|\alpha|$. Repeating this process k times we find $\#_1(\alpha)$. Output 1 if $\#_1(\alpha)$ is odd and 0 otherwise. □

All known relativizable assertions of the form $K_1 \leq_T^p K_2$ are shown in Figure 2.

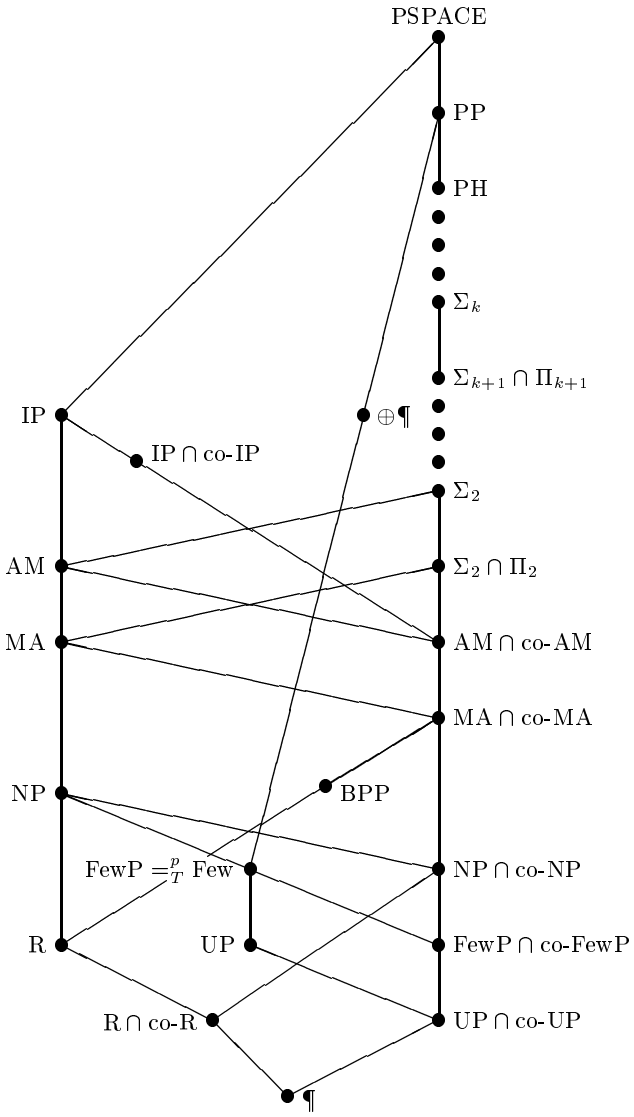
5.1. On completeness of Figure 2. It is unknown whether the Figure 2 is complete, that is, whether all true relativizable assertions of the form $K_1 \leq_T^p K_2$ are shown in Figure 2. Let us go through the following 15 assertions which must be proved to verify that Figure 2 is complete.

1. $\exists A\ \mathrm{R}^A \cap \text{co-}\mathrm{R}^A \not\leq_T^{p,A} \oplus\mathrm{P}^A$. This assertion is true and follows from the fact that the class $\oplus\mathrm{P}^A$ is downward closed under $\leq_T^{p,A}$-reductions and from the theorem $\exists A\ \mathrm{R}^A \cap \text{co-}\mathrm{R}^A \not\subseteq \oplus\mathrm{P}^A$. The fact that $\oplus\mathrm{P}^A$ is closed under $\leq_T^A$-reductions was proved in [**49**], the second theorem was proved in the previous section.

2. $\exists A\ \mathrm{UP}^A \cap \text{co-}\mathrm{UP}^A \not\leq_T^{p,A} \mathrm{BPP}^A$. This follows from the fact that the class BPP^A is downward closed under $\leq_T^{p,A}$-reductions. Indeed, in the previous section it was proved that there exists an oracle A such that $\mathrm{UP}^A \cap \text{co-}\mathrm{UP}^A \not\subseteq \mathrm{BPP}^A$.

3. $\exists A\ \mathrm{FewP}^A \cap \text{co-}\mathrm{FewP}^A \not\leq_T^{p,A} \mathrm{UP}^A$. This is true and is proved in this section.

FIGURE 2. Turing reducibility between complexity classes



4. $\exists A\ \mathrm{R}^A \not\leq_T^{p,A} \mathrm{NP}^A \cap \text{co-NP}^A$. This follows from the fact that the class $\mathrm{NP}^A \cap \text{co-NP}^A$ is downward closed under $\leq_T^{p,A}$-reductions and from the fact that $\exists A\ \mathrm{R}^A \not\subseteq \text{co-NP}^A$ (it was proved in the previous section).

5. $\exists A\ \mathrm{UP}^A \not\leq_T^{p,A} \mathrm{IP}^A \cap \text{co-IP}^A$. This follows from the fact that the class $\mathrm{IP}^A \cap \text{co-IP}^A$ is downward closed under $\leq_T^{p,A}$-reductions and from the fact $\exists A\ \mathrm{UP}^A \not\subseteq \text{co-IP}^A$ proved in the previous section.

6. $\exists A\ \Sigma_2^A \cap \Pi_2^A \not\leq_T^{p,A} \mathrm{IP}^A$. This is true and is proved in Section 6.

7. $\exists A\ \mathrm{BPP}^A \not\leq_T^{p,A} \mathrm{NP}^A$. This is proved in Section 6.

8. $\exists A \ \oplus \mathrm{P}^A \not\leq_T^{p,A} \mathrm{PH}^A$. This follows from the fact that the class PH^A is downward closed under $\leq_T^{p,A}$-reductions (the closure of the class Σ_k is included in the class Σ_{k+1}) and from the fact that $\exists A \ \oplus \mathrm{P}^A \not\subseteq \mathrm{PH}^A$.

9. $\exists A \ \mathrm{AM}^A \not\leq_T^{p,A} \Sigma_2^A \cap \Pi_2^A$. This follows from the fact that the class $\Sigma_2^A \cap \Pi_2^A$ is downward closed under $\leq_T^{p,A}$-reductions and from the fact that $\exists A \ \mathrm{AM}^A \not\subseteq \Sigma_2^A$.

10. $\exists A \ \mathrm{AM}^A \cap \text{co-}\mathrm{AM}^A \not\leq_T^{p,A} \mathrm{MA}^A$. This is true and the proof is presented in this section.

11. $\exists A \ \oplus \mathrm{P}^A \not\leq_T^{p,A} \mathrm{IP}^A$. This assertion is true and is proved in Section 6.

12. $\exists A \ \mathrm{IP}^A \cap \text{co-}\mathrm{IP}^A \not\leq_T^{p,A} \mathrm{PP}^A$. Unknown.

13. $\exists A \ \Sigma_k^A \cap \Pi_k^A \not\leq_T^{p,A} \Sigma_{k-1}^A$ $(k \geq 3)$. Unknown (for all $k \geq 3$).

14. $\exists A \ \Sigma_k^A \not\leq_T^{p,A} \Sigma_k^A \cap \Pi_k^A$ $(k \geq 3)$. This follows from the fact that the class $\Sigma_k^A \cap \Pi_k^A$ is downward closed under $\leq_T^{p,A}$-reductions and from the fact that $\exists A \ \Sigma_k^A \not\subseteq \Pi_k^A$.

15. $\exists A \ \mathrm{PH}^A \not\leq_T^{p,A} \Sigma_k^A$ $(k \geq 1)$. This assertion is true and follows from 14.

5.2. Theorems. We prove now the assertions 3 and 10.

THEOREM 16 (Joint work with An. A. Muchnik [**51**]). *There is an oracle A such that* $\mathrm{AM}^A \cap \text{co-}\mathrm{AM}^A \not\leq_T^{p,A} \mathrm{MA}^A$.

PROOF. Consider the following separation problem F. Let $\alpha = \beta\gamma$, where $\beta, \gamma \in \mathbb{F}_{2n}$, $n \in \mathbb{N}$. Then

$$F(\alpha) = \begin{cases} 1 & \text{if } \mathrm{M}_{2/3}\, x \in \mathbb{B}^n \ \exists y \in \mathbb{B}^n \ \beta(xy) = 1, \\ & \quad \mathrm{M}_{2/3}\, x \in \mathbb{B}^n \ \forall y \in \mathbb{B}^n \ \gamma(xy) = 0, \\ 0 & \text{if } \mathrm{M}_{2/3}\, x \in \mathbb{B}^n \ \forall y \in \mathbb{B}^n \ \beta(xy) = 0, \\ & \quad \mathrm{M}_{2/3}\, x \in \mathbb{B}^n \ \exists y \in \mathbb{B}^n \ \gamma(xy) = 1, \\ * & \text{otherwise.} \end{cases}$$

It is straightforward that $F \in \mathrm{AMLOGS} \cap \text{co-}\mathrm{AMLOGS}$. By Theorem 7 it suffices to prove that F is not $\preceq_T^l$-reducible to the problem F_{MA}. Recall that $F_{\mathrm{MA}}(\beta) \neq 0$ only if the norm of β, $\|\beta\| = 2k$, and in this case

$$F_{\mathrm{MA}}(\beta) = \begin{cases} 1 & \text{if } \exists r \in \mathbb{B}^k \ \ \mathrm{M}_{2/3}\, s \in \mathbb{B}^k \ \ \beta(rs) = 1, \\ 0 & \text{if } \forall r \in \mathbb{B}^k \ \ \mathrm{M}_{2/3}\, s \in \mathbb{B}^k \ \ \beta(rs) = 0, \\ * & \text{otherwise.} \end{cases}$$

The following property holds for the separation problem F_{MA} as well as for all other particular problems G under considered in the present paper. For any separation problem H, if $H \preceq_T^l G$, then there exists a pair $\langle M, f \rangle$ reducing H to G such that the following two assertions hold:

(1) the number of queries made by M for input $|\alpha|$ does not depend on the answers of the oracle and is equal to a polylogarithm of $|\alpha|$ and
(2) for all the queries '$B(u) =?$' made by M to its oracle B during the work on the input $|\alpha|$, the length of the word $f(u, \alpha)$ is the same and depends only on $|\alpha|$. That is, if we consider the pair $\langle M, f \rangle$ as a single machine, then all its queries to the oracle G during the work on the input α have the same length which depends only on $|\alpha|$.

We assume that all the pairs $\langle M, f\rangle$ considered in the sequel satisfy both properties (1) and (2).

Assume that $F \preceq^l_T F_{\mathrm{MA}}$ via a pair $\langle M, h\rangle$. Let us fix a large n (at the end of the proof we will see how large it should be). Let φ be a function from $\mathbb{B}^n$ into $\mathbb{B}^n$. Denote by $\bar\varphi$ the word of length 2^{2n} encoding the graph of φ. That is, for all $x, y \in \mathbb{B}^n$, $\bar\varphi(xy)$ is equal to 1 if $y = \varphi(x)$, and is equal to 0 otherwise. We will take words of the form $\bar\varphi\bar\psi$, where φ and ψ are partial functions from $\mathbb{B}^n$ into $\mathbb{B}^n$, as arguments of F. Let m be the number of queries to the oracle F_{MA} made by M for the input of this form. As $|\bar\varphi\bar\psi| = 2^{2n+1}$, we see that $m = \mathrm{poly}(n)$. We define a binary sequence $b_1, \dots, b_m$, partial functions $\varphi, \psi : \mathbb{B}^n \to \mathbb{B}^n$, and total functions $f_0, g_0 : \mathbb{B}^n \to \mathbb{B}^n$ such that the sequence of oracle answers to the queries made by $\langle M, h\rangle$ to the oracle F_{MA} during the work on both inputs $\bar f_0\bar\psi$ and $\bar\varphi\bar g_0$ is equal to $b_1, \dots, b_m$. The cardinalities of domains of the functions φ and ψ will be bounded by a polynomial of n, therefore, for n large enough we shall get $|\operatorname{Dom}(\varphi)|$, $|\operatorname{Dom}(\psi)| < \frac{1}{3}2^n$. Obviously, this is a contradiction because $\langle M, h\rangle$ reduces F to F_{MA} and $F(\bar f_0\bar\psi) = 1$, $F(\bar\varphi\bar g_0) = 0$.

Let $2k$ denote the norm of queries made by the pair $\langle M, h\rangle$ to the oracle F_{MA} (that is, the norm of α's such that $\langle M, h\rangle$ queries '$F_{\mathrm{MA}}(\alpha) =?$') during the work on inputs of the norm $2n + 1$ (obviously, $k \le \mathrm{poly}(n)$). Define the following auxiliary separation problem on words of the norm $2k$:

$$G(\beta) = \begin{cases} 1 & \text{if } \exists r \in \mathbb{B}^k \ \ \mathrm{M}_{1/2}\, s \in \mathbb{B}^k \ \ \beta(rs) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Obviously, G is a solution to F_{MA}.

Take arbitrary functions $f, g : \mathbb{B}^n \to \mathbb{B}^n$. Run the machine M on the input 2^{2n+1} with the oracle $G(h(\cdot, \bar f\bar g))$. Let $e(f, g)$ denote the sequence of oracle answers. Since the length of the word $e(f, g)$ is equal to m, there exists a word e_0 of length m such that

$$\frac{\big|\{\langle f, g\rangle \mid e(f, g) = e_0\}\big|}{2^{2n(2^n)}} \ge \frac{1}{2^m}.$$

Let $\mathcal{K}$ denote the set $\{\langle f, g\rangle \mid e(f, g) = e_0\}$. Obviously, for all the pairs $\langle f, g\rangle \in \mathcal{K}$ the queries to the oracle $G(h(\cdot, \bar f\bar g))$ made by M are the same. Let $v_1, \dots, v_m$ denote those queries (that is, the queries are '$G(h(v_1, \bar f\bar g)) =?$', ..., '$G(h(v_m, \bar f\bar g)) =?$'). Let $P(\alpha, v, u)$ denote the uth symbol of the word $h(v, \alpha)$, ($\alpha \in \mathbb{F}_{2n+1}$, $u \in \mathbb{B}^{2k}$). Let $b_1, \dots, b_m$ denote the bits of the word e_0.

Let I stand for the set $\{i \mid i \le m,\ b_i = 1\}$. We know that for any $i \in I$ and for all $\langle f, g\rangle \in \mathcal{K}$, there exists $r_i \in \mathbb{B}^k$ such that $\mathrm{M}_{1/2}\, s \in \mathbb{B}^k \ \ P(\bar f\bar g, v_i, r_i s) = 1$. Again, we can find a set $\mathcal{K}' \subseteq \mathcal{K}$ such that for any $i \in I$ and for all $\langle f, g\rangle \in \mathcal{K}'$, that r_i is the same and such that $\frac{|\mathcal{K}'|}{|\mathcal{K}|} \ge \frac{1}{2^{km}}$. Evidently, $\frac{|\mathcal{K}'|}{2^{2n(2^n)}} \ge \frac{1}{2^{km+m}}$. Let ε denote the number $\frac{1}{2^{km+m}}$. We consider the set $\mathcal{K}'$ as a planar set of area at least ε. Obviously, there exists a vertical section of the set $\mathcal{K}'$ of length at least ε and there exists a horizontal section of the set $\mathcal{K}'$ of length at least ε. That is, there exist functions f_0, g_0 and families of functions $\mathcal{F}'$ and $\mathcal{G}'$ such that $|\mathcal{F}'| \ge \varepsilon 2^{n\cdot 2^n}$, $|\mathcal{G}'| \ge \varepsilon 2^{n\cdot 2^n}$, $\{f_0\} \times \mathcal{G}' \subseteq \mathcal{K}'$, $\mathcal{F}' \times \{g_0\} \subseteq \mathcal{K}'$.

Define now a partial function $\varphi : \mathbb{B}^n \to \mathbb{B}^n$ and a family $\mathcal{F}$ consisting of (total) functions from $\mathbb{B}^n$ into $\mathbb{B}^n$. Assume that x, y are in $\mathbb{B}^n$. Denote by $\mathrm{popularity}_{\mathcal{F}}(x, y)$ the fraction $\big|\{f \in \mathcal{F} \mid f(x) = y\}\big| \big/ \big|\mathcal{F}\big|$. Set first $\varphi = \emptyset$, $\mathcal{F} = \mathcal{F}'$. Then, while there exists a pair $\langle x, y\rangle \in (\mathbb{B}^n \setminus \operatorname{Dom}(\varphi)) \times \mathbb{B}^n$ such that $\mathrm{popularity}_{\mathcal{F}}(x, y) \ge 2^{-n+1}$, we

choose such a pair $\langle x, y\rangle$, extend the partial function φ to x by setting $\varphi(x) = y$, and delete from $\mathcal{F}$ all the functions f such that $f(x) \neq y$.

We claim that the resulting φ, $\mathcal{F}$ have the following properties:

(1) $\mathcal{F} \subseteq \mathcal{F}'$,
(2) all the functions from the set $\mathcal{F}$ extend φ,
(3) $\mathrm{popularity}_{\mathcal{F}}(x, y) < 2^{-n+1}$ for all $\langle x, y\rangle \in (\mathbb{B}^n \setminus \mathrm{Dom}(\varphi)) \times \mathbb{B}^n$,
(4) $|\mathrm{Dom}(\varphi)| \leq -\log_2(|\mathcal{F}'|/2^{n(2^n)}) \leq km + m = \mathrm{poly}(n)$.

The properties (1)–(3) are evident. Let us prove the assertion (4). Let $\mathcal{F}_i$, φ_i, x_i, and y_i denote the values of the variables $\mathcal{F}$, φ, x, and y, respectively, after the ith iteration of the while-loop. Then

$$\frac{|\mathcal{F}_{i+1}|}{\bigl|\{f : \mathbb{B}^n \to \mathbb{B}^n \mid f \text{ extends } \varphi_{i+1}\}\bigr|} \geq \frac{2|\mathcal{F}_i|}{\bigl|\{f : \mathbb{B}^n \to \mathbb{B}^n \mid f \text{ extends } \varphi_i\}\bigr|}$$

because

$$|\mathcal{F}_{i+1}| \geq 2^{-n+1}|\mathcal{F}_i|$$

and

$$\bigl|\{f : \mathbb{B}^n \to \mathbb{B}^n \mid f \text{ extends } \varphi_{i+1}\}\bigr| = 2^{-n}\bigl|\{f : \mathbb{B}^n \to \mathbb{B}^n \mid f \text{ extends } \varphi_i\}\bigr|.$$

Since

$$|\mathcal{F}_{i+1}|/\bigl|\{f : \mathbb{B}^n \to \mathbb{B}^n \mid f \text{ extends } \varphi_{i+1}\}\bigr| \leq 1$$

for all i, the number of iterations of the while-loop is at most $-\log_2(|\mathcal{F}'|/2^{n(2^n)})$.

Apply the same procedure to the family $\mathcal{G}'$ and let $\mathcal{G}, \psi$ denote the result.

Let us prove that

$$F_{\mathrm{MA}}(h(v_i, \bar{\varphi}\bar{g}_0)) = b_i$$

for all $i \leq m$. Take an arbitrary $i \leq m$. Consider two cases.

1st case: $b_i = 1$. Then we know that

$$\mathrm{M}_{1/2}\, s \in \mathbb{B}^k \;\; P(fg_0, v_i, r_i s) = 1 \tag{48}$$

for all the $f \in \mathcal{F}$. By the definition of $\preceq^l_T$-reducibility, $F_{\mathrm{MA}}(h(v_i, \bar{\varphi}\bar{g}_0)) \neq *$ (if n is so large that $|\mathrm{Dom}(\varphi)| < \frac{1}{3}2^n$). Assume that $F_{\mathrm{MA}}(h(v_i, \bar{\varphi}\bar{g}_0)) = 0$. Then

$$\mathrm{M}_{2/3}\, s \in \mathbb{B}^k \;\; P(\bar{\varphi}\bar{g}_0, v_i, r_i s) = 0. \tag{49}$$

Let N be the machine that for any given $\alpha \in \mathbb{F}_{2n+1}$, $v \in \mathbb{B}^*$, $u \in \mathbb{B}^{2k}$, computes $P(\alpha, v, u)$ in time $\mathrm{poly}(|v|+n)$. If α has the form $\bar{\eta}\bar{\theta}$, where η, θ are partial functions from $\mathbb{B}^n$ into $\mathbb{B}$, then the queries made by N to α have one of the two following forms: '$\eta(x) = y$?' and '$\theta(x) = y$?', where $x, y \in \mathbb{B}^n$. For $x, y \in \mathbb{B}^n$, let $w_{\varphi g_0}(x, y)$ denote the fraction

$$\bigl|\{s \in \mathbb{B}^n \mid N \text{ on the input } \langle \bar{\varphi}\bar{g}_0, v_i, r_i s\rangle \text{ queries '}\varphi(x) = y\text{?'}\}\bigr|/2^n.$$

Obviously, $\sum_{x,y \in \mathbb{B}^n} w_{\varphi g_0}(x, y) \leq \mathrm{poly}(n)$. Then for any $f \in \mathcal{F}$ the assertions (48) and (49) imply that

$$\sum_{x \in \mathbb{B}^n \setminus \mathrm{Dom}(\varphi)} w_{\varphi g_0}(x, f(x)) \geq \frac{1}{6};$$

therefore,

$$\frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F},\; x \in \mathbb{B}^n \setminus \mathrm{Dom}(\varphi)} w_{\varphi g_0}(x, f(x)) \geq \frac{1}{6}.$$

Let us rewrite the left hand side of the last inequality as follows:

$$\begin{aligned}
&\frac{1}{|\mathcal{F}|} \sum_{f\in\mathcal{F},\ x\in\mathbb{B}^n\setminus \mathrm{Dom}(\varphi)} w_{\varphi g_0}(x, f(x)) \\
&= \sum_{x\in\mathbb{B}^n\setminus \mathrm{Dom}(\varphi),\ y\in\mathbb{B}^n} w_{\varphi g_0}(x,y)\cdot \mathrm{popularity}_{\mathcal{F}}(x,y) \\
&\le 2^{-n+1} \sum_{x\in\mathbb{B}^n\setminus \mathrm{Dom}(\varphi),\ y\in\mathbb{B}^n} w_{\varphi g_0}(x,y) \le 2^{-n+1}\mathrm{poly}(n).
\end{aligned}$$

If n is large enough, we get the contradiction: $2^{-n+1}\mathrm{poly}(n) \ge \frac{1}{6}$.

2nd case: $b_i = 0$. We know that $\left|\{s\in\mathbb{B}^k \mid P(\bar f \bar g_0, v_i, rs) = 1\}\right|/2^k$ is at most $1/2$ for all $r\in\mathbb{B}^k$ and for all $f\in\mathcal{F}$. Assume that $F_{\mathrm{MA}}(h(v_i, \bar\varphi\bar g_0)) = 1$, that is, there exists $r\in\mathbb{B}^k$ such that

$$\mathrm{M}_{2/3}\, s\in\mathbb{B}^k\ P(\bar\varphi\bar g_0, v_i, rs) = 1.$$

Then just as was done in the first case we can get a contradiction.

In the same way we can prove that

$$F_{\mathrm{MA}}(h(v_i, \bar f_0\bar\psi)) = b_i$$

for all $i\le m$. □

THEOREM 17 ([**51**]). *There is an oracle A such that* $\mathrm{FewP}^A \cap \text{co-}\mathrm{FewP}^A \not\subseteq_T^{p,A} \mathrm{UP}^A$.

PROOF. To demonstrate the idea let us prove first that there exists an oracle A such that $\mathrm{FewP}^A \cap \text{co-}\mathrm{FewP}^A \not\subseteq \mathrm{UP}^A$. Define the following separation problem F. If $\|\beta\| = \|\gamma\|$, then

$$F(\beta\gamma) = \begin{cases} 1 & \text{if } 1\le \#_1(\beta)\le 2,\ \#_1(\gamma) = 0,\\ 0 & \text{if } 1\le \#_1(\gamma)\le 2,\ \#_1(\beta) = 0,\\ * & \text{otherwise.}\end{cases}$$

It is straightforward that $F\in \mathrm{FewPLOGS}\cap\text{co-}\mathrm{FewPLOGS}$. By Theorem 1, it is sufficient to prove that $F\notin \mathrm{UPLOGS}$. Assume the contrary: suppose there exist a polynomial p and a polylog-time predicate P such that

$$F(\beta\gamma) = 1 \Longrightarrow \exists! r\in\mathbb{B}^{p(\|\beta\|)}\ P(\beta\gamma, r) = 1,$$
$$F(\beta\gamma) = 0 \Longrightarrow \forall r\in\mathbb{B}^{p(\|\beta\|)}\ P(\beta\gamma, r) = 0.$$

Take $\beta_0 = \gamma_0 = 0^{2^n}$, where n is large. Consider two cases.

1st case: $\exists r\in\mathbb{B}^{p(n)}\ P(\beta_0\gamma_0, r) = 1$. Pick an expert r_0 such that $P(\beta_0\gamma_0, r_0) = 1$. If n is large enough, then there exists $u\in\mathbb{B}^n$, such that r_0 does not query '$\gamma_0(u) =?$'. Set $\gamma_0(u) = 1$ and get a contradiction.

2nd case: $\forall r\ P(\beta_0\gamma_0, r) = 0$. Let us prove that if n is large enough, then there exists $\beta\in\mathbb{F}_n$ such that $\#_1(\beta) = 2$ and $|\{r\in\mathbb{B}^{p(n)} : P(\beta\gamma_0, r) = 1\}|\ge 2$. For a $u\in\mathbb{B}^n$, let β_1^u denote the word whose uth bit is 1 and other bits are 0. For all u we have $F(\beta_1^u\gamma_0) = 1$, therefore, for all $u\in\mathbb{B}^n$ there is unique $r = r_u\in\mathbb{B}^{p(n)}$ such that $P(\beta_1^u\gamma_0, r) = 1$. Call the set of all $v\in\mathbb{B}^n$ such that the expert r_u queries '$\beta_1^u(v) =?$' the 1-*base of* u, and call the set of all $v\in\mathbb{B}^n$ such that the expert r_u queries '$\beta_0(v) =?$' the 0-*base of* u. Let $B_1(u)$ and $B_0(u)$ denote the 1-base and 0-base of u, respectively.

Let us prove that if n is large enough, then there exist $u_1, u_2 \in \mathbb{B}^n$ such that $u_1 \notin B_0(u_2) \cup B_1(u_2)$, and $u_2 \notin B_1(u_1)$. Indeed, the numbers of elements in all bases are bounded by a polynomial of n, say $q(n)$. Take random u_1, u_2 (independent and uniformly distributed). We have

$$\mathrm{Prob}[u_1 \in B_0(u_2)] \le \frac{q(n)}{2^n},$$
$$\mathrm{Prob}[u_1 \in B_1(u_2)] \le \frac{q(n)}{2^n},$$
$$\mathrm{Prob}[u_2 \in B_1(u_1)] \le \frac{q(n)}{2^n}.$$

Therefore, with probability close to 1, none of the three events occurs.

Fix u_1 and u_2 such that $u_1 \notin B_0(u_2) \cup B_1(u_2)$ and $u_2 \notin B_1(u_1)$. Define the word β_2 as follows: $\beta_2(u_1) = \beta_2(u_2) = 1$ and $\beta_2(v) = 0$ for $v \ne u_1, u_2$. Then $\beta_2\gamma_0 \in \mathrm{D}(F)$ and $P(\beta_2\gamma_0, r_{u_1}) = P(\beta_1\gamma_0, r_{u_2}) = 1$ (since $u_2 \notin B_1(u_1)$, $u_1 \notin B_1(u_2)$). We have $r_{u_1} \ne r_{u_2}$ because $P(\beta_0^{u_1}\gamma_0, r_{u_1}) = 1$ and $P(\beta_0^{u_1}\gamma_0, r_{u_2}) = 0$ (since $u_1 \notin B_0(u_2)$). The contradiction shows that F is not in UPLOGS.

Let us prove now that F is not $\preceq_T^l$-reducible to F_{UP}. Recall that

$$F_{\mathrm{UP}}(\alpha) = \begin{cases} 1 & \text{if } \#_1(\alpha) = 1, \\ 0 & \text{if } \#_1(\alpha) = 0, \\ * & \text{otherwise.} \end{cases}$$

Assume that F is $\preceq_T^l$-reducible to F_{UP} via the pair $\langle M, f\rangle$. Then, by definition of the$\preceq_T^l$-reducibility we have

$$\forall \alpha \in \mathrm{D}(F)\ \forall e \in \mathbb{B}^*\ \#_1(f(e,\alpha)) \in \{0,1\}. \tag{50}$$

Fix $n \in \mathbb{N}$ and set $\alpha_0 = 0^{2^{n+1}}$. Let D_1 denote the set $\{\alpha \in \mathbb{F}_{n+1} : \#_1(\alpha) = 1\}$. Evidently, $D_1 \subseteq \mathrm{D}(F)$. We construct a set $U \subseteq \mathbb{B}^{n+1}$ having at most $\mathrm{poly}(n)$ elements such that for all α in D_1 that are equal to zero on all the elements of U, the sequence of answers to queries to the oracle F_{UP} made by $\langle M, f\rangle$ during the work on input α is the same.

Let m denote the number of queries made by M to oracle during the work on the input 2^{n+1}. Define the binary sequence $b_1, \ldots, b_m$ and the sequence $v_1, \ldots, v_m$ of binary words by induction as follows. Let v_i be the word such that the machine M asks '$F_{\mathrm{UP}}(f(v_i, \alpha)) =$?' during the work on input 2^{n+1} after getting the answers $b_1, \ldots, b_{i-1}$ to the previous questions to the oracle and let

$$b_i = \begin{cases} 1 & \text{if } \#_1(f(v_i, \alpha_0)) \ge 1, \\ 0 & \text{otherwise.} \end{cases}$$

For any i let us find a set U_i such that $F_{\mathrm{UP}}(f(v_i, \alpha)) = b_i$ for all $\alpha \in D_1$ that are equal to zero on all the elements of U_i. Then we shall set $U = \bigcup_{i=1}^m U_i$.

Fix an arbitrary i not exceeding m and construct U_i. By the definition of $\preceq_T^l$-reducibility, there exists a machine N that for any given $\langle \alpha, v_i, r\rangle$ (where $|r| = \|f(v_i, \alpha)\|$) produces rth bit of the word $f(v_i, \alpha)$ in time polylogarithmic in $|\alpha|$.

Consider two cases.

1st case: $b_i = 1$, that is, $\#_1(f(v_i, \alpha_0)) \ge 1$. Pick a word r such that $f(v_i, \alpha_0)(r) = 1$. Include in U_i all the words $u \in \mathbb{B}^n$ such that N asks '$\alpha_0(u) =$?' during the computation on the input $\langle \alpha_0, v_i, r\rangle$. Then $\#_1(f(v_i, \alpha)) \ge 1$ for all

$\alpha \in \mathbb{F}_{n+1}$ that are equal to zero on all the elements of U_i. By (50), this means that $\#_1(f(v_i, \alpha)) = 1$ for all $\alpha \in D_1$ that are equal to zero on all the elements of U_i.

2nd case: $\#_1(f(v_i, \alpha_0)) = 0$. Let $\beta_0 = \gamma_0 = 0^{2^n}$. We use all notation introduced in the proof of the first part. Let us prove that the set $V = \{u \in \mathbb{B}^n \mid \#_1(f(v_i, \beta_1^u \gamma_0)) = 1\}$ has no more than $\mathrm{poly}(n)$ elements. Namely, we claim that $|V| \leq 3q(n)$, where $q(n)$ is a polynomial upper bound for the number of queries of the form '$\alpha_0(v) =?$' made by N during the computation on any input $\langle \alpha_0, v_i, r\rangle$ (where $|r| = \|f(v_i, \alpha_0)\|$). Assume the contrary: suppose that $|V| > 3q(n)$. For a $u \in V$, let r_u denote the word r such that rth bit of word $f(v_i, \beta_1^u \gamma_0)$ is 1. Let $B_0(u)$ $[B_1(u)]$ denote the set of all v such that N queries '$\alpha_0(v) =?$' ['$\beta_1^u \gamma_0(v) =?$'] at some moment during the computation on the input $\langle \alpha_0, v_i, r_u\rangle$ $[\langle \beta_1^u \gamma_0, v_i, r_u\rangle]$. Then $|B_0(u)|, |B_1(u)| \leq q(n)$ for all $u \in V$. Take random independent u_1, u_2 uniformly distributed in V. The probability of event "$u_1 \notin B_0(u_2) \cup B_1(u_2),\ u_2 \notin B_1(u_1)$" is at least $1 - 3q(n)/|V| > 0$. Just as it was done in the proof of the first part, we can construct a word $\beta_2 \in \mathrm{D}(F)$ such that $\#_1(f(v_i, \beta_2 \gamma_0)) \geq 2$, which contradicts (50).

Similarly we can construct a set V' having $\mathrm{poly}(n)$ elements such that

$$\#_1(f(v_i, \beta_0 \gamma_1^u)) = 1$$

for all $u \in \mathbb{B}^n \setminus V'$. Set $U = V \cup V'$.

If n is so large that $2^n > |U|$, there exist $\alpha_1, \alpha_2 \in D_1$ such that $F(\alpha_1) = 1$, $F(\alpha_2) = 0$ and both α_1 and α_2 are equal to zero on all the elements of U. We have $\langle M, f\rangle^{F_{\mathrm{UP}}}(\alpha_1) = \langle M, f\rangle^{F_{\mathrm{UP}}}(\alpha_2)$. The contradiction thus obtained proves the theorem. □

6. Complete languages in particular complexity classes

It is well known that the following classes

$$\mathrm{P}^A,\ \mathrm{NP}^A,\ \text{co-NP}^A,\ \Sigma_k^A,\ \Pi_k^A,\ \mathrm{PSPACE}^A,\ \oplus\mathrm{P}^A,\ \mathrm{PP}^A. \tag{51}$$

have $\leq_m^p$-complete languages (also called m-complete languages). All the known theorems of the form "K_2^A is $\leq_m^{p,A}$-hard (or $\leq_T^{p,A}$-hard) for the class K_1^A for all A" can be obtained using the following two rules:

(1) a class K_2^A is $\leq_m^p$-hard for the class K_1^A if there is a class K^A in the list (51) such that $K_1^A \subseteq K^A \subseteq K_2^A$;
(2) a class K_2^A is $\leq_T^p$-hard for the class K_1^A if there exists a class K^A in the list (51) such that $K_1^A \leq_T^p K^A \leq_T^p K_2^A$.

It is unknown whether all true assertions of the form "K_2^A is $\leq_m^{p,A}$-hard $[\leq_T^{p,A}$-hard] for the class K_1^A for all A", where K_1^A and K_2^A are classes shown in Figure 1, can be obtained by the rules (1) and (2). We have already proved some assertions which are necessary in order to get a positive answer to the above question. Indeed, if K_2^A is $\leq_m^{p,A}$-hard for K_1^A, then $K_1^A \subseteq K_2^A$ (since all the classes under consideration are downward closed under $\leq_m^{p,A}$-reductions). Therefore, if we have proved that $K_1^A \not\subseteq K_2^A$ for some A, then we have also proved that K_2^A is not $\leq_m^{p,A}$-hard for the class K_1^A for some A. Similarly, if $K_1^A \not\leq_T^{p,A} K_2^A$ for some A, then K_2^A is not $\leq_T^{p,A}$-hard for K_1^A for the same A. Let us go through remaining assertions which must be proved to obtain the positive answer to the above question.

1. $\exists A\ \Sigma_2^A \cap \Pi_2^A$ is not $\leq_T^{p,A}$-hard for the class Few^A. It is unknown whether this is true or not (and for $\leq_m^{p,A}$ reductions too).

2. $\exists A$ IP^A is not $\leq_T^{p,A}$-hard for the class BPP^A. This was proved by An. A. Muchnik together with the author. The proof is presented in this section.

3. $\exists A$ $\mathrm{IP}^A \cap \text{co-}\mathrm{IP}^A$ is not $\leq_T^{p,A}$-hard for the class $\mathrm{R}^A \cap \text{co-}\mathrm{R}^A$. This assertion is true and was proved in [**28**].

4. $\exists A$ $\mathrm{IP}^A \cap \text{co-}\mathrm{IP}^A$ is not $\leq_T^{p,A}$-hard for the class $\mathrm{UP}^A \cap \text{co-}\mathrm{UP}^A$. This is true. The proof is presented in this section.

5. $\exists A$ $\Sigma_2^A \cap \Pi_2^A$ is not $\leq_T^{p,A}$-hard for the class BPP^A. It is unknown whether this is true or not (and for $\leq_m^{p,A}$ reductions too).

6. $\exists A$ Few^A is not $\leq_T^{p,A}$-hard for the class $\mathrm{UP}^A \cap \text{co-}\mathrm{UP}^A$. This assertion is true. In [**28**], it was proved that there exists an oracle A such that the class FewP^A is not $\leq_T^{p,A}$-hard for the class $\mathrm{UP}^A \cap \text{co-}\mathrm{UP}^A$. Since $\mathrm{Few}^A \leq_T^p \mathrm{FewP}^A$ this implies that Few^A is also not $\leq_T^{p,A}$-hard for the class $\mathrm{UP}^A \cap \text{co-}\mathrm{UP}^A$ for some A.

7. $\exists A$ $\Sigma_k^A \cap \Pi_k^A$ is not $\leq_T^{p,A}$-hard for the class $\Sigma_k^A \cap \Pi_k^A$ ($k \geq 3$). It is unknown whether this is true or not. (For $k = 1, 2$ this follows from 4 and 1, respectively.)

8. $\exists A$ PP^A is not $\leq_T^{p,A}$-hard for the class $\mathrm{IP}^A \cap \text{co-}\mathrm{IP}^A$. It is unknown whether this is true. Since PP^A has $\leq_m^p$-complete language, this assertion is equivalent to the assertion $\exists A$ $\mathrm{IP}^A \cap \text{co-}\mathrm{IP}^A \not\leq_T^{p,A} \mathrm{PP}^A$.

Let us turn to the proofs. We use the following lemma.

LEMMA 2. *If F and G are nondegenerate separation problems such that $F \notin$ n.u.PLOGS and* n.u.LOG(G) = n.u.PLOG*, then there exists an oracle A such that the class* $\mathrm{POLY}^A(G)$ *is not* $\leq_T^{p,A}$*-hard for the class* $\mathrm{POLY}^A(F)$.

PROOF. By the Theorem 8, it suffices to prove that the separation problem F is $\preceq_T^l$-reducible to no language in the class $\mathrm{LOG}(G)$. Assume that there is a language $H \in \mathrm{LOGS}(G)$ such that $F \preceq_T^l H$. Then H is in n.u.LOGS(G) = n.u.PLOG $\subseteq$ n.u.PLOGS. Therefore F is in n.u.PLOGS because the class n.u.PLOGS is downward closed under $\preceq_T^l$-reductions. □

Assertions 3 and 4 can be easily derived from the Lemma 2, Theorem 7, and the following theorem.

THEOREM 18. n.u.IPLOG $\cap$ co-n.u.IPLOG = n.u.PLOG.

We omit the proof of Theorem 18 because this proof is an easy generalization of that of Nisan's result (see [**40**]) n.u.BPPLOG = n.u.PLOG.

The assertion 6 can be proved in a similar way.

THEOREM 19 ([**51**]). n.u.FewLOG = n.u.PLOG.

PROOF. Assume F is in n.u.FewLOG. Then there are predicates P, R computable in polylogaritmic number of queries and polylogarithms $p(n)$, $q(n)$ such that

$$\left|\{r \in \mathbb{B}^{p(|\alpha|)} : P(\alpha, r) = 1\}\right| < q(|\alpha|),$$
$$F(\alpha) = R(\alpha, \left|\{r \in \mathbb{B}^{p(|\alpha|)} : P(\alpha, r) = 1\}\right|)$$

for all $\alpha \in \mathrm{D}(F)$. Let $n = |\alpha|$. Let us call words in the set $\mathbb{B}^{p(n)}$ *experts*. We say that an expert r *accepts* α if $P(\alpha, r) = 1$. We claim that probing at most polylog(n) bits of α we can find all the experts accepting α. This easily implies that $F \in$ n.u.PLOG.

There is a polylogarithm $s(n)$ such that for any expert r there is a Boolean decision tree T_r of height $s(n)$ computing $P(\alpha, r)$ for all $r \in \mathbb{F}_n$. Fix T_r for all r. Call any partial function $\varphi : \mathbb{B}^n \to \mathbb{B}$ a *segment*. Two segments are *consistent* if they have a common extension. Any decision tree T_r queries the value of α on $k = s(n)$ arguments, say $u_1, \ldots, u_k$. Call the segment $\{\langle u_i, \alpha(u_i)\rangle \mid i \le k\}$ the *information of r about α*. Call the information of r about any α accepted by r the *certificate of expert r*. A *certificate* is a certificate of some expert.

We find all experts accepting α for any given α as follows. For any subset U of $\mathbb{B}^n$ let $\Phi_U(\alpha)$ denote the set of all certificates having the same value on elements of U as α. Our goal is to construct a set U such that $\Phi_U(\alpha)$ is the set of all certificates consistent with α. Let us start with $U = \emptyset$. Repeat k times the following loop.

Take any maximal (with respect to inclusion) subset $\Psi = \{\varphi_1, \ldots, \varphi_j\}$ of $\Phi_U(\alpha)$ such that the sets $\mathrm{Dom}(\varphi_1) \setminus U, \ldots, \mathrm{Dom}(\varphi_j) \setminus U$ are pairwise disjoint. Then $j \le q(n)$ because there exists a $\beta \in \mathbb{F}_n$ that is consistent with all certificates from Ψ and $\varphi_1, \ldots, \varphi_j$ are certificates of different experts (because certificates of any expert are pairwise inconsistent). Ask the value of α on all the elements of the set $V = (\mathrm{Dom}(\varphi_1) \cup \ldots \cup \mathrm{Dom}(\varphi_j)) \setminus U$. Since Ψ is maximal, the domain of any certificate $\varphi \in \Phi_U(\alpha) \setminus \Psi$ intersects with V. Set $U = U \cup V$. Note that $|\mathrm{Dom}(\varphi) \setminus U|$ has decreased for any certificate $\varphi \in \Phi_U(\alpha) \setminus \Psi$ and $\mathrm{Dom}(\varphi) \setminus U$ has become empty for any certificate $\varphi \in \Psi$. The loop is completed.

The value $\max\{|\mathrm{Dom}(\varphi) \setminus U| \mid \varphi \in \Phi_U(\alpha)\}$ decreases or remains zero after each iteration of the above loop. Therefore, $\mathrm{Dom}(\varphi) \subseteq U$ for any $\varphi \in \Phi_U(\alpha)$ after k iterations of the loop. This means that $\Phi_U(\alpha)$ is the set of *all* certificates consistent with α. Obviously, an expert accepts α iff some its certificate is consistent with α. Hence we know all the experts accepting α. It remains to note that during each iteration of the loop we make at most $q(n) \cdot k$ queries to α. □

The assertion 2 cannot be derived from the Lemma 2 since n.u.PLOG $\subset$ n.u.NPLOG $\subseteq$ n.u.IPLOG.

THEOREM 20 (Joint work with An. A. Muchnik [**51**]). *There is an oracle A such that* IP^A *is not* $\le_T^{p,A}$*-hard for the class* BPP^A.

We prove this theorem together with the other unproved theorems of the previous section.

THEOREM 21 ([**51**]). $\exists A\ \mathrm{BPP}^A \not\le_T^{p,A} \mathrm{NP}^A$.

THEOREM 22 ([**51**]). $\exists A\ \oplus\mathrm{P}^A \not\le_T^{p,A} \mathrm{IP}^A$.

THEOREM 23 ([**51**]). $\exists A\ \Sigma_2^A \cap \Pi_2^A \not\le_T^{p,A} \mathrm{IP}^A$.

PROOF OF THEOREMS 20–23. In fact, Theorem 21 follows from Theorem 20 because the class NP^A has a $\le_m^p$-complete language and $\mathrm{NP}^A \subseteq \mathrm{IP}^A$. Nevertheless we prove Theorem 21 first. By Theorem 8 it suffices to prove that $F_{\mathrm{BPP}} \not\preceq_T^l F_{\mathrm{NP}}$.

Assume that $F_{\mathrm{BPP}} \preceq_T^l F_{\mathrm{NP}}$. Let $\langle M, f\rangle$ be a reducing pair. Fix a large integer n. Let m denote the number of queries made by M to oracle during the work on input 2^n. Obviously, $m \le \mathrm{poly}(n)$. Assume that α is in $\mathbb{F}_n$. Run the machine M with the oracle $F_{\mathrm{NP}}(f(\cdot, \alpha))$ on the input 2^n. Let $e(\alpha)$ denote the sequence of oracle answers received by M in that computation ($e(\alpha) \in \mathbb{B}^m$). Let α_0 be an $\alpha \in \mathbb{F}_n$ with lexicographical greatest $e(\alpha)$. Let $e_0 = b_1^0 \ldots b_m^0$ stand for $e(\alpha_0)$ and $v_1, \ldots, v_m$ for the queries of M to the oracle $F_{\mathrm{NP}}(f(\cdot, \alpha_0))$ (more precisely, the

queries are '$F_{\mathrm{NP}}(f(v_i,\alpha_0))=?$'). Let I be the set of all the indices $i\le k$ such that $F_{\mathrm{NP}}(f(v_i,\alpha_0))=1$, that is, $\#_1 f(v_i,\alpha_0)>0$. For each $i\in I$ fix a word t_i such that $f(v_i,\alpha_0)(t_i)=1$. Let $q(n)$ be a polynomial bounding the time of bit-computation of the function $f(v_i,\alpha)$ for $\alpha\in\mathbb{F}_n$, $i\le m$. Obviously, for any $i\in I$ there is a set $U_i\subseteq\mathbb{B}^n$ having at most $q(n)$ elements such that $f(v_i,\alpha)(t_i)=1$ for all α having the same values on all the elements of U_i as α_0 has. Set $U=\bigcup_{i\in I}U_i$. Evidently, $|U|\le mq(n)=\mathrm{poly}(n)$. We have $F_{\mathrm{NP}}(f(v_i,\alpha))=1$ for all $i\le m$ such that $b_i^0=1$ and for all $\alpha\in\mathbb{F}_n$ having the same values on all the words in U as α_0 has.

We claim that, moreover, $e(\alpha)=e(\alpha_0)$ for all $\alpha\in\mathbb{F}_n$ having the same values on all the words in U as α_0 has. Assume the contrary. Let α be a counterexample. Let $b_1\dots b_m$ be the bits of $e(\alpha)$. Let i be the least number such that $b_i\ne b_i^0$. Then, since e_0 is the lexicographically greatest word among the word of the form $e(\alpha)$, $\alpha\in\mathbb{F}_n$, we have $b_i=0$, $b_i^0=1$. As α and α_0 have the same values on all the words in U, we have $F_{\mathrm{NP}}(f(v_i,\alpha))=1$. On the other hand $b_1^0\dots b_{i-1}^0=b_1\dots b_{i-1}$, therefore the ith query to the oracle made by M during the computation on the input 2^n with the oracle $F_{\mathrm{NP}}(f(\cdot,\alpha))$ is '$F_{\mathrm{NP}}(f(v_i,\alpha))=?$'. Consequently, $F_{\mathrm{NP}}(f(v_i,\alpha))=b_i=0$. The contradiction proves the claim.

The equality $e(\alpha)=e(\alpha_0)$ implies that $\langle M,f\rangle^{F_{\mathrm{NP}}}(\alpha)=\langle M,f\rangle^{F_{\mathrm{NP}}}(\alpha_0)$. Without loss of generality we may assume that $\langle M,f\rangle^{F_{\mathrm{NP}}}(\alpha_0)=0$. Take n so large that $|U|<\frac{1}{3}2^n$. Let α be equal to α_0 on all the elements of U and to 1 on the remaining words. We have

$$1=F_{\mathrm{BPP}}(\alpha)\not\le\langle M,f\rangle^{F_{\mathrm{NP}}}(\alpha_0)=\langle M,f\rangle^{F_{\mathrm{NP}}}(\alpha)=0.$$

Theorem 21 is proved.

Let us prove Theorem 22. Since PARITY is a language, by Theorem 7, it suffices to prove that PARITY $\not\preceq^l_T$ IPLOG. Assume that PARITY is $\preceq^l_T$-reducible to a language F in the class IPLOG via a pair $\langle M,f\rangle$. Define α_0, m, $q(n)$, $v_1,\dots,v_m$, e_0 just as it was done in the previous proof. Since F is in IPLOG, there exists a polylog-time Verifier V for F. For each $i\le m$ such that $b_i^0=1$, fix a Prover P_i such that $\mathrm{Prob}[(P_i,V)(f(v_i,\alpha_0))=1]>2/3$. Let N be a machine that computes the tth bit of the word $f(v,\alpha)$ within time $\mathrm{poly}(\|\alpha\|+|v|)$ for any given $\langle\alpha,v,t\rangle$, where $|t|=\|f(v,\alpha)\|$. Let $r=\mathrm{poly}(n)$ be an upper bound for the number of queries of the form '$\alpha_0(x)=?$', where x is in $\mathbb{B}^n$, made by N in computations on inputs of the form $\langle\alpha_0,v_i,t\rangle$, where $|t|=\|f(v_i,\alpha_0)\|$. Let β_0^i denote $f(v_i,\alpha_0)$. Let $s=\mathrm{poly}(n)$ be an upper bound for the number of queries of the form '$\beta_0^i(t)=?$', where $|t|=\|\beta_0^i\|$, made by V in the dialogue with P_i on the input β_0^i. Let x be in $\mathbb{B}^n$. Let $w^i_{\alpha_0}(x)$ denote the probability of the event "there exists $t\in\mathbb{B}^{\|\beta_0^i\|}$ such that V queries '$\beta_0^i(t)=?$' in the dialogue with P_i on the input α_0, and N queries '$\alpha_0(x)=?$' during the computation on the input $\langle\alpha_0,v_i,t\rangle$". Then $\sum_{i:b_i^0=1}\sum_{x\in\mathbb{B}^n}w^i_{\alpha_0}(x)\le msr$, therefore, there exists $x_0\in\mathbb{B}^n$ such that $\sum_{i:b_i^0=1}w^i_{\alpha_0}(x_0)\le msr/2^n<1/3$ (if n is sufficiently large). Change the x_0th bit of α_0 and let α denote the resulting word. Let us prove that $e(\alpha)=e(\alpha_0)$, and therefore $\langle M,f\rangle^F(\alpha)=\langle M,f\rangle^F(\alpha_0)$. Assume that $e(\alpha)\ne e(\alpha_0)$. Let $b_1\dots b_m$ denote the bits of $e(\alpha)$. Take the least i such that $b_i\ne b_i^0$. Then $b_i=0$ and $b_i^0=1$. Therefore, $F(f(v_i,\alpha))=0$, consequently,

$$\mathrm{Prob}[(P_i,V)(f(v_i,\alpha))=1]<1/3.$$

On the other hand,

$$\mathrm{Prob}[(P_i,V)(f(v_i,\alpha_0))=1]>2/3.$$

Hence, $w^i_{\alpha_0}(x_0) > 1/3$ because α and α_0 have different values only on x_0. The contradiction thus obtained shows that $e(\alpha) = e(\alpha_0)$ and $\langle M, f\rangle^F(\alpha) = \langle M, f\rangle^F(\alpha_0)$. Since $\mathrm{PARITY}(\alpha) \neq \mathrm{PARITY}(\alpha_0)$, the theorem is proved.

Let us prove Theorem 20. We have to prove that the separation problem F_{BPP} is $\preceq^l_T$-reducible to no language F in the class IPLOG. Assume the contrary: $F_{\mathrm{BPP}} \preceq^l_T F \in$ IPLOG. We use all notation from the previous proof. Without loss of generality we may assume that $\langle M, f\rangle^F(\alpha_0) = 1$. Let α_1 be a word in the set $\{\alpha \in \mathbb{F}_n \mid e(\alpha) = e(\alpha_0)\}$ having the least number of ones. Without loss of generality we may assume that $\alpha_1 = \alpha_0$. If $\#_1(\alpha_0) < \frac{1}{3}2^n$, then the contradiction is already derived. If $\#_1(\alpha_0) \geq \frac{1}{3}2^n$, then there exists $x_0 \in \mathbb{B}^n$ such that $\alpha_0(x_0) = 1$ and $\sum_{i:b^0_i=1} w^i_{\alpha_0}(x_0) \leq \frac{msr}{(1/3)2^n} < 1/3$. Define the word α as follows: $\alpha(x_0) = 0$, $\alpha(x) = \alpha_0(x)$ for all $x \neq x_0$. Then $\#_1(\alpha) < \#_1(\alpha_0)$. Just as it was done in the previous proof we can prove that $e(\alpha) = e(\alpha_0)$. This contradicts the choice of α_0.

Let us prove Theorem 23. Let α be a partial function from $\mathbb{B}^n$ into $\mathbb{B}^n$. Let $\bar{\alpha}$ denote the word encoding the graph of α, that is $\bar{\alpha}(xy) = 1 \Leftrightarrow \alpha(x) = y$ for all $x, y \in \mathbb{B}^n$. Consider the separation problem

$$F(\gamma) = \begin{cases} 1 & \text{if } \exists n \in \mathbb{N} : \gamma = \bar{\alpha}\bar{\beta}, \text{ where } \alpha \text{ and } \beta \text{ are partial} \\ & \text{functions from } \mathbb{B}^n \text{ into } \mathbb{B}^n \text{ such that } \alpha \text{ is total and } \beta \\ & \text{is defined on all the arguments but one,} \\ 0 & \text{if } \exists n \in \mathbb{N} : \gamma = \bar{\alpha}\bar{\beta}, \text{ where } \alpha \text{ and } \beta \text{ are partial} \\ & \text{functions from } \mathbb{B}^n \text{ into } \mathbb{B}^n \text{ such that } \beta \text{ is total and } \alpha \\ & \text{is defined on all the arguments but one,} \\ * & \text{otherwise.} \end{cases}$$

Obviously, $F \in \Sigma_2\mathrm{LOGS} \cap \Pi_2\mathrm{LOGS}$. Let E_n denote the set $\{\gamma \in \mathbb{F}_{2n+1} \mid F(\gamma) \neq *\}$.

By Theorem 4, it suffices to prove that there exists no $G \in$ IPLOGS such that $F \preceq^l_T G$. Assume that such a problem G exists. Let $\langle M, f\rangle$ be pair reducing F to G. Fix a large n. We use all the notations from the previous proofs. Take a word $\gamma \in E_n$ having the lexicographical greatest $e(\gamma)$. Let α_0, β_0 be the partial functions such that $\gamma = \bar{\alpha}_0\bar{\beta}_0$. Without loss of generality we may assume that $F(\bar{\alpha}_0\bar{\beta}_0) = 1$, that is, α_0 is total. Let β_0 be undefined on the word x_1. Fix a Verifier for the solving the problem G. We enumerate bits of γ in such a way that for $x, y \in \mathbb{B}^n$, $\gamma(0xy) = \bar{\alpha}_0(xy)$, $\gamma(1xy) = \bar{\beta}_0(xy)$. For any i such that $b^0_i = 1$, define the weight $w^i_{\alpha_0\beta_0}(u)$ of the word $u \in \mathbb{B}^{2n+1}$ as follows: $w^i_{\alpha_0\beta_0}(u)$ is equal to the probability of the event "there exists $t \in \mathbb{B}^{\|f(v_i,\alpha_0)\|}$ such that V queries '$f(v_i, \alpha_0)(t) =$?' in the dialogue with P_i on input $f(v_i, \alpha_0)$, and N queries '$\alpha_0(u) =$?' during the work on input $\langle \alpha_0, v_i, t\rangle$". If n is large enough, we can find $x_0 \in \mathbb{B}^n$ such that $\sum_{i:b^0_i=1} w^i_{\alpha_0\beta_0}(0x_0\alpha_0(x_0)) < 1/6$ and we can find $y_1 \in \mathbb{B}^n$ such that $\sum_{i:b^0_i=1} w^i_{\alpha_0\beta_0}(1x_1y_1) < 1/6$.

Define the partial functions α, β bas follows:

$$\alpha(x) = \begin{cases} \alpha_0(x), & \text{if } x \neq x_0, \\ \text{undefined}, & \text{if } x = x_0, \end{cases}$$

$$\beta(x) = \begin{cases} \beta_0(x), & \text{if } x \neq x_1, \\ y_1, & \text{if } x = x_1. \end{cases}$$

Then $e(\bar{\alpha}\bar{\beta}) = e(\bar{\alpha}_0\bar{\beta}_0)$ and $F(\bar{\alpha}\bar{\beta}) = 0$. The contradiction thus obtained proves the theorem. □

7. Perceptrons and oracle separation of AM ∩ co-AM from PP

7.1. Perceptrons. In this section we prove that AM ∩ co-AM $\not\subseteq$ PP under some oracle. To prove this we need some lower bounds for perceptrons extending the well known "one-in-a-box" theorem of Minsky and Papert [**37**].

DEFINITION 9. A perceptron is a depth-2 circuit having a threshold gate at the bottom and AND-gates at the remaining level. Inputs of AND-gates are either Boolean variables or their negations. Each AND-gate is labeled by a natural number called the *weight* of the AND-gate. The *total weight* of a perceptron is the sum of absolute values of the weights on all its AND-gates. The *order* of a perceptron is the maximal fanin of its AND-gates.

Let P be a perceptron, and ϕ an assignment of values to its variables. The *weight of* ϕ, written $W_P(\phi)$, is the sum of weights on all AND's which are true on ϕ. The perceptron outputs 1 on input ϕ if $W_P(\phi)$ is greater than the threshold value of its threshold gate and 0 otherwise. Let $P(\phi)$ denote the output value.

7.2. An extension of the "one-in-a-box" theorem. Perceptrons have been studied by Minsky and Papert in [**37**]. We single out two of their results: (i) any perceptron computing the parity function of n variables must have order at least n, and (ii) (the "one-in-a-box" theorem) any perceptron recognizing whether all rows in a given Boolean matrix of size $n \times 4n^2$ contain at least one 1 has order at least n. Beigel [**6**] constructed a Boolean function of n variables that is computable by a perceptron having exponential total weight and order 1 but is not computable by perceptrons having quasipolynomial ($2^{\mathrm{polylog}(n)}$) total weight and polylogarithmic order. To be more precise, he proved the lower bound $d^2 \log w = \Omega(n)$, where d and w denote the order and the total weight, respectively, of a perceptron computing that function.

We extend Minsky and Papert's one-in-a-box theorem in the following direction. Let Π stand for the following separation problem: to separate Boolean matrices in which every row contains a 1 from matrices in which many rows (for example, 99% of them) contain zeros only. Obviously, any perceptron recognizing whether each row of a given matrix has a 1 also solves Π. Our theorem states that Π is not solvable by perceptrons of order $o(\sqrt{m})$ and of total weight $2^{o(n)}$, where n is the number of rows and m is the number of columns (Theorem 24). This implies that perceptrons of polylogarithmic order and quasipolynomial total weight cannot solve Π.

Let us proceed to precise definitions. Let M be a Boolean matrix with n rows and m columns. Then M can be defined in the usual way by means of mn Boolean values. When we say that a perceptron P has M as input we mean that these Boolean values are assigned to its input variables. In this case $P(M)$ denotes the output of P.

A matrix is called *good* if its every row contains a 1. A matrix is called *bad* if it is not good. Let $0 \leq q \leq 1$. a A matrix is called *q-bad* if the ratio of the number of rows with no 1's to the total number of rows is at least q.

The "one-in-a-box" theorem of Minsky and Papert states that the order a perceptron deciding whether an input Boolean matrix of size $n \times 4n^2$ is good must be at least n.

We say that a perceptron P separates good matrices from q-bad matrices of size $n \times m$ if $P(M) = 1$ for every good matrix of size $n \times m$ and $P(M) = 0$ for any q-bad matrix of size $n \times m$. Note that for any m, n there is a perceptron of order m and total weight m separating good matrices from 1-bad matrices of size $n \times m$ (this perceptron decides whether the first row contains a 1).

THEOREM 24 ([**53**]). *Let $0 < \varepsilon \le 1/2$. Suppose that there is a perceptron of order d and of total weight w that separates good matrices from $(1-\varepsilon)$-bad matrices of size $n \times m$, then $d \ge \sqrt{(6/13)\varepsilon m}$ or $w \ge 0.5e^{(2/15)\varepsilon n}$.*

PROOF. Let m, n be integers. Let $\mathcal{M}$ denote the set of Boolean matrices having n rows and m columns. Let M_{ij} denote the element of matrix M standing in the ith row and in the jth column. Let μ be a probability distribution on the set $\mathcal{M}$. For a property S of matrices in $\mathcal{M}$ let $\mathrm{Prob}_\mu[S(M)]$ denote the probability (with respect to μ) that a random matrix M satisfies S. Let d be an integer and μ, ν probability distributions on $\mathcal{M}$. We say that μ and ν are *d-indistinguishable* if

$$\mathrm{Prob}_\mu[M_{i_1 j_1} = b_1, \ldots, M_{i_u j_u} = b_u] = \mathrm{Prob}_\nu[M_{i_1 j_1} = b_1, \ldots, M_{i_u j_u} = b_u]$$

for any sequence $\langle i_1, j_1\rangle, \ldots, \langle i_u, j_u\rangle$ of indices such that $u \le d$ and for any sequence $b_1, \ldots, b_u$ of bits.

The theorem is an easy corollary of the following two lemmas.

LEMMA 3. *Suppose there are d-indistinguishable probability distributions μ and ν on $\mathcal{M}$ such that a random matrix is good with probability 1 with respect to μ and q-bad with probability at least $1-p$ with respect to ν. Then a perceptron of order d separating good matrices from q-bad matrices has total weight at least p^{-1}.*

LEMMA 4. *If $d < \sqrt{(6/13)\varepsilon m}$ and $0 < \varepsilon \le 1/2$, then there are d-indistinguishable probability distributions μ and ν on $\mathcal{M}$ such that a random matrix is good with probability 1 with respect to μ and a random matrix is $(1-\varepsilon)$-bad with probability at least $1 - 2e^{-(2/15)\varepsilon n}$ with respect to ν.*

PROOF OF LEMMA 3. Let d, μ, ν, q, p be as in Lemma 3. Let P be perceptron of order d and of total weight w separating good matrices from q-bad matrices in $\mathcal{M}$. Let E_μ and E_ν stand for the average with respect to μ and ν, respectively. We claim that

$$\mathrm{E}_\mu W_P(M) = \mathrm{E}_\nu W_P(M). \tag{52}$$

Indeed, let $C(M)$ denote the Boolean function computed by an AND-gate C in P. Let l be the total number of AND-gates in P, C_i the ith gate, and w_i the weight of C_i. Then $\mathrm{E}_\mu W_P(M) = \sum_{i=1}^{l} w_i \mathrm{E}_\mu C_i(M) = \sum_{i=1}^{l} w_i \mathrm{Prob}_\mu[C_i(M) = 1]$. Therefore, it suffices to prove that $\mathrm{Prob}_\mu[C(M) = 1] = \mathrm{Prob}_\nu[C(M) = 1]$ for any AND-gate C in P. Represent C as a conjunction $\bigwedge_{s=1}^{u}(M_{i_s j_s} = b_s)$, where every b_s is either 0 or 1. Then

$$\mathrm{Prob}_\mu[C(M) = 1] = \mathrm{Prob}_\mu[M_{i_1 j_1} = b_1, \ldots, M_{i_u j_u} = b_u].$$

Thus the d-indistinguishability of μ and ν implies (52).

Let t be the threshold value of the threshold-gate. Since a random matrix is good with probability 1 with respect to μ, we have $\mathrm{E}_\mu W_P(M) \ge t + 1$. On the

other hand, since a random matrix is q-bad with probability at least $1-p$ with respect to ν, we have $\mathrm{E}_\nu W_P(M) \leq (1-p)t + pw$. Therefore, $t+1 \leq (1-p)t + pw$. Thus, $t+1 \leq t+pw$, so that $1 \leq pw$. □

In the proof of Lemma 4, we will use the following lemma of Farkash, which is a version of the duality theorem from the theory of linear programming (see, for example, [**41**]).

LEMMA 5. *Let*

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1t}x_t &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2t}x_t &= b_2 \\ &\ldots \\ a_{s1}x_1 + a_{s2}x_2 + \ldots + a_{st}x_t &= b_s \end{aligned}$$

be a system of linear equations in real non-negative $x_1, x_2, \ldots, x_t$. It has a solution if and only if there are no real $y_1, y_2, \ldots, y_s$ such that

$$\begin{aligned} a_{11}y_1 + a_{21}y_2 + \ldots + a_{s1}y_s &\geq 0 \\ a_{12}y_1 + a_{22}y_2 + \ldots + a_{s2}y_s &\geq 0 \\ &\ldots \\ a_{1t}y_1 + a_{2t}y_2 + \ldots + a_{st}y_s &\geq 0 \\ b_1y_1 + b_2y_2 + \ldots + b_sy_s &< 0. \end{aligned}$$

PROOF OF LEMMA 4. Let σ be a probability distribution on the segment $\{1, 2, \ldots, m\}$ and τ a probability distribution on the segment $\{0, 1, 2, \ldots, m\}$. The distributions σ and τ will be specified later.

For $i \leq n$ and $M \in \mathcal{M}$, let k_i denote the number of 1's in the ith row of M. We define μ by the condition that $k_1, k_2, \ldots, k_n$ are independent (with respect to σ) random variables with values in $\{1, 2, \ldots, m\}$.

To choose a matrix at random with respect to μ, pick random $k_1, k_2, \ldots, k_n$ independent with respect to σ. Then for each $i \leq n$ take a random string with exactly k_i ones as the ith row of the matrix.

The distribution ν is defined in the same way, but with σ replaced by τ.

The distribution τ is in turn obtained from a distribution ρ (to be specified later) on the segment $\{1, 2, \ldots, m\}$ by means of the following transformation:

$$\mathrm{Prob}_\tau[k = a] \begin{cases} 0.6\varepsilon\, \mathrm{Prob}_\rho[k = a], & \text{if } a \in \{1, 2, \ldots, m\}; \\ 1 - 0.6\varepsilon, & \text{if } a = 0. \end{cases}$$

If ν is obtained from ρ in the way described above, then a random matrix is $(1-\varepsilon)$-bad with probability at least $1 - 2e^{-(2/15)\varepsilon n}$ with respect to ν. This is a direct corollary of the Chernoff inequality. Indeed, a row of a random matrix has only 0's with probability $p = 1 - 0.6\varepsilon$ with respect to ν. Take $\delta = 0.4\varepsilon$. Since we assume that $\varepsilon \leq 1/2$, δ and p satisfy the conditions of Theorem 10, whence we conclude that a random matrix has less than $(1-\varepsilon)n$ zero rows with probability at most

$$2e^{-\frac{\delta^2 n}{2p(1-p)}} = 2e^{-\frac{(0.4\varepsilon)^2 n}{2 \cdot 0.6\varepsilon(1-0.6\varepsilon)}} = 2e^{-\frac{0.16\varepsilon n}{1.2(1-0.6\varepsilon)}} \leq 2e^{-\frac{0.16\varepsilon n}{1.2}} = 2e^{-2/15\varepsilon n}.$$

We now specify the probability distributions σ and ρ on $\{1,2,\ldots,m\}$. We want to do this so that distributions σ and τ will have the same first d moments, that is,

$$\mathrm{E}_\sigma\, k^i = \mathrm{E}_\tau\, k^i \tag{53}$$

for all $1 \le i \le d$. Let us prove that this implies the d-indistinguishability of μ and ν.

Indeed, we claim that the value of

$$\mathrm{Prob}_\mu[M_{i_1 j_1} = b_1, \ldots, M_{i_u j_u} = b_u]$$

is a polynomial in $\mathrm{E}_\sigma\, k$, $\mathrm{E}_\sigma\, k^2$, ..., $\mathrm{E}_\sigma\, k^d$ for any sequence $\langle i_1, j_1\rangle, \ldots, \langle i_u, j_u\rangle$ of indices of length at most d and for any sequence $b_1, \ldots, b_u$ of bits. Let us prove this claim. For any $i \le n$, let c_i stand for the number of $l \in \{1,\ldots,u\}$ such that $i_l = i$ and $b_l = 1$, and let e_i stand for the number of $l \in \{1,\ldots,u\}$ such that $i_l = i$ and $b_l = 0$. Then

$$\mathrm{Prob}_\mu[M_{i_1 j_1} = b_1, \ldots, M_{i_u j_u} = b_u] = \prod_{i=1}^{n} \mathrm{E}_\sigma \frac{k(k-1)\cdots(k-c_i+1)(m-k)(m-k-1)\cdots(m-k-e_i+1)}{m(m-1)\cdots(m-c_i-e_i+1)}$$

Evidently, for any i, $\mathrm{E}_\sigma \frac{k(k-1)\cdots(k-c_i+1)(m-k)(m-k-1)\cdots(m-k-e_i+1)}{m(m-1)\cdots(m-c_i-e_i+1)}$ is a linear combination of the $\mathrm{E}_\sigma\, k^r$, $r = 0, 1, \ldots, c_i + e_i$. As $c_i + e_i \le d$, the claim is proved.

Recall that ν is obtained from τ in the same way as as μ is obtained from σ. Therefore, (53) implies the d-indistinguishability of μ and ν.

It is easy to see that

$$\mathrm{E}_\tau\, k^i = 0.6\varepsilon\, \mathrm{E}_\rho\, k^i$$

for any $i \ge 1$. Thus we have to prove that there are probability distributions σ and ρ on $\{1,2,\ldots,m\}$ satisfying

$$\mathrm{E}_\sigma\, k^i = 0.6\varepsilon\, \mathrm{E}_\rho\, k^i \quad \text{for } i = 1, 2, \ldots, d. \tag{54}$$

Here an application of the Farkash lemma is appropriate because the dual problem is easier than the original one.

LEMMA 6. *The following conditions are equivalent:*
1) there are probability distributions σ and ρ on the segment $\{1,2,\ldots,m\}$ satisfying (54); 2) there is no polynomial $p(x)$ of degree at most d such that

$$-p(0)\frac{1-0.6\varepsilon}{0.6\varepsilon} < p(j) \le 0 \quad \textit{for } j = 1, 2, \ldots, m.$$

PROOF. The condition 1) is equivalent to the existence of a non-negative solution of the system

$$\begin{aligned}
0.6\varepsilon(1^1 x_1 + 2^1 x_2 + \ldots + m^1 x_m) - (1^1 x_{m+1} + 2^1 x_{m+2} + \ldots + m^1 x_{2m}) &= 0\\
0.6\varepsilon(1^2 x_1 + 2^2 x_2 + \ldots + m^2 x_m) - (1^2 x_{m+1} + 2^2 x_{m+2} + \ldots + m^2 x_{2m}) &= 0\\
&\ldots\\
0.6\varepsilon(1^d x_1 + 2^d x_2 + \ldots + m^d x_m) - (1^d x_{m+1} + 2^d x_{m+2} + \ldots + m^d x_{2m}) &= 0\\
x_1 + x_2 + \ldots + x_m &= 1\\
x_{m+1} + x_{m+2} + \ldots + x_{2m} &= 1.
\end{aligned}$$

By the Farkash Lemma this means that there are no real $p_1, p_2, \ldots, p_d, y, z$ such that

$$0.6\varepsilon \sum_{i=1}^{d} p_i j^i + y \geq 0 \quad \text{for } j = 1, 2, \ldots, m, \tag{55}$$

$$-\sum_{i=1}^{d} p_i j^i + z \geq 0 \quad \text{for } j = 1, 2, \ldots, m, \tag{56}$$

$$y + z < 0. \tag{57}$$

We interpret the numbers $-z, p_1, p_2, \ldots, p_d$ as coefficients of a polynomial $p(u) = -z + p_1 u + \cdots + p_d u^d$. Then (56) means that $p(j) \leq 0$ for all $j = 1, 2, \ldots, m$. The existence of y satisfying (55) and (57) means that $0.6\varepsilon(p(j) - p(0)) + p(0) > 0$ for all $j = 1, 2, \ldots, m$. The last inequality can be rewritten as $-p(0)\frac{1-0.6\varepsilon}{0.6\varepsilon} < p(j)$, so 1) and 2) are equivalent. □

It remains to prove that there is no polynomial $p(x)$ of degree at most d such that $-p(0)\frac{1-0.6\varepsilon}{0.6\varepsilon} < p(j) \leq 0$ for all $j = 1, 2, \ldots, m$. We will use the following theorem of Markov (see, for example, [**12, 35, 42**]): $\|P'\| \leq d^2\|P\|$, where P is a polynomial of degree d and $\|P\|$ denotes the maximum of $|P(x)|$ over all $x \in [-1; 1]$. An easy corollary of this theorem is the following simplified version of the Ehlich and Zeller lemma [**14**]. Let

$$Y = \{-1, -1 + 2/m, -1 + 4/m, \ldots, 1 - 2/m, 1\}$$

and let $\|P_Y\|$ denote the maximum of $|P(x)|$ over all $x \in Y$. Then

$$\|P\| \left(1 - \frac{d^2}{m}\right) \leq \|P_Y\|. \tag{58}$$

Indeed, by the Lagrange theorem and by Markov's theorem we have

$$\|P\| - \|P_Y\| \leq \|P'\| \frac{1}{m} \leq \|P\| \frac{d^2}{m}.$$

This implies (58).

Suppose now that there is a polynomial $p(x)$ of degree d such that $-p(0)\frac{1-0.6\varepsilon}{0.6\varepsilon} < p(j) \leq 0$ for all $j = 1, 2, \ldots, m$. We have to prove that $d \geq \sqrt{(6/13)\varepsilon m}$.

Observe that $p(0) > 0$. Let α denote $\frac{1-0.6\varepsilon}{0.6\varepsilon}$. Let $P(x) = p(\frac{1}{2}m(x+1)) + \frac{\alpha-1}{2}p(0)$. Then $\|P_Y\| = \frac{\alpha+1}{2}p(0)$, and therefore

$$(1 - d^2/m)\|P\| \leq \frac{\alpha + 1}{2} p(0). \tag{59}$$

On the other hand, since $|P(-1 + 2/m) - P(-1)| = |p(1) - p(0)| \geq p(0)$, the Lagrange theorem implies that $\|P'\| \geq p(0)m/2$. Therefore, by Markov's theorem,

$$\|P\| \geq p(0) \frac{m}{2d^2}. \tag{60}$$

Combining (59) and (60) we obtain

$$\left(1 - \frac{2d^2}{m}\right) p(0) \frac{m}{2d^2} \leq \frac{\alpha + 1}{2} p(0).$$

Therefore

$$\frac{d^2}{m} \geq \frac{1}{\alpha + 2} = \frac{0.6\varepsilon}{1 + 0.6\varepsilon} \geq \frac{0.6\varepsilon}{1.3} = \frac{6}{13}\varepsilon. \quad \square$$

7.3. Oracle separation of AM $\cap$ co-AM **from** PP. The lower bound in Theorem 24 is sufficient to construct an oracle under which AM $\not\subseteq$ PP. To construct an oracle under which AM $\cap$ co-AM $\not\subseteq$ PP we need a lower bound for perceptrons solving another separation problem. Let us describe this problem.

Let $\mathcal{M}_n$ stand for the family of Boolean matrices of size $n \times n$ and let $\mathcal{N}_n = \mathcal{M}_n \times \mathcal{M}_n$. Let $D = \langle M_0, M_1 \rangle$ be a pair of matrices in $\mathcal{N}_n$. We say that D is of type 0 [type 1] if every row in M_0 [M_1] contains a 1 and at least $2/3$ of rows in M_1 [M_0] contain no 1's.

THEOREM 25. *There is a $\delta > 0$ such that the following holds for sufficiently large n. If there is a perceptron of order d and of total weight w separating elements in $\mathcal{N}_n$ of type 0 from elements of type 1 in $\mathcal{N}_n$, then either $d \geq \delta n^{1/2}$ or $w \geq 2^{\delta n}$.*

PROOF. Let n be an integer. Let P be a perceptron of order d and of total weight w separating elements of type 0 in $\mathcal{N}_n$ from elements in of type 1 in $\mathcal{N}_n$. Put $\varepsilon = 1/3$, $\delta = 0.01$ and $m = n$. Suppose that $d \leq \delta n^{1/2}$. Then the conditions of Lemma 4 are satisfied. Therefore, there are probability distributions μ and ν on $\mathcal{M}_n$ such that:

1) a random matrix M is good with probability 1 with respect to μ;

2) a random matrix M is not $2/3$-bad with probability at most $2e^{-(2/15)(1/3)n}$ with respect to ν;

3) μ and ν are d-indistinguishable.

Let p denote $2e^{-(2/15)(1/3)n}$. Consider the probability distributions $\kappa = \mu \times \nu$ and $\lambda = \nu \times \mu$ on $\mathcal{N}_n$. Then

$$\mathrm{Prob}_\kappa[D \text{ has type } 0\,] \geq 1 - 2p,$$
$$\mathrm{Prob}_\lambda[D \text{ has type } 1\,] \geq 1 - 2p.$$

As we have seen above, 3) implies that

$$\mathrm{E}_\kappa W_P(D) = \mathrm{E}_\lambda W_P(D).$$

Let t be the threshold value of the threshold-gate. Obviously we can assume that $|t| < w$. We have $\mathrm{E}_\kappa W_P(D) \geq (1-2p)(t+1) - 2pw$ and $\mathrm{E}_\lambda W_P(D) \leq (1-2p)t + 2pw$. Therefore, $(1-2p)(t+1) - 2pw \leq (1-2p)t + 2pw$, which implies the inequality $w \geq 1/(6p) = (1/12)e^{(2/15)(1/3)n} \geq e^{\delta n}$ for sufficiently large n. □

THEOREM 26 ([**50**]). *There is an oracle A such that* $\mathrm{AM}^A \cap \mathrm{co\text{-}AM}^A \not\subseteq \mathrm{PP}^A$.

PROOF. By Theorem 1 it is sufficient to prove that AMLOGS$\cap$co-AMLOGS $\not\subseteq$ PPLOGS. Let α be a binary word of length $2n^2$. We will view α as a pair of Boolean matrices of size $n \times n$. Consider the separation problem

$$F(\alpha) = \begin{cases} 1 & \text{if } \alpha \text{ has type 1,} \\ 0 & \text{if } \alpha \text{ has type 0,} \\ * & \text{if } \alpha \text{ has neither type 0 nor 1 or } |\alpha| \neq 2n^2. \end{cases}$$

Obviously, $F \in$ AMLOGS $\cap$ co-AMLOGS. So it is sufficient to prove that $F \notin$ PPLOGS. Assume that there are a polylog-time predicate $Q(\alpha, r)$ and a polylogarithm $l(n)$ such that

$$F(\alpha) = 1 \quad \Rightarrow \quad \mathrm{M}_{1/2}\, r \in \mathbb{B}^{l(n)}\ Q(\alpha, r) = 1,$$
$$F(\alpha) = 0 \quad \Rightarrow \quad \neg\, \mathrm{M}_{1/2}\, r \in \mathbb{B}^{l(n)}\ Q(\alpha, r) = 1.$$

Assume that the computation time of Q on inputs α, r, $|\alpha| = 2n^2$, $|r| = l(n)$, is bounded by the polylogarithm $d(n)$. For every n, let us construct a perceptron of order $d = d(n)$ and of total weight $w = 2^{d(n)+l(n)}$ such that

$$P(\alpha) = 1 \Leftrightarrow \mathrm{M}_{1/2}\, r \in \mathbb{B}^{l(n)}\ Q(\alpha, r) = 1 \tag{61}$$

for all $\alpha \in \mathbb{B}^{2n^2}$.

Let r be a random string of length $l(n)$. The value of $Q(\alpha, r)$ depends on d bits of α learned in the computation of Q on α, r. Let $v = v(1)v(2)\dots v(d)$ be a binary string of length d such that $Q(\alpha, r) = 1$ whenever the d bits of α asked in the computation of Q on α, r are $v(1)v(2)\dots v(d)$, respectively. We associate the following AND-gate C with the pair $\langle v, r\rangle$. Let $u_1, u_2, \dots, u_d$ be the indices of bits of α asked in the computation of Q on α, r. On an assignment α, the gate C produces 1 when $\alpha(u_k) = v(k)$ for all $k \in \{1, 2, \dots, d\}$. Declare the weight on C to be 1, and the threshold value of the perceptron P to be $2^{l(n)-1}$. It is easy to verify that

$$W_P(\alpha) = |\{r \in \mathbb{B}^{l(n)} \mid Q(\alpha, r) = 1\}|.$$

This implies (61).

Obviously, the order of P is $d(n) = \mathrm{polylog}n$ and the total weight of P is $2^{d(n)+l(n)} = 2^{\mathrm{polylog}n}$. Theorem 25 shows that for every sufficiently large n, P cannot separate pairs of type 0 from pairs of type 1 in $\mathcal{N}_n$ and we are done. □

7.4. Conclusion. Theorem 24 states that a perceptron of small total weight separating good matrices from q-bad ones has large order. This leaves the possibility that perceptrons of small order and arbitrary total weight can separate good matrices from q-bad ones (for some $q < 1$). Since the one-in-a-box theorem involves restrictions on the order only, a theorem stating that perceptrons of small order and arbitrary total weight cannot do the job would a better extension of the one-in-a-box theorem. Recently, R. Beigel obtained such a lower bound (personal communication). He proved that perceptrons separating good matrices of size $n \times n$ from q-bad ones must have superpolylogarithmic order in n (for any fixed $q < 1$). The problem if perceptrons of order $n^{o(1)}$ can do the job remains open. Note that Beigel's bound also suffices to separate AM from PP via oracles.

8. The universum method

We have not yet constructed so far any oracles under which some positive and negative assertions hold simultaneously, say $\mathrm{P} = \mathrm{R} \neq \mathrm{BPP}$.

Many results of this sort (when an oracle is constructed under which some Boolean combination of complexity assertions is true) have appeared in literature. The following results among them deal with the classes considered here. Rackoff in [**43**] constructed oracles A and B such that $\mathrm{P}^A = \mathrm{R}^A \neq \mathrm{NP}^A$ and $\mathrm{P}^B \neq \mathrm{R}^B = \mathrm{NP}^B$. In [**5**], it was proved that $\mathrm{P} = \mathrm{NP} \cap \text{co-NP} \neq \mathrm{NP}$ under some oracle. Homer and Selman in [**29**] showed that there is an oracle under which $\mathrm{P} \neq \mathrm{NP}$ but NP-sets are separable. This implies that reliability of all the cryptographic schemes based on the existence of one-way functions cannot be derived from $\mathrm{P} \neq \mathrm{NP}$ by relativizable arguments (since one-way functions do not exist if NP-sets are separable). We show that one cannot prove using relativizable arguments that NP-sets are inseparable even under hypothesis that both co-NP-sets are inseparable and $\mathrm{P} \neq \mathrm{R}$. The strongest our result states that there is an oracle under which $\mathrm{P} \neq \mathrm{NP}$ but NP-sets are separable, co-NP-sets are separable, and $\mathrm{P} = \mathrm{BPP}$. In other words, it

is impossible to prove by relativizable arguments even the disjunction "NP-sets are inseparable or co-NP-sets are inseparable or P $\neq$ BPP" under the P $\neq$ NP hypothesis.

Our method goes back to [**5**]. We call it "the universum method". We refine that method and apply it to prove the existence of oracles relative to which certain Boolean combinations of the assertions P = NP, P = R, P = BPP, P = NP$\cap$co-NP, P = R $\cap$ co-R, "NP-sets are P-separable", and "co-NP-sets are P-separable" hold (we are successful in constructing oracles for 13 of 17 possible combinations, thus 4 problems of this kind remain unsolved).

Roughly speaking, the method works as follows. Suppose we want to prove that there exists an oracle A such that $\mathrm{P}^A \neq \mathrm{BPP}^A$ and $\mathrm{P}^A = \mathrm{R}^A$. First, we define a subset V (called *the universum*) of the set of all oracles. Second, we choose a sufficiently powerful oracle H (in all known applications we can take any PSPACE-complete set as H). Third, we consider machines having two oracles: the oracle H and a varying oracle B ranging over V. (Thus, every machine of this type accepts a subset of $\mathbb{B}^* \times V$.) Finally, we prove that there exists a BPP-machine of this type which recognizes a subset of $\mathbb{B}^* \times V$ recognizable by no P-machine of this type and prove that for any R-machine of this type there exists a P-machine of this type recognizing the same subset of $\mathbb{B}^* \times V$.

Another general method close to ours was presented in [**15**]. An extension of that method was applied by Fortnow and Rogers in [**17**] to prove the existence of oracles relative to which certain Boolean combinations of the assertions P = NP, P = UP, P = NP $\cap$ co-NP, "NP-sets are P-separable", and "co-NP-sets are P-separable" hold. They succeeded in constructing oracles for all possible combinations.

In a sense our method (as well as the method of [**15**]) is a special kind of the forcing method. In Section 8.4, we prove two results that can be interpreted as that our method fails to prove the following two theorems: P $\neq$ R = PSPACE under some oracle [**43**] and P = NP $\neq$ PSPACE under some oracle [**32**].

We start with a sample application of the method.

8.1. A sample application. In the sequel, we will use the following notation. For a finite set $M \subset \mathbb{B}^*$ let $\mathit{maxlength}\,(M)$ denote the $\max_{y \in M} |y|$.

Assume that P is a deterministic oracle machine. Let $\mathit{Query}_P^B(x, B \oplus H)$ denote the set of all $y \in \mathbb{B}^*$ such that P asks '$B(y) =?$' during the computation on input x with oracle $B \oplus H$. Assume that N is a nondeterministic oracle machine and c is one of its computations with oracle $B \oplus H$ on some input. Then $\mathit{Query}_N^B(c, B \oplus H)$ denotes the set of all $y \in \mathbb{B}^*$ such that N asks '$B(y) =?$' during the computation c. By P- [NP-, BPP-] machine we mean a polynomial-time deterministic [nondeterministic, probabilistic] oracle machine.

DEFINITION 10. Let L_1, L_2 and L be languages. We say that L separates L_1 from L_2 if $L_1 \subseteq L$ and $L_2 \subseteq \mathbb{B}^* \setminus L$. Let $\mathcal{C}$ and $\mathcal{C}'$ be families of languages. A $\mathcal{C}$-set is a language from $\mathcal{C}$. We say that $\mathcal{C}$-sets are $\mathcal{C}'$-separable if for any two disjoint languages L_1 and L_2 in $\mathcal{C}$, there exists a language L in $\mathcal{C}'$ which separates L_1 from L_2. We say that NP^A-*sets* [co-NP^A-*sets*] are *separable* if NP^A-*sets* [co-NP^A-*sets*] are P^A-*separable*. If this is not the case, then we say that NP^A-sets [co-NP^A-sets] are *inseparable*.

THEOREM 27 ([**17, 39**]). *There exists an oracle A such that* NP^A*-sets are inseparable and* co-NP^A*-sets are separable.*

PROOF. The proof of this theorem is very close to the proof of the theorem from [**5**] stating that $\mathrm{P}^A = \mathrm{NP}^A \cap \text{co-NP}^A \neq \mathrm{NP}^A$ for some oracle A.

Define the sequence of integers n_i by induction: $n_0 = 1$, $n_{i+1} = 2^{2^{n_i}}$. Let $\mathbb{S} = \{n_i \mid i \in \mathbb{N}\}$. Consider the following set of oracles:

$$V = \{B \in \Omega \mid \text{for all } n \in \mathbb{S} \text{ there exists at most one } y \in \mathbb{B}^n \text{ such that } B(y) = 1 \text{ and for all } n \in \mathbb{N} \setminus \mathbb{S} \text{ there exist no } y \in \mathbb{B}^n \text{ such that } B(y) = 1\}.$$

Let H be a PSPACE-complete language. The oracle A will have the form $B \oplus H$, where B is in V. Thus, we have to define the oracle B.

We construct B in such a way that the following sets:

$$L_0^B = \{1^n \mid n \in \mathbb{S} \text{ and there exists } y \in \mathbb{B}^{n-1} \text{ such that } B(0y) = 1\},$$
$$L_1^B = \{1^n \mid n \in \mathbb{S} \text{ and there exists } y \in \mathbb{B}^{n-1} \text{ such that } B(1y) = 1\}$$

are P^A-inseparable. Obviously, both L_0^B and L_1^B belong to $\mathrm{NP}^{B\oplus H}$ and are disjoint for any $B \in V$.

So we have to construct an oracle $B \in V$ such that

(1) L_0^B and L_1^B are separable by no $\mathrm{P}^{B\oplus H}$-set, and

(2) any two disjoint co-$\mathrm{NP}^{B\oplus H}$-sets are separable by some $\mathrm{P}^{B\oplus H}$-set.

Let M be a deterministic or nondeterministic machine. Write $M^A(x) = 1$ if M with oracle A accepts x and write $M^A(x) = 0$ otherwise. Say that a pair $\langle N_0, N_1\rangle$ of NP-machines is *correct on* A if the languages $\{x \mid N_0^A(x) = 0\}$ and $\{x \mid N_1^A(x) = 0\}$ are disjoint. The assertion (1) means that for any P-machine P the language $\{x \mid P^{B\oplus H}(x) = 1\}$ does not separate L_0^B from L_1^B. The assertion (2) means that for any pair $\langle N_0, N_1\rangle$ of NP-machines correct on $B \oplus H$ there exists a set in $\mathrm{P}^{B\oplus H}$ separating $\{x \mid N_0^{B\oplus H}(x) = 0\}$ from $\{x \mid N_1^{B\oplus H}(x) = 0\}$. Let P_0, P_1, ..., P_i, ... be an enumeration of P-machines and $\langle N_{00}, N_{01}\rangle$, $\langle N_{10}, N_{11}\rangle, \ldots,$ $\langle N_{j0}, N_{j1}\rangle, \ldots$ be an enumeration of pairs of NP-machines.

We shall make a countable number of steps. On any step we shall define some oracle values and make some freezing. In other words, we construct a sequence of intervals $\Gamma_1 \supseteq \Gamma_2 \supseteq \Gamma_3 \supseteq \ldots$ each Γ_i intersects with V. On step $i = 2k + 1$ we find Γ_i such that language $\{x \mid P_k^{B\oplus H}(x) = 1\}$ does not separate L_0^B from L_1^B for any $B \in \Gamma_i \cap V$. On step $i = 2k + 2$ we find Γ_i such that either the pair $\langle N_{k0}, N_{k1}\rangle$ is not correct on $B \oplus H$ for any $B \in \Gamma_i \cap V$, or the languages $\{x \mid N_{k0}^{B\oplus H}(x) = 0\}$ and $\{x \mid N_{k1}^{B\oplus H}(x) = 0\}$ are separable by a set in $\mathrm{P}^{B\oplus H}$ for any $B \in \Gamma_i \cap V$. Obviously, for any oracle B in the set $V \cap \bigcap_{i=1}^{\infty} \Gamma_i$ the assertions (1) and (2) will hold.

We start with $\Gamma_0 = \Omega$. Let us explain what to do on each step. Let

$$\Gamma_{i-1} = \Gamma(\phi) = \{B \in \Omega \mid B|\,\mathrm{Dom}(\phi) = \phi\}$$

be the interval constructed on $(i-1)$th step. On the ith step we make the following. Consider two cases.

First case: $i = 2k + 1$. Pick $n \in \mathbb{S}$ greater than *maxlength* $\mathrm{Dom}(\phi)$ and so large that P_k on input 1^n makes less than 2^{n-1} queries to oracle. Let C be the oracle in Γ_{i-1} that is equal to zero on all the words not in $\mathrm{Dom}(\phi)$. Without loss of generality we may assume that $P_k^{C\oplus H}(1^n) = 0$ (the other case is entirely similar). We know that $|\mathit{Query}_{P_k}^{C}(1^n, C \oplus H)|$ is less than the number of words of length

$n-1$. Pick a word z of the form $1u$ in the set $\mathbb{B}^n \setminus \mathit{Query}_{P_k}^C(1^n, C \oplus H)$. Note that z is not in $\mathrm{Dom}(\phi)$ since $n > \mathit{maxlength}\,(\mathrm{Dom}(\phi))$. Let

$$\Gamma_i = \{B \in \Gamma_{i-1} \mid B(z) = 1,\ B(y) = C(y) \text{ for all } y \in \mathit{Query}_{P_k}^C(1^n, C \oplus H)\}.$$

Then $P_k^{B\oplus H}(1^n) = P_k^{C\oplus H}(1^n) = 0$ and $L_1^B(1^n) = 1$ for any $B \in \Gamma_i$, and $\Gamma_i \cap V$ is non-empty since $C \cup \{z\}$ is in $\Gamma_i \cap V$.

The reader can see that, in fact, we have proved the following lemma, whose analog will be used in all other proofs.

LEMMA 7. *If an interval Γ intersects with V then there exists no P-machine P such that $P^{B\oplus H}$ separates L_0^B from L_1^B for any $B \in \Gamma \cap V$.*

Second case: $i = 2k+2$. Consider two subcases.

First subcase: There exists an oracle $C \in \Gamma_{i-1} \cap V$ such that the pair $\langle N_{k0}, N_{k1}\rangle$ is not correct on $C \oplus H$. Then pick an $x \in \mathbb{B}^*$ such that $N_{k0}^{C\oplus H}(x) = N_{k1}^{C\oplus H}(x) = 0$. Let

$$\Gamma_i = \Big\{ B \in \Gamma_{i-1} \mid B(y) = C(y) \text{ for all } y \in \bigcup_{c_0} \mathit{Query}_{N_{k0}}^C(c_0, C \oplus H) \cup \bigcup_{c_1} \mathit{Query}_{N_{k1}}^C(c_1, C \oplus H) \Big\},$$

where the unions are over all the computations of N_{k0} and N_{k1}, respectively, on input 1^n with oracle $C \oplus H$.

Second subcase: the pair $\langle N_{k0}, N_{k1}\rangle$ is correct on $C \oplus H$ for any $C \in \Gamma_{i-1} \cap V$. Then let $\Gamma_i = \Gamma_{i-1}$. We have to prove that the sets $\{x \mid N_{k0}^{B\oplus H}(x) = 0\}$ and $\{x \mid N_{k1}^{B\oplus H}(x) = 0\}$ are separable by a set from $\mathrm{P}^{B\oplus H}$ for any $B \in \Gamma_i \cap V$. This assertion easily follows from the following two lemmas.

DEFINITION 11. A *polynomial* machine is a deterministic oracle Turing machine that works within polynomial-space and makes at most $\mathrm{poly}(|x|)$ queries to oracle (x is an input).

LEMMA 8. *Assume that Γ is an interval and $\langle N_0, N_1\rangle$ is a pair of* NP*-machines that is correct on $C \oplus H$ for any $C \in \Gamma \cap V$. Then there exists a polynomial machine P such that $P^B(x)$ is equal to a $j \in \{0,1\}$ for which $N_j^{B\oplus H}(x) = 1$ for any x and any $B \in V \cap \Gamma$.*

LEMMA 9. *Let P be a polynomial machine. Then there exists a P-machine M such that $P^B(x) = M^{B\oplus H}(x)$ for all $x \in \mathbb{B}^*$, $B \in \Omega$ (recall that H is a* PSPACE*-complete set).*

PROOF OF LEMMA 8. We describe the work of P on input x with oracle B in the case $B \in \Gamma \cap V$. The reader can easily modify the program of P to handle the general case. Machine P with oracle B on input x works as follows. Find first $n = n_i \in \mathbb{S}$ such that $\log_2 n \le |x| < 2^n$. Let m be so large that N_j on inputs of length greater than m cannot query oracle values on words of length n_{i+1} or greater ($j = 0, 1$). If $|x| < m$ then compute $N_0^{B\oplus H}(x)$ directly and return 0 if $N_0^{B\oplus H}(x) = 1$ and 1 otherwise.

If $|x| \ge m$ make the following. Query the value of B on all the words of length at most n_{i-1}. The number of such queries is less than $2^{n_{i-1}+1} \le 2|x|$. We know B's value on all the words that both values $N_0^{B\oplus H}(x)$ and $N_1^{B\oplus H}(x)$ depends on

FIGURE 3

```
begin
    w := Λ (the empty word);
    while result(x, w) = $
            commentary: result(x, w) is computed in time
            poly(|x|, |w|) by querying H;
        do y := question (x, w);
                commentary: question (x, w) is computed in
                time poly(|x|, |w|) by querying H;
            b := B(y);
            w := wb;
        od
    return result (x, w)
end
```

except for values on words of length n. Let

$$C(y) = \begin{cases} B(y) & \text{if } |y| \neq n; \\ 0 & \text{otherwise.} \end{cases}$$

Note that $C \in V \cap \Gamma$, therefore $N_0^{C\oplus H}(x) = 1$ or $N_1^{C\oplus H}(x) = 1$. Find an $l \in \{0,1\}$ such that $N_l^{C\oplus H}(x) = 1$, and find an accepting computation c of N_l with oracle $C \oplus H$ on input x. This can be done within polynomial-space by checking all the computations of N_0 and N_1 with oracle $C \oplus H$ on input x. All the queries made to H in those computations can be answered within polynomial-space because their lengths are bounded by $\mathrm{poly}(|x|)$ and $H \in \mathrm{PSPACE}$. Set $W = \mathit{Query}_{N_l}^{C}(c, C \oplus H) \cap \mathbb{B}^n$. Query '$B(y) =?$' for all $y \in W$. If $B(y) = 0$ for all $y \in W$, then $N_l^{B\oplus H}(x) = 1$; in this case return l. Otherwise we have found the unique word of length n on which B is equal to 1, and therefore can find both $N_0^{B\oplus H}(x)$ and $N_1^{B\oplus H}(x)$ within polynomial-space without making extra queries to B. Obviously, we have made $\mathrm{poly}(|x|)$ queries. □

PROOF OF LEMMA 9. Let P be a polynomial machine. Define the functions *question* (x, w) and *result* (x, w) as follows. Let w be a binary word of length n. Run the machine P on input x and give the answer $w(1)$ to the first query, the answer $w(2)$ to the second query and so on. There are three possibilities:

1) P makes exactly n queries and then returns a result say r; in this case set

$$\mathit{question}\,(x, w) = \$,\ \mathit{result}\,(x, w) = r;$$

2) P makes n queries and then makes $(n+1)$st query, say '$B(y) =?$'; in this case set

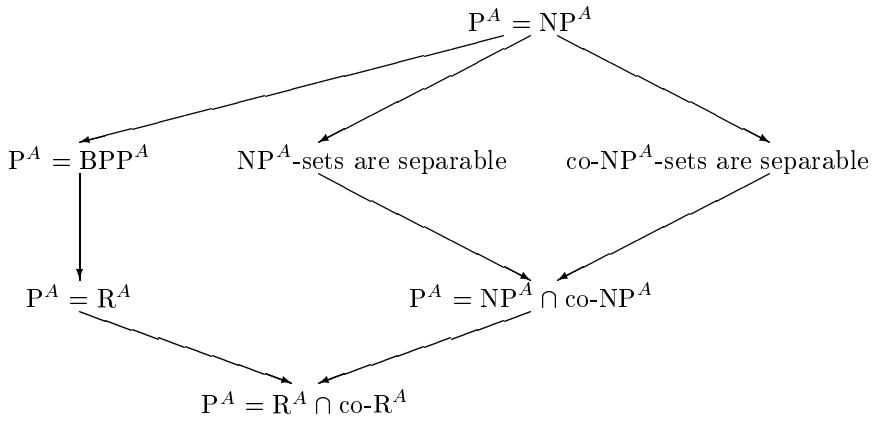
$$\mathit{question}\,(x, w) = y,\ \mathit{result}\,(x, w) = \$;$$

3) P makes less than n queries; in this case set

$$\mathit{question}\,(x, w) = \mathit{result}\,(x, w) = \$.$$

Obviously, both functions *question* and *result* are computable within polynomial-space. Therefore, they can be computed by a polynomial-time machine with oracle H. Let the machine M work according the program shown on Figure 3. □

FIGURE 4. The edges of the drawn directed graph represent relativizable implications. For example, the implication $\mathrm{P}^A = \mathrm{NP}^A \Rightarrow \mathrm{P}^A = \mathrm{BPP}^A$ is true since $\mathrm{BPP}^A \subset \Sigma_2^A$.



The proof of Theorem 27 is finished. □

All other theorems are proved according to the presented scheme. Namely, first a set V of oracles is defined (which is called the universum). The oracle under which the desired Boolean combination of complexity assertions holds always has the form $B \oplus H$, where H is a PSPACE-complete set. The desired properties of B are represented as a countable family of requirements on B and then a diagonal construction is used to satisfy all the requirements. On the ith step, an interval Γ_i is constructed such that the ith requirement holds for any $B \in \Gamma_i \cap V$. The requirements are of two types: "negative" ones and "positive" ones (in the above example the requirements satisfied on odd steps are negative ones and the requirements satisfied on even steps are positive ones). Negative requirements are satisfied by using an appropriate analog of Lemma 7. Its proof is always easy, therefore we will only present the analog of languages L_0^B and L_1^B. The positive requirements will be satisfied by trying first to make the current pair of machines (or a single machine in the case of classes BPP and R) incorrect. The notion of correctness of course will be specific in each case. If this fails, then we use an analog of Lemma 8, which combined with Lemma 9 (common for all the applications of the method) will complete the proof.

Thus, the proof of any specific theorem in the sequel will consist of the definition of the appropriate universum, the definition of analogs of languages L_0^B and L_1^B and the proof(s) of the appropriate analog(s) of Lemma 8.

8.2. Other applications of the universum method. The assertions on complexity classes in which we are interested are shown in Figure 4. We prove the existence of oracles under which one or another combination of assertions that label the nodes of the shown graph holds. There are 17 possible combinations of those assertions. They are listed in Table 1.

We are able to prove the existence of oracles under which the combinations of all the lines but the lines number 3, 4, 9 and 10 are true. In fact, we do not know the answer to the following question.

TABLE 1. The signs "+" and "⊕" put in a line of the table indicate that the corresponding assertion is true. The signs "−" and "⊖" indicate that the corresponding assertion is false. The difference between the signs "+" and "⊕" is that the truth of assertions labeled by "+" follows from the truth of assertions labeled by "⊕" but the truth of any assertion labeled by "⊕" does not follow from the truth of the other assertions. The same difference is between the signs "−" and "⊖". The commentary ending each line includes information about where the combination present in the line is proved.

	P = NP	NP-sets separable	co-NP-sets separable	P = NP ∩co-NP	P = BPP	P = R	P = R ∩co-R	Commentary
1	⊕	+	+	+	+	+	+	[**5**]
2	⊖	⊕	⊕	+	⊕	+	+	Theorem 39
3	−	⊕	⊕	+	⊖	⊕	+	Unknown
4	−	⊕	⊕	+	−	⊖	+	Unknown
5	−	⊕	⊖	+	⊕	+	+	Theorem 30
6	−	⊕	⊖	+	⊖	⊕	+	Th. 33
7	−	⊕	⊖	+	−	⊖	+	Th. 34
8	−	⊖	⊕	+	⊕	+	+	Theorem 28
9	−	⊖	⊕	+	⊖	⊕	+	Unknown
10	−	⊖	⊕	+	−	⊖	+	Unknown
11	−	⊖	⊖	⊕	⊕	+	+	Th. 32
12	−	⊖	⊖	⊕	⊖	⊕	+	Theorem 35
13	−	⊖	⊖	⊕	+	⊖	+	Theorem 36
14	−	−	−	⊖	⊕	+	+	Th. 29
15	−	−	−	⊖	⊖	⊕	+	Th. 37
16	−	−	−	⊖	−	⊖	⊕	Th. 38
17	−	−	−	−	−	−	⊖	Well known

QUESTION. Is there an oracle under which co-NP-sets are separable and P $\neq$ BPP?

We shall use only the universums of the form

$$V = V(Z) = \{B \in \Omega \mid \forall n \notin \mathbb{S}\ B|\mathbb{B}^n = \mathbf{0},\ \forall n \in \mathbb{S}\ B|\mathbb{B}^n \in Z\},$$

where Z is a subfamily of $\mathbb{F}$ (recall that $\mathbb{F}$ denotes $\bigcup_{n=0}^{\infty} \mathbb{F}_n$, where $\mathbb{F}_n$ is the set of all functions from $\mathbb{B}^n$ to $\mathbb{B}$), and $\mathbf{0}$ is an identically zero function. The set Z is called the *base* of $V(Z)$.

We will use the following five *standard* bases:

- $Z(\leq 1) = \{\alpha \in \mathbb{F} \mid \#_1\alpha \leq 1\}$; this base was already used in the proof of Theorem 27
- $Z(=1) = \{\alpha \in \mathbb{F} \mid \#_1\alpha = 1\}$
- $Z(\geq 1) = \{\alpha \in \mathbb{F} \mid \#_1\alpha \geq 1\}$
- $Z(\mathrm{BPP}) = \bigcup_{n\in\mathbb{S}}\{\alpha \in \mathbb{F}_n \mid \#_1\alpha/2^n \notin [1/3; 2/3]\}$
- $Z(\mathrm{R}) = \bigcup_{n\in\mathbb{S}}\{\alpha \in \mathbb{F}_n \mid \#_1\alpha/2^n \notin (0; 2/3]\}$

Other bases will be built from the standard bases by the following operation + on bases: $Z' + Z'' = \{\alpha \in \mathbb{F}_{n_i} \mid i \text{ is even and } \alpha \in Z'\} \cup \{\alpha \in \mathbb{F}_{n_i} \mid i \text{ is odd and } \alpha \in Z''\}$.

Now, we formulate the five analogs of Lemma 7, which will be used to satisfy negative requirements in the proofs of the next theorems. Their proofs are straightforward and therefore we omit them.

LEMMA 10. *Let Z' be any base and let an interval Γ intersect with the universum $V = V(Z(\leq 1)+Z')$. Then there exists no P-machine M such that $M^{B\oplus H}$ separates the language*

$$\{1^n \mid n = n_i \in \mathbb{S},\ i \text{ is even and there exists } y \in \mathbb{B}^{n-1} \text{ such that } B(0y) = 1\}$$

from the language

$$\{1^n \mid n = n_i \in \mathbb{S},\ i \text{ is odd and there exists } y \in \mathbb{B}^{n-1} \text{ such that } B(1y) = 1\}$$

for any $B \in \Gamma \cap V$. These languages are in $\mathrm{NP}^{B\oplus H}$ and are disjoint for any $B \in V$.

LEMMA 11. *Let Z' be any base and let an interval Γ intersect with the universum $V = V(Z(\geq 1)+Z')$. Then there exists no P-machine M such that $M^{B\oplus H}$ separates the language*

$$\{1^n \mid n = n_i \in \mathbb{S},\ i \text{ is even and } B(0y) = 0 \text{ for all } y \in \mathbb{B}^{n-1}\}$$

from the language

$$\{1^n \mid n = n_i \in \mathbb{S},\ i \text{ is odd and } B(1y) = 0 \text{ for all } y \in \mathbb{B}^{n-1}\}$$

for any $B \in \Gamma \cap V$. These languages are in $\text{co-NP}^{B\oplus H}$ and are disjoint for any $B \in V$.

LEMMA 12. *Let Z' be any base and let an interval Γ intersect with the universum $V = V(Z(=1)+Z')$. Then there exists no P-machine M such that $M^{B\oplus H}$ recognizes the language*

$$\{1^n \mid n = n_i \in \mathbb{S},\ i \text{ is even and there exists } y \in \mathbb{B}^{n-1} \text{ such that } B(0y) = 1\}$$

for any $B \in \Gamma \cap V$. This language is in $\mathrm{NP}^{B\oplus H} \cap \text{co-NP}^{B\oplus H}$ for any $B \in V$.

LEMMA 13. *Let Z' be any base and let an interval Γ intersect with the universum $V = V(Z(\mathrm{BPP})+Z')$. Then there exists no P-machine M such that $M^{B\oplus H}$ recognizes the language*

$$\{1^n \mid n = n_i \in \mathbb{S},\ i \text{ is even and } \#_1(B|\mathbb{B}^n) > (2/3)2^n\}$$

for any $B \in \Gamma \cap V$. This language is in $\mathrm{BPP}^{B\oplus H}$ for any $B \in V$.

LEMMA 14. *Let Z' be any base and let an interval Γ intersect with the universum $V = V(Z(\mathrm{R})+Z')$. Then there exists no P-machine M such that $M^{B\oplus H}$ recognizes the language*

$$\{1^n \mid n = n_i \in \mathbb{S},\ i \text{ is even and } \#_1(B|\mathbb{B}^n) > (2/3)2^n\}$$

for any $B \in \Gamma \cap V$. This language is in $\mathrm{R}^{B\oplus H}$ for any $B \in V$.

Now, we are going to consider all the lines in the table except lines numbered 3, 4, 9 and 10. The existence of oracles under which the combinations in the first and in the last lines hold is well known so we skip those lines.

THEOREM 28 ([**39**]). NP-*sets are inseparable,* co-NP-*sets are separable and* $\mathrm{P} = \mathrm{BPP}$ *under some oracle (8th line in the table).*

PROOF. This theorem strengthens Theorem 27 and its proof uses the same universum $V = V(Z(\leq 1))$. All we have to do is to prove the analog of Lemma 8 for BPP-machines. We say that a BPP-machine M is correct on an oracle A if M^A accepts each input with a probability lying outside the segment $[1/3;2/3]$.

LEMMA 15. *Assume that Γ is an interval and M is a BPP-machine that is correct on $C \oplus H$ for any $C \in \Gamma \cap V$. Then there exists a polynomial machine P that recognizes with oracle B the same language as M does with oracle $B \oplus H$ for any $B \in V \cap \Gamma$.*

PROOF. Let us construct P. Let x be an input to P. In fact, the beginning of the proof of all analogs of Lemma 8 is common. We first find an $n = n_i \in \mathbb{S}$ such that $\log_2 n \leq |x| < 2^n$, query B's values on words of length at most n_{i-1}, then compute the value $\mathrm{Prob}[M^{B\oplus H}(x)]$ directly if $|x|$ is so small that $M^{B\oplus H}(x)$ may depend on $B|\mathbb{B}^{\geq n_{i+1}}$. It remains to construct a polynomial machine P' that on input $\langle x, B|\mathbb{B}^{\leq n_{i-1}}\rangle$, where $B \in V \cap \Gamma$, decides whether $\mathrm{Prob}[M^{B\oplus H}(x)] > 2/3$ provided x is so long that $M^{B\oplus H}$ on input x cannot query B's value on words of length n_{i+1} or greater.

Let P' work as follows. Find first the probability p of the event "$M^{C\oplus H}(x) = 1$", where C is the oracle that is equal to B on words of length different from n and to zero on remaining words. Note that we know all the values of C needed to find p. Without loss of generality we may assume that $p \geq 1/2$ (the case $p \leq 1/2$ is entirely similar).[3]

For an oracle D, let $w_D(y)$ denote the probability of the event "$M^{D\oplus H}$ at some moment in the computation on input x queries '$D(y) =?$'". Then

$$\sum_{y\in\mathbb{B}^*} w_D(y) \leq \mathrm{poly}(|x|)$$

for any D. Let W denote the set of all $y \in \mathbb{B}^n$ such that $w_C(y) \geq 1/6$. Obviously, $|W| \leq \mathrm{poly}(|x|)$.

Find W and query '$B(y) =?$' for all $y \in W$. Consider two cases.

First case: $B(y) = 0$ for all $y \in W$. Let us prove that then $\mathrm{Prob}[M^{B\oplus H}(x) = 1] > 2/3$. Since M is correct on $B\oplus H$ the probability q of the event "$M^{B\oplus H}(x) = 1$" is either greater than $2/3$ or less than $1/3$. We claim that the first alternative holds. Indeed, if $B|\mathbb{B}^n = C|\mathbb{B}^n$, then $q = p \geq 1/2$, therefore $q > 2/3$. Otherwise denote by y the unique word of length n such that $B(y) = 1$. Then $w_C(y) < 1/6$ because $y \notin W$. Therefore, $\left|\mathrm{Prob}[M^{B\oplus H}(x) = 1] - \mathrm{Prob}[M^{C\oplus H}(x) = 1]\right| \leq w_C(y) < 1/6$, therefore, $\mathrm{Prob}[M^{B\oplus H}(x) = 1] > 1/2 - 1/6 = 1/3$. Return 1 in the first case.

Second case: $B(y) = 1$ for some $y \in W$. In this case we know all the values of B needed to compute $\mathrm{Prob}[M^{B\oplus H}(x) = 1]$. □

THEOREM 29 ([**39**]). *There exists an oracle A such that $\mathrm{NP}^A \cap \text{co-NP}^A \neq \mathrm{P}^A$ and $\mathrm{BPP}^A = \mathrm{P}^A$ (14th line in the table).*

PROOF. Take the base $Z = Z(=1)$. The analog of Lemma 8 for BPP-machines is already proved (see the proof of Lemma 15 and Footnote 3). □

[3]Since M is correct on $C \oplus H$, we know that in this case $p > 2/3$. However we shall not use this fact because we want the proof to be valid for the case $V = V(Z(=1))$, in this case p can belong to the segment $[1/3; 2/3]$.

THEOREM 30 ([**39**]). *There exists an oracle A such that* NP^A*-sets are separable,* co-NP^A*-sets are inseparable and* $\mathrm{BPP}^A = \mathrm{P}^A$ *(5th line in the table).*

PROOF. Let $V = V(Z(\geq 1))$.

Call a pair $\langle N_0, N_1\rangle$ of NP-machines correct on A if $N_0^A(x) = 0$ or $N_1^A(x) = 0$ for all x. To ensure the separability of NP^A-sets we shall prove the following analog of Lemma 8.

LEMMA 16. *Let $\langle N_0, N_1\rangle$ be a pair of NP-machines that is correct on $B \oplus H$ for any $B \in V \cap \Gamma$. Then there exists a good machine P that for all $B \in V \cap \Gamma$ on input x with oracle B finds an $l \in \{0,1\}$ for which $N_l^{B\oplus H}(x) = 0$.*

PROOF. Let x be an input word. Let $n = n_i$ be defined as in the proof of Lemma 8. Assume that the length of x is so large that both machines N_0, N_1 on input x cannot query oracle values on words of length $\geq n_{i+1}$ and that n is greater than lengths of words defining Γ. Assume that we already know $B|\mathbb{B}^{\leq n_i}$. We have to find an l such that $N_l^{B\oplus H}(x) = 0$.

To this end we shall use the technique from [**9**]. Let $l = 0, 1$. An *l-certificate* is a function having the form $C|$ *Query* $_{N_l}^{C}(c, C \oplus H)$, where C is an oracle agreeing with B on all words of length different from n, $N_l^{C\oplus H}(x) = 1$ and c is an accepting computation of $N_l^{C\oplus H}$ on x.

Note that if γ is an l-certificate and C continues γ, then $N_l^{C\oplus H}(x) = 1$. Obviously, the number of elements in the domain of any l-certificate is polynomial in $|x|$, let $p(|x|)$ denote this polynomial. Assume that x is so long that $2^n > 2p(|x|)$. We claim that then any 0-certificate ϕ is inconsistent with any 1-certificate ψ. Indeed, assume that a 0-certificate ϕ and a 1-certificate ψ are consistent. Then there exists an oracle C agreeing with B on words of length different from n that continues both ϕ and ψ. As $|\mathrm{Dom}(\phi)| + |\mathrm{Dom}(\psi)| < 2^n$ we may assume that there exists $y \in \mathbb{B}^n$ such that $C(y) = 1$, that is, C is in V.[4] Since C continues both ϕ and ψ we have $N_0^{C\oplus H}(x) = N_1^{C\oplus H}(x) = 1$. Thus the pair N_0, N_1 is incorrect on $C \oplus H$ and C is in $V \cap \Gamma$. The contradiction proves the claim. □

Let $\mathcal{C}_0$ [$\mathcal{C}_1$] be the set of all 0-certificates [1-certificates]. Let $U = \emptyset$. Repeat $p(|x|)$ times the following loop. Pick a 0-certificate ϕ in $\mathcal{C}_0$ (if $\mathcal{C}_0$ is empty then return 0 and halt). Query '$B(y) =$?' for all $y \in \mathrm{Dom}(\phi)$ and remove from $\mathcal{C}_0$ and $\mathcal{C}_1$ all certificates that are inconsistent with $B|\,\mathrm{Dom}(\phi)$. Include in U all the elements of $\mathrm{Dom}(\phi)$. (We will explain further how to perform the described program within polynomial-space.)

Before and after each iteration of the loop all the certificates in $\mathcal{C}_0 \cup \mathcal{C}_1$ agree with each other on U. On the other hand, in each iteration, any certificate ψ in $\mathcal{C}_1$ is inconsistent with the picked 0-certificate ϕ, therefore its domain intersects with $\mathrm{Dom}(\phi) \setminus U$. Hence the number of elements of the set $\mathrm{Dom}(\psi) \setminus U$ decreases after each iteration of the loop for any 1-certificate ψ in $\mathcal{C}_1$.

Thus, after $p(|x|)$ iterations, U includes the domains of all the certificates in $\mathcal{C}_1$. If $\mathcal{C}_1$ becomes empty, then $N_1^{B\oplus H}(x) = 0$. Otherwise $\mathcal{C}_0$ becomes empty, therefore $N_0^{B\oplus H}(x) = 0$. Obviously, we have made at most $p(|x|)^2$ queries to B.

[4] In the next theorem we shall need this lemma for $V = V(Z(\mathrm{BPP}))$. In this case we need the inequality $|\mathrm{Dom}(\phi)| + |\mathrm{Dom}(\psi)| < (1/3)2^n$. Assuming this inequality we can find an oracle C consistent with both ϕ and ψ, agreeing with B on words of length different from n and such that the number of words of length n in C is greater than $(2/3)2^n$, that is, we can find $C \in V \cap \Gamma$ continuing both ϕ and ψ.

Let us prove now that the described program can be run within polynomial-space. We do not need to write $\mathcal{C}_0$ or $\mathcal{C}_1$. It suffices to write the set U and the B's value on elements of U. For given U and $B|U$ we can decide whether there is a 0-certificate [1-certificate] consistent with $B|U$ by checking all the computations of N_0 [N_1] on input x. If a query '$B(y) =?$' is made during one of computations we answer '$B(y)$' if $|y| \le n_{i-1}$ or y is in U (note that we know B's value on such words), 0 if $n_{i-1} < |y| < n$ or $n < |y|$ and try all the answers otherwise. As the number of queries does not exceed $\mathrm{poly}(|x|)$ the amount of written information is $\mathrm{poly}(|x|)$. □

To ensure the equality $\mathrm{BPP}^A = \mathrm{P}^A$ we shall prove the following analog of Lemma 8.

LEMMA 17. *Let M be a BPP-machine that is correct on $B \oplus H$ for any $B \in V \cap \Gamma$. Then there exists a polynomial machine P such that P^B recognizes the same language as $M^{B\oplus H}$ does for any $B \in V \cap \Gamma$.*

PROOF. We use some ideas from [**30**] and [**40**]. By Lemma 16 it suffices to construct a pair $\langle N_0, N_1\rangle$ of NP-machines such that

$$\begin{aligned} \mathrm{Prob}[M^{B\oplus H}(x) = 1] > 2/3 \quad &\Rightarrow \quad N_1^{B\oplus H}(x) = 1,\ N_0^{B\oplus H}(x) = 0 \\ \mathrm{Prob}[M^{B\oplus H}(x) = 1] < 1/3 \quad &\Rightarrow \quad N_1^{B\oplus H}(x) = 0,\ N_0^{B\oplus H}(x) = 1 \end{aligned}$$

for any x and any $B \in V \cap \Gamma$. We shall construct the machine N_1 (and the machine N_0 can be constructed in the similar way).

Let x be an input and let B be in $V \cap \Gamma$. Let n be defined as usually and let x be so long that M on input x cannot query B's value on words of length n_{i+1} or greater and that n is greater than lengths of words defining Γ. In the rest of the proof of the lemma we shall consider only oracles that agree with B an all the words of length different from n. Note that an oracle C with this property is in $V \cap \Gamma$ iff $\#_1(C|\mathbb{B}^n) > 0$.

Let k denote the maximal number of queries that M can make on input x. Obviously, $k \le \mathrm{poly}(|x|)$.

For a $y \in \mathbb{B}^n$ and an oracle D let $w_D(y)$ denote the probability of the event "$M^{D\oplus H}$ on input x queries '$D(y) =?$'". Let

$$W = \{y \in \mathbb{B}^n \mid w_B(y) \ge 1/(9k+3)\}.$$

Since $\sum_{y\in\mathbb{B}^*} w_B(y) \le k$, the set W has at most $(9k+3)k$ elements.

CLAIM. If $\mathrm{Prob}[M^{B\oplus H}(x) = 1] > 2/3$ then $\mathrm{Prob}[M^{C\oplus H}(x) = 1] > 2/3$ for any oracle C that agrees with B on W.

PROOF. Assume the contrary: there exists C agreeing with B on W and such that $\mathrm{Prob}[M^{C\oplus H}(x) = 1] < 1/3$. Choose a C satisfying these conditions and differing from B on the least number of words. Let $U = \{y \in \mathbb{B}^n \mid B(y) \ne C(y)\}$.

Let us prove that $w_C(y) > 1/3$ for any y in U possibly but one. Let y be an element of U. Let C_y denote the oracle obtained from C by changing the value on y. We distinguish two cases.

First case: $C_y \in V$. As C_y differs from B on less arguments than C does and $C_y \in V \cap \Gamma$, we have $\mathrm{Prob}[M^{C\oplus H}(x) = 1] > 2/3$. Therefore, $w_C(y) \ge \mathrm{Prob}[M^{C_y\oplus H}(x) = 1] - \mathrm{Prob}[M^{C\oplus H}(x) = 1] > 2/3 - 1/3 = 1/3$.

Second case: $C_y \notin V$, that is $\#_1(C_y|\mathbb{B}^n) = 0$. This may happen only if $\#_1(C|\mathbb{B}^n) = 1$ and therefore this case can occur for the unique y.

As $\sum_{y\in\mathbb{B}^*} w_C(y) \le k$, we have $|U| \le 3k+1$. Since $U \cap W = \emptyset$, we have $w_B(y) < \frac{1}{9k+3}$ for any $y \in U$. Hence $\sum_{y\in U} w_B(y) < \frac{1}{(9k+3)}(3k+1) = \frac{1}{3}$. On the other hand,

$$\sum_{y\in U} w_B(y) \ge \mathrm{Prob}[M^{B\oplus H}(x)=1] - \mathrm{Prob}[M^{C\oplus H}(x)=1] > 2/3 - 1/3 = 1/3.$$

The contradiction proves the claim. □

In a similar way we can prove that if $\mathrm{Prob}[M^{B\oplus H}(x)=1] < 1/3$ then $\mathrm{Prob}[M^{C\oplus H}(x)=1] < 1/3$ for any oracle $C \in V$ agreeing with B on W.

Thus there is a nondeterministic oracle machine that accepts x with all oracles $B \oplus H$ such that $B \in V$ and $\mathrm{Prob}[M^{B\oplus H}(x)=1] > 2/3$ and rejects x with oracles $B \oplus H$ such that $B \in V$ and $\mathrm{Prob}[M^{B\oplus H}(x)=1] < 1/3$, and that makes to oracle $\mathrm{poly}(|x|)$ queries. This machine guesses a set W having at most $k(9k+3)$ elements, learns B's value on W and then accepts iff $\mathrm{Prob}[M^{C\oplus H}(x)=1] > 2/3$ for all oracles $C \in V$ that agree with B on W. However this machine is not a polynomial one, as it does not work within polynomial-space. To convert this machine into a polynomial-space machine we make the following.

Assume that $2(9k+3)k < 2^n$.

Claim. $\mathrm{Prob}[M^{B\oplus H}(x)=1] > 2/3$ if and only if

(∗) there is U such that $|U| \le (9k+3)k$ and $\mathrm{Prob}[M^{C\oplus H}(x)=1] > 2/3$ for any C agreeing with B on U and equal to zero on at most $2(9k+3)k$ words of length n.

Proof. The implication from the left to the right follows from the the above claim since we can take W as U. Let us prove the implication from the right to the left. Assume that $\mathrm{Prob}[M^{B\oplus H}(x)=1] < 1/3$ but there is U such that $|U| \le (9k+3)k$ and $\mathrm{Prob}[M^{C\oplus H}(x)=1] > 2/3$ for any C agreeing with B on U and equal to zero on at most $2(9k+3)k$ words of length n. Choose such U. Let D denote the oracle agreeing with B on $U \cup W$ and equal to 1 on all the words from $\mathbb{B}^n \setminus (U \cup W)$. As D agrees with B on W, we have $\mathrm{Prob}[M^{D\oplus H}(x)=1] < 1/3$. On the other hand, D agrees with B on U and is equal to zero on at most $2(9k+3)k$ words of length n. We conclude that $\mathrm{Prob}[M^{D\oplus H}(x)=1] > 2/3$. The contradiction shows that $\mathrm{Prob}[M^{B\oplus H}(x)=1] > 2/3$ if and only if (∗) is true. □

Assume that a subset U of $\mathbb{B}^n$ has at most $(9k+3)k$ elements. The values of an oracle C that is equal to zero on at most $2(9k+3)k$ words of length n can be identified by means of a polynomial amount of information. Thus for given U, $B|B^{\le n_{i-1}}$ and $B|U$ we can decide within polynomial-space whether (∗) is true. Therefore we can decide this in polynomial-time using the oracle H. The machine $N_1^{B\oplus H}$ on input x works as follows. Query the value of B on all the words of length at most n_{i-1}. Then guess a set $U \subset \mathbb{B}^n$ having $(9k+3)k$ elements and accept if (∗) is true. □

Theorem 31 ([**39**]). *There exists an oracle A such that NP^A-sets are separable, $\mathrm{BPP}^A \ne \mathrm{P}^A$, and $\mathrm{R}^A = \mathrm{P}^A$.*

Proof. Let $V = V(Z(\mathrm{BPP}))$. To ensure separability of NP^A-sets we need the following analog of Lemma 8.

Call a pair $\langle N_0, N_1\rangle$ of NP-machines correct on A if $N_0^A(x) = 0$ or $N_1^A(x) = 0$ for all x.

LEMMA 18. *Let $\langle N_0, N_1\rangle$ be a pair of NP-machines that is correct on $B \oplus H$ for any $B \in V \cap \Gamma$. Then there is a polynomial machine P that for any $B \in V \cap \Gamma$ on input x with oracle B finds an $l \in \{0,1\}$ for which $N_l^{B\oplus H}(x) = 0$*

PROOF. This lemma can be proven just as Lemma 16. The only difference is that we have to take x so large that $(1/3)2^n > 2p(|x|)$ (and not $2^n > 2p(|x|)$ as in that proof). □

Let us say that a probabilistic oracle Turing machine M is correct on A if $\mathrm{Prob}[M^A(x) = 1]$ either is equal to 0 or is greater than $2/3$ for any x.

We need the following analog of of Lemma 8.

LEMMA 19. *Assume that M is a probabilistic polynomial-time oracle machine correct on oracle $B \oplus H$ for any $B \in V \cap \Gamma$. Then there exists a polynomial machine P that with any oracle $B \in V \cap \Gamma$ recognizes the same language as M does with the oracle $B \oplus H$.*

PROOF. By Lemma 18 it suffices to construct a pair $\langle N_0, N_1\rangle$ of NP-machines such that

$$\begin{aligned} \mathrm{Prob}[M^{B\oplus H}(x) = 1] > 2/3 \quad &\Rightarrow \quad N_1^{B\oplus H}(x) = 1,\ N_0^{B\oplus H}(x) = 0 \\ \mathrm{Prob}[M^{B\oplus H}(x) = 1] = 0 \quad &\Rightarrow \quad N_1^{B\oplus H}(x) = 0,\ N_0^{B\oplus H}(x) = 1 \end{aligned}$$

for any x and any $B \in \Gamma \cap V$.

It is obvious that there exists an NP-machine N_1 satisfying this requirement.

Thus, we have to construct N_0. Let x be an input to N_0 and let B be in $V \cap \Gamma$. Let $n = n_i$ be defined as earlier. Assume that M on input x cannot query oracle values on words of length $\geq n_{i+1}$. In the rest of the proof of the lemma we shall consider only oracles that agree with B an all the words of length different from n.

Let $k = \mathrm{poly}(|x|)$ be the maximal number of queries to B which machine M can make during the work on input x. Denote by $w_C(y)$ the probability of the event "$M^{C\oplus H}(x)$ on input x queries '$C(y)$ =?'". Let $W = \{y \in \mathbb{B}^n \mid w_B(y) > 1/4k\}$. Note that $|W| < 4k^2$.

CLAIM. Assume that $\mathrm{Prob}[M^{B\oplus H}(x) = 1] = 0$. Then $\mathrm{Prob}[M^{C\oplus H}(x) = 1] = 0$ for any C agreeing with B on W.

PROOF. Assume the contrary and let C be a counterexample. Then for at least one random string, $M^{C\oplus H}(x) = 1$. Denote by U the set of all the $y \in \mathbb{B}^n$ such that the query '$C(y)$ =?' is made during the computation of $M^{C\oplus H}$ on x for that random string. Obviously, $|U| \leq k$. Let D be the oracle agreeing with C on U and with B on remaining words. If n is large enough, then D is in Γ. If D belonged to V we would obtain a contradiction: we know that probability of the event "$M^{D\oplus H}(x) = 1$" is positive, hence, this probability would be greater than $2/3$. Therefore, $\sum_{D(y)\neq B(y)} w_B(y)$ would be greater than $2/3$. On the other hand, $\{y \mid D(y) \neq B(y)\} \subset U \setminus W$, consequently,

$$\sum_{D(y)\neq B(y)} w_B(y) \leq |U|\frac{1}{4k} \leq \frac{k}{4k} = \frac{1}{4}.$$

Now we have to explain what to do if $D \notin V$, that is,

$$\#_1(D|\mathbb{B}^n) \in [(1/3)2^n, (2/3)2^n].$$

We know that $B \in V$, that is, $\#_1(B|\mathbb{B}^n) \notin [(1/3)2^n, (2/3)2^n]$. Without loss of generality we may assume that $\#_1(B|\mathbb{B}^n) > (2/3)2^n$. Then $\#_1(D|\mathbb{B}^n) > (2/3)2^n - |U|$. We have $|U| \le k = \mathrm{poly}(|x|)$. Therefore, we may assume that $2|U| + |W| \le (1/3)2^n$.

As $(1/3)2^n \le \#_0(D|\mathbb{B}^n)$, there exists a set $T \subset \mathbb{B}^n$ having exactly $|U|$ elements, disjoint from $U \cup W$ and such that $D(y) = 0$ for any $y \in T$. Choose such a T and change the value of D on all the words from T. Now we have $\#_1(D|\mathbb{B}^n) > (2/3)2^n$, therefore D is in V. As $D(y) = C(y)$ for any $y \in U$, we have $\mathrm{Prob}[M^{D\oplus H}(x) = 1] > 0$. Therefore, $\mathrm{Prob}[M^{D\oplus H}(x) = 1] > 2/3$.

Recall that $\mathrm{Prob}[M^{B\oplus H}(x) = 1] = 0$. Hence,

$$\sum_{D(y)\neq B(y)} w_B(y) > 2/3.$$

On the other hand, $\sum_{D(y)\neq B(y)} w_B(y) \le (|U|+|T|)\frac{1}{4k} \le \frac{2k}{4k} = \frac{1}{2}$. The contradiction proves the claim. □

The rest is as in the proof of Lemma 17. Assume that $|x|$ is so large that $4k^2 + k < \frac{1}{3}2^n$.

CLAIM. $\mathrm{Prob}[M^{B\oplus H}(x) = 1] = 0$ if and only if

(∗) there is U such that $|U| \le 4k^2$ and $\mathrm{Prob}[M^{C\oplus H}(x) = 1] = 0$ for any C agreeing with B on U and equal to zero on at most $4k^2 + k$ words of length n.

PROOF. The implication from the left to the right follows from the above claim. Let us prove the implication from the right to the left.

Assume that $\mathrm{Prob}[M^{B\oplus H}(x) = 1] > 2/3$ but there is U such that $|U| \le 4k^2$ and $\mathrm{Prob}[M^{C\oplus H}(x) = 1] = 0$ for any C agreeing with B on U and equal to zero on at most $4k^2 + k$ words of length n. Choose such a U. Fix any random string for which $M^{B\oplus H}(x) = 1$ and let R denote the set of $y \in \mathbb{B}^n$ such that the query '$B(y) =?$' was made during the computation of $M^{B\oplus H}$ on input x for that string. Let D denote the oracle agreeing with B on $U \cup R$ and equal to 1 on all the words in $\mathbb{B}^n \setminus (U \cup R)$. Then $\mathrm{Prob}[M^{D\oplus H}(x) = 1] > 0$ since D agrees with B on R. On the other hand, D agrees with B on U and is equal to zero on at most $4k^2 + k$ words of length n. Hence $\mathrm{Prob}[M^{D\oplus H}(x) = 1] = 0$. The contradiction proves the claim. □

So machine $N_0^{B\oplus H}$ on input x works as follows. Query the value of B on all the words of length at most n_{i-1}. Then guess a set $U \subset \mathbb{B}^n$ having $4k^2$ elements and accept if (∗) is true. □

Thus the theorem is proved. □

The above proven facts on the five standard universums are shown in Table 2. The last line in the table contains unproven facts about the universum $V(\mathbb{F})$ presented for the sake of completeness. They all except inseparability of NP-sets are straightforward. The latter fact was proven by An. Muchnik (personal communication). We present the sketch of his proof. Consider the a binary string f of length $k(2^k + 1)$. We view f as a function form the set $M = \{1, \ldots, 2^k + 1\}$ into $\mathbb{B}^k$. By the Pigeon Hole Principle there are different $i, j \in M$ such that $f(i) = f(j)$. Let the first co-NP-set consist of those f for which there are no different $i, j \in M$ such

TABLE 2. The sign "+" in a line of the table indicates that the analog of Lemma 8 is true for the corresponding universum. The sign "–" in a line of the table indicates that the analog of Lemma 7 is true for the corresponding universum. The letter "o" means that the proof was omitted (because we do not need the corresponding assertion).

	NP-sep.	o-NP-sep.	**NP** ∩ co-**NP**	**BPP**	**R**
$V(Z(\leq 1))$	−	+	+	+	+
$V(Z(=1))$	−	−	−	+	+
$V(Z(\geq 1))$	+	−	+	+	+
$V(Z(\mathrm{BPP}))$	+	− (o)	+	−	+
$V(Z(\mathrm{R}))$	+	− (o)	+	−	−
$V(\mathbb{F})$	+ (o)	− (o)	+ (o)	+ (o)	+ (o)

that $f(i) = f(j)$ and the first symbol of $f(i)$ is 0 and the second co-NP-set of those f for which there are no different $i, j \in M$ such that $f(i) = f(j) = 1$ and the first symbol of $f(i)$ is 1. It is clear that to separate these sets we need to learn at least $2^k + 1$ bits of f.

In the proofs of the following theorems we use bases obtained by addition from the five standard bases.

THEOREM 32 ([**39**]). *There exists an oracle A such that* NP^A*-sets and* co-NP^A*-sets are inseparable,* $\mathrm{NP}^A \cap \text{co-}\mathrm{NP}^A = \mathrm{P}^A$ *and* $\mathrm{BPP}^A = \mathrm{P}^A$ *(11th line in the table).*

PROOF. Let $V = V(Z(\leq 1)+Z(\geq 1))$. Let us prove first the analog of Lemma 8 for NP ∩ co-NP-machines.

LEMMA 20. *Let $\langle N_0, N_1\rangle$ be a pair of NP-machines such that the languages accepted by $N_0^{B\oplus H}$ and $N_1^{B\oplus H}$ are complementary for any $B \in V \cap \Gamma$. Then there exists a polynomial machine P that with any oracle $B \in V \cap \Gamma$ accepts the same language as N_0 does with oracle $B \oplus H$.*

PROOF. Machine P works as follows. Let x be an input. Let $n = n_i \in \mathbb{S}$ be defined by inequalities $\log_2 n \leq |x| < 2^n$. If i is even then by definition

$$\alpha \in Z \Leftrightarrow \#_1\alpha \leq 1$$

for any $\alpha \in \mathbb{F}_n$. In this case we consider $\langle N_0, N_1\rangle$ as a pair defining a problem of separation of co-NP-sets and argument as in the proof of Lemma 8.

If i is odd, then by definition

$$\alpha \in Z \Leftrightarrow \#_1\alpha \geq 1$$

for any $\alpha \in \mathbb{F}_n$. In this case we consider $\langle N_0, N_1\rangle$ as a pair defining a problem of separation of NP-sets and reason as in the proof of Lemma 16. □

The analog of Lemma 8 for BPP-machines can be proved similarly. □

To prove the next Theorems 33–38 we do not need any new ideas. Therefore we shall only present the bases used in their proofs.

THEOREM 33 ([**39**]). *There exists an oracle A such that* NP^A*-sets are separable,* co-NP^A*-sets are inseparable,* $\mathrm{BPP}^A \neq \mathrm{P}^A$, $\mathrm{R}^A = \mathrm{P}^A$ *(6th line in the table).*

PROOF. Take the base $Z = Z(\mathrm{BPP}) + Z(\geq 1)$. □

THEOREM 34 ([**39**]). *There exists an oracle A such that* NP^A*-sets are separable,* co-NP^A*-sets are inseparable, and* $\mathrm{P}^A \neq \mathrm{R}^A$ *(7th line in the table).*

PROOF. Take the base $Z = Z(\mathrm{R}) + Z(\geq 1)$. □

THEOREM 35 ([**39**]). *There exists an oracle A such that* NP^A*-sets are inseparable,* co-NP^A*-sets are inseparable,* $\mathrm{BPP}^A \neq \mathrm{P}^A$, $\mathrm{NP}^A \cap \text{co-}\mathrm{NP}^A = \mathrm{P}^A$, *and* $\mathrm{R}^A = \mathrm{P}^A$ *(12th line in the table).*

PROOF. Take the base $Z = Z(\leq 1) + Z(\geq 1) + Z(\mathrm{BPP})$. □

THEOREM 36 ([**39**]). *There exists an oracle A such that* NP^A*-sets are inseparable,* co-NP^A*-sets are inseparable,* $\mathrm{R}^A \neq \mathrm{P}^A$, *and* $\mathrm{NP}^A \cap \text{co-}\mathrm{NP}^A = \mathrm{P}^A$ *(13th line in the table).*

PROOF. Take the base $Z = Z(\leq 1) + Z(\geq 1) + Z(\mathrm{R})$. □

THEOREM 37 ([**39**]). *There exists an oracle A such that* $\mathrm{NP}^A \cap \text{co-}\mathrm{NP}^A \neq \mathrm{P}^A$, $\mathrm{BPP}^A \neq \mathrm{P}^A$, *and* $\mathrm{R}^A = \mathrm{P}^A$ *(15th line in the table).*

PROOF. Take the base $Z = Z(=1) + Z(\mathrm{BPP})$. □

THEOREM 38 ([**39**]). *There exists an oracle A such that* $\mathrm{NP}^A \cap \text{co-}\mathrm{NP}^A \neq \mathrm{P}^A$, $\mathrm{R}^A \neq \mathrm{P}^A$, *and* $\mathrm{R}^A \cap \text{co-}\mathrm{R}^A = \mathrm{P}^A$ *(16th line in the table).*

PROOF. Take the base $Z = Z(=1) + Z(\mathrm{R})$. □

The next theorem completes theorems shown in the Table 1. To prove it we need many universums.

THEOREM 39 ([**39**]). *There exists an oracle A such that* $\mathrm{P}^A \neq \mathrm{NP}^A$, NP^A*-sets are separable,* co-NP^A*-sets are separable, and* $\mathrm{BPP}^A = \mathrm{P}^A$ *(second line in the table).*

PROOF. We use a diagonal construction as in the proof Theorem 27 but instead of a chain of intervals we construct a chain of universums

$$V_0 \supseteq V_1 \supseteq \dots$$

Any V_i will be specified by an interval $\Gamma = \Gamma(\alpha)$ and a positive integer j and will consist of all oracles $B \in \Gamma$ such that $B|\mathbb{B}^n \equiv 0$ for all $n \notin \mathbb{S}$ and

$$\#_1(B|\mathbb{B}^n) \leq n/j$$

for all n in $\mathbb{S}$ of length greater than *maxlength* $(\mathrm{Dom}\,\alpha)$. Let $V(\alpha, j)$ stand for the set of B specified in this way by α, j. Obviously, $V(\alpha, j) = \emptyset$ if $\alpha(y) \neq 0$ for some $y \in \mathrm{Dom}\,\alpha$ such that $|y| \notin \mathbb{S}$. So we shall assume that this is not the case. Since any set $V(\alpha, j)$ is closed in Baire's topology, the intersection $\bigcap_{i=0}^{\infty} V_i$ will be non-empty provided all V_i's are non-empty. (We recall that Baire's topology is the topology whose base consists of intervals.)

The oracle A will have the form $B \oplus H$, where H is a PSPACE-complete set. The set in $\mathrm{NP}^A \setminus \mathrm{P}^A$ will be $L^B = \{1^n \mid n \in \mathbb{S},\ \exists u \in \mathbb{B}^n\ B(u) = 1\}$.

We do not present the whole diagonal construction but instead present only specific points. The steps on which we satisfy the requirement $L^B \notin \mathrm{P}^{B \oplus H}$ are made as in all previous proofs, on those steps we make some freezing, thus the current universum $V(\alpha, j)$ is replaced by universum $V(\alpha', j)$ where α' extends α.

On steps on which we satisfy the requirement of separability of NP-sets we use the following analog of Lemma 8.

LEMMA 21. *Assume that N_0, N_1 are NP-machines such that the languages accepted by $N_0^{B\oplus H}$ and $N_1^{B\oplus H}$ are disjoint for any $B \in V(\alpha, j)$. Then there exists a polynomial machine separating those languages for any $B \in V(\alpha, 2j)$.*

PROOF. The polynomial machine separating $\{x \mid N_0^{B\oplus H} = 1\}$ form $\{x \mid N_1^{B\oplus H}\}$ works on input x as follows. Let $n = n_i$ be defined as in previous proofs. An *l-certificate* ($l = 0, 1$) is a function of the form $C|(Query\,{}^{C}_{N_l}(c, C \oplus H) \cap \mathbb{B}^n)$, where C is an oracle in $V(\alpha, 2j)$ agreeing with B on words of length different from n, $N_l^{C\oplus H}(x) = 1$ and c is an accepting computation of $N_l^{C\oplus H}$ on input x.

Let us prove that every 0-certificate is inconsistent with every 1-certificate. Assume the contrary: some 0-certificate ϕ is consistent with some 1-certificate ψ. Let

$$C(y) = \begin{cases} \phi(y) & \text{if } y \in \mathrm{Dom}(\phi); \\ \psi(y) & \text{if } y \in \mathrm{Dom}(\psi); \\ B(y) & \text{if } |y| \neq n; \\ 0 & \text{otherwise.} \end{cases}$$

As $\#_1\phi \leq \frac{n}{2j}$ and $\#_1\psi \leq \frac{n}{2j}$, we have $\#_1(C|\mathbb{B}^n) \leq \frac{n}{2j} + \frac{n}{2j} = \frac{n}{j}$, hence $C \in V(\alpha, j)$. On the other hand $N_0^{C\oplus H}(x) = N_1^{C\oplus H}(x) = 1$. The contradiction shows that each 0-certificate is inconsistent with each 1-certificate.

Further we can reason just as in the proof of Lemma 16. □

LEMMA 22. *Assume that N_0, N_1 are* NP*-machines such that the languages accepted by $N_0^{B\oplus H}$ and $N_1^{B\oplus H}$ span $\mathbb{B}^*$ for all $B \in V(\alpha, j)$. Then there exists a polynomial machine P that on input x with any oracle $B \in V(\alpha, j)$ finds an $l \in \{0, 1\}$ such that $N_l^{B\oplus H}(x) = 1$.*

PROOF. Let the machine P work on input x as follows. Start with the oracle C that is equal to B on words of length different from n and to zero on remaining words. Find an l such that $N_l^{C\oplus H}(x) = 1$ (such l does exist since C is in $V(\alpha, j)$). Then we either discover that $N_l^{B\oplus H}(x) = N_l^{C\oplus H}(x)$ or find a $u \in \mathbb{B}^n$ such that $B(u) = 1$. In the latter case include u in C and repeat the process.

After at most $k = [n/j] + 1$ iterations we will halt since $\#_1(B|\mathbb{B}^n) < k$. □

We need also the analog of Lemma 8 for BPP-machines:

LEMMA 23. *Let M be a BPP-machine that is correct on $B \oplus H$ for any $B \in V(\alpha, j)$. Then there exists a polynomial machine P such that P^B recognizes the same language as $M^{B\oplus H}$ for any $B \in V(\alpha, j)$.*

PROOF. Let the machine P work as follows. Let x be the input. Let n be defined as in Lemma 8. Let the oracle C be equal to B on words of length different from n and to zero on remaining words. Let $U = \{u \in \mathbb{B}^n \mid w_C(u) > 1/(3k)\}$, where k stands for a polynomial upper bound for the number of queries made by M on input x and $w_C(u)$ denotes the probability of event "$M^{C\oplus H}$ on input x queries '$C(u) =?$'". Query '$B(u) =?$' for all $u \in U$. If there is no $u \in U$ such that $B(u) = 1$ then

$$\Big|\mathrm{Prob}[M^{B\oplus H}(x) = 1] - \mathrm{Prob}[M^{C\oplus H}(x) = 1]\Big| \leq \sum_{y:C(y)\neq D(y)} w_C(y) \leq k \cdot \frac{1}{3k} = 1/3.$$

Therefore, in this case $M^{B\oplus H}$ accepts x iff $M^{C\oplus H}$ accepts x.

Otherwise include in C all those $u \in U$ which are in B and repeat the process. After at most $l = [n/j] + 1$ iterations we will halt since $\#_1(B|\mathbb{B}^n) < l$. □

Let us present one more application of the universum method consisting in a new proof of a known theorem.

THEOREM 40 ([**27**]). *There exists an oracle A such that* $\mathrm{P}^A \neq \mathrm{NP}^A \cap \text{co-}\mathrm{NP}^A \neq \mathrm{NP}^A$ *and the class* $\mathrm{NP}^A \cap \text{co-}\mathrm{NP}^A$ *has an m-complete language.*

PROOF. It is sufficient to construct an oracle A such that $\mathrm{NP}^A \cap \text{co-}\mathrm{NP}^A \neq \mathrm{P}^A$, $\mathrm{NP}^A \not\subseteq \text{co-}\mathrm{NP}^A$ and the class $\mathrm{NP}^A \cap \text{co-}\mathrm{NP}^A$ has an m-complete language. The oracle A will have the form $B \oplus H$, where H is a PSPACE-complete set. Thus, we have to construct the oracle B.

Take the universum

$$V = \{A \in \Omega \mid \#_1(A|\mathbb{B}^n) = 1 \text{ for any even } n \text{ and } \#_1(A|\mathbb{B}^n) \leq 1 \text{ for any odd } n\}.$$

The language in $\mathrm{NP}^A \setminus \text{co-}\mathrm{NP}^A$ will be

$$L_1^B = \{1^n \mid n \text{ is odd and } \exists u \in \mathbb{B}^n \ B(u) = 1\}.$$

The language in $\mathrm{NP}^A \cap \text{co-}\mathrm{NP}^A \setminus \mathrm{P}^A$ will be

$$L_2^B = \{1^n \mid n \text{ is even and } \exists u \in \mathbb{B}^{n-1} B(1u) = 1\}.$$

Obviously, $L_1^B \in \mathrm{NP}^A$ and $L_2^B \in \mathrm{NP}^A \cap \text{co-}\mathrm{NP}^A$ for any $B \in V$ (recall that $A = B \oplus H$). Thus, we have to construct an oracle $B \in V$ such that

(1) $L_1^B \notin \text{co-}\mathrm{NP}^A$;
(2) $L_2^B \notin \mathrm{P}^A$;
(3) $\mathrm{NP}^A \cap \text{co-}\mathrm{NP}^A$ has an m-complete language.

To this end let us enumerate all the polynomial-time deterministic and nondeterministic oracle machines and all the pairs of nondeterministic polynomial-time oracle machines. It is easy to see that there is a chain

$$\Gamma_0 \supseteq \Gamma_1 \supseteq \Gamma_2 \supseteq \Gamma_3 \supseteq \dots$$

of intervals such that each Γ_i intersects with V and the following holds. If $i = 3k$, then kth nondeterministic machine does not accept the language $\{0,1\}^* \setminus L_1^B$ for any $B \in \Gamma_i \cap V$. If $i = 3k+1$ then kth deterministic machine does not recognize the set L_2^B for any $B \in \Gamma_i \cap V$. And if $i = 3k+2$ then either the languages accepted by the nondeterministic machines in kth pair are complementary for any $B \in \Gamma_i \cap V$ or those languages are not complementary for any $B \in \Gamma_i \cap V$.

Take any oracle B in $\bigcap_{i=1}^{\infty} \Gamma_i \cap V$. The assertions (1) and (2) are true. It remains to prove that $\mathrm{NP}^A \cap \text{co-}\mathrm{NP}^A$ has a complete language. Denote by N_j the jth nondeterministic polynomial-time Turing machine and by $p_j(|x|)$ a polynomial bounding its running time.

For a $C \in \Omega$ let $C^{\leq n}$ denote the word of length $2^{n+1} - 1$ encoding C's value on words of length at most n in lexicographic order.

Let us note that a pair $\langle N_j, N_k \rangle$ of NP-machines defines a language in $\mathrm{NP}^{B \oplus H} \cap \text{co-}\mathrm{NP}^{B \oplus H}$ iff $N_j^{B \oplus H}(x) + N_k^{B \oplus H}(x) = 1$ for any x. As a complete language we take the following language:

$$L^B = \{\langle j, k, B^{\leq n}, x, 0^{p_j(|x|)+p_k(|x|)} \rangle \mid j,k,n \in \mathbb{N},\ N_j(x, B \oplus H) = 1$$
$$\text{and } N_j(x, C \oplus H) + N_k(x, C \oplus H) = 1 \text{ for any } C \in V \cap \Gamma(B \mid \mathbb{B}^{\leq n})\}.$$

Let us prove that L^B is in $\mathrm{NP}^A \cap \text{co-NP}^A$. To this end let us prove that L^B is in NP^A (the remaining part $L^B \in \text{co-NP}^A$ can be proved entirely similar).

Let us construct first a nondeterministic polynomial space oracle machine that accepts L^B and makes polynomial number of queries.

Let w be an input word. Decide first whether w has the form

$$\langle j, k, D^{\leq n}, x, 0^{p_j(|x|)+p_k(|x|)}\rangle$$

for some $D \in V$ and some j, k, n. Then decide whether $B^{\leq n} = D^{\leq n}$. This can be done within space $|D^{\leq n}| \leq |w|$ and requires $|D^{\leq n}| \leq |w|$ queries to D. Then decide whether $N_j(x, C \oplus H) + N_k(x, C \oplus H) = 1$ for all $C \in V$ such that $C^{\leq n} = D^{\leq n}$. This can be done within polynomial-space since both values $N_j(x, C \oplus H)$ and $N_k(x, C \oplus H)$ depend only on value of C on words of length at most $p_j(|x|) + p_k(|x|)$ and $B|\mathbb{B}^i$ can be described by at most i bits for any $B \in V$, therefore all the needed information about C can be written using polynomial-space. If this is not the case, then reject. Otherwise run N_j on input x with oracle $B \oplus H$ and accept if $N_j(x, B \oplus H) = 1$.

As in the proof of Lemma 9 we can convert the constructed nondeterministic polynomial-space machine into a nondeterministic polynomial-time machine with oracle H.

Thus, it remains to prove that L^B is complete in $\mathrm{NP}^A \cap \text{co-NP}^A$. Let a language L be in $\mathrm{NP}^A \cap \text{co-NP}^A$. Let $\langle N_j, N_k\rangle$ be a pair of nondeterministic polynomial-time oracle machines such that $L(x) = N_j(x, B \oplus H) = 1 - N_k(x, B \oplus H)$ for any x. The construction of the oracle ensures that there exists n such that $N_j(x, C \oplus H) + N_k(x, C \oplus H) = 1$ for any $C \in V \cap \Gamma(B \mid \mathbb{B}^{\leq n})$. Let us fix such an n. The map $x \mapsto \langle j, k, B^{\leq n}, x, 0^{p_j(|x|)+p_k(|x|)}\rangle$ reduces L to L^B. □

Remark 6. In a similar way we could prove all the theorems form this section in a stronger form: we could add the assertion that all the involved classes have m-complete problems.

8.3. General theorems. In this section we formalize the method applied in the previous section. We assume that our goal is to construct an oracle under which a certain Boolean combination of assertions of the form $K_1 \subseteq K_2$ or K_1-sets are K_2-separable holds. As K_1, K_2 we will consider classes of the form $\mathrm{POLY}(F)$.

We need some new definitions.

Definition 12. A *description* is a mapping from the set $\mathbb{B}^* \times \Omega$ into the set $\{0, 1, \#, *\}$. A description D is called an *oracle machine* if $D(x, A) \neq$ for all x, A. A description D is called *correct on* an oracle A if $D(x, A) \neq \#$ for all $x \in \mathbb{B}^*$. For a description D correct an oracle A, let D^A denote the separation problem $x \mapsto D(x, A)$. For a given class $\mathcal{D}$ of descriptions and an oracle A, let $\mathcal{D}^A$ stand for the set $\{D^A \mid D \in \mathcal{D}$ and D is correct on $A\}$. For classes K_1 and K_2 of separation problems, we write $K_1 \leq K_2$ if for any $P_1 \in K_1$ there exists $P_2 \in K_2$ such that $P_1 \leq P_2$.

Note that if K_1 is a class of languages, then $K_1 \leq K_2$ means the same as $K_1 \subseteq K_2$.

Let F be a separation problem. Any function f polynomial-time bit-computable relative to an oracle defines the map $\langle x, A\rangle \mapsto F(f^A(x))$. Let $\mathbf{POLY}(F)$ stand for the class of all oracle machines of this form. It is easy to see that $\mathbf{POLY}^A(F) =$

$\mathrm{POLY}^A(F)$ for any A. Let F be a separation problem. Any pair $\langle f^A, g^A\rangle$ of functions polynomial-time bit-computable relative to A defines the description

$$D(x,A)=\begin{cases}1 & \text{if } F(f^A(x))=1,\ F(g^A(x))=0,\\ 0 & \text{if } F(f^A(x))=0,\ F(g^A(x))=1,\\ * & \text{if } F(f^A(x))=0,\ F(g^A(x))=0,\\ \# & \text{otherwise.}\end{cases} \tag{62}$$

Let $\mathrm{POLY}(F)$-separation denote the class of descriptions D of this form. It is easy to see that $\mathrm{POLY}^A(F)$-sets are $\mathrm{POLY}^A(G)$-separable if $(\mathrm{POLY}(F)\text{-separation})^A \le \mathrm{POLY}^A(G)$. *From now on we shall consider only classes of descriptions having either the form* $\mathbf{POLY}(F)$ *or the form* $\mathrm{POLY}(F)$*-separation.*

Let

$$\mathcal{K}_1,\dots,\mathcal{K}_n,\ \mathcal{L}_1,\dots,\mathcal{L}_n,\ \mathcal{M}_1,\dots,\mathcal{M}_m,\ \mathcal{N}_1,\dots,\mathcal{N}_m$$

be classes of descriptions. We want to prove that there exists an oracle A such that

$$\mathcal{K}_i^A \not\le \mathcal{L}_i^A \quad \text{for } i=1,\dots,n \text{ and}$$
$$\mathcal{M}_j^A \le \mathcal{N}_j^A \quad \text{for } j=1,\dots,m.$$

We shall consider for simplicity of notation the case $m=n=1$.

Let us call a *universum* any nonempty subset V of Ω.

Definition 13. A *superuniversum* is any countable family $\mathcal{V}$ of universums having a largest universum up to inclusion and such that the following two assertions hold:

1. For any $V\in\mathcal{V}$ and for any interval Γ intersecting with V there exists $V'\in\mathcal{V}$ such that $V'\subseteq V\cap\Gamma$.
2. For any countable chain $V_1\supseteq V_2\supseteq V_3\supseteq\dots$ of elements of $\mathcal{V}$ the intersection $\bigcap_{i=1}^{\infty} V_i$ is non-empty.

The reader can see that in all the theorems proven in the previous section the elements of $\mathcal{V}$ have the form $V\cap\Gamma$, where Γ is an interval and V is closed in Baire's topology. Any set of this form is closed in Baire's topology, this implies Condition 2 because the space Ω is compact. For example, if V is closed in Baire's topology, then the family

$$\mathcal{V}(V)=\{V\cap\Gamma \mid \Gamma \text{ is an interval intersecting with } V\}$$

is a superuniversum. Such universums were used in the proofs of Theorems 27–38, 40. The only exception was Theorem 39.

Thus, we wish to prove that there exists an oracle A for which

$$\mathcal{K}^A\not\le\mathcal{L}^A,\qquad \mathcal{M}^A\le\mathcal{N}^A.$$

Notation. Let H be an oracle and $\mathcal{D}$ a class of description. Then $\mathcal{D}_H$ is the class of descriptions $\{\langle x,A\rangle\mapsto D(x,A\oplus H)\mid D\in\mathcal{D}\}$.

For example, $\mathbf{POLY}(F_{\mathrm{BPP}})_H$ is the class of descriptions of the form

$$M(x,A)=\begin{cases}1 & \text{if } \mathrm{Prob}[M^A(x)=1]>2/3;\\ 0 & \text{if } \mathrm{Prob}[M^A(x)=1]<1/3;\\ \# & \text{otherwise,}\end{cases}$$

where M is a polynomial-time probabilistic oracle machine having an extra oracle H. In general, if $\mathcal{K}$ is a class of oracle machines of a certain type, then $\mathcal{K}_H$ is the class of machines of that type having the extra oracle H.

Let $\mathrm{l}(\mathcal{V})$ stand for the largest universum in $\mathcal{V}$. Assume that for a superuniversum $\mathcal{V}$ and for an oracle H the following two assertions are true:

(a) There is a description K in $\mathcal{K}_H$ that is correct on any oracle in $\mathrm{l}(\mathcal{V})$ and such that there are no $L \in \mathcal{L}_H$ and $V \in \mathcal{V}$ such that $K(x,A) \le L(x,A)$ for any x and any $A \in V$;

(b) for any $V \in \mathcal{V}$ and any $M \in \mathcal{M}_H$ correct on any oracle in V there are $N \in \mathcal{N}_H$ and $V' \in \mathcal{V}$ such that $V' \subseteq V$ and $M(x,A) \le N(x,A)$ for any x and any $A \in V'$.

We claim that in this case there is an oracle A such that $\mathcal{K}^A \not\le \mathcal{L}^A$ and $\mathcal{M}^A \le \mathcal{N}^A$.

THEOREM 41 ([**38**]). *Assume that there are H and $\mathcal{V}$ such that (a) and (b) are true. Then there is an oracle A such that $\mathcal{K}^A \not\le \mathcal{L}^A$ and $\mathcal{M}^A \le \mathcal{N}^A$.*

PROOF. Let K satisfy (a). We shall construct an oracle $B \in \mathrm{l}(\mathcal{V})$ such that the separation problem K^B is not easier than any problem in $(\mathcal{L}_H)^B$ and such that $(\mathcal{M}_H)^B \le (\mathcal{N}_H)^B$. Then for the oracle $A = B \oplus H$ the assertions $\mathcal{K}^A \not\le \mathcal{L}^A$ and $\mathcal{M}^A \le \mathcal{N}^A$ will be true.

Let M_0, M_1, ..., M_i, ... be an enumeration of $\mathcal{M}_H$ and let L_0, L_1, ..., L_j, ... an enumeration of $\mathcal{L}_H$. We have to satisfy countably many requirements of two types: for each $i \in \mathbb{N}$ we have to satisfy the requirement

$$K^B \not\le L_i^B$$

and for each $i \in \mathbb{N}$ the requirement

$$M_i \text{ is incorrect on } B \text{ or } M_i^B \le N^B \text{ for some } N \in \mathcal{N}_H. \tag{63}$$

We make a countable number of steps. On step j we find a universum V_j so that $V_1 \supseteq V_2 \supseteq V_3 \supseteq \ldots$. More exactly, on step $j = 2i+1$ we construct a universum V_j such that $K^B \not\le L_i^B$ for any $B \in V_j$. On step $j = 2i+2$ we ensure the ith condition of the form (63), that is, we construct a universum V_j such that M_i is incorrect on B or $M_i^B \le N^B$ for some $N \in \mathcal{N}_H$ and for all $B \in V_j$.

As B we take any oracle from the set $\bigcap_{i=1}^{\infty} V_i$. Set $V_0 = \mathrm{l}(\mathcal{V})$. It remains to describe each step. Let j be the number of the current step. Consider two cases.

First case: $j = 2i+1$. Since K satisfies (a), there are $x \in \mathbb{B}^*$ and $C \in V_{j-1}$ such that $K(x,C) \not\le L_i(x,C)$. Choose x and C satisfying this inequality. There is an interval Γ including C such that $K(x,B) = K(x,C)$ and $L_i(x,B) = L_i(x,C)$ for all $B \in \Gamma$. We have, therefore, $K(x,B) \not\le L_i(x,B)$ for all $B \in \Gamma$. By Condition 1 in the definition of a superuniversum, there is a universum $V \in \mathcal{V}$ such that $V \subseteq V_{j-1} \cap \Gamma$. Let $V_j = V$. Obviously, $K^B \not\le L_i^B$ for all $B \in V_j$.

Second case: $j = 2i+2$. Assume first that M_i is correct on V_{j-1}. By Condition (b) there are a description $N \in \mathcal{N}_H$ and a universum $V' \subseteq V_{j-1}$, $V' \in \mathcal{V}$, such that $M_i(x,B) \le N(x,B)$ for all $x \in \mathbb{B}^*$ and all $B \in V'$. Then we can set $V_j = V'$. Obviously, for all $B \in V_j$ the assertion (63) holds. Otherwise (when M_i is incorrect on V_{j-1}) we can reason as in the first case because "to be incorrect" is a local property. □

Thus to construct an oracle under which $\mathcal{K} \not\leq \mathcal{L}$ and $\mathcal{M} \leq \mathcal{N}$ it suffices to find a superuniversum $\mathcal{V}$ and an oracle H such that the conditions (a) and (b) above are fulfilled. In this form, the method is universal. Indeed, if $\mathcal{K}^A \not\leq \mathcal{L}^A$ and $\mathcal{M}^A \leq \mathcal{N}^A$ then (a) and (b) are fulfilled for $\mathcal{V} = \{\{A\}\}, H = \emptyset$ as well as for $\mathcal{V} = \{\{\emptyset\}\}, H = A$. Thus Theorem 41 does not capture the essence of the method. Roughly speaking to find a specific oracle we need to find a universum (or a family of universums) with specific inner structure. And the oracle H should not be specific. Both the universums $\{A\}$, $\{\emptyset\}$ do not satisfy this requirement, they have very simple inner structure—they consist of a single oracle. We shall present another general theorem, which captures the essence of the universum method.

Given a class $\mathcal{D}$ of descriptions, we define a nonuniform counterpart of the class $\mathcal{D}$ as follows.

DEFINITION 14. The nonuniform counterpart of a class $\mathcal{D}$ of descriptions is the class

$$\text{n.u.}\mathcal{D} = \bigcup_{C \in \Omega} \mathcal{D}_C.$$

REMARK 7. If $\mathcal{K} = \mathbf{POLY}(F)$ then there is an equivalent definition of n.u.$\mathcal{K}$ using decision trees. Recall that $F^{\leq n}$ denotes the binary word of length $2^{n+1} - 1$ encoding the value of F on the words of length $\leq n$. We say that a function $f^A(x)$ of A and x is bit-computable in polynomial number of queries if there are two families, T_x and $S_{x,i}$, of decision trees such that (i) the height of both T_x and S_x is polynomial in $|x|$ and (ii) $T_x(A^{\leq p(|x|)}) = |f^A(x)| \leq p(|x|)$ and $S_{x,i}(A^{\leq p(|x|)}) =$ (ith bit of $f^A(x)$) for some polynomial p and for any x and $i \leq |f^A(x)|$. We claim that n.u.$\mathbf{POLY}(F)$ is the class of descriptions of the form $\langle x, A\rangle \mapsto F(f^A(x))$, where $f^A(x)$ is a function bit-computable in polynomial number of queries. Let us verify this for $F = F_{\mathrm{P}}$. We have to prove that a description D is in n.u.$\mathbf{P}$ iff there are a family $\{T_x \mid x \in \mathbb{B}^*\}$ of Boolean decision trees and a polynomial p such that $T_x(A^{\leq p(|x|)}) = D(x, A)$ and the height of T_x does not exceed $p(|x|)$ for all x.

Assume that D is in n.u.$\mathbf{P}$, say $D(x, A) = M(x, A \oplus C)$, where M is a polynomial-time oracle machine and C is an oracle. It is clear that for any x the value $M(x, A \oplus C)$ can be computed by a Boolean decision tree of height $p(|x|)$ in the variables $A(y)$, $|y| \leq \mathrm{poly}(|x|)$. Conversely, assume that a description D is computable by a family of decision trees, $D(x, A) = T_x(A^{\leq p(|x|)})$, where p is a polynomial and the height of T_x is at most $p(|x|)$. We may assume that the nodes of T_x are binary words (the root is the empty word, and $v0$ and $v1$ are the left and the right sons, respectively, of the node v). Then take as C the oracle encoding for all x and all vertices v of T_x the label of v and the indication whether v is a leaf in T_x or not. Then there is a polynomial-time machine computing $D(x, A)$ given x as input and $A \oplus C$ as oracle.

Similarly, n.u.$\mathbf{POLY}(F)$-separation consists of descriptions of the form (62), where both $f^A(x), g^A(x)$ are bit-computable in polynomial number of queries.

Now we are able to present the second general theorem which is the essence of the universum method. Consider the following two conditions:

(a′) There is a description $K \in \text{n.u.}\mathcal{K}$ that is correct on any oracle in $\mathrm{l}(\mathcal{V})$ and such that there are no $L \in \text{n.u.}\mathcal{L}$ and $V \in \mathcal{V}$ such that $K(x, A) \leq L(x, A)$ for any x and any $A \in V$;

(b′) for any $V \in \mathcal{V}$ and any description $M \in$ n.u.$\mathcal{M}$ correct on any oracle in V there are $N \in$ n.u.$\mathcal{N}$ and $V' \in \mathcal{V}$ such that $V' \subseteq V$ and $M(x, A) \leq N(x, A)$ for any x and any $A \in V'$.

Note that (a′) and (b′) are obtained from (a) and (b), respectively, by replacing uniform classes relativized with H by the corresponding nonuniform classes.

THEOREM 42 ([**38**]). *Assume that (a′) and (b′) are true. Then there is an oracle H such that (a) and (b) hold.*

PROOF. Assume that both (a′) and (b′) are true. Let D be an oracle such that $K \in \mathcal{K}_D$. Then (a) holds for all oracles H to which D is polynomial-time Turing reducible. Thus, it suffices to construct an oracle H such that (b) is true and such that D is reducible to H.

We shall assume that the classes $\mathcal{M}, \mathcal{N}$ have the form $\mathbf{POLY}(F)$, $\mathcal{M} = \mathbf{POLY}(F)$ and $\mathcal{N} = \mathbf{POLY}(G)$. Other cases are entirely similar.

Let f_j^A, $j = 0, 1, 2, \ldots$, be an enumeration of functions that are bit-computable in polynomial number of queries. Then

$$\mathcal{M}^A = \{F(f_j^A(x)) \mid F(f_j^A(x)) \text{ is correct on } A\},$$
$$\mathcal{N}^A = \{G(f_m^A(x)) \mid G(f_m^A(x)) \text{ is correct on } A\}.$$

Let $\mathcal{V} = \{V_0, V_1, V_2, \ldots\}$.

Let $j, l \in \mathbb{N}$. There is a polynomial p_j such that $F(f_j^A(x))$ depends only on x and $A|\mathbb{B}^{\leq p_j(|x|)}$. Let x be in $\mathbb{B}^*$ and F in Ω. Consider the set $U = U(j, l, x, F^{\leq p_j(|x|)})$ consisting of all the pairs $\langle k, B\rangle$, $k \in \mathbb{N}$, $B \in \Omega$, such that

$$F(f_j^{A\oplus F}(x)) \leq G(f_k^{A\oplus B}(x))$$

for all $x \in \mathbb{B}^*$ and for all $A \in V_l$. Take a pair $\langle k, B\rangle$ from U having the minimal sum

$$k + (\text{time of bit-computation of } f_k \text{ on input } x).$$

Let C be an oracle encoding in a natural way the map $\langle j, l, x, F^{\leq p_j(|x|)}\rangle \mapsto \langle k, B\rangle$. More exactly, with oracle C it is possible to compute k in time $\mathrm{poly}(j + l + |x| + 2^{p_j(|x|)} + k)$ for any given $j, l, x, F^{\leq p_j(|x|)}$ and it is possible to compute $B(y)$ in time $\mathrm{poly}(j + l + |x| + 2^{p_j(|x|)} + k + |y|)$ for any given $j, l, x, F^{\leq p_j(|x|)}, y$. Let H be a language that is complete in $\mathrm{EXP}^{C\oplus D}$. As D is reducible to H, the condition (a′) is true. Let us prove (b′). Fix i, j such that the description $F(f_j^{A\oplus H}(x))$ is correct on any oracle $A \in V_i$.

We have to prove that there exist a universum $V_l \subseteq V_i$ and k' such that

$$F(f_j^{A\oplus H}(x)) \leq G(f_{k'}^{A\oplus H}(x))$$

for all $x \in \mathbb{B}^*$ and all $A \in V_l$. By (b′) there exist k', l, and an oracle B' such that $V_l \subseteq V_i$ and

$$F(f_j^{A\oplus H}(x)) \leq G(f_{k'}^{A\oplus B'}(x))$$

for all $A \in V_l$ and all x. Let us choose such k', l, B'.

Fixing j, l and $F = H$ in the map $j, l, x, F^{\leq p_j(|x|)} \mapsto \langle k, B\rangle$ we obtain two functions: $k(x)$ and $B(x, y)$ (the value of B on y). Thus, we have

$$F(f_j^{A\oplus H}(x)) \leq G(f_{k(x)}^{A\oplus B(x)}(x))$$

for all $A \in V_l$ and all x.

LEMMA 24. *Both functions $k(x)$ and $B(x,y)$ are polynomial-time computable relative to H.*

PROOF. Since the pair $\langle k', B' \rangle$ belongs to U and

$$k' + (\text{time of bit-computation of } f_{k'} \text{ on input } x) \le \text{poly}(|x|).$$

we can conclude that

$$k(x) + (\text{time of bit-computation of } f_{k(x)} \text{ on input } x) \le \text{poly}(|x|). \tag{64}$$

Let us prove first that $k(x)$ is polynomial-time computable (with oracle H). By (64), we have $k(x) \le \text{poly}(|x|)$, therefore it suffices to prove that $k(x)$ can be computed in time $2^{\text{poly}(|x|)}$ with oracle $C \oplus D$. This can be done as follows. Find first $H^{\le p_j(|x|)}$. To this end compute $H(z)$ for all z with length at most $p_j(|x|)$ using an exponential machine with oracle $C \oplus D$ recognizing H. (As $|z| \le p_j(|x|)$, each $H(z)$ can be computed in time $2^{\text{poly}(|x|)}$.) Then query C to find $k(x)$. The queries to C have length at most $2^{\text{poly}(|x|)}$.

The polynomial-time computability of the function $B(x,y)$ can be proved similarly. □

We may assume that the enumeration $\{f_n\}$ is so that the polynomial-time computability with oracle H of the functions $k(x)$ and $B(x,y)$ implies that $f_{k(x)}^{A \oplus B(x)}(x) = f_m^{A \oplus H}(x)$ for some m and for all x, A. Consequently, we have $F(f_j^{A \oplus H}(x)) \le G(f_m^{A \oplus H}(x))$ for all $A \in V_l$ and all x. □

8.4. When the universum method cannot be used. We say that the universum method can be applied to prove that there is an oracle A such that $\mathcal{K}^A \not\le \mathcal{L}^A$ and $\mathcal{M}^A \le \mathcal{N}^A$ if there is a superuniversum $\mathcal{V}$ such that (a′) and (b′) are true. In this section we present two theorems that cannot be proven by the universum method.

THEOREM 43 ([**32**]). *There exists an oracle A such that*

$$\mathrm{P}^A = \mathrm{NP}^A \ne \mathrm{PSPACE}^A.$$

THEOREM 44 ([**43**]). *There exists an oracle A such that*

$$\mathrm{P}^A \ne \mathrm{R}^A = \mathrm{PSPACE}^A.$$

In the proofs of both Theorems 43 and 44, some difficult-to-compute information is encoded via oracle values, to ensure the truth of positive assertion ($\mathcal{M}^A \le \mathcal{N}^A$). To prove Theorem 43, one needs a lower bound of [**25, 55**] for complexity of computation of PARITY by means of AND,OR-circuits of bounded depth, which has a rather complicated proof. Theorem 44 was proved in [**43**] (in fact, the weaker assertion that $\mathrm{P} \ne \mathrm{R} = \mathrm{NP}$ under some oracle was proved there but the proof holds good also for our case).

Actually, we can prove that the following corollaries of Theorems 43 and 44 cannot be proven by the universum method.

COROLLARY 45. *There exists an oracle A such that*

$$\mathrm{NP}^A = \text{co-NP}^A \ne \mathrm{PSPACE}^A$$

COROLLARY 46. *There exists an oracle A such that*

$$\mathrm{P}^A \ne \mathrm{PSPACE}^A \text{ and } \text{co-NP}^A \subseteq \mathrm{R}^A.$$

Let **P**, **R**, **NP**, co-**NP**, **PSPACE** denote $\mathbf{POLY}(F_{\mathrm{P}})$, $\mathbf{POLY}(F_{\mathrm{R}})$, $\mathbf{POLY}(F_{\mathrm{NP}})$, $\mathbf{POLY}(F_{\text{co-NP}})$, $\mathbf{POLY}(F_{\mathrm{PSPACE}})$, respectively.

THEOREM 47 ([**38**]). *The universum method cannot be applied to prove Corollary 45, that is, there is no superuniversum $\mathcal{V}$ such that (a′) and (b′) are true for $\mathcal{K} = \mathbf{PSPACE}$, $\mathcal{L} = \mathbf{NP}$ and $\mathcal{M} = \text{co-}\mathbf{NP}$, $\mathcal{N} = \mathbf{NP}$.*

PROOF. Let V be a universum. Say that an interval Γ *n-isolates an oracle A in V* if $A \in \Gamma$ and $A|\mathbb{B}^n = B|\mathbb{B}^n$ for any $B \in V \cap \Gamma$. A set $W \subseteq \mathbb{B}^*$ *n-isolates an oracle A in V* if the interval $\Gamma(A|W)$ n-isolates A in V. We define the size of a finite set of words as the sum of the lengths of all its elements. The size of an interval $\Gamma = \Gamma(\phi)$ is defined as the size of $\mathrm{Dom}(\phi)$. We say that V is *thin* if there exists a polynomial $p(n)$ such that for all $A \in V$ and all $n \in \mathbb{N}$, there exists a set $W \subseteq \mathbb{B}^*$ of size $p(n)$ that n-isolates A in V. Consider two cases.

First case: there exists a universum V in $\mathcal{V}$ that is thin. Let $p(n)$ be the corresponding polynomial. Let us prove that (a′) is false in this case. Indeed, let K be an oracle machine from n.u.**PSPACE**. Let us construct a machine L in n.u.**NP** such that $K^A = L^A$ for all $A \in V$. The machine L on input $\langle x, A\rangle$ works as follows. Let the length of queries of K to oracle on input x be bounded by the polynomial $q(|x|)$. For every $i \le q(|x|)$ guess a set $W_i \subseteq \mathbb{B}^*$ such that $size(W_i) \le p(i)$. Ask '$A(y) =?$' for all y in W_i. If there is no $B \in V$ such that the interval $\Gamma(A|W_i)$ i-isolates B in V for all $i \le q(|x|)$, then reject. Otherwise, choose such B. Note that if A is in V, then $A|\mathbb{B}^{\le q(|x|)} = B|\mathbb{B}^{\le q(|x|)}$, and therefore $K(x, A) = K(x, B)$. Then accept iff $K(x, B) = 1$.

The total number of queries to A is $\sum_{i=0}^{q(|x|)} p(i) = \mathrm{poly}(|x|)$. The maximal length of a query is $\max_{i \le q(|x|)} p(i) = \mathrm{poly}(|x|)$. Thus, $L \in$ n.u.**NP**.

Second case: all the universums in $\mathcal{V}$ are not thin. We claim that in this case (b′) is false for $\mathcal{M} = \text{co-}\mathbf{NP}$, $\mathcal{N} = \mathbf{NP}$. Let $V_0, V_1, V_2, \ldots, V_i, \ldots$ be an enumeration of universums in $\mathcal{V}$. For an oracle A, let $m^i_A(n)$ denote the minimal size of set n-isolating A in V_i. For all $i \in \mathbb{N}$, fix a sequence $\{B^{in}\}$, $n \in \mathbb{N}$, of oracles such that $B^{in} \in V_i$ and such that for any i, $m^i_{B^{in}}(n)$ is not bounded by any polynomial of n. The set of natural numbers can be partitioned into subsets $Q_0, Q_1, Q_2, \ldots, Q_i, \ldots$ such for any $i \in \mathbb{N}$, $m^i_{B^{in}}(n)$ is not bounded by any polynomial of n when n ranges over Q_i. Obviously, there is a description $M \in$ n.u.co-**NP** such that

$$M(1^n, A) = \begin{cases} 1 & \text{if } \forall y \in \mathbb{B}^n\, A(y) = B^{in}(y), \text{ where } i \\ & \text{is the number such that } n \in Q_i; \\ 0 & \text{otherwise.} \end{cases}$$

Let us prove that there are no $N \in$ n.u.**NP** and $i \in \mathbb{N}$ such that $M^A = N^A$ for all $A \in V_i$. Suppose the contrary: such N, i exist. Let $p(n)$ denote the polynomial bounding the size of the set $Query_N(1^n, A)$. Then for all $A \in V_i$a nd all $n \in Q_i$,

$$A|\mathbb{B}^n = B^{in}|\mathbb{B}^n \Leftrightarrow M(1^n, A) = 1 \Leftrightarrow N(1^n, A) = 1.$$

In particular, $N(1^n, B^{in}) = 1$. Pick an accepting computation c of N on input $\langle 1^n, B^{in}\rangle$ and set $W = Query_N(c, B^{in})$. Then $size(W) \le p(n)$ and $N(1^n, A) = 1$ for all $A \in \Gamma(B^{in}|W)$. Therefore, $A|\mathbb{B}^n = B^{in}|\mathbb{B}^n$ for all $A \in V_i \cap \Gamma(B^{in}|W)$ that is, W n-isolates B^{in} in V^i. Consequently, $m^i_{B^{in}} \le p(n)$ for all $n \in Q_i$. The contradiction thus obtained proves the theorem. □

THEOREM 48 ([**38**]). *Corollary 46 cannot be proven by the universum method, that is, there exists no superuniversum $\mathcal{V}$ such that (a′) and (b′) hold for $\mathcal{K} = \mathbf{PSPACE}$, $\mathcal{L} = \mathbf{P}$ and $\mathcal{M} = \text{co-}\mathbf{NP}$, $\mathcal{N} = \mathbf{R}$.*

PROOF. Assume that T is a decision tree in variables $A(x)$, $x \in \mathbb{B}^{\leq m}$, for some m. Let $A \in V$. We say that T *n-identifies A in V* if the set of values of A learned by T in the computation on input A^m, *Query* $_T(A^m)$, n-isolates A in V. A universum V is called *identifiable* if there is a family P_n, $n \in \mathbb{N}$, of decision trees and a polynomial p such that (i) P_n is a tree in variables $A(x)$, $|x| \leq p(n)$, (ii) the height of P_n is at most $p(n)$, and (ii) P_n n-identifies A in V for all $n \in \mathbb{N}$ and all $A \in V$. We say that a universum V is *randomly identifiable* if there are a family $P_{n,r}$, $n \in \mathbb{N}$, $r \in \mathbb{B}^{\text{poly}(n)}$, of decision trees and a polynomial p such that (i) $P_{x,r}$ is a decision tree in variables $A(x)$, $x \in \mathbb{B}^{\leq p(n)}$, (ii) the height of $P_{n,r}$ is at most $p(n)$ and

$$\text{Prob}\,[P_{nr} \text{ } n\text{-identifies } A \text{ in } V] > \frac{1}{2} \tag{iii}$$

(with respect to the uniform distribution in r's) for all $n \in \mathbb{N}$ and all $A \in V$. Obviously

$$\begin{aligned} (V \text{ is identifiable}) &\Rightarrow (V \text{ is randomly identifiable}) \\ &\Rightarrow (V \text{ is thin}). \end{aligned}$$

LEMMA 25. *Any randomly identifiable universum is identifiable.*

PROOF. Assume that V is randomly identifiable. Let $P_{x,r}$ be a family of decision trees and p a polynomial satisfying (i), (ii) and (iii) above.

For an $n \in \mathbb{N}$, let Z_n denote the set $\{B|\mathbb{B}^{\leq n} \mid B \in V\}$. We claim that $|Z_n| \leq 2^{\text{poly}(n)}$. Indeed, for all $n \in \mathbb{N}$ and all $B \in V$, there is a set $W \subseteq \mathbb{B}^{\leq p(n)}$ having at most $p(n)$ elements and such that $C|\mathbb{B}^{\leq n} = B|\mathbb{B}^{\leq n}$ for any $C \in V \cap \Gamma(B|W)$. Thus, $|Z_n|$ is not greater than the number of elements in the set

$$\{B|W \mid W \subseteq \mathbb{B}^{\leq p(n)},\ |W| \leq p(n),\ B \in \Omega\}.$$

This number does not exceed $(2^{p(n)+1})^{p(n)}$. Thus, $|Z_n| \leq (2^{p(n)+1})^{p(n)} 2^{p(n)} = 2^{\text{poly}(n)}$.

Let for a partial function $\alpha\colon \mathbb{B}^* \to \mathbb{B}$, let $\hat{\alpha}$ denote the oracle

$$\hat{\alpha}(z) = \begin{cases} \alpha(z) & \text{if } z \in \text{Dom}(\alpha); \\ 0 & \text{if } z \in \mathbb{B}^* \setminus \text{Dom}(\alpha). \end{cases}$$

Obviously, for any A, n and r,

$$T_{n,r} \text{ } n\text{-identifies } A \text{ in } V \Leftrightarrow T_{n,r} \text{ } n\text{-identifies } \widehat{A|B^{p(n)}} \text{ in } V. \tag{65}$$

Therefore, we have

$$\text{Prob}\,[T_{n,r} \text{ } n\text{-identifies } \hat{\alpha} \text{ in } V] > 1/2$$

for any $\alpha \in Z_{p(n)}$. We conclude that

$$\text{Prob}\,[T_{n,r} \text{ } n\text{-identifies } \hat{\alpha} \text{ in } V] > 1/2$$

with respect to the uniform distribution in r's and in $\alpha \in Z_{p(n)}$. Therefore, there exists r_1 such that

$$\text{Prob}\,[T_{n,r_1} \text{ } n\text{-identifies } \hat{\alpha} \text{ in } V] > 1/2$$

with respect to the uniform distribution in $\alpha \in Z_{p(n)}$. Let

$$Z' = \{\alpha \in Z_{p(n)} \mid T_{n,r_1} \text{ does not } n\text{-identify } \hat{\alpha} \text{ in } V\}.$$

Reasoning in a similar way we can prove that there exists r_2 such that

$$\mathrm{Prob}\,[T_{n,r_2}\ n\text{-identifies } \hat{\alpha} \text{ in } V] > 1/2$$

with respect to the uniform distribution in $\alpha \in Z'$. Let

$$Z'' = \{\alpha \in Z' \mid T_{n,r_2} \text{ does not } n\text{-identify } \hat{\alpha} \text{ in } V\}$$

and so on. We thus define the words r_1, r_2, ..., r_i, ... and the sets Z', Z'', ..., $Z^{(i)}$, ... Evidently,

$$|Z^{(i+1)}| < (1/2)|Z^{(i)}|$$

for all i, therefore for some polynomial $t(n)$ we get $Z^{t(n)} = \emptyset$. Thus for any $\alpha \in Z_{p(n)}$ there exists an $i \le t(n)$ such that T_{n,r_i} n-identifies $\hat{\alpha}$ in V. By (65), this means that for any $A \in V$ there exists $i \le t(n)$ such that T_{nr} n-identifies A in V.

The decision tree T_n that n-identifies $A \in V$ in V works as follows. It just executes $T_{n,r_1}, \ldots, T_{n,r_{t(n)}}$. The height of T_n is $t(n)p(n)$ and its input variables are $A(x)$, $|x| \le p(n)$. □

Let us continue the proof of the theorem. Consider two cases.

First case: there exists a randomly identifiable universum $V \in \mathcal{V}$. Then by the above lemma the universum V is identifiable. Therefore, for any $K \in$ n.u.**PSPACE** there is an $L \in$ n.u.**P** such that $K(x, A) = L(x, A)$ for all x and all $A \in V$ (first identify the oracle A in V and then find $K(x, A)$). Therefore, (a′) is false for $\mathcal{K} = \mathbf{PSPACE}$, $\mathcal{L} = \mathbf{P}$.

Second case: all universums $V \in \mathcal{V}$ are not randomly identifiable. Let us prove that (b′) is false for $\mathcal{M} =$ co-**NP**, $\mathcal{N} = \mathbf{R}$.

Let V_0, V_1, ..., V_i, ... be an enumeration of $\mathcal{V}$. Let $m^i_B(n)$ denote the least $s \in \mathbb{N}$ such that there exists a family T_r, $r \in \mathbb{B}^s$, of decision trees of height at most s in variables $A(x)$, $|x| \le s$, such that $\mathrm{Prob}\,[T_r\ n\text{-identifies } B \text{ in } V_i] > 1/2$. We know that for any i there exists a sequence $\{B^{in}\}$, $n \in \mathbb{N}$, of oracles in V_i such that the function $n \mapsto m^i_{B^{in}}(n)$ is not bounded by any polynomial of n. Obviously, the set $\mathbb{N}$ can partitioned into subsets Q_0, Q_1, ..., Q_i, ... such that for each $i \in \mathbb{N}$ the function $m^i_{B^{in}}(n)$ is not bounded by any polynomial of n when n ranges over Q_i. Obviously, there exists a description $M \in$ n.u.co-**NP** such that

$$M(1^n, A) = 1 \Leftrightarrow A|\mathbb{B}^n = B^{in}|\mathbb{B}^n.$$

for any $i \in \mathbb{N}$, any $n \in Q_i$ and any A. Assume that there are a description N in n.u.**R** and $i \in \mathbb{N}$ such that $M(x, A) = N(x, A)$ for all x and all $A \in V_i$.

Let $q(n)$ be the length of random strings used by N on inputs of the form $\langle 1^n, A\rangle$ ($A \in \Omega$). For any $r \in \mathbb{B}^{q(n)}$, let $N(1^n, r, A)$ denote the output of N on input $\langle 1^n, A\rangle$ and random input r, and $Query_N(1^n, r, A)$ the set of queries to A made by N during the work on the input $\langle 1^n, A\rangle$ and the random input r. Then

$$\begin{aligned} &A|\mathbb{B}^n \ne B^{in}|\mathbb{B}^n \Rightarrow M(1^n, A) = 0 \Rightarrow \mathrm{Prob}_{r\in\mathbb{B}^{q(n)}}\,[N(1^n, r, A) = 1] = 0, \\ &A|\mathbb{B}^n = B^{in}|\mathbb{B}^n \Rightarrow M(1^n, A) = 1 \Rightarrow \mathrm{Prob}_{r\in\mathbb{B}^{q(n)}}\,[N(1^n, r, A) = 1] > 1/2 \end{aligned} \tag{66}$$

for any $A \in V_i$ and any n. In particular,

$$\mathrm{Prob}_{r\in\mathbb{B}^{q(n)}}\,\left[N(1^n, r, B^{ni}) = 1\right] > 1/2$$

for all $n \in Q_i$. Assertion (66) implies that if $A \in V_i$ and $A|\mathbb{B}^n \neq B^{ni}|\mathbb{B}^n$, then $N(1^n, r, A) = 0$ for all $r \in \mathbb{B}^{q(n)}$.

Thus, the set $Query_N(1^n, r, B^{ni})$ n-isolates B^{ni} in V_i for any $r \in \mathbb{B}^{q(n)}$ such that $N(1^n, r, B^{ni}) = 1$. Therefore $m^i_{B^{in}}(n)$ grows polynomially when n ranges Q_i. The contradiction thus obtained finishes the proof. □

9. Relationships between complexity classes relativized with a random oracle

The study of Complexity theory relative to a random oracle was initiated in [**8**]. It was proved that $\mathrm{P}^A \neq \mathrm{NP}^A \neq \text{co-NP}^A$ for a random A. Another result of [**8**] states that, for a random A, there exists an infinite P^A-immune NP^A-set. (The relevant notions are defined below, see Definition 15.)

Let us look at these results from the point of view of the analogy between recursion theory and complexity theory. According to this analogy, P-sets correspond to decidable sets, NP-sets to recursively enumerable sets, and co-NP-sets to complements of recursively enumerable sets. More precisely, we shall consider complexity theory relative to a random oracle. Then decidable sets correspond to P^A-sets, recursively enumerable sets to NP^A-sets and complements of recursively enumerable sets to co-NP^A-sets, where A is a random oracle. Thus, analogs of the following theorems are true in complexity theory: the theorem on the existence of a recursively enumerable undecidable set, the theorem on the existence of recursively enumerable set whose complement is not recursively enumerable. Analog of the theorem that any infinite recursively enumerable set has an infinite decidable subset is false.

In this section we show that analogs of the following theorems are true: the theorem on the existence of recursively enumerable inseparable sets, the theorem on the existence of a simple set (recursively enumerable set whose complement is infinite but has no infinite recursively enumerable subsets). We show also that the analog the theorem on separability of sets having recursively enumerable complements is false.

It is unknown whether $\mathrm{NP}^A \cap \text{co-NP}^A = \mathrm{P}^A$ is true for a random A (this is analogous to Post's theorem, which asserts that a set X is decidable whenever both X and its complement are recursively enumerable). According to [**9**], a positive answer would imply that AM $\cap$ co-AM = BPP, and therefore that the problem of graph isomorphism is in BPP. Thus, there is no hope of proving that $\mathrm{NP}^A \cap \text{co-NP}^A = \mathrm{P}^A$ for a random A. On the other hand, no absolute corollary of the assertion "$\mathrm{NP}^A \cap \text{co-NP}^A \neq \mathrm{P}^A$ for a random A" is known. Thus one may hope to prove that $\mathrm{NP}^A \cap \text{co-NP}^A \neq \mathrm{P}^A$ for a random A.

All the result cited above are presented in the Table 3. The analogous results on generic oracle are also shown there for comparison.

DEFINITION 15. A language L is called $\mathcal{C}$-immune if L is infinite but no infinite subset of L belongs to $\mathcal{C}$.

Let $S(A)$ be a property of an oracle A. We say that $S(A)$ holds for a random A (or for almost all A) if the uniform measure of the set $\{A \mid S(A)\}$ is equal to 1.

THEOREM 49 ([**54**]). *Relative to a random oracle, there are $L \in$ P and an* NP*-set $L_1 \subseteq L$ such that both L_1 and $L \setminus L_1$ are infinite, L_1 has no infinite* co-NP*-subsets and $L \setminus L_1$ has no infinite* NP*-subsets.*

TABLE 3

	Relative to random oracle	Relative to generic oracle	Recursion theory
$P \neq NP$	+	+	+
$NP \neq$ co-NP	+	+	+
NP-sets are P-inseparable	+	?	+
co-NP-sets are P-inseparable	+	+	−
$P = NP \cap$ co-NP	?	?	+
NP has P-immune sets	+	−	−
NP has co-NP-immune sets	+	−	−
co-NP has NP-immune sets	+	+	+

PROOF. Define a sequence $\{t_i\}$ of integers by induction: $t_0 = 1$, $t_{i+1} = 2^{t_i}$. Let i be a natural number. An *i-block* is a set of the form

$$B_w = \{wv \mid v \in \mathbb{B}^*,\ |v| = \log_2 t_i\},$$

where w is a binary word of length t_i. Thus, every i-block consists of t_i words (of length $t_i + \log_2 t_i$). Let

$$L = \{1^{t_i} \mid\ i \in \mathbb{N}\}.$$

Let B be a block. We say that an oracle A is *identically zero in B* if $\forall u \in B\ A(u) = 0$. For any oracle A let

$$\begin{aligned} L_1^A &= \{1^{t_i} \mid\ i \in \mathbb{N} \text{ and } A \text{ is identically zero in some } i\text{-block}\}, \\ L_0^A &= L \setminus L_1^A. \end{aligned}$$

Obviously, L is in P^A and L_1^A is in NP^A for all A. Both languages L_0^A and L_1^A are infinite for almost all A, because

$$\text{Prob}[1^{t_i} \in L_0^A] = 1 - \text{Prob}[1^{t_i} \in L_1^A] = (1 - 2^{-t_i})^{2^{t_i}} \to e^{-1}$$

as $i \to \infty$. Let us prove that

1) L_0^A has no infinite NP^A-subsets for almost all A, and
2) L_1^A has no infinite co-NP^A-subsets for almost all A.

We claim that instead of 1) it suffices to prove the following assertion:

1′) there exists $c < 1$ such that for any nondeterministic machine N, the probability of the event "L_{N^A} is infinite and $L_{N^A} \subseteq L_0^A$" is less than c,

and that, instead of 2), it suffices to prove the following assertion:

2′) there exists $c < 1$ such that for any nondeterministic machine N, the probability of the event "$\mathbb{B}^* \setminus L_{N^A}$ is infinite and $(\mathbb{B}^* \setminus L_{N^A}) \subseteq L_1^A$" is less than c.

We first prove the implication 1′) $\Rightarrow$ 1). Assume that 1′) is true but the probability of event "L_0^A has an infinite NP^A-subset" is positive. Then there exists a nondeterministic machine N such that the probability of the event "L_{N^A} is infinite and $L_{N^A} \subseteq L_0^A$ " is positive. A simple theorem from measure theory states that if a set S of oracles has positive measure and $c < 1$, then there exists an interval Γ such that

$$\text{Prob}[A \in S \mid A \in \Gamma] > c. \tag{67}$$

Applying this theorem to the set

$$S = \{A \mid L_{N^A} \text{ is infinite and } L_{N^A} \subseteq L_0^A\}$$

we find an interval Γ such that (67) is true. Choose $w_1, \ldots, w_j \in \mathbb{B}^*, b_1, \ldots, b_j \in \mathbb{B}$ such that $\Gamma = \{A \mid A(w_1) = b_1, \ldots, A(w_j) = b_j\}$. Given an oracle A, let us define a new oracle A' as follows:

$$A'(u) = \begin{cases} b_l, & \text{if } u = w_l, \text{ where } l \in \{1, \ldots, j\}; \\ A(u), & \text{if } u \notin \{w_1, \ldots, w_j\}. \end{cases}$$

It is easy to see that

$$\mathrm{Prob}[A' \in S] = \mathrm{Prob}[A \in S \mid A \in \Gamma] > c.$$

Let $k = \max_{m \le j} |w_m|$. We can easily construct a nondeterministic machine N_1 such that

$$N_1^A(x) = \begin{cases} 0, & \text{if } |x| + \log_2 |x| \le k; \\ N^{A'}(x), & \text{otherwise.} \end{cases}$$

We claim that if $A' \in S$, then $L_{N_1^A}$ is infinite and $L_{N_1^A} \subseteq L_0^A$. Indeed, suppose that A' is in S, in other words, that $L_{N^{A'}}$ is infinite and $L_{N^{A'}} \subseteq L_0^{A'}$. Then, obviously, $L_{N_1^A}$ is infinite. To prove the inclusion $L_{N_1^A} \subseteq L_0^A$, suppose that $N_1^A(x) = 1$. Then $|x| + \log_2 |x| > k$ and $N^{A'}(x) = 1$. It follows that $x \in L_0^{A'}$. Consequently, $x \in L_0^A$ (because $|x| + \log_2 |x| > k$) and our claim is proved. We have, therefore,

$$\mathrm{Prob}\big[L_{N_1^A} \text{ is infinite and } L_{N_1^A} \subseteq L_0^A\big] \ge \mathrm{Prob}[A' \in S] > c.$$

This contradiction proves the implication $1') \Rightarrow 1)$. The implication $2') \Rightarrow 2)$ is proved in a similar way.

We now have to prove $1')$ and $2')$. Let N be a nondeterministic machine. We claim that the probability of either of the following two events

$$\text{“}L_{N^A} \text{ is infinite and } L_{N^A} \subseteq L_0^A\text{”} \tag{68}$$

$$\text{“}\mathbb{B}^* \setminus L_{N^A} \text{ is infinite and } \mathbb{B}^* \setminus L_{N^A} \subseteq L_1^A\text{”} \tag{69}$$

does not exceed 0.9. Obviously, if L_{N^A} is infinite and $L_{N^A} \subseteq L_0^A$, then there are infinitely many i such that $N^A(1^{t_i}) = 1$. Therefore, to prove that $\mathrm{Prob}[(68)] \le 0.9$ it suffices to prove the implication

$$\mathrm{Prob}\big[\exists^\infty i \; N^A(1^{t_i}) = 1\big] > 0.9 \Rightarrow \tag{70}$$

$$\mathrm{Prob}\big[\exists i \; (N^A(1^{t_i}) = 1 \;\&\; 1^{t_i} \in L_1^A)\big] \ge 0.1 \tag{71}$$

where $\exists^\infty i$ means “there exist infinitely many i such that”. Similarly, to prove that $\mathrm{Prob}[(69)] \le 0.9$ it suffices to prove the implication

$$\mathrm{Prob}\big[\exists^\infty i \; N^A(1^{t_i}) = 0\big] > 0.9 \Rightarrow \tag{72}$$

$$\mathrm{Prob}\big[\exists i \; (N^A(1^{t_i}) = 0 \;\&\; 1^{t_i} \in L_0^A)\big] \ge 0.1. \tag{73}$$

We shall prove these two implications in a unified way. Namely, we prove that for any $\delta \in \{0, 1\}$, the following implication holds

$$\mathrm{Prob}\big[\exists^\infty i \; N^A(1^{t_i}) = \delta\big] > 0.9 \Rightarrow \tag{74}$$

$$\mathrm{Prob}\big[\exists i \; (N^A(1^{t_i}) = \delta \;\&\; 1^{t_i} \in L_\delta^A)\big] \ge 0.1. \tag{75}$$

Indeed, the assumption (74) implies that

(76)
$$\sum_{i=k}^{\infty} \mathrm{Prob}\big[N^A(1^{t_i}) = \delta \ \&\ N^A(1^{t_{i-1}}) \neq \delta, N^A(1^{t_{i-2}}) \neq \delta, \ldots, N^A(1^{t_k}) \neq \delta\big] > 0.9$$

for all $k \in \mathbb{N}$.

Let $P_{ik\delta}(A)$ denote the event

$$N^A(1^{t_{i-1}}) \neq \delta, N^A(1^{t_{i-2}}) \neq \delta, \ldots, N^A(1^{t_k}) \neq \delta.$$

LEMMA 26. *For sufficiently large k and for all $i \geq k$, we have*

$$\mathrm{Prob}\big[1^{t_i} \in L^A_\delta \ \&\ N^A(1^{t_i}) = \delta \ \&\ P_{ik\delta}(A)\big] > (1/3)\,\mathrm{Prob}\big[N^A(1^{t_i}) = \delta \ \&\ P_{ik\delta}(A)\big] - \epsilon_i, \tag{77}$$

where ϵ_i is a sequence such that the series $\sum_{i=1}^{\infty} \epsilon_i$ converges.

We shall prove the lemma later. Let us now continue the proof of the theorem. Summing (77) over $i \geq k$, we get

$$\sum_{i=k}^{\infty} \mathrm{Prob}\big[1^{t_i} \in L^A_\delta \ \&\ N^A(1^{t_i}) = \delta \ \&\ P_{ik\delta}(A))\big] > (1/3)\sum_{i=k}^{\infty} \mathrm{Prob}\big[N^A(1^{t_i}) = \delta \ \&\ P_{ik\delta}(A)\big] - \sum_{i=k}^{\infty} \epsilon_i. \tag{78}$$

Combining (76) and (78), we get

$$\sum_{i=k}^{\infty} \mathrm{Prob}\big[1^{t_i} \in L^A_\delta \ \&\ \mathbb{N}^A(1^{t_i}) = \delta \ \&\ P_{ik\delta}(A)\big] > 0.3 - \sum_{i=k}^{\infty} \epsilon_i. \tag{79}$$

Two events whose probabilities are summed on the left hand side of (79) are pairwise disjoint and are included in the event

$$\exists i \ (N^A(1^{t_i}) = \delta \ \&\ 1^{t_i} \in L^A_\delta). \tag{80}$$

Therefore, the probability of (80) is greater then $0.3 - \sum_{i=k}^{\infty} \epsilon_i$. Taking k so large that $\sum_{i=k}^{\infty} \epsilon_i < 0.2$, we get (75).

Thus the proof of the theorem is complete. However, it remains to prove Lemma 26. □

PROOF OF THE LEMMA 26. Fix a large number k (to be specified later). Let $i \geq k$. Let D_i denote the set of all binary words of length less than $t_i + \log_2 t_i$ and F_i the set of all functions from D_i into $\mathbb{B}$. For every j, the event $N^A(1^{t_j}) \neq \delta$ depends only on the values of A on words of length bounded by a polynomial in t_j (because the length of the questions put by N to the oracle on input 1^{t_j} is bounded by a polynomial in t_j). As $t_i = 2^{t_{i-1}}$, it follows that we can choose k so that the event $P_{ik\delta}(A)$ depends only on $A|D_i$ for all $i \geq k$. This is the choice of k we referred to. ($f|S$ denotes the restriction of function f to the set S.)

We claim that, for all $f \in F_i$,

$$\mathrm{Prob}\big[1^{t_i} \in L^A_\delta \ \&\ N^A(1^{t_i}) = \delta \mid A|D_i = f\big] > (1/3)\,\mathrm{Prob}\big[N^A(1^{t_i}) = \delta \mid A|D_i = f\big] - \epsilon_i, \tag{81}$$

where $\{\epsilon_i\}$ is a sequence such that the series $\sum_{i=1}^{\infty} \epsilon_i$ converges. Let us first prove that this implies (77). Assume that (81) is true for any $f \in F_i$. Then, multiplying (81) by $\mathrm{Prob}[A|D_i = f]$, we obtain

$$\begin{aligned}(82)\quad &\mathrm{Prob}\big[1^{t_i} \in L_\delta^A \ \&\ N^A(1^{t_i}) = \delta \ \&\ A|D_i = f\big] \\ &\qquad > (1/3)\,\mathrm{Prob}\big[N^A(1^{t_i}) = \delta \ \&\ A|D_i = f\big] - \epsilon_i\,\mathrm{Prob}\big[A|D_i = f\big].\end{aligned}$$

Summing (82) over all $f \in F_i$ such that $P_{ik\delta}(A)$ is true whenever $A|D_i = f$, we obtain

$$\begin{aligned}&\mathrm{Prob}\big[1^{t_i} \in L_\delta^A \ \&\ N^A(1^{t_i}) = \delta \ \&\ P_{ik\delta}(A)\big] \\ &\qquad > (1/3)\,\mathrm{Prob}\big[N^A(1^{t_i}) = \delta \ \&\ P_{ik\delta}(A)\big] - \epsilon_i\,\mathrm{Prob}\big[P_{ik\delta}(A)\big] \\ &\qquad\qquad \geq (1/3)\,\mathrm{Prob}\big[N^A(1^{t_i}) = \delta \ \&\ P_{ik\delta}(A)\big] - \epsilon_i.\end{aligned}$$

Thus we get (77).

It remains to prove that (81) is true for all $f \in F_i$. Fix an $f \in F_i$. Obviously, the number of queries put by M to oracle during possible computations on input 1^{t_i}, is bounded by a polynomial in t_i. Let $q(t_i)$ denote this polynomial.

To proceed further we have to handle the cases $\delta = 0$ and $\delta = 1$ separately.

Lemma 27.

$$\begin{aligned}(83)\quad &\mathrm{Prob}\big[1^{t_i} \in L_1^A \ \&\ N^A(1^{t_i}) = 1 \mid A|D_i = f\big] \\ &\qquad > \big(1 - (1 - 2^{-t_i})^{2^{t_i} - q(t_i)}\big)\,\mathrm{Prob}\big[N^A(1^{t_i}) = 1 \mid A|D_i = f\big].\end{aligned}$$

Proof. Let $r = q(t_i)$. An *r-neighborhood* is a set of oracles of the form

$$(84)\qquad \{A \mid A|D_i = f,\ A|B_1 = f_1, A|B_2 = f_2, \ldots, A|B_r = f_r,\ A|E = g\},$$

where $B_1, \ldots, B_r$ are i-blocks, $f_1, \ldots, f_r$ are functions from $B_1, \ldots, B_r$, respectively, into $\mathbb{B}$, E is a finite set disjoint from D_i and all the i-blocks, and g is a function from E into $\mathbb{B}$. Let

$$W = \{A \mid N^A(1^{t_i}) = 1, A|D_i = f\}.$$

Lemma 28. *The set W can be represented as a finite union of r-neighborhoods.*

Proof. We first prove that the set W can be represented as a finite union of intervals of the form $\{A \mid A(w_1) = b_1, \ldots, A(w_r) = b_r,\ A|D_i = f\}$. Let A be an oracle such that $A|D_i = f$. Assume that $N^A(1^{t_i}) = 1$. Fix an accepting computation of N^A on input 1^{t_i}. Let the oracle be queried about the value on the words $w_1, \ldots, w_r$. Then the set W contains the whole interval $\{B \mid B(w_1) = A(w_1), \ldots, B(w_r) = A(w_r),\ B|D_i = f\}$. Since the number of possible computations of N on input 1^{t_j} with all possible oracles is finite, we are done.

Thus, the set W can be represented as a finite union of intervals of the form $\{A \mid A(w_1) = b_1, \ldots, A(w_r) = b_r,\ A|D_i = f\}$. Obviously any interval of this form is a finite union of r-neighborhoods. □

Let $\Gamma_1, \ldots, \Gamma_n$ be r-neighborhoods such that $W = \Gamma_1 \cup \cdots \cup \Gamma_n$. We have to prove that

$$(85)\qquad \mathrm{Prob}\big[1^{t_i} \in L_1^A \mid A \in \Gamma_1 \cup \cdots \cup \Gamma_n\big] > 1 - (1 - 2^{-t_i})^{2^{t_i} - r}.$$

Call an r-neighborhood (84) *bad* if f_l is identically zero for some $l \le r$. Every oracle A from a bad r-neighborhood satisfies $1^{t_i} \in L_1^A$. Therefore, if we remove all bad r-neighborhoods from $\Gamma_1 \cup \cdots \cup \Gamma_n$, then the probability

$$\mathrm{Prob}\big[1^{t_i} \in L_1^A \mid A \in \Gamma_1 \cup \cdots \cup \Gamma_n\big]$$

will not increase. Hence, it suffices to prove (85) in the case when all $\Gamma_1, \ldots, \Gamma_n$ are not bad. Clearly, it suffices to prove that

$$\mathrm{Prob}[A \text{ is identically zero in some } i\text{-block} \mid A \in \Gamma_m \setminus (\Gamma_1 \cup \cdots \cup \Gamma_{m-1})] > 1 - (1 - 2^{-t_i})^{2^{t_i} - r} \tag{86}$$

for all $m \le n$ such that $\Gamma_m \setminus (\Gamma_1 \cup \cdots \cup \Gamma_{m-1})$ is not empty.

Fix $m \le n$. Represent Γ_m in the form (84). Let $C_1, \ldots, C_{2^{t_i}-r}$ denote all the i-blocks that do not belong to the set $\{B_1, \ldots, B_r\}$. For any $l \le 2^{t_i} - r$ let

$$\begin{aligned} p_l = \mathrm{Prob}[&A \text{ is identically zero in } C_l \mid \\ &A \in \Gamma_m \setminus (\Gamma_1 \cup \cdots \cup \Gamma_{m-1}) \text{ and } A \text{ is not identically zero} \\ &\text{in all the blocks } C_1, \ldots, C_{l-1}]. \end{aligned} \tag{87}$$

Of course, it may happen that some of p_l's are undefined because the set

$$\{A \in \Gamma_m \setminus (\Gamma_1 \cup \cdots \cup \Gamma_{m-1}) \mid A \text{ is not identically zero in all the blocks } C_1, \ldots, C_{l-1}\}$$

is empty. This means that every $A \in \Gamma_m \setminus (\Gamma_1 \cup \cdots \cup \Gamma_{m-1})$ is identically zero in some of the blocks $C_1, \ldots, C_{l-1}$. Then the probability on the left-hand side of (86) is equal to 1, and the inequality (86) is automatic.

In general, the probability on the left-hand side of (86) is equal to $1-(1-p_1)(1-p_2)\cdots(1-p_{2^{t_i}-r})$. Thus to prove Lemma 27 it suffices to prove the following.

LEMMA 29. *$p_l \ge 2^{-t_i}$ for any $l \le 2^{t_i} - r$ such that p_l is defined.*

PROOF. Fix an $l \le 2^{t_i} - r$ such that p_l is defined. For any oracle A, define a new oracle A' as follows:

$$A'(u) = \begin{cases} 0, & \text{if } u \in C_l; \\ A(u), & \text{otherwise.} \end{cases}$$

A set of oracles U is said to be *monotone* if $A \in U$ implies $A' \in U$. We claim that the set

$$U = \{A \mid A \in \Gamma_m \setminus (\Gamma_1 \cup \cdots \cup \Gamma_{m-1}) \text{ and } A \text{ is not identically zero in all the blocks } C_1, \ldots, C_{l-1}\}$$

is monotone. Recall that

$$\Gamma_m = \{A \mid A|D_i = f,\ A|B_1 = f_1, \ldots,\ A|B_r = f_r,\ A|E = g\}$$

and that C_l does not belong to the set $\{B_1, \ldots, B_r\}$. It follows that Γ_m is monotone. Since all the sets $\Gamma_1, \ldots, \Gamma_{m-1}$ are not bad, the complement of the set $\Gamma_1 \cup \cdots \cup \Gamma_{m-1}$ is also monotone. Obviously, the set

$$\{A \mid A \text{ is not identically zero in all the blocks } C_1, \ldots, C_{l-1}\}$$

is monotone. Therefore, U is monotone, being an intersection of monotone sets.

We see that Lemma 29 is easily implied by the following.

LEMMA 30. *Any monotone set V of positive measure satisfies*

$$\mathrm{Prob}[A \textit{ is identically zero in } C_l \mid A \in V] \geq 2^{-t_i}.$$

PROOF. Let V satisfy the assumptions of the Lemma 30. Let H denote the set of all the functions from C_l into $\mathbb{B}$. Let $\mathbf{0}$ stand for the identically zero function in H. Let h be an arbitrary function in H. We claim that there exists a measure-preserving one-to-one function from the set

$$\mathcal{B} = \{A \in V \mid A|C_l = h\}$$

into the set

$$\mathcal{D} = \{A \in V \mid A|C_l = \mathbf{0}\}.$$

Indeed, let π be a permutation of H such that $\pi(h) = \mathbf{0}$. Then the map $A \mapsto A''$, where

$$A''(u) = \begin{cases} \pi(A|C_l)(u) & \text{if } u \in C_l; \\ A(u) & \text{otherwise.} \end{cases}$$

takes $\mathcal{B}$ into $\mathcal{D}$ because $A'' = A'$ for $A \in \mathcal{B}$ and V is monotone. Thus

$$\mathrm{Prob}[A \in V \ \&\ A|C_l = h] \leq \mathrm{Prob}[A \in V \ \&\ A|C_l = \mathbf{0}]$$

for all h. Therefore,

$$\mathrm{Prob}[A \in V] = \sum_h \mathrm{Prob}[A \in V \ \&\ A|C_l = h] \leq 2^{|C_l|}\,\mathrm{Prob}[A \in V \ \&\ A|C_l = \mathbf{0}]. \quad \square$$

This proves Lemma 29 and Lemma 27. $\square$

It is easy to verify that $\lim_{i\to\infty}\left(1 - (1 - 2^{-t_i})^{2^{t_i} - q(t_i)}\right) = 1 - e^{-1} > 1/3$. Therefore Lemma 27 implies Lemma 26 in the case $\delta = 1$ (when we set $\epsilon_i = 0$).

It remains to consider the case $\delta = 0$. Let $a_i = (1 - 2^{-t_i})^{2^{t_i}} = \mathrm{Prob}[1^{t_i} \in L_0^A]$ and $\epsilon_i = (1 - 2^{-t_i})^{2^{t_i} - q(t_i)} - a_i$. Obviously,

$$\epsilon_i = O(q(t_i)2^{-t_i}).$$

Thus the series $\sum_{i=1}^{\infty} \epsilon_i$ converges. The case $\delta = 0$ of Lemma 26 is an immediate corollary of the following.

LEMMA 31.

$$\mathrm{Prob}\left[1^{t_i} \in L_0^A \ \&\ N^A(1^{t_i}) = 0 \mid A|D_i = f\right] > a_i\,\mathrm{Prob}\left[N^A(1^{t_i}) = 0 \mid A|D_i = f\right] - \epsilon_i. \tag{88}$$

PROOF. By Lemma 27,

$$\mathrm{Prob}\left[1^{t_i} \in L_1^A,\ N^A(1^{t_i}) = 1 \mid A|D_i = f\right] > (1 - a_i - \epsilon_i)\,\mathrm{Prob}\left[N^A(1^{t_i}) = 1 \mid A|D_i = f\right]. \tag{89}$$

We omit the condition $A|D_i = f$ in the following computations to make them readable. We have

$$\begin{aligned}
&\mathrm{Prob}[1^{t_i} \in L_1^A,\ N^A(1^{t_i}) = 0] \\
&\quad = \mathrm{Prob}[1^{t_i} \in L_1^A] - \mathrm{Prob}[1^{t_i} \in L_1^A,\ N^A(1^{t_i}) = 1] \\
&\quad \le 1 - a_i - (1 - a_i - \epsilon_i)\,\mathrm{Prob}[N^A(1^{t_i}) = 1] \\
&\quad = (1 - a_i)\,\mathrm{Prob}[N^A(1^{t_i}) = 0] + \epsilon_i\,\mathrm{Prob}[N^A(1^{t_i}) = 1] \\
&\quad \le (1 - a_i)\,\mathrm{Prob}[N^A(1^{t_i}) = 0] + \epsilon_i. \\
&\mathrm{Prob}[1^{t_i} \in L_0^A,\ N^A(1^{t_i}) = 0] \\
&\quad = \mathrm{Prob}[N^A(1^{t_i}) = 0] - \mathrm{Prob}[1^{t_i} \in L_1^A,\ N^A(1^{t_i}) = 0] \\
&\quad \ge \mathrm{Prob}[N^A(1^{t_i}) = 0] - (1 - a_i)\,\mathrm{Prob}[N^A(1^{t_i}) = 0] - \epsilon_i \\
&\quad = a_i\,\mathrm{Prob}[N^A(1^{t_i}) = 0] - \epsilon_i.
\end{aligned}$$

The Lemma 31 is proved. □

COROLLARY 50. [**52**] *For a random A, there is an infinite* NP^A*-set with no infinite* co-NP^A*-subsets.*

COROLLARY 51 ([**54**]). *For random A there exists an infinite* co-NP^A*-set with no infinite* NP^A*-subsets.*

THEOREM 52 ([**52**]). NP^A*-languages are not* P^A*-separable for a random A.*

PROOF. We use the sequence $\{t_i\}$ defined in the proof of the previous theorem. An i-block is a set of the form

$$B_w = \{wv \mid v \in \mathbb{B}^*, |v| = \log_2 t_i\},$$

where w is a binary word of length $2t_i$. Each B_w consists of t_i binary words of length $2t_i + \log_2 t_i$. Arrange B_w's according to the lexicographical order on w's.

Fix a sequence $\{s_i\}$ of natural numbers (to be specified later). Call the first s_i i-blocks $i,0$-*blocks* and the following s_i blocks $i,1$-*blocks* (s_i will be chosen to satisfy the inequality $2s_i \le 2^{2t_i}$). For an oracle A, we define the following NP^A-languages

$$L_0^A = \{1^{t_i} \mid i \in \mathbb{N} \text{ and } A \text{ is identically zero in some } i,0\text{-block}\},$$

$$L_1^A = \{1^{t_i} \mid i \in \mathbb{N} \text{ and } A \text{ is identically zero in some } i,1\text{-block}\}.$$

We shall now specify the sequence $\{s_i\}$. We want to do this so that the probability of the event "$1^{t_i} \in L_0^A$" is about $1/i$. Let us find the probability of this event. For any i-block B, the probability of the event "A is identically zero in B" is equal to 2^{-t_i}. Therefore, we have

$$\mathrm{Prob}\big[1^{t_i} \in L_0^A\big] = \mathrm{Prob}\big[1^{t_i} \in L_1^A\big] = 1 - (1 - 2^{-t_i})^{s_i}.$$

Let $s_i = [2^{t_i}/i]$. Obviously, we have $1 - (1 - 2^{-t_i})^{s_i} = 1 - e^{-2^{-t_i} s_i (1 + o(1))}$ as $i \to \infty$. It follows that $2^{-t_i} s_i = 2^{-t_i}[2^{t_i}/i] = \frac{1}{i}(1 + o(1))$. Therefore $1 - (1 - 2^{-t_i})^{s_i} = 1 - e^{-\frac{1}{i}(1+o(1))} = 1 - (1 - 1/i + o(1/i)) = 1/i + o(1/i)$.

We claim that the set $L_0^A \cap L_1^A$ is finite with probability 1. Indeed, the events $1^{t_i} \in L_0^A$ and $1^{t_i} \in L_1^A$ are mutually independent for all $i \in \mathbb{N}$. Therefore

$$\mathrm{Prob}\big[1^{t_i} \in L_0^A \cap L_1^A\big] = (1/i + o(1/i))^2 = 1/i^2 + o(1/i^2).$$

for all $i \in \mathbb{N}$. The series $\sum(1/i^2 + o(1/i^2))$ converges. By the Borel–Cantelli lemma, there are only finitely many i such that $1^{t_i} \in L_0^A \cap L_1^A$ with probability 1.

Let $C^A = L_1^A \setminus L_0^A$. Then $C^A \in \mathrm{NP}^A$ with probability 1 (as C^A is different from the language $L_1^A \in \mathrm{NP}^A$ only on a finite number of words for almost all A). In addition, C^A is disjoint from L_0^A for all A. Therefore, it suffices to prove that C^A and L_0^A are not P^A-separable for almost all A. To prove this it is sufficient to prove that for every deterministic polynomial-time oracle machine M the probability of the event

$$\exists x \in \mathbb{B}^* \ \left(M^A(x) = 1 \ \&\ x \in L_0^A \text{ or } M^A(x) = 0 \ \&\ x \in C^A\right)$$

is equal to 1. Fix a deterministic polynomial-time oracle machine M. It is sufficient to prove that there are infinitely many i such that

$$M^A(1^{t_i}) = 1 \ \&\ 1^{t_i} \in L_0^A \text{ or } M^A(1^{t_i}) = 0 \ \&\ 1^{t_i} \in L_1^A$$

for almost all A. Let $P_i(A)$ stand for the displayed event. Note that the events $P_i(A)$ can be dependent for $i \in \mathbb{N}$.

We claim that the series

$$\sum_{i=k+1}^{\infty} \mathrm{Prob}[P_i(A) \mid \neg P_{i-1}(A) \ \wedge\ \neg P_{i-2}(A) \ \&\ \cdots \wedge\ \ \neg P_{k+1}(A) \wedge \neg P_k(A)] \tag{90}$$

diverges for all $k \in \mathbb{N}$. Assume this claim is true. Then for almost all A, there are infinitely many i such that $P_i(A)$. Indeed,

$$\begin{aligned}
\mathrm{Prob}[\exists i \geq k \ P_i(A)] &= 1 - \mathrm{Prob}[\forall i \geq k \ \neg P_i(A)] \\
&= 1 - \mathrm{Prob}[\neg P_k(A)] \prod_{i=k+1}^{\infty} \mathrm{Prob}[\neg P_i(A) \mid \neg P_{i-1}(A) \& \cdots \& \neg P_k(A)] \\
&= 1 - \mathrm{Prob}[\neg P_k(A)] \prod_{i=k+1}^{\infty} \left(1 - \mathrm{Prob}[P_i(A) \mid \neg P_{i-1}(A) \& \cdots \& \neg P_k(A)]\right)
\end{aligned}$$

for all $k \in \mathbb{N}$. As (90) diverges, we see that the last infinite product is equal to 0. Therefore $\mathrm{Prob}[\exists i \geq k \ P_i(A)] = 1$ for all k. Consequently, $\mathrm{Prob}[\forall k \ \exists i \geq k \ P_i(A)] = 1$ (intersection of a countable family of sets of measure 1 has measure 1).

To prove that the series (90) diverges, we shall prove that for all sufficiently large i,

$$\mathrm{Prob}[P_i(A) \mid \neg P_{i-1}(A) \& \cdots \& \neg P_k(A)] > 1/2i.$$

Let D_i denote the set of all binary words of length less than $2t_i + \log_2 t_i$ and F_i the set of all functions from D_i into $\mathbb{B}$.

partition of disjoint For every j, the event $P_j(A)$ depends on the values of A on the words of length bounded by a polynomial in t_j. As $t_i = 2^{t_{i-1}}$, it follows that for sufficiently large i, the event $\neg P_{i-1}(A) \& \cdots \& \neg P_k(A)$ depends only on $A|D_i$. It suffices to prove, therefore, that for sufficiently large i and for all $f \in F_i$, the conditional probability $\mathrm{Prob}[P_i(A) \mid A|D_i = f]$ is greater than $1/2i$. Fix an $i \in \mathbb{N}$ and an $f \in F_i$. In the sequel, we shall consider only oracles A such that $A|D_i = f$.

Run M with oracle A on input 1^{t_i}. Assume that the number of queries to oracle made in this computation is l and the k-th query is "$A(u_k)$=?". Delete from the sequence $u_1, \ldots, u_l$ all words of length less than $2t_i + \log_2 t_i$. Let $w_1, \ldots, w_j$ denote the resulting sequence. We call the sequence of pairs $\langle w_1, A(w_1)\rangle, \ldots, \langle w_j, A(w_j)\rangle$ the *computational protocol on* A, written $C(A)$. Let Π be the set of all computational protocols, that is,

$$\Pi = \{C(A) \mid A|D_i = f\}. \tag{91}$$

Assume $Z = (\langle w_1, b_1\rangle, \ldots, \langle w_j, b_j\rangle)$ is a protocol from Π, where $w_1, \ldots, w_j \in \mathbb{B}^*$, $b_1, \ldots, b_j \in \mathbb{B}$. We call an oracle A *consistent with* Z if $A(w_1) = b_1, \ldots, A(w_j) = b_j$. Obviously, Z is a computational protocol on A if and only if A is consistent with Z. Consequently, the family of sets

$$\big\{\, \{A \mid A|D_i = f \text{ and } A \text{ is consistent with } Z \;\big|\; Z \in \Pi\big\}$$

is a partition of the set $\{A \mid A|D_i = f\}$. It is sufficient to prove, therefore, that

$$\text{Prob}[P_i(A) \mid A \text{ is consistent with } Z,\ A|D_i = f] > \frac{1}{2i} \tag{92}$$

for all $Z \in \Pi$.

Fix a $Z \in \Pi$. Obviously, if both A' and A'' are consistent with Z then $M^{A'}(1^{t_i}) = M^{A''}(1^{t_i})$. Without loss of generality we may assume that $M^A(1^{t_i}) = 1$ whenever A is consistent with Z. Then we have

$$\begin{aligned}\text{Prob}[P_i(A) \mid A \text{ is consistent with } Z,\ A|D_i = f]&\\ = \text{Prob}\big[1^{t_i} \in L_0^A \mid A \text{ is consistent with } Z,\ A|D_i = f\big]&\\ = \text{Prob}\big[1^{t_i} \in L_0^A \mid A \text{ is consistent with } Z\big].&\end{aligned}$$

Assume that Z consists of j pairs. Then we have

$$\text{Prob}\big[1^{t_i} \in L_0^A \mid A \text{ is consistent with } Z\big] \geq 1 - (1 - 2^{-t_i})^{s_i - j}.$$

(Recall that $1^{t_i} \in L_0^A$ means that A is identically zero in some $i,0$-block and the number of $i,0$-blocks is s_i.) As $j \leq \text{poly}(t_i)$ and $s_i = [2^{t_i}/i] \geq [2^{t_i}/t_i]$, it follows that

$$1 - (1 - 2^{-t_i})^{s_i - j} = \frac{1}{i}(1 + o(1)),$$

as $i \to \infty$. Consequently, for sufficiently large i, we have

$$\text{Prob}[P_i(A) \mid A \text{ is consistent with } Z,\ A|D_i = f] > \frac{1}{2i}. \quad \square$$

THEOREM 53 ([**52**]). co-NP^A*-languages are not* P^A*-separable for a random* A.

PROOF. We shall use the notions and notation from the previous proof. We make the following changes. We now set $s_i = [c2^{t_i} \log i]$, where c is a rational constant (to be specified later). Let

$$L_0^A = \{1^{t_i} \mid\ i \in \mathbb{N} \text{ and } A \text{ is identically zero in no } i,0\text{-block}\},$$
$$L_1^A = \{1^{t_i} \mid\ i \in \mathbb{N} \text{ and } A \text{ is identically zero in no } i,1\text{-block}\}.$$

Obviously, s_i can be computed in time poly(t_i). It follows that $L_0^A \in$ co-NP^A and $L_1^A \in$ co-NP^A. It is easy to see that

$$\begin{aligned}\text{Prob}\big[1^{t_i} \in L_0^A\big] = \text{Prob}\big[1^{t_i} \in L_1^A\big] &= (1 - 2^{-t_i})^{s_i}\\ &= e^{-2^{-t_i} s_i (1 + o(1))} = e^{-c \log i (1 + o(1))} = i^{-c \log e (1 + o(1))}.\end{aligned}$$

Choose c so that $6/8 < c \log e < 7/8$. Then we have

$$\text{Prob}\big[1^{t_i} \in L_0^A\big] = \text{Prob}\big[1^{t_i} \in L_1^A\big] < i^{-5/8}$$

for all sufficiently large i. Therefore, we obtain $\text{Prob}\big[1^{t_i} \in L_0^A \cap L_1^A\big] < i^{-5/4}$ for all sufficiently large i. As the series $\sum i^{-5/4}$ converges, we see that the set $L_0^A \cap L_1^A$ is finite for almost all A.

Let $C^A = L_1^A \setminus L_0^A$. We shall prove that C^A and L_0^A are not P^A-separable for almost all A. To this end it is sufficient to prove that for every deterministic polynomial-time oracle machine M, the probability of the event

$$\exists x \in \mathbb{B}^* \ \left(M^A(x) = 1 \ \&\ x \in L_0^A \quad \text{or} \quad M^A(x) = 0 \ \&\ x \in C^A\right)$$

is equal to 1. Fix a deterministic polynomial-time oracle machine M. It suffices to prove that for almost all A, there are infinitely many i such that

$$M^A(1^{t_i}) = 1 \ \&\ 1^{t_i} \in L_0^A \quad \text{or} \quad M^A(1^{t_i}) = 0 \ \&\ 1^{t_i} \in L_1^A.$$

Let $P_i(A)$ stand for the displayed event. It suffices to prove that the series

$$\sum_{i=k+1}^{\infty} \mathrm{Prob}[P_i(A) \mid \neg P_{i-1}(A) \ \&\ \neg P_{i-2}(A) \ \&\cdots\ \&\ \neg P_{k+1}(A) \wedge \neg P_k(A)] \tag{93}$$

diverges for all $k \in \mathbb{N}$. To prove that the series (93) diverges we shall prove that

$$\mathrm{Prob}[P_i(A) \mid \neg P_{i-1}(A) \& \cdots \& \neg P_k(A)] > i^{-7/8} - 2^{-t_i}\mathrm{poly}(t_i).$$

Let D_i denote the set of all binary words of length less than $2t_i + \log_2 t_i$ and F_i the set of all functions from D_i into $\mathbb{B}$.

For all sufficiently large i, the event $\neg P_{i-1}(A) \& \cdots \& \neg P_k(A)$ depends only on $A|D_i$. It suffices to show, therefore, that the conditional probability $\mathrm{Prob}[P_i(A) \mid A|D_i = f]$ is greater than $i^{-1} - 2^{-t_i}\mathrm{poly}(t_i)$ for all $f \in F_i$. Fix an $i \in \mathbb{N}$ and an $f \in F_i$. In the sequel, we shall consider only oracles A such that $A|D_i = f$.

We have to prove that

$$\mathrm{Prob}[P_i(A) \mid A|D_i = f] > i^{-1} - 2^{-t_i}\mathrm{poly}(t_i).$$

More precisely, we shall prove the inequality

$$\mathrm{Prob}[\neg P_i(A) \mid A|D_i = f] < 1 - i^{-1} + 2^{-t_i}\mathrm{poly}(t_i).$$

The event $\neg P_i(A)$ means that A is identically zero in an $i, (1 - M^A(1^{t_i}))$-block. Let $Z = (\langle w_1, b_1\rangle, \ldots, \langle w_j, b_j\rangle)$ be a computational protocol on A. We call an i-block B *A-free* [*A-occupied*] if B is an $i, (1 - M^A(1^{t_i}))$-block and B is disjoint from the set $\{w_1, \ldots, w_j\}$ [B intersects with the set $\{w_1, \ldots, w_j\}$, respectively]. Obviously, $P_i(A)$ holds if and only if A is identically zero either in an A-free block or in an A-occupied block. Let $Q'(A)$ denote the event "A is identically zero in an A-free block" and $Q''(A)$ the event "A is identically zero in an A-occupied block".

We shall prove that

$$\mathrm{Prob}\big[Q'(A) \ \big|\ A|D_i = f\big] < 1 - i^{-1}, \tag{94}$$

$$\mathrm{Prob}\big[Q''(A) \ \big|\ A|D_i = f\big] \le 2^{-t_i}\mathrm{poly}(t_i), \tag{95}$$

for all sufficiently large i. We first prove the inequality (94). It suffices to prove that

$$\mathrm{Prob}[Q'(A) \mid A \text{ is consistent with } Z,\ A|D_i = f] < 1 - i^{-1}$$

for all sufficiently large i and all $Z \in \Pi$, We claim that

$$\mathrm{Prob}[Q'(A) \mid A \text{ is consistent with } Z,\ A|D_i = f] \le 1 - (1 - 2^{-t_i})^{s_i}$$

for all $Z \in \Pi$. Indeed, fix a protocol $Z = (\langle w_1, b_1\rangle, \ldots, \langle w_j, b_j\rangle)$ from Π. (Recall that the set Π is defined by Equation (91).) Without loss of generality we may assume that $M^A(1^{t_i}) = 1$ for A consistent with Z. Then $Q'(A)$ means that A is identically zero in an $i, 0$-block disjoint from the set $\{w_1, \ldots, w_j\}$. Let k stand

for the number of $i,0$-blocks disjoint from the set $\{w_1,\dots,w_j\}$. Then we have $\mathrm{Prob}[Q'(A) \mid A \text{ is consistent with } Z] = 1-(1-2^{-t_i})^k$. As $k \le s_i$, the claim is proved. The definition of s_i implies that $1-(1-2^{-t_i})^{s_i} < 1-i^{-1}$ for all sufficiently large i. The inequality (94) is proved.

Let us prove the inequality (95). (Recall that $Q''(A)$ means that A is identically zero in some A-occupied block.) If a word u belongs to an i-block then let $B(u)$ denote the i-block u belongs to. Let $R(A)$ stand for the event "there are a protocol $Z = (\langle w_1,b_1\rangle,\dots,\langle w_j,b_j\rangle)$ from Π and an $m \le j$ such that the following assertions hold:

(a) w_m belongs to an i-block and A is identically zero in $B(w_m)$;
(b) $\forall i < m\ w_i \notin B(w_m)$;
(c) $\forall i < m\ A(w_i) = b_i$".

The assertion $Q''(A)$ implies $R(A)$. Indeed, assume that $Q''(A)$ holds. Let Z be the computational protocol on A and m the least m' such that A is identically zero in $B(w_{m'})$. All the three assertions (a), (b) and (c) hold for these Z,m. Let us prove that $\mathrm{Prob}[R(A) \mid A|D_i = f] \le 2^{-t_i}\mathrm{poly}(t_i)$. Fix a protocol $Z = (\langle w_1,b_1\rangle,\dots,\langle w_j,b_j\rangle)$ and an $m \le j$. We shall prove that

$$\mathrm{Prob}[\text{(a) \& (b) \& (c)}] \le 2^{-m+1}2^{-t_i}. \tag{96}$$

Note that the truth value of (b) depends only on m and Z. If (b) is false then we are done, because the probability on the left hand side of (96) is equal to zero. If w_m belongs to no i-block, then we are also done. Assume that (b) is true and that w_m belongs to an i-block. Then (a) and (c) are independent (we assume that $w_1,\dots,w_j$ are distinct). We have, therefore,

$$\mathrm{Prob}[\text{(a) \& (b) \& (c)}] = \mathrm{Prob}[\text{(a)}]\,\mathrm{Prob}[\text{(c)}] = 2^{-t_i}2^{-m+1}.$$

Fix an arbitrary m. Let us prove that the probability of the event "there is a protocol $Z = (\langle w_1,b_1\rangle,\dots,\langle w_j,b_j\rangle)$ such that $m \le j$ and (a)&(b)&(c)" is at most 2^{-t_i}. Let $R_m(A)$ denote this event. If $Z = (\langle w_1,b_1\rangle,\dots,\langle w_j,b_j\rangle)$ is a protocol such that $j \ge m$, then we call the sequence $Z_m = (\langle w_1,b_1\rangle,\dots,\langle w_{m-1},b_{m-1}\rangle,w_m)$ the *m-prefix* of Z. The set $\mathcal{Z}_m = \{Z_m \mid Z \in \Pi\}$ has at most 2^{m-1} elements, since Z_m is defined completely by the tuple $\langle b_1,\dots,b_{m-1}\rangle$. The truth values of the assertions (a), (b), and (c) depend only on A and Z_m. We have, therefore,

$$\begin{aligned}\mathrm{Prob}[R_m(A) \mid A|D_i = f] &\le \sum_{Y\in\mathcal{Z}_m} \mathrm{Prob}[\text{(a), (b), and (c) hold for } Y]\,\mathrm{Prob}[C(A)_m = Y \mid A|D_i = f]\\ &\le 2^{m-1}(2^{-t_i}2^{-m+1}) = 2^{-t_i}.\end{aligned}$$

Let k be the maximal length of a protocol from Π. Obviously, $k \le \mathrm{poly}(t_i)$. We have, therefore,

$$\mathrm{Prob}[R(A) \mid A|D_i = f] \le \sum_{m=1}^{k} \mathrm{Prob}[R_m(A) \mid A|D_i = f] \le 2^{-t_i}k \le 2^{-t_i}\mathrm{poly}(t_i). \quad \square$$

References

[1] W. Aiello, S. Goldwasser, and J. Håstad. *On the power of interaction.* Proc. 27th Ann. IEEE Symp. Foundations of Computer Science, pp. 368–379, 1986.

[2] N. Ajtai. *Σ^1_1-formulae on finite structures.* Ann. Pure and Applied Logic, **24**(1983), 1–48.

[3] K. Ambos-Spies. *A note on complete problems for complexity classes.* Inf. Proc. Lett., **23**(1986), 227–230.
[4] L. Babai. *Trading group theory for randomness.* Proc. 17th Ann. ACM Symp. Theory of Computing, pp. 421–429, 1985.
[5] T. Baker, J. Gill, and R. Solovay. *Relativization of P=?NP question.* SIAM J. Computing, **4**(1975), no. 4, 431–442.
[6] R. Beigel. *Perceptrons, PP, and the polynomial time hierarchy.* Proc. 7th Ann. IEEE Conf. Structure in Complexity Theory, pp. 14–19, 1992.
[7] R. Beigel, N. Reingold, and D. Spielman. PP *is closed under intersection.* Proc. 23th Ann. ACM Symp. Theory of Computing, pp. 1–9, 1991.
[8] C. H. Bennet and J. Gill. *Relative to a random oracle $P \neq NP \neq coNP$ with probability 1.* SIAM J. Computing, **10**(1981), 96–113.
[9] M. Blum and R. Impagliazzo. *General oracle and oracle classes.* Proc. 28th Ann. IEEE Symp. Foundations of Computer Science, pp. 118–126, 1987.
[10] D. P. Bovet, P. Creszenzi, and R. Silvestri. *A uniform approach to define complexity classes.* Theor. Comp. Sci., **104**(1992), 263–283.
[11] J. Cai and L. Hemachandra. *On the power of parity polynomial time.* Math. Syst. The., **23**(1990), no. 2, 95–106.
[12] E. W. Cheney. *Approximation Theory.* McGraw-Hill, 1966.
[13] H. Chernoff. *A measure of assymptotic efficiency for tests of a hypothesis based on the sum of observations.* Ann. of Math. Statistics, **23**(1952), 493–509.
[14] H. Ehlich and K. Zeller. *Schwankung von Polynomen zwischen Gitterpunkten.* Math. Zeitschrift, **86**(1964), 41–44.
[15] S. Fenner, L. Fortnow, S. A. Kurtz, and L. Li. *An oracle builder's toolkit.* Proc. 8th Ann. IEEE Conf. Structure in Complexity Theory, pp. 120–131, 1993.
[16] L. Fortnow and N. Reingold. PP *is closed under truth table reductions.* Proc. 6th Ann. Conf. Structure in Complexity Theory, pp. 13–15, 1991.
[17] L. Fortnow and J. Rogers. *Separability and one-way functions.* Manuscript, 1994.
[18] L. Fortnow and M. Sipser. *Are there interactive protocols for coNP languages?* Inf. Proc. Lett., **28**(1988), 249–251.
[19] B. Fu. *Separating PH from PP by relativisation.* Preprint, 1990.
[20] N. Furst, J. Saxe, and M. Sipser. *Parity, circuits and the polynomial time hierarchy.* Math. Syst. The., **17**(1984), 13–27.
[21] S. Goldwasser, S. Micali, and C. Rackoff. *The knowledge complexity of interactive proof-systems.* SIAM J. Computing, **18**(1989), 186–208.
[22] S. Goldwasser and M. Sipser. *Private coins versus public coins in interactive proof systems.* Proc. 18th Ann. ACM Symp. Theory of Computing, pp. 59–68, 1986.
[23] J. Grollman and A. L. Selman. *Complexity measures for public-key cryptosystems.* SIAM J. Computing, **17**(1988), no. 2, 309–335.
[24] Y. Gurevich. *Algebras of feasible functions.* Proc. 24th Ann. IEEE Symp. Foundations of Computer Science, pp. 210–214, 1983.
[25] J. Håstad. *Almost optimal lower bounds for small depth circuits.* Proc. 18th Ann. ACM Symp. Theory of Computing, pp. 6–20, 1986.
[26] J. Hartmanis and L. Hemachandra. *Complexity classes without machines: On complete languages for UP.* Theor. Comp. Sci., **58**(1988), 129–142.
[27] J. Hartmanis and N. Immerman. *On complete problems for $NP \cap coNP$.* Proc. Intern. Colloq. Automata, Languages and Programming, 1985. Lect. Notes Comp. Sci., **194**, pp. 250–259, 1985.
[28] L. Hemaspaandra, S. Jain, and N. Vereshchagin. *Banishing robust turing completeness.* Intern. J. Found. Comp. Sci., **4**(1993), no. 3, 245–265.
[29] S. Homer, A. L. Selman. *Oracles for structural properties: The isomorphism problem and public-key cryptography.* J. Comp. and Syst. Sci., **44**(1992), no. 2, 287–301.
[30] R. Impagliazzo and M. Naor. *Decision trees and downward closures.* Proc. 3rd Ann. IEEE Conf. Structure in Complexity Theory, pp. 29–38, 1988.
[31] J. Köbler, U. Shöning, S. Toda, and J. Toran. *Turing machines with few accepting computations and low sets for PP.* Proc. 4th Ann. IEEE Conf. Structure in Complexity Theory, 1989.

[32] K.-I. Ko. *Relativized polynomial-time hierarchies having exactly k levels.* SIAM J. Computing, **18**(1989), no. 2, 392–408.
[33] S. Kurtz, S. Mahaney, and J. Royer. *Average Dependence and Random Oracle.* Proc. 7th Ann. IEEE Conf. Structure in Complexity Theory, pp. 306–317, 1992.
[34] C. Lautemann. *BPP and the polynomial hierarchy.* Inf. Proc. Lett., **17**(1983), no. 4, 215–217.
[35] G. G. Lorentz. *Approximation of Functions.* Hort, Rinehart and Winston, New York, 1966.
[36] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. *The polynomial time hierarchy has interactive proofs.* Proc. 31th Ann. IEEE Symp. Foundations of Computer Science, pp. 2–10, 1990.
[37] M. Minsky and S. Papert. *Perceptrons.* MIT Press, Cambridge, MA, 1988. (Expanded edition, first edition appeared in 1967.).
[38] A. A. Muchnik and N. K. Vereshchagin. *A general method to construct oracles realizing given relationships between complexity classes.* TR 500, Comp. Sci. Dept, Univ. of Rochester, 1994.
[39] ———. *A General Method to Construct Oracles Realizing Given Relationships between Complexity Classes.* Theor. Comp. Sci., **157**(1996), 227–258.
[40] N. Nisan. *Probabilistic versus deterministic decision trees and CREW PRAM complexity.* Proc. 21th Ann. ACM Symp. Theory of Computing, pp. 327–335, 1989.
[41] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
[42] G. Pólya and G. Szegö. *Problems and Theorems in Analysis.* Springer Verlag, 1972.
[43] C. Rackoff. *Relativized questions involving probabilistic algorithms.* Proc. 10th Ann. ACM Symp. Theory of Computing, pp. 338–342, 1978.
[44] M. Santha. *Relativized Arthur-Merlin versus Merlin-Arthur games.* Inform. and Computation, **80**(1989), 44–49.
[45] A. Shamir. *IP=PSPACE.* Proc. 31th Ann. IEEE Symp. Foundations of Computer Science, pp. 11–15, 1990.
[46] R. Silvestri. *Complexity classes and relativizations.* Ph. D. thesis. Dipartimento di Matematica, Università degli studi di Roma "La Sapienza", 1992.
[47] M. Sipser. *On relativizations and the existence of complete sets.* Proc. Intern. Colloq. Automata, Languages and Programming, 1982. Lect. Notes Comp. Sci., **140**, pp. 523–531, 1982.
[48] ———. *A complexity theoretic approach for randomness.* Proc. 15th Ann. ACM Symp. Theory of Computing, pp. 330–335, 1983.
[49] S. Toda. *On the computational power of PP and $\oplus P$.* Proc. 30th Ann. IEEE Symp. Foundations of Computer Science, pp. 514–519, 1989.
[50] N. K. Vereshchagin. *On the power of PP.* Proc. 7th Conf. Structure in Complexity Theory, pp. 138–143, 1992.
[51] ———. *Relativizable and Non-Relativizable theorems in Polynomial Theory of Algorithms.* Izv. RAN, Ser. Mat, **57**(1993),no. 2, 51–90; English transl. Russian Acad. Sci. Izv. Math., **42**(1994), no. 2, 261–298.
[52] ———. *Relationships between NP-sets, Co-NP-sets and P-sets relative to random oracles.* Izv. Vysshykh Uchebnykh Zavedenij., Ser. Mat., 1993, no. 3, 31–39; Preliminary version Proc. of 8th Ann. IEEE Conf. Structure in Complexity Theory, pp. 132–138, 1993.
[53] ———. *Lower bounds for perceptrons solving some separation problems and oracle separation of AM from PP.* Proc. 3rd Israel Symp. Theory of Computing and Systems, pp. 46–51, 1995.
[54] ———. *NP-sets are Co-NP-immune relative to a random oracle.* Proc. 3rd Israel Symp. Theory of Computing and Systems, pp. 40–45, 1995.
[55] A. Yao. *Separating the polynomial hierarchy by oracles.* Proc. 26th Ann. IEEE Symp. Foundations of Computer Science, pp. 1–10, 1985.

Dept. of Mathematical Logic and Theory of Algorithms, Moscow State University, Vorobjevy Gory, Moscow 119899, Russia
E-mail address: ver@mech.math.msu.su

Translated by THE AUTHOR