RELATIVIZABLE AND NONRELATIVIZABLE THEOREMS IN THE POLYNOMIAL THEORY OF ALGORITHMS

Nikolai K. Vereshchagin

Abstract. Starting with the paper of Baker, Gill and Solovay [BGS 75] 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 of a certain type or for machines with polylogarithmic restrictions on time. The following question arises: Are these methods of proving "relativized" results universal? In the first part of the present paper 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, as easy consequences of the known results on boolean decision trees, some new "relativized" results and new proofs of some known results. In the second part of the present paper we apply these general criteria to many particular cases. For example, for many of the complexity classes studied in the literature all relativizable inclusions between the classes are found.

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 [BGS 75], 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 don't know which of the two assertions $\mathrm{P} = \mathrm{NP}$ and $\mathrm{P} \neq \mathrm{NP}$ is true, neither of them is relativizable. After [BGS 75], many theorems of the following kind were proved (for pairs of relativizable complexity classes K_1, K_2): there exist 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 [BGS 75] as the second oracle because in fact $\mathrm{P}^B = \mathrm{PSPACE}^B$ is true for that oracle. In 1989 the first non-relativizable theorems in complexity theory appeared. The first of them was the theorem from [LFKN 89]: $\mathrm{PH} \subseteq \mathrm{IP}$. Earlier, in [FS 88], it was proved that $\exists A\ \text{Co-NP}^A \not\subseteq \mathrm{IP}^A$.

All known proofs of results having 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 the oracle step by step), which is the same in all proofs, and the specific "combinatorial" part,

1991 *Mathematics Subject Classification.* Primary 68Q15.

Translated from Izvestija Rossijskoj Akademii Nauk, Vol. 57 (1993), No. 2, pp. 51–90. The translation should appear in July 1994.

Received March 16, 1990 and December 18, 1991

Note added in proof. Results similar to those in Sections 3–6 were obtained in the paper [BCS 92].

Typeset by $\mathcal{AMS}$-TEX

in which it is proved that every step can be made. The first result of the present paper is the formalization of this statement. The proof of Theorem 1 in Section 3 is a general formulation of the diagonal part of such proofs. Corollary 1 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. The first paper of this kind known to the author is [S 82]. In that paper it is proved that there exists an oracle A for which the class $\mathrm{NP}^A \cap \text{Co-NP}^A$ has no Karp complete language (more precisely, no language complete under polynomial many-one reductions relative to A), and there exists 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 $\exists A\ K_1^A \nsubseteq K_2^A$ is true for proofs of nonexistence of complete languages in complexity classes. Theorem 2 in Section 4 provides the diagonal part of such proofs in general form.

Both Theorem 1 and Theorem 2 give the criteria. Theorem 1 is the criterion of whether

$$\forall A\ K_1^A \subseteq K_2^A, \tag{1.1}$$

while Theorem 2 is the criterion of whether

$$\forall A\ (K_2^A \text{ has a Karp complete problem for the class } K_1^A). \tag{1.2}$$

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. Denote by KLOGS the resulting "counterpart" of the class K. Then assertion (1.1) is equivalent to the absolute inclusion $K_1\mathrm{LOGS} \subseteq K_2\mathrm{LOGS}$, and assertion (1.2) is true iff the class K_2LOGS has a language complete for the class K_1LOGS. The analysis of proofs of relativizable assertions of the form (1.1) (for example, $\mathrm{BPP} \subseteq \Sigma_2 \cap \Pi_2$ from [S 83]) shows that the more natural formulations of such assertions have the form $K_1\mathrm{LOGS} \subseteq K_2\mathrm{LOGS}$.

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

$$\forall A\ (\text{ the class } K_2^A \text{ has Cook complete language for the class } K_1^A) \tag{1.3}$$

and

$$\forall A\ (\forall L_1 \in K_1^A\ \exists L_2 \in K_2^A :\ L_1 \text{ is Cook reducible to } L_2), \tag{1.4}$$

i.e. "K_1^A is Cook reducible to K_2^A".

These criteria are formulated in Sections 5 and 6.

The new approach to relativizable theorems makes the solving of problems of the forms (1.1)–(1.4) easier in both the psychological and technical sense. In Sections 7, 8 and 9 we ascertain, for several known classes K_1, K_2 between P and PSPACE to which the proposed criteria can be applied, which of the two assertions—(1.1) or the negation of (1.1)—is true or is unknown. We do the same thing also for assertions of the form (1.2), (1.3) and (1.4). Some new positive and negative results results of this type are proved (we call positive results of the form (1.1)–(1.4)). Some problems of this kind remain open.

Acknowledgements. The author is sincerely grateful to O. V. Verbitsky, An. A. Muchnik, A. A. Razborov, A. Kh. Shen and other participants of the Kolmogorov seminar in Moscow Lomonosov University and the Complexity seminar in Steklov institute for useful comments and to Fred Green for the help in translation into English.

2. Basic definitions and notation

We denote the set of all words over an alphabet A by A^*. By $\mathbf{B}$ we denote the set $\{0, 1\}$.

A separation problem is any function from the set $\mathbf{B}^*$ 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 $\{x \mid F(x) = 0\}$. Denote by $\mathrm{D}(F)$ the set $\{x \in \mathbf{B}^* \mid F(x) \neq *\}$.

We will identify every language $L \subseteq \mathbf{B}^*$ 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. The length of the word x is denoted by $|x|$.

Denote $\lceil \log_2 n \rceil$ by $\log n$ and let $\log(0)$ be 0. Functions of the form $p(\log n)$, where p is a polynomial, will be called polylogarithms. Expressions $\mathrm{poly}(n)$ and $\mathrm{polylog}(n)$ will denote a polynomial and a polylogarithm, respectively.

We shall study complexity classes defined by Turing machines whose running time is bounded by a polylogarithm of the length of the input. An ordinary Turing machine in polylogarithmic time can read only a prefix of the input word having polylogarithmic length. Therefore, we will use the model of Turing machines which is commonly used when time restrictions are so small. In this model, the input word is given as an oracle. More precisely, 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[1]. The machine may at any moment of a computation ask a question of the form '$x(i) =$?', i.e., it can write down on the input tape the number $i \leq |x|$ and then receive the ith symbol of x, denoted by $x(i)$, 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 doesn't get 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. By $M(x)$ we will denote the output of M on the input word x.

Our first goal is to give the definition of the polylogarithmic counterpart of a complexity class. As an example, we first define polylogarithmic counterparts of three well known classes, P, NP and R, and then give the general definition. The polylogarithmic counterpart of a complexity class is always a class of separation problems. If K denotes a complexity class accepted in the literature, then the polylogarithmic counterpart of this class is denoted by KLOGS, for example, PLOGS, NPLOGS and RLOGS.

Thus, let F be a separation problem. Then by definition $F \in$ PLOGS, if there exists a deterministic Turing machine M whose computation time is restricted by a polylogarithm of the size of the input such that $M(\alpha) = F(\alpha)$ for all $\alpha \in \mathrm{D}(F)$.

By a polylogarithmic nondeterministic machine we mean any nondeterministic Turing machine all of whose computations on input α have no more than

[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.

polylog($|\alpha|$) steps. By definition, $F \in$ NPLOGS if there exists a polylogarithmic nondeterministic machine M such that if $F(\alpha) = 1$, then M accepts α, and if $F(\alpha) = 0$, then M rejects α.

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

Let us turn to the definition of the notion of polylogarithmic counterpart of a complexity class. To this end we have to fix a general framework, according to which most complexity classes between P and PSPACE are defined.

To this end consider the definitions of two particular complexity classes (NP and BPP) in a convenient form.

2.1 $L \in \mathrm{NP} \iff$ there exists a polynomial time computable function $s : \mathbf{B}^* \to \mathbf{N}$ and a polynomial time predicate $P(x, i)$ such that $x \in L \Leftrightarrow \exists i \leq s(x) \, : \, P(x, i)$,

2.2 $L \in \mathrm{BPP} \iff$ there exists a polynomial time computable function $s : \mathbf{B}^* \to \mathbf{N}$ and a polynomial time predicate $P(x, i)$ such that if $x \in L$, then the ratio $\frac{\left|\{i \in \mathbf{N} | 1 \leq i \leq (x), P(x,i)\}\right|}{s(x)}$ is greater than 2/3 and if $x \notin L$, then this ratio is less than 1/3.[2]

Let us denote in both definitions by $f(x)$ 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 bounded by a polynomial of $|x|$ given its number. Now we come to the following definition.

Let f be a function from $\mathbf{B}^*$ into $\mathbf{B}^*$, and $t : \mathbf{N} \to \mathbf{N}$.

Definition 1. A function f is weakly computable in time t if

(1) the function $x \mapsto |f(x)|$ is computable in time $t(|x|)$,
(2) the partial binary predicate $P(x, i) = (i$thbit of the word $f(x))$ can be computed by a machine M which for all $x \in \mathbf{B}^*$ and all $i \leq |f(x)|$ works in time not exceeding $t(|x|)$.

Functions that are weakly computable in time poly(n), (polylog(n) and $2^{O(n)}$, respectively) are called *weakly polynomial* (*weakly polylogarithmic* and *weakly exponential*, respectively) . For example, the function $f(x) = 0^{2^{|x|}}$ is weakly polynomial (by 0^n we denote the word consisting of n zeros) and the function $f(x) = x$ is weakly polylogarithmic.

Both definitions 2.1 and 2.2 have the following form. For a fixed separation problem F we declare that a language L is in the class if there exists a weakly polynomial function f such that $L(x) = F(f(x))$ for all $x \in \mathbf{B}^*$. Let POLY(F) denote the class defined in this way by means of separation problem F. We say that a class K is *generated by a separation problem* F if $K =$ POLY(F). For example, the class NP is generated by the following separation problem F_{NP}:

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

[2] $|M|$ denotes the cardinality of the set M.

To generate 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 ones in the binary word x.

It is easy to verify that all the classes P, NP, R, BPP, UP, FewP, Σ_k, $\oplus$P, PP, PSPACE, MA, AM, IP (without private coin tossing) have the form POLY(F) for some F.

Let us define a partial ordering on the set $\{0, 1, *\}$ assuming that $* < 0$, $* < 1$. Define LOGS(F) as the class of all separation problems G such that for some weakly polylogarithmic function f the following is true: $\forall\alpha \in \mathbf{B}^*$ $G(\alpha) \leq F(f(\alpha))$, and define LOG$(F)$ to be the class of all the languages in LOGS(F). The class LOGS(F) is just called the *polylogarithmic counterpart* of the class POLY(F). More precisely, separation problem F defines the pair—the class POLY(F) and its polylogarithmic counterpart LOGS(F) (as we see later, the class LOGS(F) is not uniquely determined by the class POLY(F)).

Let us turn out to relativized classes. 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 machine works as usual two tape Turing machine with one exception. If oracle machine gets into a certain state, then the word u written on 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 oracle to provide its answer is assumed to be 1.

Let M be an oracle machine and let A be an oracle. Denote by $M^A(x)$ the output produced by M supplied with oracle A on input x, and by $t_{M^A}(x)$ the running time necessary to provide this output. Call an oracle machine M *polynomial* [or *exponential*] if there exists a polynomial $q(n)$ [a constant c] such that $t_{M^A}(x) \leq q(|x|)$ [$t_{M^A}(x) \leq 2^{c|x|+c}$] for all $x \in \mathbf{B}^*$ and all $A \subseteq \mathbf{B}^*$. 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).

Let A be an oracle. We want to relativize the definition of the class POLY(F). Let us define the notion of the weak computability relative to oracle A. Namely, in the definition of weak computability we allow machine M to call oracle A and in item (1) we allow the function $|f(x)|$ to be computable in time $t(|x|)$ by a machine with oracle A.

Definition 2. POLY$^A(F)$ is the class of all languages L such that $L(x) = F(f(x))$ for all $x \in \mathbf{B}^*$ for some function f being weakly polynomial relative to A.

3. A criterion of relativizable inclusion of one complexity class into another complexity class

The single theorem of this section claims that a polynomial complexity class K_1^A is included in a polynomial complexity class K_2^A for all oracles A if and only if the (absolute: no oracles) inclusion between their polylog-counterparts holds.

That theorem is valid for all the classes of the form $\mathrm{POLY}^A(F)$ provided that the separation problem F is nondegenerate in the following sense:

3.1 there exists a weakly polynomial function $f : \mathbf{N} \to \mathbf{B}^*$ such that $|f(n)| = n$ and $F(f(n)) \neq *$ for all $n \in \mathbf{N}$;

3.2 there are two words (denote them zero_F and one_F) such that $F(\mathit{zero}_F) = 0$, $F(\mathit{one}_F) = 1$.

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

Theorem 1. *If separation problem F satisfies the condition* 3.1 *and separation problem G satisfies the condition* 3.2*, then the following are equivalent:*

3.3 $\mathrm{LOGS}(F) \subseteq \mathrm{LOGS}(G)$*,*

3.4 $F \in \mathrm{LOGS}(G)$*, and*

3.5 $\mathrm{POLY}^A(F) \subseteq \mathrm{POLY}^A(G)$ *for all* $A \subseteq \mathbf{B}^*$*.*

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

3.6 $\mathrm{LOG}(F) \subseteq \mathrm{LOG}(G)$*.*

Proof. Obviously, 3.3 implies 3.4. Let us prove that 3.4 implies 3.3. Let F be in the class $\mathrm{LOGS}(G)$, and let g be a weakly polylogarithmic function such that $F(\alpha) \leq G(g(\alpha))$. Let us prove that $\mathrm{LOGS}(F) \subseteq \mathrm{LOGS}(G)$. Let H be in $\mathrm{LOGS}(F)$ and f be a weakly polylogarithmic function such that $H(\alpha) \leq F(f(\alpha))$. Then $H(\alpha) \leq G(g(f(\alpha)))$ for all $\alpha \in \mathbf{B}^*$. It is easy to see that $g(f(\alpha))$ is a weakly polylogarithmic function (the class of weakly polylogarithmic function is closed under superpositions), therefore, H belongs to $\mathrm{LOGS}(G)$.

Evidently, the assertion 3.3 implies the assertion 3.6, and if F is a language, then 3.6 implies 3.4.

Let us prove that 3.4 implies 3.5. Let f be a weakly polylogarithmic function such that $F(\alpha) \leq G(f(\alpha))$. Assume that A is a subset of $\mathbf{B}^*$ and L is an element of $\mathrm{POLY}^A(F)$, that is, there exists a function g being weakly polynomial relative to A such that $L(x) = F(g(x))$. Consequently, $L(x) = G(f(g(x)))$. It is easy to see that the function $f(g(x))$ is weakly polynomial relative to A (superposition of a weakly polylogarithmic function and of a function being weakly polynomial relative to A is weakly polynomial relative to A). Hence, L belongs to $\mathrm{POLY}^A(G)$.

Let us prove that if 3.4 is not true, then 3.5 is not true also. Assume that F is not in $\mathrm{LOGS}(G)$. This means that for any separation problem $H \in \mathrm{LOGS}(G)$ there exists an $\alpha \in \mathbf{B}^*$ such that $F(\alpha) \not\leq H(\alpha)$. Let us prove that, moreover, for any separation problem $H \in \mathrm{LOGS}(G)$ there exist infinitely many $\alpha \in \mathbf{B}^*$ such that $F(\alpha) \not\leq H(\alpha)$. Assume that it is not true, i.e., there exist a number n and a weakly polylogarithmic function f such that $F(\alpha) \leq G(f(\alpha))$ for all $\alpha \in \mathbf{B}^*$, $|\alpha| > n$. Then the function

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

is weakly polylogarithmic and $F(\alpha) \leq G(f_1(\alpha))$ for all $\alpha \in \mathbf{B}^*$.

Let us fix a function encoding pairs of words by words in the following way. Assume that x is in $\mathbf{B}^*$. Let us double all the bits of x and add the word “01” to the end of the resulting word. Denote the resulting word by $\bar{x}$ (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 given $\bar{x}y$ we can in polynomial time find x and y and for given word u we can decide in polynomial time whether u has the form $\bar{x}y$. For an oracle A and $n \in \mathbf{N}$, denote by A_n the word of length n, whose ith bit is equal to $A(\bar{n}i)$.[3]

We will 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:

(G) $\forall n \in \mathbf{N}\ F(A_n) \neq *$.

If (G) is true, then $L^A(n) = F(A_n)$ for all n. Since the function $h(n) = A_n$ is weakly polynomial relative to A, the assertion (G) implies that the language L^A is in $\mathrm{POLY}^A(F)$.

Let us enumerate all the functions being weakly polynomial relative to oracles. This means that we enumerate pairs of oracle machines involved in the definition of polynomial weak computability relative to an oracle. Denote ith function by $f_i^A(x)$ (A is considered as the second argument of the function). Let E be a polynomial-time decidable language such that $F(E_n) \neq *$ for all $n \in \mathbf{N}$. Such a language exists because F satisfies the condition 3.1. We start with $A = E$ to make the condition (G) true. Then we make countable number of steps numbered by $1, 2, \ldots$. On the ith step we change the value of A on a finite set of words to satisfy the following local condition

(L_i) $\exists n \in \mathbf{N}\ F(A_n) \neq G(f_i^A(n))$,

being careful not to injure the condition (G).

Then we *fix* all the values of A needed to ensure the truth of the assertion (L_i) and also all the values of A that were changed. This is to be understood as follows. Evidently, there exists a finite set U of words such that for all $A' \subseteq \mathbf{B}^*$, if A' and A have the same values on all the elements of U, then (L_i) is true for A'. We find such a U and "label" all its elements and all the elements on which A's value was changed. The values of A on labeled words are called "fixed" and cannot be changed later. Thus, when we will make ω steps, we will obtain an oracle A such that the condition (G) is true and the condition (L_i) is true for all $i \in \mathbf{N}$. Evidently, $L^A \in \mathrm{POLY}^A(F) \setminus \mathrm{POLY}^A(G)$ for that A.

So we have to describe ith step. Let A be the oracle constructed on $(i-1)$th step (with some fixed values).

Assume that α is in $\mathbf{B}^*$ and $|\alpha| = n$. Denote by $A[\alpha]$ the oracle where A_n is replaced by α, that is,

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

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

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 weakly polylogarithmic, therefore, $H \in \mathrm{LOGS}(G)$. Consequently, there exist infinitely many $\alpha \in \mathbf{B}^*$ such that $F(\alpha) \not\leq H(\alpha)$. Hence, there exists an $\alpha \in \mathbf{B}^*$ such that $F(\alpha) \not\leq H(\alpha)$ and no value of A on a

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

word of the form $\overline{|\alpha|}i$, $i \le |\alpha|$ is fixed. Pick such an α and replace A with $A[\alpha]$. Now the assertion (L_i) is true for $n = |\alpha|$ because $F(A_n) = F(\alpha) \not\le H(\alpha) = G(f_i^A(n))$.

Fix the values of A ensuring the truth of condition (L_i). Note that the assertion (G) is not injured because $F(A_n) = F(\alpha)$ and $F(\alpha) \ne *$ (since $F(\alpha) \not\le H(\alpha)$ and $*$ is the least element in the set $\{0, 1, *\}$). The implication 3.5⇒3.4 is proved.

Remark 1. All the separation problems F defining complexity classes studied in the literature have the following property. If in the definition of the class $\mathrm{POLY}(F)$ we add the extra requirement $|f(x)| = 2^{\mathrm{poly}(|x|)}$, (the definition of polynomial weak computability implies only that $|f(x)| \le 2^{\mathrm{poly}(|x|)}$), then the class $\mathrm{POLY}(F)$ does not change. Moreover, all those problems F have the following property. For a separation problem F, define the new separation problem

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

Then for all the classes studied in the literature, the corresponding separation problems F satisfy the following condition:

$$F \in \mathrm{LOGS}(\bar{F}). \tag{3.7}$$

Note that (3.7) implies $\mathrm{POLY}^A(\bar{F}) = \mathrm{POLY}^A(F)$ for all A (by Theorem 1).

If a separation problem F has the property (3.7), then the conditions 3.3, 3.4, and 3.5 are equivalent to the condition

3.8 $\mathrm{EXP}^A(F) \subseteq \mathrm{EXP}^A(G)$ for all A,

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

Indeed, the implication 3.4⇒3.8 is true because if $f(\alpha)$ is a weakly polylogarithmic function and $g(x)$ is a function weakly exponential relative to A, then the function $f(g(x))$ is weakly exponential relative to A (because $\mathrm{polylog}(2^{2^{O(n)}}) = \mathrm{poly}(2^{O(n)}) = 2^{O(n)}$). Conversely, let us prove the implication 3.8⇒3.4. Let F have the property (3.7) and let 3.4 be false. Then $\bar{F} \notin \mathrm{LOGS}(G)$. Applying the same arguments as those in the proof of implication ¬3.4⇒ ¬3.5, we can construct an oracle A such that the language $L^A = \{n \mid F(A_{2^n}) = 1\}$ is in $\mathrm{EXP}^A(F) \setminus \mathrm{EXP}^A(G)$.

Let us call a mapping $A \mapsto \mathrm{POLY}^A(F)$ the *manifold generated by* F. In general, any mapping from the set of all oracles into the set of families of languages will be called a *manifold.* For a family $\mathcal{F}$ of separation problems, define the manifold $\mathrm{POLY}^A(\mathcal{F}) = \bigcup_{F \in \mathcal{F}} \mathrm{POLY}^A(F)$. Define $\mathrm{LOGS}(\mathcal{F}) = \bigcup_{F \in \mathcal{F}} \mathrm{LOGS}(F)$.

It is easy to see that Theorem 1 can be generalized to families of separation problems.

Corollary 1. *If all the elements of a family $\mathcal{F}$ of separation problems have the property* 3.1 *and all the elements of a family $\mathcal{G}$ of separation problems have the property* 3.2, *then the following are equivalent:*

3.8 $\mathrm{LOGS}(\mathcal{F}) \subseteq \mathrm{LOGS}(\mathcal{G})$

3.9 $\mathrm{POLY}^A(\mathcal{F}) \subseteq \mathrm{POLY}^A(\mathcal{G})$ *for all A.*

Any manifold of the form $\mathrm{POLY}^A(\mathcal{F})$, where $\mathcal{F}$ is a family of non-degenerate separation problems, is called *regular* and is called *strongly regular* if $\mathcal{F}$ is one-element. Corollary 1 implies that a regular manifold $\mathrm{POLY}^A(F)$ defines family F

uniquely up to the weak polylogarithmic 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 exist 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 nonrelativizable assertion of the form $\mathrm{POLY}(F_1) = \mathrm{POLY}(F_2)$, namely the assertion IP = PSPACE proven by Shamir in [Sh 90]. Both the classes IP and PSPACE can be defined in our framework as shown in §7.

Consider the following application of Theorem 1 (it appeared in fact in [BGS 75]). Suppose we wish to prove that there exists an oracle A such that $\mathrm{P}^A \neq \mathrm{NP}^A$. According to Theorem 1, it is enough to prove that F_{NP} is not in PLOG. In other words, we have to prove that no machine can in time polylogarithmic of $|\alpha|$ recognize if one occurs in α. Assume that a polylogarithmic-time machine M recognizes whether one occurs in α. Run the machine M on the input word containing only zeros and long enough (its length n should be greater than the running time of M on words of length n; such an n does exist because $n - \mathrm{polylog}(n) \to +\infty$). The output of the machine should be 0. But since M had not queried at least one bit of α, we can fool it by changing that bit of α to 1.

In this proof we have used only that the number of bits queried by the machine M working on input α is restricted by a polylogarithm of $|\alpha|$, and the running time can be arbitrary large. This is true for all the known proofs of the results of the form $\exists A\ K_1^A \not\subseteq K_2^A$. Let us formalize this claim. Replace in the Definition 1 the restrictions for time with the restrictions for the number of queried bits of x and denote by n.u.LOGS(G) the class obtained from the class LOGS(G) after this replacement. 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 n.u.LOGS(G) $\supseteq$ LOGS(G). Assertions concerned with the number of queries can be usually proved by counting arguments.

Let us give the formal definition of the class n.u.LOGS(F) using another model of computation, namely, decision trees.

Let $x_1, \dots, x_n$ be boolean variables and let M be a set. An $(M, x_1, \dots, x_n)$-tree is a finite binary rooted tree whose leaves are labeled by elements of M, whose internal vertices are labeled by variables from the set $\{x_1, \dots, x_n\}$, and for every internal vertex, one of the two edges going from that vertex to its sons is labeled by 0 and the other is labeled by 1. An $(M, x_1, \dots, x_n)$-tree T computes the function $f : \mathbf{B}^n \to M$ defined as follows. Let $b_1 \dots b_n$ belong to $\mathbf{B}$. Evidently, there exists a single path in the tree starting at the root and going to a leaf such that for every pair $\langle u, v\rangle$ of consequent vertices in this path if u is labeled by x_i, then the edge $\langle u, v\rangle$ is labeled by b_i. The value $f(b_1 \dots b_n)$ is defined as the label of the end leaf in this path. We will denote the defined function by the same letter as the tree itself, i.e., $T(x_1 \dots x_n)$. The complexity of a tree is defined as its height.

A partial function $f : \mathbf{B}^n \to M$ *is computable in t queries* if there exists an $(M, x_1, \dots, x_n)$-tree T of height at most t such that the function $T(x_1, \dots, x_n)$ extends the function $f(x_1 \dots x_n)$. Replace in Definition 1 the notion of computability in time $t(|x|)$ with the notion of computability in $t(|x|)$ queries. The resulting notion is called the *non-uniform weak computability in time $t(n)$*.

Definition 3. n.u.LOGS(G) is the class of all the separation problems F such that $F(\alpha) \leq G(f(\alpha))$ for some non-uniform weakly polylogarithmic function f and for all $\alpha \in \mathbf{B}^*$.

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

Corollary 2. *If*

$$F \notin \mathrm{n.u.LOGS}(G), \tag{3.11}$$

then the negation of 3.5 *is true.*

It is the assertion (3.11) that is proved by counting arguments in all the known proofs of theorems of the form

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

4. The criterion of relativizable existence of an m-complete language in a complexity class

Denote the polynomial many-one reducibility (Karp reducibility) by $\leq_m^p$. Recall that $L_1 \leq_m^p L_2$ 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 by 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 any 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. Call a class K_1 of separation problems $\leq$*-hard* for a class K_2 of separation problems, if K_1 has a problem being $\leq$-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. To make its formulation more natural let us introduce the notion of weak polylogarithmic reducibility, which is denoted by $\preceq_m^l$. We say that $F \preceq_m^l G$ if $F \in \mathrm{LOGS}(G)$, that is, reducing functions are the polylogarithmic ones. It is easy to see that the relation $\preceq_m^l$ is reflexive and transitive and that every separation problem F is $\preceq_m^l$-complete in the class $\mathrm{LOGS}(F)$. We say that a separation problem G *solves* a separation problem F if $F(x) \leq G(x)$ for all $x \in \mathbf{B}^*$.

Theorem 2. *If a separation problem F satisfies the condition* 3.1 *and a separation problem G satisfies the condition* 3.2*, then the following are equivalent:*

4.1 $\mathrm{LOG}(G)$ $\preceq_m^l$*-hard for* $\mathrm{LOGS}(F)$*,*

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

4.3 *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:

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

Proof. Let us prove the implication 4.1⇒4.2. Assume that 4.1 is true, that is, there exists a separation problem $H \in \mathrm{LOGS}(G)$ such that any separation problem in the class $\mathrm{LOGS}(F)$ is $\preceq_m^l$-reducible to H. Then $F \preceq_m^l H$. Let $g : \mathbf{B}^* \to \mathbf{B}^*$ be a function reducing F to H. Then the language $H(g(\alpha))$ solves F and belongs to $\mathrm{LOG}(G)$.

Let us prove the implication 4.2⇒4.1. Assume that a language $H \in \mathrm{LOG}(G)$ solves F. Then the language H is $\preceq_m^l$-hard for the class $\mathrm{LOGS}(F)$ because the problem F is $\preceq_m^l$-complete in $\mathrm{LOGS}(F)$.

Evidently, 4.1 implies 4.4. The implication 4.4⇒4.2 in the case when F is a language can be proved just as the implication 4.1⇒4.2 is proved because $F \in \mathrm{LOG}(F)$ in this case.

Let us prove the implication 4.2⇒4.3.

Let F have a solution $H \in \mathrm{LOG}(G)$. Assume that $A \subseteq \mathbf{B}^*$. 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 3.4⇒3.5 we did not used conditions 3.1 and 3.2). Therefore, it suffices to prove that the class $\mathrm{POLY}^A(H)$ is $\le_m^{p,A}$-hard for the class $\mathrm{POLY}^A(F)$. In fact, we will prove that the class $\mathrm{POLY}^A(H)$ has an $\le_m^p$-complete language. Let g_0^A, g_1^A, g_2^A, ... be an enumeration of all the functions being weakly polynomial relative to A. Set $L_i^A(x) = H(g_i^A(x))$. By definition, $\mathrm{POLY}^A(H) = \{L_i^A \mid i \in \mathbf{N}\}$.

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

Let us prove the existence of such a function f^A. Let M^A be a machine that in time $p_i(|x|)$ computes the length of the word $g_i^A(x)$ for any given $\bar{i}\bar{x}$, and let N^A be a machine that in time $p_i(|x|)$ computes the jth bit of the word $g_i^A(x)$ 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 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 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 4.2 is false, then 4.3 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 $\le_m^{p,A}$-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 being weakly polynomial relative to oracle A and let m_0^A, m_1^A, ..., m_j^A, ... be an enumeration of all the $\le_m^{p,A}$-reducing functions (that is, all the functions of the type $\mathbf{B}^* \to \mathbf{B}^*$ being polynomial relative to A). Assume that $A \subseteq \mathbf{B}^*$. Call the language $A^i = \{x \mid \bar{i}x \in A\}$ the ith *component of* A and denote by $L_i(A)$ the language $\{n \mid F((A^i)_n) = 1\}$. Recall that for $C \subseteq \mathbf{B}^*$ C_n stands for the word of length n, whose jth bit is equal to $C(\bar{n}j)$. It's clear that it suffices to construct an oracle A such that for all $i \in \mathbf{N}$, at least one of the following two assertions is true:

(L$_1^i$) $G(f_i^A(y)) = *$ for some $y \in \mathbf{B}^*$;

and

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

The condition (L$_i^1$) is local, therefore we denote it by (L$_i^1$). To make the condition (∗) true it suffices to ensure one global condition

(G$_i$) $F((A^i)_n) \neq *$ for all $n \in \mathbf{N}$

and the following countable family of local assertions

(L^2_{ij}) $\exists n \in \mathbf{N}\ F((A^i)_n) \neq G(f_i^A(m_j^A(1^n)))$, $j \in \mathbf{N}$.

Thus we have to construct an oracle A such that for all pairs $(i,j) \in \mathbf{N}^2$ at least one of the two assertions (L^1_i) and (G_i)&(L^2_{ij}) is true.

Let us start with the oracle A being a polynomial-time decidable language such that for all i the assertion (G_i) is true. Then we fix an enumeration of the set $\mathbf{N}^2$ and make countable number of steps enumerated 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 (L^1_i) or the assertion (L^2_{ij}) true. Evidently, if for some i there exists j such that we have satisfied the condition (L^1_i) on the step (i,j), then we can skip the remaining steps (i,j'). On each step we will fix the value of A on some words.

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

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

2nd case: for any changes of non-fixed values of A^i the assertion (L^1_i) remains false. Assume that $\alpha \in \mathbf{B}^*$. Let $B \subseteq \mathbf{B}^*$ stand for the oracle such that $B^{i'} = A^{i'}$ for $i' \neq i$ and $B^i = (A^i)[\alpha]$ (let us remind that the notation $C[\alpha]$ is defined in the proof Theorem 1). Denote B by $A[\alpha,i]$. Consider the language

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

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

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

is weakly polylogarithmic, and $H(\alpha) = G(g(\alpha))$ for all $\alpha \in \mathbf{B}^*$. Hence $H \in \mathrm{LOG}(G)$.

Thus, there exist 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 fix all the values of A which the value of $f_i^A(m_j^A(n))$ depends on, and fix 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 (L^2_{ij}) true. And the assertion (G_i) was not injured because $F(\alpha) \neq *$. Since we have redefined only ith component of A all other assertions of the form ($\mathrm{G}_{i'}$) were not injured.

The implication 4.3⇒4.2 is proved.

Corollary 3. *If F is a language, then the class* $\mathrm{POLY}^A(F)$ *has a* $\le_m^p$*-complete language.*

Remark 2. It is clear from the proof of Theorem 2 that in the condition 4.3, we can replace the $\le_m^{p,A}$-reducibility by the $\le_m^p$-reducibility.

Remark 3. It is clear from the proof of Theorem 2 that for any sequence $\{\langle F_i, G_i\rangle\}$, $i = 0, 1, 2, \dots$ 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 $\le_m^{p,A}$-hard for the class $\mathrm{POLY}^A(F_i)$ for all i. To do so we have to consider for all i the countable number of components $A^{i,j} = \{x \in \mathbf{B}^* \mid \overline{i}\,\overline{j}x \in A\}$, $j \in \mathbf{N}$. The same is true for Theorem 1 and for Theorems 3 and 4 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 $\le_m^{p,B_j}$-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 4. *If for nondegenerate separation problems F and G the assertion*

$$F\text{has no solution in the class n.u.LOGS}(G), \tag{4.5}$$

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

The assertion (4.5) is the assertion usually proved by counting arguments when one proves that there exists A such that the class $\mathrm{POLY}^A(G)$ is not $\le_m^{p,A}$-hard for the class $\mathrm{POLY}^A(F)$.

Example. In [N 89], it was proved that n.u.BPPLOG = n.u.PLOG. Obviously, the separation problem F_{R} defining the class R has no solution in the class n.u.PLOG. Consequently, there exists an oracle A such that the class BPP^A has no $\le_m^{p,A}$-hard language for the class R^A.

Remark 4. If we replace in the statement of Theorem 2 the separation problems F and G by countable classes $\mathcal{F}$ and $\mathcal{G}$ of separation problems then the implication 4.3⇒4.1 remains true. To keep the implication 4.1⇒4.3 true, we have to strengthen the condition 4.1 as follows. Replace the condition 4.1 by the following condition: "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 weakly polylogarithmic and reduces the ith separation problem in $\mathcal{F}$ to H".

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

Denote by $\le_T^p$ the polynomial Turing reducibility (Cook reducibility) and denote by $\le_T^{p,A}$ the polynomial Turing reducibility relative to oracle A. Recall that $L_1 \le_T^{p,A} L_2$ if there exists a polynomial-time Turing machine M having two oracles A and L_2 and recognizing L_1.

Let $\le$ stand for some type of reducibility. Let us call a class K_1 to be $\le$-reducible to a class K_2 (notation: $K_1 \le K_2$) if $\forall L_1 \in K_1\ \exists L_2 \in K_2 \quad L_1 \le L_2$.

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

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

$$G(f(y,\alpha)) \neq * \text{ for all } y \in \mathbf{B}^*, \tag{5.1}$$

$$F(\alpha) = M^{G(f(\cdot,\alpha))}(|\alpha|), \tag{5.2}$$

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

We call a pair $\langle M, f\rangle$ a pair *reducing* F to H if the conditions (5.1) and (5.2) are true for all $\alpha \in \mathrm{D}(F)$. Note that if there exists a pair $\langle M, f\rangle$ such that the conditions (5.1) and (5.2) are true for all but finitely many $\alpha \in \mathrm{D}(F)$, then $F \preceq_T^l G$. We denote by $\langle M, f\rangle^G(\alpha)$ the output of M on input $|\alpha|$ with oracle $G(f(\cdot, \alpha))$.

Obviously, the binary relation $\preceq_T^l$ is reflexive and transitive. It is clear that $F \preceq_m^l G \Rightarrow F \preceq_T^l G$.

Theorem 3. *If a separation problem F satisfies the condition* 3.1 *and a separation problem G satisfies the condition* 3.2*, then the following are equivalent:*

5.3 $\mathrm{LOGS}(F) \preceq_T^l \mathrm{LOGS}(G)$,

5.4 $F \preceq_T^l G$,

5.5 $\mathrm{POLY}^A(F) \leq_T^{p,A} \mathrm{POLY}^A(G)$ *for all oracles A.*

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

5.6 $\mathrm{LOG}(F) \preceq_T^l \mathrm{LOG}(G)$.

Proof. Evidently, the conditions 5.3 and 5.4 are equivalent.

Assume that F is a language. Then the implication 5.6⇒5.4 is true. On the other hand, assume that 5.6 is true, that is, $F \preceq_T^l G$. 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 $n \in \mathbf{N}$. 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 $\mathrm{D}(F) = \mathbf{B}^*$, we have $G(f(x, \alpha)) \neq *$ for all $x, \alpha \in \mathbf{B}^*$. Therefore, $H(\beta) = G(h(\beta))$, where

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

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

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

Let us prove that 5.4 implies 5.5. Assume that $F \preceq_T^l G$. Denote by $\langle M, f\rangle$ a pair reducing F to G. Let A be an oracle and let L a language in the class $\mathrm{POLY}^A(F)$. Let g be a weakly polynomial relative to A function such that $L(x) = F(g(x))$. Then $L(x) = M^{G(f(\cdot, g(x)))}(|g(x)|)$ for all $x \in \mathbf{B}^*$. Since the function $|g(x)|$ is polynomial-time computable relative to A, the language L is $\leq_T^{p,A}$-reducible to the language

$\{\bar{y}g(x) \mid G(f(y,g(x))) = 1\}$, which is in $\mathrm{POLY}^A(G)$ because $G(f(y,g(x))) \neq *$ for all $x, y \in \mathbf{B}^*$ and the function $\bar{y}x \mapsto f(y, g(x))$ is weakly polynomial relative to A.

Let us prove the implication $\neg 5.4 \Rightarrow \neg 5.5$. Assume that $F \not\preceq^l_T G$. Let us prove that 5.5 is false. Note that in the assertion 5.5 the $\leq^{p,A}_T$-reducibility can be replaced by the $\leq^p_T$-reducibility. Indeed, if a language L_1 is $\leq^{p,A}_T$-reducible to a language L in $\mathrm{POLY}^A(G)$, then L_1 is $\leq^p_T$-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 the condition 3.2 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 are true:

(G) $A_n \in \mathrm{D}(F)$ for all n,

and

(L) the language $\{n \mid F(A_n) = 1\}$ is not $\leq^p_T$-reducible to any language in $\mathrm{POLY}^A(G)$.

Let M^B_1, M^B_2, ..., M^B_j, ... be an enumeration of all the polynomial-time oracle Turing machines. Let $f^A_1(x)$, $f^A_2(x)$, ..., $f^A_i(x)$, ... be an enumeration of all the weakly polynomial relative to A functions. We want to construct an oracle A such that the following assertion (L_{ij}) is true for all $i, j \in \mathbf{N}$:

(L_{ij}) $\exists n \in \mathbf{N}\ F(A_n) \neq M_j^{G(f_i^A(\cdot))}(n) \ \vee\ \exists y\ G(f_i^A(y)) = *$.

At first, let A be equal to a polynomial-time decidable language satisfying the condition (G). Make ω steps enumerated by pairs $(i,j) \in \mathbf{N}^2$.

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

1st case: there exist free $\alpha \in \mathrm{D}(F)$ and $y \in \mathbf{B}^*$ such that $F(\alpha) \neq *$ and $G(f_i^{A[\alpha]}(y)) = *$. Then replace A by $A[\alpha]$ and fix finite number of values of A to guarantee the validity of the assertion (L_{ij}). Note that the condition (G) has not been injured.

2nd case: $G(f_i^{A[\alpha]}(y)) \neq *$ for all $y \in \mathbf{B}^*$ for all free $\alpha \in \mathrm{D}(F)$. Let us prove that there exists 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 weakly computable in time $\mathrm{poly}(|y| + \log|\alpha|)$ and for the pair $\langle M, g\rangle$ the conditions (5.1) and (5.2) are fulfilled for all the free $\alpha \in \mathrm{D}(F)$. Therefore, $F \preceq^l_T G$ and we get a contradiction. After that the proof goes similar to the proof of Theorem 1.

6. The criterion of whether a complexity class has Turing hard language for another complexity class

Theorem 4. *If a separation problem F satisfies the condition* 3.1 *and a separation problem G satisfies the condition* 3.2, *then the following are equivalent:*

6.1 *the class* $\mathrm{LOG}(G)$ *is $\preceq^l_T$-hard for the class* $\mathrm{LOGS}(F)$,

6.2 *the class* $\mathrm{LOG}(G)$ *has a language which F is $\preceq^l_T$-reducible to,* 6.3 *the class* $\mathrm{POLY}^A(G)$ *is* $\le_T^{p,A}$*-hard for the class* $\mathrm{POLY}^A(F)$ *for all oracles* A.

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

6.4 $\mathrm{LOG}(G)$ *is* $\preceq_T^l$*-hard for* $\mathrm{LOG}(F)$.

Proof. Evidently, the assertions 6.1 and 6.2 are equivalent and if F is a language, then they both are equivalent to the assertion 6.4.

Let us prove the implication 6.2⇒6.3. Assume that $F \preceq_T^l H \in \mathrm{LOG}(G)$. If H does not satisfy the condition 3.2, then $F \in \mathrm{PLOG}$ and therefore the assertion 6.3 is true. Otherwise, Theorem 2 implies that for any oracle A the class $\mathrm{POLY}^A(H)$ has a $\le_m^p$-complete language. Theorem 3 implies that $\mathrm{POLY}^A(F) \le_T^{p,A} \mathrm{POLY}^A(H)$, consequently, the class $\mathrm{POLY}^A(G)$ is $\le_T^{p,A}$-hard for the class $\mathrm{POLY}^A(F)$.

Let us prove that the assertion 6.3 implies the assertion 6.2. Similar to Theorem 4, we can replace $\le_T^{p,A}$-reducibility by the $\le_T^p$-reducibility in 6.3.

Assume that 6.2 is false, that is, F is $\preceq_T^l$-reducible to no language in the class $\mathrm{LOG}(G)$.

We construct an oracle A such that the class $\mathrm{POLY}^A(G)$ has no language being $\le_T^p$-hard for the class $\mathrm{POLY}^A(F)$. Let $f_0^A(y), f_1^A(y), \ldots, f_i^A(y), \ldots$ be an enumeration of all the weakly polynomial relative to A functions. Split A into components $A^i = \{x \mid \bar{i}x \in A\}$. It suffices to define A in such a way that for any $i \in \mathbf{N}$ at least one of the following two assertions holds:

(L_i^1) $G(f_i^A(y)) = *$ for some $y \in \mathbf{B}^*$,

and

($*$) the language $L_i(A) = \{n \mid F(A_n^i) = 1\}$ is in the class $\mathrm{POLY}^A(F)$ and is not $\le_T^p$-reducible to the separation problem $G(f_i^A(y))$.

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

To make the assertion ($*$) true it suffices to satisfy the following requirement (G_i):

(G_i) $F(A_n^i) \ne *$ for all n,

and at the same time to satisfy the following condition (L_{ij}^2) for all $j \in \mathbf{N}$:

(L_{ij}^2) $\exists n \in \mathbf{N}\ F(A_n^i) \ne M_j^{G(f_i^A(\cdot))}(n)$.

To construct an oracle A satisfying (L_i^1) or (G_i)&(L_{ij}^2) for all pairs (i,j) we can follow the proof of Theorem 2. The only difference appears in the second case when the step (i,j) is described. Recall that in the second case $G(f_i^A(y)) \ne *$ for all $y \in \mathbf{B}^*$ and for all variations of non-fixed values of A^i. We call a word $\alpha \in \mathbf{B}^*$ free if no value of A^i on a word of the form $\overline{|\alpha|}j$, $j \le |\alpha|$, is fixed. We have to prove that there exists a free $\alpha \in \mathrm{D}(F)$ such that $F(\alpha) \ne M_j^{G(f_i^{A[\alpha,i]}(\cdot))}(|\alpha|)$. Assume that there exists no such α. Denote by $l(n)$ a polylogarithmic upper bound for the length of queries made by the machine M on input n. Consider the language

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

and the function $g(y,\alpha) = \bar{y}\alpha$. Since $G(f_i^{A[\alpha,i]}(y)) \ne *$ for all free α and for all $y \in \mathbf{B}^*$, the language H is in $\mathrm{LOG}(G)$. Then for the pair $\langle M_j, g\rangle$ assertions (5.1) and (5.2) are true for all free $\alpha \in \mathbf{B}^*$. Therefore, $F \preceq_T^l H$. This contradiction finishes the proof.

Corollary 5. *If $F \not\leq_T^l$ n.u.LOG(G), then there exists an oracle A such that the class* POLY$^A(G)$ *is not $\leq_T^{p,A}$-hard for the class* POLY$^A(F)$.

Remark 5. Let K_1, K_2 be classes of languages and let A be an oracle. In the paper [A-S 86] 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 which all the languages in K_1 are $\leq_T^{p,A}$-reducible to. Then consider the language

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

where $M_0, M_1, \dots$ 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, $L_1 \in K_2$ holds.

7. Relativizable inclusions between particular complexity classes

In this section we consider many of the regular manifolds lying between P^A and PSPACE^A (the only exception is the manifold Few^A; the author does not know whether this manifold is regular). As it was mentioned in Corollary 1, all the particular complexity classes studied in the literature can be generated by means of separation problems which are not equal to $*$ only on the words of length 2^n, $n \in \mathbf{N}$. To simplify the notation, we consider in the sequel only separation problems satisfying this requirement. Denote $\mathbf{B}^{2^n}$ by $\mathbf{F}_n$ and $\bigcup_{i=0}^{\infty} \mathbf{F}_n$ by $\mathbf{F}$. We enumerate the bits of a word $\alpha \in \mathbf{F}_n$ by binary words of length n rather than by the numbers from 1 to 2^n. For a word α in $\mathbf{F}$ by $\|\alpha\|$ we mean $\log_2 |\alpha|$. We call $\|\alpha\|$ the *norm* of α. While defining particular separation problems we keep the following agreement: if the problem under consideration is defined only on a set $M \subseteq \mathbf{B}^*$, then its value on all the words from $\mathbf{B}^* \setminus M$ is equal to $*$ (that is, the default value is $*$).

Consider the following relativized complexity classes: UP^A, Co-UP^A, UP$^A \cap$ Co-UP^A, FewP^A, Co-FewP^A, FewP$^A \cap$ Co-FewP^A, Few^A, $\oplus$P^A, R^A, Co-R^A, R$^A \cap$ Co-R^A, NP^A, Co-NP^A, NP$^A \cap$ Co-NP^A, BPP^A, MA^A, Co-MA^A, MA$^A \cap$ Co-MA^A, AM^A, Co-AM^A, 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, IP$^A \cap$ Co-IP^A, PH^A, PSPACE^A.

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

1. UP^A is the manifold generated by the following separation problem:

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

2. FewP^A $\rightleftharpoons$ POLY$^A(\mathcal{F})$, where $\mathcal{F}$ consists of all the separation problems F such that

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

where p is a polynomial.

3. Few^A is the class defined in the paper [CH 90] as follows:
a language L is in the class Few^A if there exist a function f^A being weakly polynomial relative to A, a polynomial q and a predicate R^A defined on the set $\mathbf{B}^* \times \mathbf{N}$ being polynomial-time computable relative to A, such that $L(x) = R^A(x, \#_1 f^A(x))$ and $\#_1(f^A(x)) \le q(|x|)$ for all $x \in \mathbf{B}^*$. It is unknown if the manifold Few^A is regular.

4. $\oplus\mathrm{P}^A \rightleftharpoons \mathrm{POLY}^A(\mathrm{PARITY})$, where

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

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

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

where uv stands for the concatenation of words u and v. Denote the class $\mathrm{LOGS}(F_{\mathrm{AM}})$ by AMLOGS.

6. MA^A is the class generated by the separation problem

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

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

7. Let us prove that the manifold PSPACE^A has the form $\mathrm{POLY}^A(F)$.

It is well known that any language L in PSPACE^A can be represented as follows:

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

where $P^A(x,u)$ is a predicate being polynomial-time computable relative to A and $p(m)$ is a polynomial.

The converse is true, too. Therefore, we can take the separation problem

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

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

8. Let us prove that the manifold IP^A can be represented in the form $\mathrm{POLY}^A(F)$.

Take the following separation problem F_{IP}: on words $\alpha \in \mathbf{F}$ of length 2^{2n^2} it is defined as follows

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

(where the probability is considered with respect to the uniform distribution in $r_1 \cdots r_n$).

Then $\mathrm{POLY}^A(F_{\mathrm{IP}}) = \mathrm{IP}^A$.

To explain the intuitive meaning of the definition of F_{IP}, let us remind the definition of the class IP^A according to [B 85] and convert it to a convenient form. By a *Verifier* we mean a pair $V = (q, Q)$, where Q is a polynomial-time computable predicate on $\mathbf{B}^* \times \mathbf{B}^* \times \mathbf{B}^*$ and $q : \mathbf{N} \to \mathbf{N}$ is a polynomial. Any function $P : \mathbf{B}^* \to \mathbf{B}^*$ is called a *Prover.* Assume that $x \in \mathbf{B}^*$, $|x| = m$. For a sequence $r_1, \dots, r_{q(m)}$ of $q(m)$ words of length $q(m)$, define the *answer of* $(P,\ V)$ *on the input* x *and random inputs* $r_1, \dots, r_{q(m)}$ as follows. For all $i \le q(m)$ set

$$p_i = P(r_1 \cdots r_i).$$

We say that the answer of (P, V) on input x and random inputs $r_1, \dots, r_{q(m)}$ is equal to 1 if lengths of all the words p_i are equal to $q(m)$ and $Q(x, r_1 \cdots r_{q(m)}, p_1 \cdots p_{q(m)}) = 1$; otherwise answer is equal to 0. Denote the answer of (P, V) on input x and random inputs $r_1, \dots, r_{q(m)}$ by $(P, V)(x)_{r_1 \cdots r_{q(m)}}$. We say that a language L belongs to IP, if there exists a Verifier V such that the following two assertions are true:

$$\forall x \in L\ \exists P\ \mathrm{Prob}\,[(P, V)(x)_{r_1 \cdots r_{q(|x|)}} = 1] > 2/3$$
$$\forall x \notin L\ \forall P\ \mathrm{Prob}\,[(P, V)(x)_{r_1 \cdots r_{q(|x|)}} = 0] > 2/3,$$

where the probability is considered with respect to the uniform distribution in $r_1 \cdots r_{q(|x|)}$.

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 with private coins (see, for example [GMR 85, GMR 89] does not fit into our framework. However, as proven in [GS 86], these two definitions are equivalent and the proof of the equivalence is relativizable.

A language L is in $\mathrm{LOG}(F_{\mathrm{IP}})$ if there exists a polylogarithmic-time Verifier for which the above assertion holds. Let us denote the class $\mathrm{LOG}(F_{\mathrm{IP}})$ by IPLOG.

9. On classes of the form Co-K^A and $K^A \cap$Co-K^A. Note that if the manifold K^A is [strongly] regular, then the manifold Co-$K^A = \{\mathbf{B}^* \setminus L \mid L \in K^A\}$ is [strongly] regular. If K_1^A, K_2^A are strongly regular, say $K_i^A = \mathrm{POLY}^A(F_i)$, $i = 1, 2$, then the manifold $K_1^A \cap K_2^A$ is strongly regular. Indeed, take the following separation problem F:

$$F(\alpha) = \begin{cases} 1, & \text{if } \alpha = \overline{|\alpha_1|}\alpha_1\alpha_2, \text{where } F_1(\alpha_1) = F_2(\alpha_2) = 1, \\ 0, & \text{if } \alpha = \overline{|\alpha_1|}\alpha_1\alpha_2, \text{where } F_1(\alpha_1) = F_2(\alpha_2) = 0, \\ *, & \text{if } \alpha \text{ has not such form.} \end{cases}$$

Obviously this separation problem F 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 inclusions between the manifolds under consideration are shown at Figure 1 (a manifold K_1^A is included in a manifold K_2^A if $K_1^A \subseteq K_2^A$ for all A). That is, all the known relativizable inclusions between the classes under consideration are shown at Figure 1. A line segment connects a class K_1^A with a class K_2^A if the

FIGURE 1. Relativizable inclusions between complexity classes.

class K_1^A is included in the class K_2^A, and the class K_2^A is positioned higher than the class K_1^A.

7.1 Historical references. The nontrivial inclusions on the Figure 1 were proved by the following authors.

7.1.1. The assertion $\mathrm{MA}^A \subseteq \Sigma_2^A \cap \Pi_2^A$ follows from Gács' result (published in [S 83]) stating that $\mathrm{BPP}^A \subseteq \Sigma_2^A \cap \Pi_2^A$. Namely, in [S 83] a separation problem $G(\alpha)$ is constructed such that $G(\alpha)$ is a solution of F_{BPP} and

$$G(\alpha) = 1 \Longleftrightarrow \forall y \in \mathbf{B}^{p(\|\alpha\|)}\ \exists z \in \mathbf{B}^{p(\|\alpha\|)} Q(\alpha, y, z) \tag{7.1}$$

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

7.1.2. The assertion $\mathrm{AM}^A \subseteq \Pi_2^A$ follows from the cited Gács' result. However, for this assertion, it is important that in (7.1) the predicate $Q(\alpha, y, z)$ is monotone in α (that is, if α' can by obtained from α be replacing some zeros by ones, then $Q(\alpha, y, z) \Rightarrow Q(\alpha', y, z)$).

7.1.3. The assertion $\mathrm{MA}^A \subseteq \mathrm{AM}^A$ was proved in [B 85].

7.1.4. $\mathrm{Few}^A \subseteq \oplus\mathrm{P}^A$ was proved in [CH 90].

7.1.5. The assertion $\mathrm{MA}^A \subseteq \mathrm{PP}^A$ can be proved rather easily. Besides that, it easily follows from the assertion $\mathrm{PP}^{\mathrm{BPP}} = \mathrm{PP}$ proven in [KSTT 89]. Indeed, $\mathrm{MA} \subseteq \mathrm{NP}^{\mathrm{BPP}} \subseteq \mathrm{PP}^{\mathrm{BPP}} = \mathrm{PP}$.

7.1.6. The assertion $\mathrm{Few}^A \subseteq \Sigma_2^A \cap \Pi_2^A$ follows from the assertion $\forall A\ \mathrm{Few}^A \leq_T^p \mathrm{NP}^A$, the latter assertion is easy and well known. For the sake of completeness, let us prove it here.

As noted, it suffices to prove that $\forall A\ \mathrm{Few}^A \leq_T^{p,A} \mathrm{NP}^A$. Fix $A \subseteq \mathbf{B}^*$. 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,

$$L(x) = R^A(x, \left|\{y \in \mathbf{B}^{p(|x|)} \mid Q^A(x,y)\}\right|),$$
$$\left|\{y \in \mathbf{B}^{p(|x|)} \mid Q^A(x,y)\}\right| \leq q(|x|).$$

Let us prove that having an NP^A-complete language as oracle, we can compute in polynomial time for any given x the cardinality of the set $\{y \in \mathbf{B}^{p(|x|)} : Q^A(x,y)\}$. The procedure is as follows. For a given x, check first if there exists a set $M \subseteq \mathbf{B}^{p(|x|)}$ of cardinality exactly $q(|x|)$ such that $\forall y \in M,\ Q^A(x,y)$. This can be done by querying the NP^A-complete language (since $|M|$ is polynomial bounded). If such a set M exists, then $\left|\{y \in \mathbf{B}^{p(|x|)} \mid Q^A(x,y)\}\right| = q(|x|)$. If not, then check if there exists a set $M \subseteq \mathbf{B}^{p(|x|)}$ of cardinality exactly $q(|x|) - 1$ such that $\forall y \in M\ Q(x,y)$. Repeat this procedure $q(|x|)$ times.

7.1.7. The assertion $\mathrm{Few}^A \subseteq \mathrm{PP}^A$ was proved in the paper [KSTT 89].

7.2 Is Figure 1 complete? We claim that it is the case, that is, all true relativizable inclusions are shown at 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

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

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 at Figure 1.

1. $\exists A\ \mathrm{UP}^A \cap \text{Co-UP}^A \not\subseteq \mathrm{BPP}^A$
2. $\exists A\ \mathrm{R}^A \cap \text{Co-R}^A \not\subseteq \oplus\mathrm{P}^A$
3. $\exists A\ \text{Co-UP}^A \not\subseteq \oplus\mathrm{IP}^A$
4. $\exists A\ \mathrm{FewP}^A \cap \text{Co-FewP}^A \not\subseteq \mathrm{UP}^A$
5. $\exists A\ \text{Co-R}^A \not\subseteq \mathrm{NP}^A$
6. $\exists A\ \mathrm{IP}^A \cap \text{Co-IP}^A \not\subseteq \mathrm{PH}^A$
7. $\exists A\ \mathrm{AM}^A \cap \text{Co-AM}^A \not\subseteq \mathrm{PP}^A$
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 $k \geq 3$

7.3 Proving the completeness of Figure 1. We give the proofs of all the assertions in the above list whose proofs do not require a lot of space and give references for all other assertions.

7.3.1 *Assertion* $\exists A\ \mathrm{UP}^A \cap \text{Co-UP}^A \not\subseteq \mathrm{BPP}^A$.

Theorem 5. $\exists A\ \mathrm{UP}^A \cap \text{Co-UP}^A \not\subseteq \mathrm{BPP}^A$

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

> for all $F \in \mathrm{LOGS}(G)$ there exists a weakly polylogarithmic function f such that $F(\alpha) \le G(f(\alpha))$ and $\|f(\alpha)\|$ depends only on $\|\alpha\|$ being equal to a polynomial $p(\|\alpha\|)$.

Assume that $F \in \mathrm{LOGS}(G)$ and let f be a weakly polylogarithmic 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 being elements of the set $\mathbf{B}^{p(\|\alpha\|)}$ are called *experts* (for f and $\|\alpha\|$), and the rth bit of $f(\alpha)$ is called the *opinion of* r *about* α. Let us fix a polylogarithmic machine M that computes the rth bit of the word $f(\alpha)$ for a given α and $r \in \mathbf{B}^{p(\|\alpha\|)}$. We say that *expert* r *queries* $\alpha(u)$ (where $u \in \mathbf{B}^{\|\alpha\|}$), if M queries the uth bit of α during the work on the input $\langle \alpha, r\rangle$. It is clear that for all α and all $r \in \mathbf{B}^{p(\|\alpha\|)}$ there exists at most $\mathrm{poly}(\|\alpha\|)$ $u \in \mathbf{B}^{\|\alpha\|}$ such that r queries $\alpha(u)$. Call the fraction $\bigl|\{r \in \mathbf{B}^{p(\|\alpha\|)} \mid r \text{ queries } \alpha(u)\}\bigr| bigm/2^{p(\|\alpha\|)}$ the *weight of* u *relative to* α. Denote the weight of u relative to α by $w_\alpha(u)$. 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: $\sum_{u \in \mathbf{B}^{\|\alpha\|}} w_\alpha(u) \le q(\|\alpha\|)$, where q is the polynomial restricting the number of queries of every expert $r \in \mathbf{B}^{p(\|\alpha\|)}$.

Now let us start with the proof of Theorem 5. 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, $\mathrm{POLY}^A(F) = \mathrm{UP}^A \cap \text{Co-UP}^A$).

Assume the contrary: suppose there exist a polynomial p and a polylogarithmic predicate P such that $\forall n\ \forall \beta, \gamma \in \mathbf{F}_n$,

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

Let us fix a value of n. Let $\beta_0 \in \mathbf{F}_n$, $\gamma_0 \in \mathbf{F}_n$ be the words containing only zeros. Without loss of generality we may assume that the fraction $\bigl|\{r \in \mathbf{B}^{p(n)} \mid P(\beta_0\gamma_0, r) = 1\}\bigr|/2^{p(n)}$ is greater than or equal to 1/2. We shall enumerate bits in the first half β of the word $\beta\gamma$ (where $\beta, \gamma \in \mathbf{F}_n$) by the words of the form $0u$, $u \in \mathbf{B}^n$, 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 restricted by $k = \mathrm{poly}(n)$.

Then $\sum_{u \in \mathbf{B}^n} w_{\beta_0\gamma_0}(1u) \le k$, therefore, there exists $u_0 \in \mathbf{B}^n$ such that $w_{\beta_0\gamma_0}(1u_0) \le \frac{k}{2^n} < \frac{1}{6}$ (if n is large enough). Denote by γ_1 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, the fraction $\bigl|\{r \in \mathbf{B}^{p(n)} \mid P(\beta_0\gamma_1, r) = 1\}\bigr|/2^{p(n)}$ is greater than 1/3. As $F(\beta_0\gamma_1) = 0$, we get the contradiction.

7.3.2 *Assertion* $\exists A\ R^A \cap \text{Co-R}^A \not\subseteq \oplus\mathrm{P}^A$.

Theorem 6. $\exists A\ R^A \cap \text{Co-R}^A \not\subseteq \oplus\text{P}^A$

Proof. Evidently, the manifold $\text{R}^A \cap \text{Co-R}^A$ can be generated by the following separation problem F. If $\gamma \in \mathbf{F}_1$, then $F(\gamma) = *$. If $\gamma \in \mathbf{F}_{n+1}$, denote by α the first half of γ and by β the second half of γ. Then

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

By the Theorem 1, it suffices to prove that $F \not\leq^l_m$ PARITY. Assume the contrary: suppose there exist a polynomial p and polylogarithmic predicate P such that

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

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

Let us fix a polylogarithmic machine M computing the predicate P and a sufficiently large n. Let the number of queries to the word γ made by M on inputs of the form $\langle \gamma, r\rangle$, $r \in \mathbf{B}^{p(n)}$, be bounded by $k = \text{poly}(n)$. Let us prove that for any fixed $r \in \mathbf{B}^{p(n)}$ the function $P(\gamma, r)$ is a polynomial of degree $\leq k$ (in the field of residues modulo 2) of variables $\gamma(v)$, $v \in \mathbf{B}^n$. Indeed,

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

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

Therefore, the function $\sum_{r \in \mathbf{B}^{p(n)}} P(\gamma, r)$ is a polynomial of degree at most k of variables $\gamma(v)$. Denote this polynomial by Q. Divide the variables $\gamma(v)$, $v \in \mathbf{B}^{n+1}$ into two groups $\alpha(u)$, $u \in \mathbf{B}^n$ and $\beta(u)$, $u \in \mathbf{B}^n$, 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 $u \in \mathbf{B}^n$ and set $\alpha(0^n) = 0$. Denote the resulting polynomial of degree at most $k = \text{poly}(n)$ by R. The polynomial R has $2^n - 1$ variables, has zero constant term and is equal to 1, if at least 2^{n-1} variables are equal to 1. Let us derive a contradiction from the existence of such a polynomial. Consider the set A consisting of all the 2^{n-1}-dimensional boolean vectors having exactly 2^{n-1} ones. The cardinality of the set A is equal to $\binom{2^{n-1}}{2^n-1}$. Let us prove that this number is odd. We shall use a well known criterion of whether $\binom{m}{l}$ is odd.

Lemma 1. *$\binom{m}{l}$ is odd iff any bit of the binary representation of the number m is greater than or equal to the corresponding bit of the number l.*

Proof. Let $i = m - l$. Then $\binom{m}{l} = \frac{(i+l)!}{i!l!}$. For an integer k, denote by $t(k)$ the greatest integer j such that 2^j divides k. Obviously, $t(j!) = \left[\frac{j}{2}\right] + \left[\frac{j}{4}\right] + \ldots$. Therefore

$$t\Big(\binom{m}{l}\Big) = \Big(\Big[\frac{i+l}{2}\Big] - \Big[\frac{i}{2}\Big] - \Big[\frac{l}{2}\Big]\Big) + \Big(\Big[\frac{i+l}{4}\Big] - \Big[\frac{i}{4}\Big] - \Big[\frac{l}{4}\Big]\Big) + \ldots$$

Each term in this sum is nonnegative and $\left[\frac{i+l}{2^s}\right]-\left[\frac{i}{2^s}\right]-\left[\frac{l}{2^s}\right]=0$ iff $i \bmod 2^s + l \bmod 2^s < 2^s$. Thus $\binom{m}{l}$ is odd if $i \bmod 2^s + l \bmod 2^s < 2^s$ for all s. This means that the sth bit of i or the s-bit of l is equal to zero for all s.

By this lemma, the number $\binom{2^{n-1}}{2^n-1}$ is odd. For any $\boldsymbol{a} \in A$, $R(\boldsymbol{a}) = 1$, therefore, $\sum_{\boldsymbol{a}\in A} R(\boldsymbol{a}) = 1$. Consider an arbitrary monomial T in R. Let us prove $\sum_{\boldsymbol{a}\in A} T(\boldsymbol{a}) = 0$ to get a contradiction. Let T be equal to $\alpha(u_1)\cdots\alpha(u_i)$, where $i \le k$ and $u_1, \dots, u_i$ are different. Since R has no constant term, we have $i \ge 1$. Let us prove that the number $\boldsymbol{a} \in A$ such that $\boldsymbol{a}(u_j) = 1$ for all $j \le i$, is even. Obviously, this number is equal to $\binom{2^{n-1}-i}{2^n-1-i}$ (we assume that $i < 2^{n-1}$; since $i \le k = \text{poly}(n)$, this is true if n is large enough). Let s be the number of the lowest bit of the binary representation of i being equal to 1. Then the sth bit of the number $2^n - 1 - i$ is equal to 0, and the sth bit of the number $2^{n-1} - i$ is equal to 1. Lemma 1 implies that the number $\binom{2^{n-1}-i}{2^n-1-i}$ is even.

We have to consider also the second case (the constant term in Q is equal to 1). But this case can be reduced to the first case by adding 1 to Q.

7.3.3 *Assertion* $\forall A$ Co-UP$^A \not\subseteq$ IP^A. This assertion was in fact proved in [FS 88] (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 7 (Fortnow, Sipser). $\exists A$ 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 polylogarithmic-time Verifier V such that

$$\#_1(\alpha) = 0 \Rightarrow \exists P \text{ Prob}\left[(P,V)(\alpha) = 1\right] > 2/3,$$
$$\#_1(\alpha) = 1 \Rightarrow \forall P \text{ Prob}\left[(P,V)(\alpha) = 1\right] < 1/3,$$

where $(P,V)(\alpha)$ stands the answer 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 Prover P such that 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. Let us call different outcomes of coin tossing *experts* and let us call the queries to α_0 made by the Verifier during the dialogue with the Prover P on input α_0 and outcome r of coin tossing *the queries of the expert r to α.* For a given $u \in \mathbf{B}^n$ call the fraction $\left|\{r \in \mathbf{B}^{p(n)} \mid \text{makes the query `}\alpha_0(u) =?\text{'}\}\right|/2^{p(n)}$ 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; denote the resulting word by α_1. Since Prob $[(P,V)(\alpha_0) = 1] > 2/3$, we obtain Prob $[(P,V)(\alpha_1) = 1] > 2/3 - 1/3 = 1/3$. On the other hand, this probability should be less than 1/3. Contradiction.

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

7.3.5 *Assertion* 5. $\exists A$ $\text{Co-R}^A \not\subseteq \mathrm{NP}^A$.

Theorem 8. $\exists A$ $\text{Co-R}^A \not\subseteq \mathrm{NP}^A$.

Proof. Assume the contrary: suppose there exist a polynomial p and a polylogarithmic time predicate $P(\alpha, r)$ such that $\forall \alpha \in \mathbf{F}$,

$$\#_1(\alpha) = 0 \Rightarrow \exists r \in \mathbf{B}^{p(\|\alpha\|)} P(\alpha, r) = 1$$
$$\#_1(\alpha) > 2/3|\alpha| \Rightarrow \forall r \in \mathbf{B}^{p(\|\alpha\|)} P(\alpha, r) = 0.$$

Let us find α such that $\#_1(\alpha) > (2/3)|\alpha|$ and $\exists r \in \mathbf{B}^{p(\|\alpha\|)}$ $P(\alpha, r) = 1$. Take $\alpha_0 = 0^{2^n}$, where n is large enough. Then there exists $r_0 \in \mathbf{B}^{p(n)}$ such that $P(\alpha_0, r_0)$. Change the value of α_0 on all u such that the polylogarithmic machine computing $P(\alpha_0, r_0)$ does not query '$\alpha_0(u) = ?$'. The resulting word α satisfies the desired conditions.

7.3.6 *Assertion* $\exists A$ $\mathrm{IP}^A \cap \text{Co-IP}^A \not\subseteq \mathrm{PH}^A$. In the paper [AGH 86] it was proved that $\exists A$ $\mathrm{IP}^A \not\subseteq \mathrm{PH}^A$. Minor changes in that proof allows us to prove that there exists an oracle A such that $\mathrm{IP}^A \cap \text{Co-IP}^A \not\subseteq \mathrm{PH}^A$.

7.3.7 *Assertion* $\exists A$ $\mathrm{AM}^A \cap \text{Co-AM}^A \not\subseteq \mathrm{PP}^A$. This assertion is proved in the paper [V 92].

7.3.8 *Assertion* $\exists A$ $\mathrm{AM}^A \not\subseteq \Sigma_2^A$. This assertion is proved in the paper [Sa 89].

7.3.9 *Assertion* $\exists A$ $\mathrm{PP}^A \not\subseteq \mathrm{PH}^A$. This assertion follows from the fact that there exists no $k \in \mathbf{N}$ such that the function $\mathrm{MAJORITY}(x_1, \ldots, x_n)$ can be represented in the following 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 $f_{i_1 \ldots i_{2k}}(x_1, \ldots, x_n)$ is a variable or the negation of a variable ([FSS 84], [A 83], [Y 85], [H 86]).

7.3.10 *Assertion* $\exists A$ $\oplus \mathrm{P}^A \not\subseteq \mathrm{PH}^A$. This assertion is proved in the papers [FSS 84], [A 83], [Y 85], [H 86].

7.3.11 *Assertion* $\exists A$ $\oplus \mathrm{P}^A \not\subseteq \mathrm{PP}^A$. This assertion is proved in [BG 81]. In fact, this theorem easily follows from the assertion PARITY $\not\leq_m^l$ MAJORITY proven in [MP 88].

7.3.12 *Assertion* $\forall k \geq 3$ $\exists A$ $\Pi_k^A \not\subseteq \Sigma_k^A$. The first superpolynomial lower bounds for the size Σ_k-circuits necessary for the computation of Π_k-functions where obtained by M. Sipser. We need the lower bound $(2^{f(n)})$, where f grows faster than any polylogarithm. Such a bound is obtained in the paper [H 86].

8. 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, and vice versa.
(2) $\oplus\mathrm{P}^A \leq_T^p \mathrm{PP}^A$.
(3) $\mathrm{Few}^A \leq_T^p \mathrm{NP}^A$.
(4) $\mathrm{PH}^A \leq_T^p \mathrm{PP}^A$.

The assertion (1) is evident. Both assertions (2) and (3) are simple. The assertion (3) will be proved in §7, and the assertion (2) will be proved right now. The assertion (4) was proved in the paper [T 89].

Theorem 9. $\oplus\mathrm{P}^A \leq_T^p \mathrm{PP}^A$ *for any oracle* A.

Proof. By Theorem 3 it suffices to prove that the language PARITY(α) is $\preceq_T^l$-reducible to the language

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

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($|\alpha|$) for any given α 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 else.

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

8.1 On completeness of Figure 2. It is unknown if the Figure 2 is complete, i.e., if all the relativizable theorems of the form $K_1 \leq_T^p K_2$ are shown at Figure 2. Let us go through the following 15 assertions which should be proved to prove that Figure 2 is complete.

8.1.1. $\exists A\ R^A \cap \mathrm{Co}\text{-}\mathrm{R}^A \not\leq_T^{p,A} \oplus\mathrm{P}^A$. 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 $\exists A\ \mathrm{R}^A \cap \mathrm{Co}\text{-}\mathrm{R}^A \not\subseteq \oplus\mathrm{P}^A$. The closeness of the class $\oplus\mathrm{P}^A$ under $\leq_T^A$-reductions was proved in [T 89], the second theorem was proved in the previous section.

FIGURE 2. Turing reducibility between complexity classes.

8.1.2. $\mathrm{UP}^A \cap \text{Co-}\mathrm{UP}^A \not\leq_T^{p,A} \mathrm{BPP}^A$. This assertion is true and follows from the fact that the class BPP^A is downward closed under $\leq_T^{p,A}$-reductions (for all A). Indeed, in the previous section it was proved that there exists an oracle A such that $\mathrm{UP}^A \cap \text{Co-}\mathrm{UP}^A \not\subseteq \mathrm{BPP}^A$.

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

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

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

8.1.6. $\exists A$ $\Sigma_2^A \cap \Pi_2^A \not\leq_T^{p,A} \mathrm{IP}^A$. This assertion is true and is proved in §9.

8.1.7. $\exists A$ $\mathrm{BPP}^A \not\leq_T^{p,A} \mathrm{NP}^A$. This assertion is true and is proved in §9.

8.1.8. $\exists A$ $\oplus\mathrm{P}^A \not\leq_T^{p,A} \mathrm{PH}^A$. This assertion is true and 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 $\exists A$ $\oplus\mathrm{P}^A \not\subseteq \mathrm{PH}^A$.

8.1.9. $\exists A$ $\mathrm{AM}^A \not\leq_T^{p,A} \Sigma_2^A \cap \Pi_2^A$. This assertion is true and 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 $\exists A$ $\mathrm{AM}^A \not\subseteq \Sigma_2^A$.

8.1.10. $\exists A$ $\mathrm{AM}^A \cap \text{Co-}\mathrm{AM}^A \not\leq_T^{p,A} \mathrm{MA}^A$. This assertion was proved by the author together with An. A. Muchnik. The proof is presented in this section.

8.1.11. $\exists A$ $\oplus\mathrm{P}^A \not\leq_T^{p,A} \mathrm{IP}^A$. This assertion is true and is proved in §9.

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

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

8.1.14. $\exists A$ $\Sigma_k^A \not\leq_T^{p,A} \Sigma_k^A \cap \Pi_k^A$ $(k \geq 3)$. This assertion is true and 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 $\exists A$ $\Sigma_k^A \not\subseteq \Pi_k^A$.

8.1.15. $\exists A$ $\mathrm{PH}^A \not\leq_T^{p,A} \Sigma_k^A$ $(k \geq 1)$. This assertion is true and follows from the assertion 8.1.14.

8.2 Theorems. We prove now the assertions 8.1.3 and 8.1.10.

Theorem 10. (Joint work with An. A. Muchnik.) $\exists A$ $\mathrm{AM}^A \cap \text{Co-}\mathrm{AM}^A \not\leq_T^{p,A} \mathrm{MA}^A$.

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

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

By Theorem 3 it suffices to prove that F is not $\preceq_T^l$-reducible to the problem F_{MA}. Recall that $F_{\mathrm{MA}}(\beta) \neq *$ only if the norm of β is even and that for $\|\beta\| = 2k$

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

The following property holds for the separation problem F_{MA} as well as for all other particular problems G 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 n does not depend on the answers of the oracle and is equal to a polynomial of n 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 $|\alpha|$.

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

Assume that $F \preceq_T^l F_{\mathrm{MA}}$ via the 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 $\mathbf{B}^n$ into $\mathbf{B}^n$. Denote by $\bar{\varphi}$ the word of length 2^{2n} encoding the graph of φ. That is, for all $x, y \in \mathbf{B}^n$, $\bar{\varphi}(xy)$ 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 the set $\mathbf{B}^n$ into the set $\mathbf{B}^n$, as arguments of F.

Let $m = \mathrm{poly}(n)$ be the number of queries made by M to the oracle on input 2^{2n+1}. We shall define a binary sequence $b_1, \dots, b_m$, partial functions $\varphi, \psi : \mathbf{B}^n \to \mathbf{B}^n$, and total functions $f_0, g_0 : \mathbf{B}^n \to \mathbf{B}^n$ 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 the input $\bar{f}_0\bar{\psi}$ is equal to $b_1, \dots, b_m$ and the sequence of oracle answers to the queries made by (M, h) to the oracle F_{MA} during the work on the input $\bar{\varphi}\bar{g}_0$ is also 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 large enough n we shall get $|\mathrm{Dom}(\varphi)|$, $|\mathrm{Dom}(\psi)| < \frac{1}{3}2^n$. Obviously, we shall get a contradiction because $\langle M, h\rangle$ reduces F to F_{MA} and $F(\bar{f}_0\bar{\psi}) = 1$, $F(\bar{\varphi}\bar{g}_0) = 0$.

Denote by $2k$ the norm of queries made by the pair $\langle M, h\rangle$ to the oracle F_{MA} (i.e., 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 $2k$:

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

Obviously, G solves F_{MA}.

Take arbitrary functions $f, g : \mathbf{B}^n \to \mathbf{B}^n$. Run the machine M on the input 2^{2n+1} with the oracle $G(h(\cdot, \bar{f}\bar{g}))$. Denote by $e(f, g)$ 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 the fraction $\frac{\left|\{\langle f,g\rangle | e(f,g)=e_0\}\right|}{2^{n(2^n)}}$ is at least $\frac{1}{2^m}$. Denote the set $\{\langle f, g\rangle \mid e(f, g) = e_0\}$ by $\mathcal{K}$. 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. Denote those queries by $v_1, \dots, v_m$ (i.e., 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 $h(v, \alpha)$, ($\alpha \in \mathbf{F}_{2n+1}$, $u \in \mathbf{B}^{2k}$). Denote the bits of the word e_0 by $b_1, \dots, b_m$.

Denote by I the set $\{i \mid i \le m,\ b_i = 1\}$. We know that if $i \in I$, then for all $\langle f, g\rangle \in \mathcal{K}$ there exists $r_i \in \mathbf{B}^k$ such that $\mathrm{M}_{1/2} s \in \mathbf{B}^k\ P(\bar{f}\bar{g}, v_i, r_i s) = 1$. 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, $\frac{|\mathcal{K}'|}{2^{n(2^n)}} \ge \frac{1}{2^{km+m}}$. Denote the number $\frac{1}{2^{km+m}}$ by ε. We consider the set $\mathcal{K}'$ as a planar set of the area not smaller than ε. Obviously, there exists a vertical section of the set $\mathcal{K}'$ of length not smaller than ε and there exists a horizontal section of the set $\mathcal{K}'$ of length not smaller than ε.

That is, there exist functions f_0, g_0 and families of functions $\mathcal{F}'$ and $\mathcal{G}'$ such that $|\mathcal{F}'| \geq \varepsilon 2^{(n\cdot 2^n)}$, $|\mathcal{G}'| \geq \varepsilon 2^{(n\cdot 2^n)}$, $\{f_0\} \times \mathcal{G}' \subseteq \mathcal{K}'$, $\mathcal{F}' \times \{g_0\} \subseteq \mathcal{K}'$.

Define now a partial function $\varphi : \mathbf{B}^n \to \mathbf{B}^n$ and a family $\mathcal{F}$ consisting of (total) functions from $\mathbf{B}^n$ into $\mathbf{B}^n$. Assume that x, y are in $\mathbf{B}^n$. Denote by $\mathit{popularity}_{\mathcal{F}}(x,y)$ the fraction $\big|\{f \in \mathcal{F} \mid f(x) = y\}\big|/|\mathcal{F}|$. Set first $\varphi = \emptyset$, $\mathcal{F} = \mathcal{F}'$. Then, while there exists a pair $\langle x,y\rangle \in (\mathbf{B}^n \setminus \mathrm{Dom}(\varphi)) \times \mathbf{B}^n$ such that $\mathit{popularity}_{\mathcal{F}}\langle x,y\rangle \geq 2^{-n+1}$, pick 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) $\mathit{popularity}_{\mathcal{F}}(x,y) < 2^{-n+1}$ for all $\langle x,y\rangle \in (\mathbf{B}^n \setminus \mathrm{Dom}(\varphi)) \times \mathbf{B}^n$,
(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 value of the variables $\mathcal{F}$, φ, x, and y after ith iteration of the while-loop. Then

$$|\mathcal{F}_{i+1}|/\big|\{f : \mathbf{B}^n \to \mathbf{B}^n \mid f \text{ extends } \varphi_{i+1}\}\big| \geq 2|\mathcal{F}_i|/\big|\{f : \mathbf{B}^n \to \mathbf{B}^n \mid f \text{ extends } \varphi_i\}\big|$$

because

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

and

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

Since

$$|\mathcal{F}_{i+1}|/\big|\{f : \mathbf{B}^n \to \mathbf{B}^n \mid f \text{ extends } \varphi_{i+1}\}\big| \leq 1,$$

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 denote by $\mathcal{G}, \psi$ the resulting functions.

Let us prove that for all $i \leq m$,

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

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

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

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

for all the $f \in \mathcal{F}$. By 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

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

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

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

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

$$\sum_{x\in\mathbf{B}^n\setminus\mathrm{Dom}(\varphi)} w_{\varphi g_0}(x, f(x)) \ge \frac{1}{6}, \text{ therefore}$$
$$\frac{1}{|\mathcal{F}|} \sum_{f\in\mathcal{F},\ x\in\mathbf{B}^n\setminus\mathrm{Dom}(\varphi)} w_{\varphi g_0}(x, f(x)) \ge \frac{1}{6}.$$

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

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

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 $\big|\{s \in \mathbf{B}^k \mid P(\bar{f}\bar{g}_0, v_i, rs) = 0\}\big|/2^k$ is at most $1/2$ for all $r \in \mathbf{B}^k$ 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 $r \in \mathbf{B}^k$ such that

$$\mathrm{M}_{2/3} s \in \mathbf{B}^k\ P(\bar{\varphi}\bar{g}_0, v_i, rs) = 1.$$

Then just as it was done in the first case we can get a contradiction. In the same way we can prove that $\forall i \le m$,

$$F_{\mathrm{MA}}(h(v_i, \bar{f}_0\bar{\psi})) = b_i.$$

Theorem 11. *There is an oracle A such that* $\mathrm{FewP}^A \cap \text{Co-FewP}^A \not\le_T^{p,A} \mathrm{UP}^A$.

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

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

By the Theorem 1, it is sufficient to prove that $F \notin$ UPLOGS. Assume the contrary: suppose there exist a polynomial p and a polylogarithmic-time predicate P such that

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

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

1st case: $\exists r \in \mathbf{B}^{p(n)}\ P(\beta_0\gamma_0, r) = 1$.

Pick an expert r_0 such that $P(\beta_0\gamma_0, r_0) = 1$. If n is large enough, then there exists $u \in \mathbf{B}^n$, 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 $\beta_1 \in \mathbf{F}_n$ such that $\#_1(\beta_1) = 2$ and $\#\{r \in \mathbf{B}^{p(n)} : P(\beta_1\gamma_0, r) = 1\} \geq 2$. For a $u \in \mathbf{B}^n$ denote by β_0^u the word whose uth bit is 1 and other bits are 0. For all u we have $F(\beta_0^u\gamma_0) = 1$, therefore, $\forall u \in \mathbf{B}^n\ \exists! r \in \mathbf{B}^{p(n)}\ P(\beta_0^u\gamma_0, r) = 1$. Denote that r by r_u. Call the set of all $v \in \mathbf{B}^n$ such that the expert r_u queries '$\beta_0^u(v) =?$' the 1-*base of* u, and call the set of all $v \in \mathbf{B}^n$ such that the expert r_u queries '$\beta_0(v) =?$' the 0-*base of* u. Denote the bases of u by $B_1(u)$ and $B_0(u)$ respectively.

Let us prove that if n is large enough, then there exist $u_1, u_2 \in \mathbf{B}^n$ 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

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

Therefore, all three events do not happen with probability close to 1.

Fix u_1 and u_2 such that u_1 is not in $B_0(u_2) \cup B_1(u_2)$ and u_2 is not in $B_1(u_1)$. Let us define the word β_1 as follows: $\beta_1(u_1) = \beta_1(u_2) = 1$ and $\beta_1(v) = 0$ for $v \neq u_1, u_2$. Then $\beta_1\gamma_0 \in \mathrm{D}(F)$ and $P(\beta_1\gamma_0, r_{u_1}) = P(\beta_1\gamma_0, r_{u_2}) = 1$ (since $u_2 \notin B_1(u_1)$, $u_1 \notin B_1(u_2)$). We have $r_{u_1} \neq r_{u_2}$ 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 $\preceq_T^l$-reducibility we have

$$(*) \qquad \forall \alpha \in \mathrm{D}(F)\ \forall e \in \mathbf{B}^*\ \#_1(f(e, \alpha)) \in \{0, 1\}$$

Fix $n \in \mathbf{N}$ and set $\alpha_0 = 0^{2^{n+1}}$. Denote by D_1 the set $\{\alpha \in \mathbf{F}_{n+1} : \#_1(\alpha) = 1\}$. Evidently, $D_1 \subseteq \mathrm{D}(F)$. We construct a set $U \subseteq \mathbf{B}^{n+1}$ having at most 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 for queries to oracle F_{UP} made by $\langle M, f\rangle$ during the work on input α is the same.

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

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

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

Let us fix any i not exceeding m and construct U_i. By 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 of $|\alpha|$.

Consider two cases.

1st case: $b_i = 1$, that is, $\#_1(f(v_i, \alpha_0)) \geq 1$. Pick a word r such that $f(v_i, \alpha_0)(r) = 1$. Include in U_i all the words $u \in \mathbf{B}^n$ such that N asks '$\alpha_0(u) =$?' during the computation on input $\langle \alpha_0, v_i, r \rangle$. Then $\#_1(f(v_i, \alpha)) \geq 1$ for all $\alpha \in \mathbf{F}_{n+1}$ being equal to zero on all the elements of U_i. By $(*)$, this means that $\#_1(f(v_i, \alpha)) = 1$ for all $\alpha \in D_1$ being 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 during the proof of the first part. Let us prove that the set $V = \{u \in \mathbf{B}^n \mid \#_1(f(v_i, \beta_0^u \gamma_0)) = 1\}$ 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 $u \in V$ denote by r_u the word r such that rth bit of word $f(v_i, \beta_0^u \gamma_0)$ is 1. Denote by $B_0(u)$ $[B_1(u)]$ the set of all v such that N queries '$\alpha_0(v) =$?' ['$\beta_0^u \gamma_0(v) =$?'] at some moment during the computation on the input $\langle \alpha_0, v_i, r_u \rangle$ $[\langle \beta_0^u \gamma_0, v_i, r_u \rangle]$. Then $|B_0(u)|, |B_1(u)| \leq q(n)$ for all $u \in V$. Take random independent u_1, u_2 being 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 $\beta_1 \in \mathrm{D}(F)$ such that $\#_1(f(v_i, \beta_1 \gamma_0)) \geq 2$, which contradicts to $(*)$.

Likewise we can construct a set V' having $\mathrm{poly}(n)$ elements such that $\#_1(f(v_i, \beta_0 \gamma_0^u)) = 1$ for all $u \in \mathbf{B}^n \setminus V'$. Set $U_i = V \cup V'$.

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 obtained contradiction proves the theorem.

9. Complete languages in particular complexity classes

It is known that the following classes

$$\mathrm{P}^A,\ \mathrm{NP}^A,\ \text{Co-NP}^A,\ \Sigma_k^A,\ \Pi_k^A,\ \mathrm{PSPACE}^A,\ \oplus\mathrm{P}^A,\ \mathrm{PP}^A. \tag{9.1}$$

have $\leq_m^p$-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 exists a class K^A in the list (9.1) 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 (9.1) such that $K_1^A \leq_T^p K^A \subseteq K_2^A$.

9.1 Are the rules (1) and (2) complete? It is unknown if all true assertions of the form "K_2^A is $\le_m^{p,A}$-hard [$\le_T^{p,A}$-hard] for the class K_1^A for all A", where K_1^A and K_2^A are classes shown at Figure 1, can be obtained by the rules (1) and (2). We have proved some assertions which are necessary to prove in order to get positive answer to the above question. Indeed, if K_2^A is $\le_m^{p,A}$-hard for the class K_1^A, then $K_1^A \subseteq K_2^A$ (since all the classes under consideration are downward closed under $\le_m^{p,A}$-reductions). Therefore, if we have proved that $\exists A\ K_1^A \not\subseteq K_2^A$, then we have also proved that $\exists A\ K_2^A$ is not $\le_m^{p,A}$-hard for the class K_1^A. Analogously, if we have proved that $\exists A\ K_1^A \not\le_T^{p,A} K_2^A$, then we have also proved that $\exists A\ K_2^A$ is not $\le_T^{p,A}$-hard for the class K_1^A. Let us go through remaining assertions which should be proved to obtain the positive answer to the above question. We divide the list of those assertions into two parts. The first part contains all the assertions of the form "$\exists A\ K_2^A$ is not $\le_T^{p,A}$-hard for the class K_1^A" such that it is unknown if $\exists A\ K_2^A \le_T^{p,A} K_1^A$, the second part contains all the remaining assertions.

The first part of the list.

1. $\exists A$ PP^A is not $\le_T^{p,A}$-hard for the class IP$^A \cap$ Co-IP^A. It is unknown whether this is true. Since PP^A has $\le_m^p$-complete language, this assertion is equivalent to the assertion $\exists A$ IP$^A \cap$ Co-IP$^A \not\le_T^{p,A}$ PP^A.

2. $\exists A\ \Sigma_k^A$ is not $\le_T^{p,A}$-hard for the class $\Sigma_{k+1}^A \cap \Pi_{k+1}^A$. It is unknown whether this is true. Since Σ_k^A has a $\le_m^p$-complete language, this assertion is equivalent to the assertion $\exists A\ \Sigma_{k+1}^A \cap \Pi_{k+1}^A \not\le_T^{p,A} \Sigma_k^A$.

The second part of the list.

1. $\exists A\ \Sigma_k^A \cap \Pi_k^A$ is not $\le_T^{p,A}$-hard for the class $\Sigma_k^A \cap \Pi_k^A$ $(k \ge 3)$. It is unknown whether this is true or not.

2. $\exists A$ IP^A is not $\le_T^{p,A}$-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. $\exists A$ IP$^A \cap$ Co-IP^A is not $\le_T^{p,A}$-hard for the class R$^A \cap$ Co-R^A. This assertion is true and was proved in [HJV 92].

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

5. $\exists A\ \Sigma_2^A \cap \Pi_2^A$ is not $\le_T^{p,A}$-hard for the class BPP^A. It is unknown whether this is true or not.

6. $\exists A$ Few^A is not $\le_T^{p,A}$-hard for the class UP$^A \cap$Co-UP^A. This assertion is true. In the paper [HJV 92], it was proved that there exists an oracle A such that the class FewP^A is not $\le_T^{p,A}$-hard for the class UP$^A \cap$ Co-UP^A. In the present paper we prove that Few^A is not $\le_T^{p,A}$-hard for the class UP$^A \cap$ Co-UP^A for some A.

7. $\exists A\ \Sigma_2^A \cap \Pi_2^A$ is not $\le_T^{p,A}$-hard for the class Few^A. It is unknown whether this is true or not.

The listed assertions are "maximal" possible assertions of the form "$\exists A\ K_2^A$ is not $\le_T^{p,A}$-hard for the class K_1^A" (this means that if we replace the class K_1^A by some lower class in the Figure 1 or replace the class K_2^A by some upper class in the Figure 1, then the assertion becomes false). Let us give other assertions of this form proven earlier. In the paper [S 82] it is proved that $\exists A$ R^A has no $\le_m^{p,A}$-complete language, this theorem is strengthened in the paper [HJV 92] to prove that $\exists A$ R^A has no $\le_T^{p,A}$-complete language; in the paper [S 82] it is proved that $\exists A$ $\mathrm{NP}^A \cap \text{Co-NP}^A$ has no $\le_m^{p,A}$-complete language; in the paper [HH 88] it is proved that $\exists A$ BPP^A has no $\le_m^{p,A}$-complete language, in the papers [A-S 86], [G 83], [HI 85] both results are strengthened to prove that $\exists A$ $\mathrm{NP}^A \cap \text{Co-NP}^A$ has no $\le_T^{p,A}$-complete language and $\exists A$ BPP^A has no $\le_T^{p,A}$-complete language; in the paper [HH 88] it is proved that $\exists A$ UP^A has no $\le_m^{p,A}$-complete language, this theorem is strengthened in the paper [HJV 92] to prove that $\exists A$ UP^A has no $\le_T^{p,A}$-complete language.

9.2 Theorems on non-completeness. Let us turn to the proofs. We use the following lemma.

Lemma 2. *If F and G are nondegenerate separation problems such that*

(9.1) $F \notin$ n.u.PLOGS *and*

(9.2) n.u.LOGS$(G) =$ n.u.PLOG,

then there exists an oracle A such that the class $\mathrm{POLY}^A(G)$ *is not $\le_T^{p,A}$-hard for the class* $\mathrm{POLY}^A(F)$.

Proof. By the Theorem 4, it suffices to prove that the separation problem F is $\preceq_T^l$-reducible to no language in the class LOG(G). Assume that there exists a language $H \in$ 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 3, and the following theorem.

Theorem 12. n.u.IPLOG $\cap$ Co-n.u.IPLOG = n.u.PLOG.

We omit the proof of Theorem 12 because its proof is an easy generalization of Nisan's result (see [N 89]) n.u.BPPLOG = n.u.PLOG. Independently, Theorem 12 was proved by the author it the first version of the present paper.

The assertion 6 can be proved in similar way. Formally, we cannot use Lemma 2 because we do not know whether the manifold Few^A is regular.

Theorem 13. *If F is a nondegenerate separation problem and F is not in* n.u.PLOGS, *then there exists an oracle A such that the class* Few^A *is not $\le_T^{p,A}$-hard for the class* $\mathrm{POLY}^A(F)$.

Proof. We can apply the diagonal construction used in the proof of Theorem 3. It is clear that it suffices to prove the following lemma.

Lemma 3. *Let $P(\alpha, r)$ be a predicate being defined on the set $\mathbf{F} \times \mathbf{B}^*$ and computable in* poly$(\|\alpha\|, |r|)$ *queries to α and let $p(n)$, $q(n)$ be polynomials such that* $\forall \alpha \in \mathbf{F}$ $\left|\{r \in \mathbf{B}^{p(\|\alpha\|)} : P(\alpha, r) = 1\}\right| \le q(\|\alpha\|)$. *Then the function*

$$f(\alpha) = \left|\{r \in \mathbf{B}^{p(\|\alpha\|)} : \ P(\alpha, r) = 1\}\right|$$

is non-uniformly polylogarithmic.

Proof. Let us fix a polynomial $s(\|\alpha\|, |r|)$ and a machine M such that M computes $P(\alpha, r)$ in time $s(\|\alpha\|, |r|)$ for any given $\langle \alpha, r \rangle$. Let n be an integer. Denote $p(n)$ by

m and $s(n,m)$ by k. Let us call words in the set $\mathbf{B}^m$ *experts*. We say that an expert r *accepts* $\alpha \in \mathbf{F}_n$ if $P(\alpha, r) = 1$. For any $\alpha \in \mathbf{F}_n$ let $f(\alpha) = \{r \in \mathbf{B}^m \mid r \text{ accepts } \alpha\}$.

It is sufficient to prove that the function $f(\alpha)$ can can be computed in $q(n)k^2$ queries.

Call any partial function $\varphi : \mathbf{B}^n \to \mathbf{B}$ a *segment*. Two segments are *consistent* if they have common extension. Any expert for a given $\alpha : \mathbf{B}^n \to \mathbf{B}$ queries the value of α on k arguments, say $u_1, \dots, u_k$. 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 a *certificate of expert* r. A *certificate* is a certificate of some expert.

We find all experts accepting α for any given $\alpha \in \mathbf{F}_n$ as follows. For any subset U of $\mathbf{B}^n$ denote by $\Phi_U(\alpha)$ the set of all certificates having the same value on elements of U as α has. 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 $\Psi = \{\varphi_1, \dots, \varphi_j\}$ of $\Phi_U(\alpha)$ such that the sets $\mathrm{Dom}(\varphi_1) \setminus U, \dots, \mathrm{Dom}(\varphi_j) \setminus U$ are pairwise disjoint. Then $j \le q(n)$ because there exists $\beta \in \mathbf{F}_n$ being consistent with all certificates in Ψ and $\varphi_1, \dots, \varphi_j$ 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 $V = (\mathrm{Dom}(\varphi_1) \cup \dots \cup \mathrm{Dom}(\varphi_j)) \setminus U$. 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 $|\mathrm{Dom}(\varphi) \setminus U|$ is decreased for any certificate $\varphi \in \Phi_U(\alpha) \setminus \Psi$ and $\mathrm{Dom}(\varphi) \setminus U$ becomes empty for any certificate $\varphi \in \Psi$ after this setting. 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.IPLOG $\supseteq$ n.u.NPLOG $\supset$■ n.u.PLOG.

Theorem 14. (Joint work with An. A. Muchnik). *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 yet unproved theorems from the previous section.

Theorem 15. $\exists A\ \mathrm{BPP}^A \not\le_T^{p,A} \mathrm{NP}^A$.

Theorem 16. $\exists A\ \oplus \mathrm{P}^A \not\le_T^{p,A} \mathrm{IP}^A$.

Theorem 17. $\exists A\ \Sigma_2^A \cap \Pi_2^A \not\le_T^{p,A} \mathrm{IP}^A$.

Proofs of Theorems 14–17. In fact, Theorem 15 follows from Theorem 14 because the class NP^A has a $\le_m^p$-complete language and $\mathrm{NP}^A \subseteq \mathrm{IP}^A$. Nevertheless we prove first Theorem 15. By Theorem 4 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 n. Denote by m 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 $\mathbf{F}_n$. Run the machine M supplied with the oracle $F_{\mathrm{NP}}(f(\cdot, \alpha))$ on the input 2^n. Denote by $e(\alpha)$ the sequence of oracle answers received by M in that computation ($e(\alpha) \in \mathbf{B}^m$). Take an $\alpha \in \mathbf{F}_n$ having lexicographical greatest $e(\alpha)$, denote that α by α_0. Denote $e(\alpha_0)$ by $e_0 = b_1^0 \cdots b_m^0$, and denote the queries of M to the oracle $F_{\mathrm{NP}}(f(\cdot, \alpha_0))$ by $v_1, \dots, v_m$ (more precisely, the queries are '$F_{\mathrm{NP}}(f(v_i, \alpha_0)) =?$'). Let I be the set of all the indices $i \leq k$ 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 weak computation of the function $f(v_i, \alpha)$ for $\alpha \in \mathbf{F}_n$, $i \leq m$. Obviously, for any $i \in I$ there exists a set $U_i \subseteq \mathbf{B}^n$ 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, $|U| \leq mq(n) = \mathrm{poly}(n)$. We have $F_{\mathrm{NP}}(f(v_i, \alpha)) = 1$ for all $i \leq m$ such that $b_i^0 = 1$ and for all $\alpha \in \mathbf{F}_n$ 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 $\alpha \in \mathbf{F}_n$ having the same values on all the words in U as α_0 has. Assume the contrary. Let α be a counterexample. Let $b_1 \cdots b_m$ be the bits of $e(\alpha)$. Let i be the least number such that $b_i \neq b_i^0$. Then, since the word e_0 is the lexicographical greatest word among the word of the form $e(\alpha)$, $\alpha \in \mathbf{F}_n$, 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 $b_1^0 \cdots b_{i-1}^0 = b_1 \cdots b_{i-1}$, 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, $F_{\mathrm{NP}}(f(v_i, \alpha)) = b_i$. 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 all the elements of $\mathbf{B}^n \setminus U$. We have $F_{\mathrm{BPP}}(\alpha) \not\leq \langle M, f \rangle^{F_{\mathrm{NP}}}(\alpha_0) = \langle M, f \rangle^{F_{\mathrm{NP}}}(\alpha)$. Theorem 15 is proved.

Let us prove Theorem 16. Since PARITY is a language, by Theorem 3, it suffices to prove that PARITY $\not\preceq_T^l$ IPLOG. Assume that PARITY is $\preceq_T^l$-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 polylogarithmic Verifier V for F. For each $i \leq m$ 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 $r = \mathrm{poly}(n)$ is an upper bound for the number of queries of the form '$\alpha_0(x) =?$', where x is in $\mathbf{B}^n$, made by N in computations on inputs of the form $\langle \alpha_0, v_i, t \rangle$, where $|t| = \|f(v_i, \alpha_0)\|$. Denote $f(v_i, \alpha_0)$ by β_0^i. 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 dialogue with P_i on input β_0^i. Let x be in $\mathbf{B}^n$. Denote by $w_{\alpha_0}^i(x)$ the probability of the event "there exists $t \in \mathbf{B}^{\|\beta_0^i\|}$ such that V queries '$\beta_0^i(t) =?$' in the dialogue with P_i on input α_0 and N queries '$\alpha_0(x) =?$' during the computation on the input $\langle \alpha_0, v_i, t \rangle$". Then $\sum_{i : b_i^0 = 1} \sum_{x \in \mathbf{B}^n} w_{\alpha_0}^i(x) \leq msr$, therefore, there exists $x_0 \in \mathbf{B}^n$ such that $\sum_{i : b_i^0 = 1} w_{\alpha_0}^i(x_0) \leq msr/2^n < 1/3$ (if n is sufficiently large). Change the x_0th bit of α_0 and denote the resulting word by α. 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 $e(\alpha) \neq e(\alpha_0)$. Denote by $b_1 \cdots b_m$ the bits of $e(\alpha)$. Take the least i such that $b_i \neq b_i^0$. 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,

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

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

Let us prove Theorem 14. 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: $F_{\text{BPP}} \preceq^l_T F \in$ IPLOG. We use all notations 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 $\{\alpha\in\mathbf{F}_n \mid e(\alpha)=e(\alpha_0)\}$ 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 $x_0\in\mathbf{B}^n$ such that $\sum_{i:b^0_i=1} w^i_{\alpha_0}(x_0)\leq\frac{msr}{(1/3)2^n}<1/3$ and $\alpha_0(x_0)=1$. Define the word α as follows: $\alpha(x_0)=0$, $\alpha(x)=\alpha_0(x)$ for $x\neq x_0$. 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 with the choice of α_0.

Let us prove Theorem 17. Let α be a partial function from $\mathbf{B}^n$ into $\mathbf{B}^n$. Denote by $\bar\alpha$ the word encoding the graph of α ($\bar\alpha\in\mathbf{B}^{2^{2n}}$). Consider the separation problem

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

Denote by E_n the set $\{\gamma\in\mathbf{F}_{2n+1}\mid F(\gamma)\neq *\}$.

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 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 $x,y\in\mathbf{B}^n$, $\gamma(0xy)=\bar\alpha_0(xy)$, $\gamma(1xy)=\bar\beta_0(xy)$. For an i such that $b^0_i=1$, define the weight $w^i_{\alpha_0\beta_0}(u)$ of word $u\in\mathbf{B}^{2n+1}$ as follows: $w^i_{\alpha_0\beta_0}(u)$ is equal to the probability of the event "there exists $t\in\mathbf{B}^{\|f(v_i,\alpha_0)\|}$ 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 $x_0\in\mathbf{B}^n$ 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 $y_1\in\mathbf{B}^n$ 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 obtained contradiction proves the theorem.

References

[A 83] N. Ajtai, Σ_1^1*-formulae on finite structures*, Ann. Pure Appl. Logic **24** (1983), 1–48.

[AGH 86] W. Aiello, S. Goldwasser, J. Håstad, *On the Power of Interaction*, Proc. 26th Annual IEEE Symp. on Foundations of Comp. Scie., 1986, pp. 368–379.

[A-S 86] K. Ambos-Spies, *A Note on Complete Problems for Complexity Classes*, Inf. Proc. Lett. **23** (1986), 227–230.

[B 85] L. Babai, *Trading Group Theory for Randomness Proc. 17th Annual ACM Symp. on Theory of Computing.* (1985), 421–429.

[BG 81] C. H. Bennett, J. Gill, *Relative to a random oracle* $\mathrm{P}^A \neq \mathrm{NP}^A \neq \text{Co-NP}^A$ *with probability 1*, SIAM Journ. on Computing **10** (1981), 96–113.

[BGS 75] T. Baker, J. Gill and R. Solovay, *Relativization of* $\mathrm{P} \overset{?}{=} \mathrm{NP}$ *Question*, SIAM Journ. on Computing **4** (1975), 431–442.

[CH 90] J. Cai, L. Hemachandra, *On the Power of Parity Polynomial Time*, Mathem. Systems Theory **23** (1990), no. 2, 95–106.

[FS 88] L. Fortnow, M. Sipser, *Are There Interactive Protocols for* Co-NP*-Languages?* Inf. Proc. Lett. **28** (1988), 249–251.

[FSS 84] N. Furst, J. Saxe, M. Sipser, *Parity, circuits and the polynomial time hierarchy*, Mathem. Systems Theory **17** (1984), 13–27.

[G 83] Y. Gurevich, *Algebras of Feasible Functions*, Proc. 24th Annual IEEE Symp. on Foundations of Comp. Scie., 1983, pp. 210–214.

[GMR 85] S. Goldwasser, S. Micali, C. Rackoff, *Knowledge Complexity of Interactive Proofs*, Proc. of Ann. ACM Proc. 17th Annual ACM Symp. on Theory of Computing, 1985, pp. 291–305.

[GMR 89] S. Goldwasser, S. Micali, C. Rackoff The Knowledge Complexity of Interactive Proof-Systems, SIAM Journ. on Computing **18** (1989), 186–208.

[GS 86] S. Goldwasser, M. Sipser, *Private Coins Versus Public Coins in Interactive Proof Systems*, Proc. 18th Annual ACM Symp. on Theory of Computing, 1986, pp. 59–68.

[H 86] J. Håstad, *Almost Optimal lower bounds for small depth circuits*, Proc. 18th Annual ACM Symp. on Theory of Computing, 1986, pp. 6–20.

[HH 88] J. Hartmanis, L. Hemachandra, *Complexity classes without machines: On complete languages for* UP, Lect. Notes in Comp. Scie., vol. 226, 1986, pp. 123–135; Theor. Comp. Scie. **58** (1988), 129–142.

[HI 85] J. Hartmanis, N.Immerman, *On complete problems for* NP ∩ Co-NP, Intern. Colloq. on Automata, Languages and Programming, 1985; Lect. Notes Comp. Scie., vol. 194, 1985, pp. 250–259.

[HJV 92] L. Hemachandra, S. Jain, N. Vereshchagin, *Banishing Robust Turing Completeness*, Proc. Logic at TVER' 92, Symp. on Logical Found. of Comp. Scie.; Lect. Notes in Comp. Scie., vol. 620, 1992, pp. 186–197.

[KSTT 89] J. Köbler, U. Shöning, S. Toda, J. Tóran, *Turing machines with few accepting computations and low sets for* PP, Proc. of Annual Conf. on Structure in Complexity Theory, 1989, pp. 208–216.

[LFK 90] C. Lund, L. Fortnow, H. Karloff, N. Nisan, *The Polynomial Time Hierarchy has Interactive Proofs*, Proc. 31th Annual IEEE Symp. on Foundations of Comp. Scie., 1990, pp. 2–10.

[MP 88] M. Minsky, S. Papert, *Perceptrons*, MIT Press, Cambridge, MA:, 1988, (Expanded edition. The first edition appeared in 1969.).

[N 89] N. Nisan, *Probabilistic versus Deterministic Decision Trees and CREW PRAM Complexity*, Proc. 21th Annual ACM Symp. on Theory of Computing, 1989, pp. 327–335.

[Sa 89] M. Santha, *Relativized Arthur-Merlin versus Merlin-Arthur Games*, Inform. and Comput. **80** (1989), 44–49.

[Sh 90] A. Shamir IP = PSPACE, Proc. 31th Annual IEEE Symp. on Foundations of Comp. Scie., 1990, pp. 11–15.

[S 82] M. Sipser, *On Relativizations and the Existence of Complete Sets*, Proc. 9th Intern. Colloq. on Automata, Languages and Programming, 1982; Lect. Notes Comp. Scie., vol. 140, 1982, pp. 523–531.

[S 83] M. Sipser, *A complexity Theoretic Approach for Randomness*, Proc. 15th Annual ACM Symp. on Theory of Computing., 1983, pp. 330–335.

[T 89] S. Toda, *On the computational power of* PP *and* $\oplus$P, Proc. 30th Annual IEEE Symp. on Foundations of Comp. Scie., 1989, pp. 514–519.

[V 92] N. K. Vereshchagin, *On the Power of* PP, Proc. 7th Annual IEEE Conf. on Structure in Complexity Theory, 1992, pp. 138–143.

[Y 85] A. Yao, *Separating the Polynomial Hierarchy by Oracles*, Proc. 26th Annual IEEE Symp. on Foundations of Comp. Scie., 1985, pp. 1–10.

[BCS 92] D. P. Bovet, P. Crescenzi, R. Silvestri, *A uniform approach to define complexity classes*, Theor. Comp. Scie. **104** (1992), 263–283.

Translated by THE AUTHOR

Institute of New Technologies, 11 Kirovogradskaja Street, Moscow, Russia 113587

E-mail address: ver@math.imu.msk.su