A General Method to Construct Oracles Realizing Given Relationships between Complexity Classes

Andrei A. Muchnik*†
amuchnik@int.glas.apc.org

Nikolai K. Vereshchagin*‡§¶
ver@ium.ac.msk.su ver@math.math.msu.su

Abstract

We present a method to prove oracle theorems of the following type. Let $K_1, \ldots, K_{2n}$ and $L_1, \ldots, L_{2m}$ be complexity classes. The method provides a general framework for constructing an oracle A such that $K_{2i-1}^A \neq K_{2i}^A$ for $i = 1, \ldots, n$ and $L_{2j-1}^A \neq L_{2j}^A$ for $j = 1, \ldots, m$. Using that method we obtain several results of this kind. The hardest of them is the existence of an oracle A such that $\mathbf{P}^A \neq \mathbf{NP}^A$, $\mathbf{P}^A = \mathbf{BPP}^A$ and both $\mathbf{Co\text{-}NP}^A$-sets and $\mathbf{NP}^A$-sets are $\mathbf{P}^A$-separable. We exhibit also two theorems that cannot be proved by that method.

1 Introduction

When people realized that $\mathbf{P} \neq \mathbf{NP}$ is likely true but hard to prove (and impossible to prove by relativizable arguments [BGS 75]), they began to prove interesting theorems under $\mathbf{P} \neq \mathbf{NP}$ hypothesis. Interesting problems of this kind arise in cryptography, where the reliability of all known protocols is based on complexity assumptions even stronger than $\mathbf{P} \neq \mathbf{NP}$. A very challenging problem is to construct cryptographic protocols which are provably secure if $\mathbf{P} \neq \mathbf{NP}$. In the present paper we ascertain that many complexity assertions cannot be proved by relativizable arguments even under the $\mathbf{P} \neq \mathbf{NP}$ assumption and under stronger assumptions. In other words, we construct oracles relative to which certain Boolean combinations of the $\mathbf{P} \neq \mathbf{NP}$ assumption and stronger ones are true. Moreover, we investigate a general powerful method to

*This research was in part supported by a grant from the American Mathematical Society.

†Institute of New Technologies, Nizhnjaa Radishchevskaja 10, Moscow, Russia 109004

‡This research was in part supported by the grant MQT000 from the International Science Foundation, a NAS/NRC COBASE grant, and NSF grant CCR-8957604. Work done in part while visiting the University of Rochester.

§Dept. of Mathematical Logic, Moscow State University, Moscow 119899, Russia

¶All the results in this paper are joint results by both authors. The text was written by the second author.

prove such theorems and in the last section we present some theorems which cannot be proved by that method.

Many results of this sort (when an oracle is constructed under which some Boolean combination of complexity assertions is true) have appeared in literature. The following results among them deal with the classes considered in the present paper. Rackoff in [R 78] constructed oracles A and B such that $\mathbf{P}^A = \mathbf{R}^A \neq \mathbf{NP}^A$ and $\mathbf{P}^B \neq \mathbf{R}^B = \mathbf{NP}^B$. In [BGS 75], it was proved that $\mathbf{P} = \mathbf{NP} \cap \mathbf{Co\text{-}NP} \neq \mathbf{NP}$ under some oracle.

In the present paper we prove, for example, that there exists an oracle under which $\mathbf{P} \neq \mathbf{NP}$ and **NP**-sets are separable, thus we solve the problem left open in [GS 88].[1] This implies that reliability of all the cryptographic schemes based on the existence of one-way functions cannot be derived from $\mathbf{P} \neq \mathbf{NP}$ by relativizable arguments (since one-way functions do not exist if **NP**-sets are separable). Moreover, we show that one cannot prove using relativizable arguments that **NP**-sets are inseparable even under hypothesis that both **Co-NP**-sets are inseparable and $\mathbf{P} \neq \mathbf{R}$. The strongest result of our paper states that there exists an oracle under which $\mathbf{P} \neq \mathbf{NP}$ and **NP**-sets are separable and **Co-NP**-sets are separable and $\mathbf{P} = \mathbf{BPP}$. In other words, it is impossible to prove by relativizable arguments even the disjunction "**NP**-sets are inseparable or **Co-NP**-sets are inseparable or $\mathbf{P} \neq \mathbf{BPP}$" under the $\mathbf{P} \neq \mathbf{NP}$ hypothesis.

The method used in the present paper goes back to [BGS 75]. We call it "the universum method". In the present paper we refine that method and apply it to prove the existence of oracles relative to which certain Boolean combinations of the assertions $\mathbf{P} = \mathbf{NP}$, $\mathbf{P} = \mathbf{R}$, $\mathbf{P} = \mathbf{BPP}$, $\mathbf{P} = \mathbf{NP} \cap \mathbf{Co\text{-}NP}$, $\mathbf{P} = \mathbf{R} \cap \mathbf{Co\text{-}R}$, "**NP**-sets are **P**-separable", and "**Co-NP**-sets are **P**-separable" hold (we are successful to construct oracles for 13 of 17 possible combinations, thus 4 problems of this kind remain unsolved).

Roughly speaking, the method works as follows. Suppose we want to prove that there exists an oracle A such that $\mathbf{P}^A \neq \mathbf{BPP}^A$ and $\mathbf{P}^A = \mathbf{R}^A$. First, we define a subset V (called *the universum*) of the set of all oracles. Second, we choose a sufficiently powerful oracle H (in all known applications we can take any **PSPACE**-complete set as H). Third, we consider machines having two oracles: the oracle H and a varying oracle B ranging over V. (Thus, every machine of this type accepts a subset of $\mathbf{B}^* \times V$, where $\mathbf{B} = \{0, 1\}$ is the input alphabet.) Finally, we prove that there exists a BPP-machine of this type which recognizes a subset of $\mathbf{B}^* \times V$ recognizable by no P-machine of this type and prove that for any R-machine of this type there exists a P-machine of this type recognizing the same subset of $\mathbf{B}^* \times V$.

Another general method close to ours was presented in the paper [FFKL 93]. The extension of that method was applied by Fortnow and Rogers in [FR 94] to prove the existence of oracles relative to which certain Boolean combinations of the assertions $\mathbf{P} = \mathbf{NP}$, $\mathbf{P} = \mathbf{UP}$, $\mathbf{P} = \mathbf{NP} \cap \mathbf{Co\text{-}NP}$, "**NP**-sets

[1] Independently, this was proven in [FR 94].

are **P**-separable", and "**Co-NP**-sets are **P**-separable" hold. They succeeded in constructing oracles for all possible combinations.

In a sense our method (as well as the method of [FFKL 93]) is a special case of the forcing method (see Section 5.2.1). In Section 6, we prove two negative results that can be interpreted as that both methods fail to prove the following two theorems: the theorem that $\mathbf{P} \neq \mathbf{R} = \mathbf{PSPACE}$ under some oracle [R 78] and the theorem proven by Ko in [Ko 89] that $\mathbf{P} = \mathbf{NP} \neq \mathbf{PSPACE}$ under some oracle.

In the next section we introduce some notation. In Section 3, we prove a sample theorem in full detail to exhibit all main ideas of the method in the simplest situation. In Section 4 we obtain the main results applying our method. We give there also a simple proof of the result by [HI 86] that there exists an oracle relative to which $\mathbf{P} \neq \mathbf{NP} \cap \mathbf{Co\text{-}NP} \neq \mathbf{NP}$ and the class $\mathbf{NP} \cap \mathbf{Co\text{-}NP}$ has a complete problem. In Section 5, we formalize the universum method and formulate two general theorems on the method. This makes possible to formulate what means that the method fails to prove the two above cited theorems. All theorems in Sections 5 and 6 are presented without proofs. The proofs can be found in [MV 94].

2 Preliminaries

We restrict ourselves to binary alphabet $\mathbf{B} = \{0, 1\}$. The set of all binary words is denoted by $\mathbf{B}^*$. The set of all functions from $\mathbf{B}^*$ into $\mathbf{B}$ is denoted by Ω. A language $L \subset \mathbf{B}^*$ is identified with its characteristic function. We say that a language L *separates* languages L_0 and L_1 if $L_0(x) = 1 \Rightarrow L(x) = 0$ and $L_1(x) = 1 \Rightarrow L(x) = 1$ for any $x \in \mathbf{B}^*$. Call any language in a class C of languages a *C-set*. An *oracle* is an element of Ω.

Denote by $|y|$ the length of word y.

Let A be an oracle.

A language L belongs to the class $\mathbf{P}^A$ if there is a polynomial time Turing machine M^A with oracle A that recognizes L.

A language L belongs to the class $\mathbf{NP}^A$ if there is a polynomial time nondeterministic Turing machine N^A with oracle A that accepts L.

A language L belongs to the class $\mathbf{Co\text{-}NP}^A$ if $\mathbf{B}^* \setminus L$ is in $\mathbf{NP}^A$.

A language L belongs to the class $\mathbf{BPP}^A$ if there is a polynomial time probabilistic Turing machine M^A with oracle A such that $\mathrm{Prob}[M^A(x) = 1] > 2/3$ for all $x \in L$ and $\mathrm{Prob}[M^A(x) = 1] < 1/3$ for all $x \notin L$.

A language L belongs to the class $\mathbf{R}^A$ if there is a polynomial time probabilistic Turing machine M^A with oracle A such that $\mathrm{Prob}[M^A(x) = 1] > 1/2$ for all $x \in L$ and $\mathrm{Prob}[M^A(x) = 1] = 0$ for all $x \notin L$.

A language L belongs to the class $\mathbf{PSPACE}^A$ if there is a polynomial space Turing machine M^A with oracle A that recognizes L.

We say that $\mathbf{NP}^A$-*sets* ($\mathbf{Co\text{-}NP}^A$-*sets*) are *separable* if any two disjoint $\mathbf{NP}^A$-sets ($\mathbf{Co\text{-}NP}^A$-sets) are separable by a $\mathbf{P}^A$-set. If this is not the case, then we say that $\mathbf{NP}^A$-sets ($\mathbf{Co\text{-}NP}^A$-sets) are *inseparable*.

Let $\mathbf{N}$ denote the set of nonnegative integers. Denote by $\mathbf{B}^n$ the set of all binary words of length n. Denote by $\mathbf{B}^{\leq n}$ ($\mathbf{B}^{\geq n}$) the set of all binary words of length at most n (at least n, respectively).

Denote by $f|D$ the restriction of function f to set D.

Let us define the binary operation $\oplus$ on languages as follows: $B \oplus C = \{0u \mid u \in B\} \cup \{1v \mid v \in C\}$ (we denote by xy the concatenation of words x and y).

An *interval* is a set of oracles having the form

$$\Gamma(\varphi) = \{A \in \Omega \mid \forall y \in \mathrm{Dom}(\varphi)\ A(y) = \varphi(y)\},$$

where φ is a function from a finite subset of $\mathbf{B}^*$ into $\mathbf{B}$, and $\mathrm{Dom}(\varphi)$ denotes the domain of φ.

For a finite set $M \subset \mathbf{B}^*$ denote by $maxlength(M)$ the $\max_{y \in M} |y|$ and denote by $|M|$ the number of elements in M.

Let P be a deterministic oracle machine. Denote by $Query_P^B(x, B \oplus H)$ the set of all $y \in \mathbf{B}^*$ such that P asks '$B(y) =$?' during the computation on input x with oracle $B \oplus H$. Let N be a nondeterministic oracle machine and c be one of its computations with oracle $B \oplus H$ on some input. Denote by $Query_N^B(c, B \oplus H)$ the set of all $y \in \mathbf{B}^*$ such that N asks '$B(y) =$?' during the computation c. By P- [NP-, BPP-] machine we mean a polynomial-time deterministic [nondeterministic, probabilistic] oracle machine.

For a function ψ defined on a finite set D denote by $\#_1\psi$ the number of $y \in D$ such that $\psi(y) = 1$ (the notation $\#_0\psi$ is defined in the similar way).

3 A sample application

Theorem 1 *There exists an oracle A such that $\mathbf{NP}^A$-sets are inseparable and $\mathbf{Co\text{-}NP}^A$-sets are separable.*

Proof. The proof of this theorem is very close to the proof of the theorem from [BGS 75] stating that $\mathbf{P}^A = \mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A \neq \mathbf{NP}^A$ for some oracle A.

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

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

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

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

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

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

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

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

2) any two disjoint $\mathbf{Co\text{-}NP}^{B\oplus H}$-sets are separable by some $\mathbf{P}^{B\oplus H}$-set.

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

In step i, we construct an interval Γ_i intersecting with V in such a way that $\Gamma_1 \supset \Gamma_2 \supset \Gamma_3 \supset \ldots$ and the following holds:

if $i = 2k + 1$, then the language $\{x \mid P_k^{B\oplus H}(x) = 1\}$ does not separate L_0^B from L_1^B for any $B \in \Gamma_i \cap V$ and

if $i = 2k+2$, then either the pair $\langle N_{k0}, N_{k1}\rangle$ is not correct on $B \oplus H$ for any $B \in \Gamma_i \cap V$, or the languages $\{x \mid N_{k0}^{B\oplus H}(x) = 0\}$ and $\{x \mid N_{k1}^{B\oplus H}(x) = 0\}$ are separable by a $\mathbf{P}^{B\oplus H}$-set for any $B \in \Gamma_i \cap V$.

Obviously for any oracle B in the set $V \cap \bigcap_{i=1}^{\infty} \Gamma_i$ the assertions 1) and 2) will hold.

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

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

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

First case: $i = 2k + 1$. Pick $n \in \mathbf{S}$ greater than $maxlength\,\mathrm{Dom}(\varphi)$ and so large that P_k on input 1^n makes less than 2^{n-1} queries to oracle. Let C be the oracle in Γ_{i-1} being equal to zero on all the words not in $\mathrm{Dom}(\varphi)$. Without loss of generality we may assume that $P_k^{C\oplus H}(1^n) = 0$ (other case is entirely similar). We know that $|\,Query_{P_k}^C(1^n, C \oplus H)|$ is less than the number of words of length $n-1$. Pick a word z of the form $1u$ in the set $\mathbf{B}^n \setminus Query_{P_k}^C(1^n, C \oplus H)$. Note that z is not in $\mathrm{Dom}(\varphi)$ since $n > maxlength(\mathrm{Dom}(\varphi))$. Let

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

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

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

Lemma 1.1 If an interval Γ intersects with V then there exists no P-machine P such that $P^{B\oplus H}$ separates L_0^B from L_1^B for any $B \in \Gamma \cap V$.

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

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

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

The union is over all the computations of N_{k0} (N_{k1}) on input 1^n with oracle $C \oplus H$.

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

Definition 1 A *good* machine is a deterministic oracle Turing machine working within polynomial space and which on input x makes at most poly($|x|$) queries.

Lemma 1.2 If Γ is an interval and $\langle N_0, N_1\rangle$ is a pair of NP-machines being correct on $C \oplus H$ for any $C \in \Gamma \cap V$, then there exists a good machine P such that for any x and any $B \in V \cap \Gamma$, $P^B(x)$ is equal to a $j \in \{0,1\}$ for which $N_j^{B\oplus H}(x) = 1$.

Lemma 1.3 Let P be a good machine. Then there exists a P-machine M such that $P^B(x) = M^{B\oplus H}(x)$ for all $x \in \mathbf{B}^*$, $B \in \Omega$ (recall that H is a **PSPACE**-complete set).

Proof of Lemma 1.2. We describe the work of P on input x with oracle B in the case $B \in \Gamma \cap V$. The reader can easily modify the program of P to handle the general case. Machine P with oracle B on input x works as follows. Find first $n = n_i \in \mathbf{S}$ such that $\log_2 n \le |x| < 2^n$. Let m be so large that N_j on inputs of length greater than m cannot query oracle values on words of length n_{i+1} or greater ($j = 0, 1$). If $|x| < m$ then compute $N_0^{B\oplus H}(x)$ directly and return 0 if $N_0^{B\oplus H}(x) = 1$ and 1 else.

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

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

Note that $C \in V \cap \Gamma$, therefore $N_0^{C\oplus H}(x) = 1$ or $N_1^{C\oplus H}(x) = 1$. Find an $l \in \{0,1\}$ such that $N_l^{C\oplus H}(x) = 1$ and find an accepting computation c of N_l

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

Obviously, we have made $\mathrm{poly}(|x|)$ queries. □

Proof of Lemma 1.3. Let P be a good machine. Define the functions $question(x, w)$ and $result(x, w)$ as follows. Let w be a binary word of length n. For any $i \leq n$ denote by $w(i)$ the ith symbol of w. Run the machine P on input x and give the answer $w(1)$ to the first query, the answer $w(2)$ to the second query and so on. There are three possibilities:

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

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

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

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

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

$$question(x, w) = result(x, w) = \$.$$

Obviously, both functions *question* and *result* are computable within polynomial space. Therefore, they can be computed by a polynomial-time machine with oracle H. Let machine M work according the program shown on Figure 1. □

The proof of Theorem 1 is finished. □

All other theorems in this paper are proved according to the presented scheme. Namely, first a set V of oracles is defined (which is called the universum).

The oracle under which the desired Boolean combination of complexity assertions holds always has the form $B \oplus H$, where H is a **PSPACE**-complete set.

The desired properties of B are represented as a countable family of requirements on B and then the diagonal construction is used to satisfy all the requirements. On the ith step, an interval Γ_i is constructed such that the ith requirement holds for any $B \in \Gamma_i \cap V$. The requirements are of two types: "negative" ones and "positive" ones (in the above example the requirements satisfied on odd steps are negative ones and the requirements satisfied on even steps are

Figure 1:

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

positive ones). Negative requirements are satisfied by using an appropriate analog of Lemma 1.1. Its proof is always easy, therefore we will only present the analog of languages L_0^B and L_1^B. The positive requirements will be satisfied by trying first to make the current pair of machines (or single machine in the case of classes **BPP** and **R**) incorrect. The notion of correctness of course will be specific in each case. If this fails, then we use an analog of Lemma 1.2, which combined with Lemma 1.3 (common for all the applications of the method) will complete the proof.

Thus, the proof of any specific theorem in the sequel will consist of the definition of the universum, the definition of analog of languages L_0^B and L_1^B and the proof(s) of the appropriate analog(s) of Lemma 1.2.

4 The applications of the universum method

The assertions on complexity classes to which the method is applied are shown on the Figure 2. We apply the universum method to prove the existence of oracles under which one or another combination of assertions that label the nodes of the graph holds. There are 17 possible combinations of those assertions. They are listed in the Table 1.

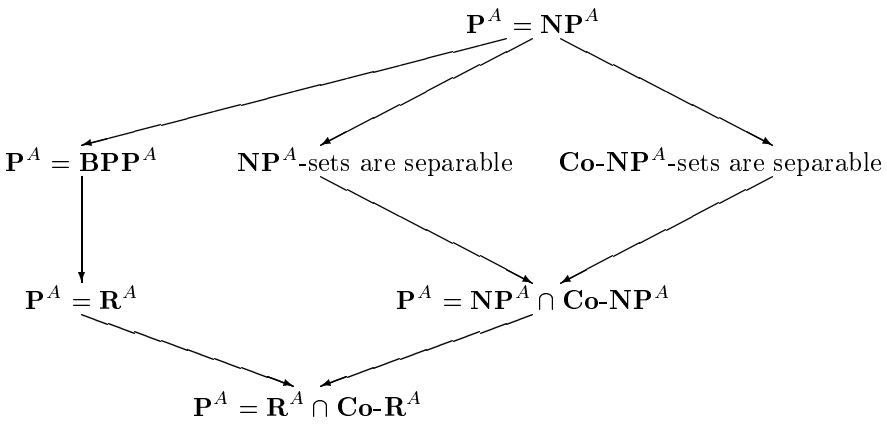
We are able to prove the existence of oracles under which the combinations of all the lines but the lines number 3, 4, 9 and 10 are true. In fact, we do not know the answer to the following question.

Question Is there an oracle under which **Co-NP**-sets are separable and $\mathbf{P} \neq \mathbf{BPP}$?

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

	P = NP	**NP**-sets separable	**Co-NP**-sets separable	**P = NP ∩Co-NP**	**P = BPP**	**P = R**	**P = R ∩Co-R**	Commentary
1	$\oplus$	+	+	+	+	+	+	[BGS 75]
2	$\ominus$	$\oplus$	$\oplus$	+	$\oplus$	+	+	Theorem 13
3	$-$	$\oplus$	$\oplus$	+	$\ominus$	$\oplus$	+	Unknown
4	$-$	$\oplus$	$\oplus$	+	$-$	$\ominus$	+	Unknown
5	$-$	$\oplus$	$\ominus$	+	$\oplus$	+	+	Theorem 4
6	$-$	$\oplus$	$\ominus$	+	$\ominus$	$\oplus$	+	Th. 7
7	$-$	$\oplus$	$\ominus$	+	$-$	$\ominus$	+	Th. 8
8	$-$	$\ominus$	$\oplus$	+	$\oplus$	+	+	Theorem 2
9	$-$	$\ominus$	$\oplus$	+	$\ominus$	$\oplus$	+	Unknown
10	$-$	$\ominus$	$\oplus$	+	$-$	$\ominus$	+	Unknown
11	$-$	$\ominus$	$\ominus$	$\oplus$	$\oplus$	+	+	Th. 6
12	$-$	$\ominus$	$\ominus$	$\oplus$	$\ominus$	$\oplus$	+	Theorem 9
13	$-$	$\ominus$	$\ominus$	$\oplus$	+	$\ominus$	+	Theorem 10
14	$-$	$-$	$-$	$\ominus$	$\oplus$	+	+	Th. 3
15	$-$	$-$	$-$	$\ominus$	$\ominus$	$\oplus$	+	Th. 11
16	$-$	$-$	$-$	$\ominus$	$-$	$\ominus$	$\oplus$	Th. 12
17	$-$	$-$	$-$	$-$	$-$	$-$	$\ominus$	Well known

Figure 2: The edges of the drawn directed graph represent relativizable implications (i.e., implications that are true under any oracle). For example, the implication $\mathbf{P}^A = \mathbf{NP}^A \Rightarrow \mathbf{P}^A = \mathbf{BPP}^A$ is true by the well-known result of Sipser $\mathbf{BPP}^A \subset \Sigma_4^A$ (P.Gács improved that result to $\mathbf{BPP}^A \subset \Sigma_2^A$ [S 83], see also [L 83] for the simplified proof).



We shall use only the universums of the form

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

where Z is a subfamily of the family $\mathbf{F}$ of all the functions having the type $\mathbf{B}^n \to \mathbf{B}$ for some $n \in \mathbf{S}$. In the sequel we denote by $\mathbf{F}_n$ the set $\mathbf{B}^{\mathbf{B}^n}$. The set Z is called the *base* of $V(Z)$.

The following five standard bases are important for the present investigation:

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

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

Now, we formulate the five analogs of Lemma 1.1, which will be used to satisfy negative requirements in the proofs of the next theorems. Their proofs are straightforward and therefore we omit them.

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

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

from the language

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

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

Lemma 2 *Let Z' be any base and let an interval Γ intersect with the universum $V = V(Z(\geq 1)+Z')$. Then there exists no* P*-machine M such that $M^{B\oplus H}$ separates the language*

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

from the language

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

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

Lemma 3 *Let Z' be any base and let an interval Γ intersect with the universum $V = V(Z(=1)+Z')$. Then there exists no* P*-machine M such that $M^{B\oplus H}$ recognizes the language*

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

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

Lemma 4 *Let Z' be any base and let an interval Γ intersect with the universum $V = V(Z(\mathbf{BPP})+Z')$. Then there exists no* P*-machine M such that $M^{B\oplus H}$ recognizes the language*

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

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

Lemma 5 *Let Z' be any base and let an interval Γ intersect with the universum $V = V(Z(\mathbf{R})+Z')$. Then there exists no* P*-machine M such that $M^{B\oplus H}$ recognizes the language*

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

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

Now, we are going to consider all the lines in the table except lines numbered 3, 4, 9 and 10.

The existence of oracles under which the combinations in the first and in the last lines hold is well known so we skip those lines.

Theorem 2 **NP**-*sets are inseparable,* **Co-NP**-*sets are separable and* $\mathbf{P} = \mathbf{BPP}$ *under some oracle (8th line in the table).*

Proof. This theorem strengthens Theorem 1 and its proof uses the same universum $V = V(Z(\leq 1))$. All we have to do is to prove the analog of Lemma 1.2 for BPP-machines. We say that a BPP-machine M is correct on an oracle A if M^A accepts any input with probability lying outside the segment $[1/3;2/3]$.

Lemma 2.1 Assume that Γ is an interval and M is a BPP-machine being correct on $C \oplus H$ for any $C \in \Gamma \cap V$. Then there exists a good machine P that recognizes with oracle B the same language as M does with oracle $B \oplus H$ for any $B \in V \cap \Gamma$.

Proof. Let M and Γ satisfy the conditions of the lemma. Let us construct P. Let x be an input to P. In fact, the beginning of the proof of all analogs of Lemma 1.2 is common. We first find an $n = n_i \in \mathbf{S}$ such that $\log_2 n \leq |x| < 2^n$, query B's values on words of length at most n_{i-1}, then compute the value $\mathrm{Prob}[M^{B\oplus H}(x)]$ directly if $|x|$ is so small that $M^{B\oplus H}(x)$ may depend on $B|\mathbf{B}^{\geq n_{i+1}}$. It remains to construct a good machine P' that on input $\langle x, B|\mathbf{B}^{\leq n_{i-1}}\rangle$, where $B \in V \cap \Gamma$, decides if $\mathrm{Prob}[M^{B\oplus H}(x)] > 2/3$ provided x is so long that $M^{B\oplus H}$ on input x cannot query B's value on words of length n_{i+1} or greater.

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

We will use the notion of weight introduced in [S 82]. For an oracle D let $w_D(y)$ (the *weight* of y *relative* to D) denote the probability of the event "$M^{D\oplus H}$ at some moment in the computation on input x queries '$D(y) =?$'". It is easy to see that

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

for any D.

Denote by W the set of all $y \in \mathbf{B}^n$ such that $w_C(y) \geq 1/6$. Obviously, $|W| \leq \mathrm{poly}(|x|)$.

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

First case: $\forall y \in W\ B(y) = 0$. Let us prove that then $\mathrm{Prob}[M^{B\oplus H}(x) = 1] > 2/3$. Since M is correct on $B \oplus H$ the probability q of the event "$M^{B\oplus H}(x) = 1$"

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

is either greater than $2/3$ or less than $1/3$. We claim that the first alternative holds. Indeed, if $B|\mathbf{B}^n = C|\mathbf{B}^n$, then $q = p \geq 1/2$, therefore $q > 2/3$. Otherwise denote by y the unique word of length n such that $B(y) = 1$. Then $w_C(y) < 1/6$ because $y \notin W$. Let us make use of the following

Lemma 2.2 For any oracles D and E,

$$\left|\mathrm{Prob}[M^D(x) = 1] - \mathrm{Prob}[M^E(x) = 1]\right| \leq \sum_{z \in \mathbf{B}^*: D(z) \neq E(z)} w_D(z).$$

Proof. Obvious.

By the above lemma, $\left|\mathrm{Prob}[M^{B\oplus H}(x) = 1] - \mathrm{Prob}[M^{C\oplus H}(x) = 1]\right| \leq w_C(y) < 1/6$, therefore, $\mathrm{Prob}[M^{B\oplus H}(x) = 1] > 1/2 - 1/6 = 1/3$. Thus $\mathrm{Prob}[M^{B\oplus H}(x) = 1] > 2/3$. Return 1 in the first case.

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

Theorem 3 *There exists an oracle A such that $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A \neq \mathbf{P}^A$ and $\mathbf{BPP}^A = \mathbf{P}^A$ (14th line in the table).*

Proof. Take the base $Z = Z(=1)$. The analog of Lemma 1.2 for BPP-machines is already proved (see the proof of Lemma 2.1 and the footnote on page 12). □

Theorem 4 *There exists an oracle A such that $\mathbf{NP}^A$-sets are separable, $\mathbf{Co\text{-}NP}^A$-sets are inseparable and $\mathbf{BPP}^A = \mathbf{P}^A$ (5th line in the table).*

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

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

To ensure separability of $\mathbf{NP}^A$-sets we shall prove the following analog of Lemma 1.2.

Lemma 4.1 Let $\langle N_0, N_1\rangle$ be a pair of NP-machines being correct on $B \oplus H$ for any $B \in V \cap \Gamma$. Then there exists a good machine P that for all $B \in V \cap \Gamma$ on input x with oracle B finds an $l \in \{0, 1\}$ for which $N_l^{B\oplus H}(x) = 0$.

Proof. Let x be the input word. Let $n = n_i$ be defined as in the proof of Lemma 1.2. Assume that the length of x is so large that both machines N_0, N_1 on input x cannot query oracle values on words of length $\geq n_{i+1}$ and that n is greater than lengths of words defining Γ. Assume that we already know $B|\mathbf{B}^{\leq n_i}$. We have to find an l such that $N_l^{B\oplus H}(x) = 0$.

To this end we shall use the technique from [BI 87]. We need the notion of a certificate. Let $l = 0, 1$. An *l-certificate* is a function having the form $C|\, Query_{N_l}^C(c, C \oplus H)$, where C is an oracle agreeing with B on all words of length different from n, $N_l^{C\oplus H}(x) = 1$ and c is an accepting computation of $N_l^{C\oplus H}$ on x.

Note that if γ is an l-certificate and C continues γ, then $N_l^{C\oplus H}(x) = 1$. Obviously, the cardinality of domain of any l-certificate is bounded by a polynomial of $|x|$, say $p(|x|)$. Assume that x is so long that $2^n > 2p(|x|)$. We claim that then any 0-certificate φ is inconsistent with any 1-certificate ψ (that is, $\varphi(y) \neq \psi(y)$ for some $y \in \mathrm{Dom}(\varphi) \cap \mathrm{Dom}(\psi)$).

Indeed, assume that a 0-certificate φ and a 1-certificate ψ are consistent. Then there exists an oracle C agreeing with B on words of length different from n that continues both φ and ψ. As $|\mathrm{Dom}(\varphi)| + |\mathrm{Dom}(\psi)| < 2^n$ we may assume that there exists $y \in \mathbf{B}^n$ such that $C(y) = 1$, that is, C is in V.[3] Since C continues both φ and ψ we have $N_0^{C\oplus H}(x) = N_1^{C\oplus H}(x) = 1$. Thus the pair N_0, N_1 is incorrect on $C \oplus H$ and C is in $V \cap \Gamma$. The contradiction proves the claim.

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

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

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

Obviously, we have made at most $p(|x|)^2$ queries to B.

Let us prove now that the described program can be run within polynomial space. We do not need to store $\mathcal{C}_0$ or $\mathcal{C}_1$. It suffices to store the set U and the B's value on elements of U.

Having U and $B|U$ we can decide if there is a 0-certificate [1-certificate] consistent with $B|U$ by checking all the computations of N_0 [N_1] on input x. If a query '$B(y) =?$' is made during one of computations we answer '$B(y)$' if $|y| \leq n_{i-1}$ or y is in U (note that we know B's value on such words), 0 if $n_{i-1} < |y| < n$ or $n < |y|$ and try all the answers otherwise. As the number of queries does not exceed $\mathrm{poly}(|x|)$ the amount of stored information is $\mathrm{poly}(|x|)$. □

To ensure the equality $\mathbf{BPP}^A = \mathbf{P}^A$ we shall prove the following analog of Lemma 1.2.

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

Lemma 4.2 Let M be a BPP-machine being correct on $B \oplus H$ for any $B \in V \cap \Gamma$. Then there exists a good machine P such that for any $B \in V \cap \Gamma$, P^B recognizes the same language as $M^{B \oplus H}$ does.

Proof. We use some ideas from [N 89] and [IN 88].

By Lemma 4.1 it suffices to construct a pair $\langle N_0, N_1 \rangle$ of NP-machines such that

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

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

Let x be an input and let B be in $V \cap \Gamma$. Let n be defined as usually and let x be so long that M on input x cannot query B's value on words of length n_{i+1} or greater and that n is greater than lengths of words defining Γ.

Denote by k the maximal number of queries that M can make on input x. It is important that $k \leq \mathrm{poly}(|x|)$. Let in the sequel $Q(B)$ denote the set of all the oracles in V agreeing with B on all the words of length different from n. Note that $Q \subset V \cap \Gamma$ so M is correct on $C \oplus H$ for any $C \in Q(B)$.

For $y \in \mathbf{B}^n$ and $D \in Q(B)$ denote by $w_D(y)$ the probability of the event "$M^{D\oplus H}$ on input x queries '$D(y) =?$'".

Let

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

Let us note that since $\sum_{y \in \mathbf{B}^*} w_B(y) \leq k$ the set W has at most $(9k+3)k$ elements.

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

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

Let us prove that $w_C(y) > 1/3$ for any y in U possibly but one. Let y be an element of U. Denote by C_y the oracle obtained from C by changing the value on y. Then C_y differs from B on less arguments than C does. Since $C_y|W = B|W$, we have $\mathrm{Prob}[M^{C\oplus H}(x) = 1] > 2/3$ provided C_y is in $Q(B)$. Let us distinguish two cases.

First case: C_y is in $Q(B)$. Then $\mathrm{Prob}[M^{C_y\oplus H}(x) = 1] > 2/3$. Therefore, by Lemma 2.2, $w_C(y) \geq \mathrm{Prob}[M^{C_y\oplus H}(x) = 1] - \mathrm{Prob}[M^{C\oplus H}(x) = 1] > 2/3 - 1/3 = 1/3$.

Second case: $C_y \notin Q(B)$. Since C_y agrees with B on all the word of length different from n this means that $\#_1(C_y|\mathbf{B}^n) = 0$. This may happen only if $\#_1(C|\mathbf{B}^n) = 1$ and therefore this case can occur for the single y.

As $\sum_{y\in\mathbf{B}^*} w_C(y) \le k$, we have $|U| \le 3k+1$. Since $U \cap W = \emptyset$, we have $w_B(y) < \frac{1}{9k+3}$ for any $y \in U$. Hence $\sum_{y\in U} w_B(y) < \frac{1}{(9k+3)}(3k+1) = \frac{1}{3}$. On the other hand, by Lemma 2.2

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

The contradiction proves the claim.

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

For a set $U \subset \mathbf{B}^n$ let $O(B|U)$ denote the set of all oracles C agreeing with B on all the words of length different from n and on U and such that the number of words in $\mathbf{B}^n \setminus U$ on which C is equal to zero is at most $(9k+3)k$. Assume that $|x|$ is so large that $2(9k+3)k < 2^n$. Then $O(B|U) \subset Q(B)$ for any U having at most $(9k+3)k$ elements.

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

$(*)$ there is U such that $|U| \le (9k+3)k$ and $\mathrm{Prob}[M^{C\oplus H}(x)=1] > 2/3$ for any $C \in O(B|U)$.

The implication from the left to the right follows from the above claim since we can take W as U. Let us prove the implication from the right to the left.

Assume that $\mathrm{Prob}[M^{B\oplus H}(x)=1] < 1/3$ but there is U such that $|U| \le (9k+3)k$ and $\mathrm{Prob}[M^{C\oplus H}(x)=1] > 2/3$ for any $C \in O(B|U)$. Take such U. Let D denote the oracle in $Q(B)$ agreeing with B on $U \cup W$ and being equal to 1 on all the words in $\mathbf{B}^n \setminus (U \cup W)$. Then $\mathrm{Prob}[M^{D\oplus H}(x)=1] < 1/3$ since D agrees with B on W. On the other hand D is in $O(B|U)$ hence $\mathrm{Prob}[M^{D\oplus H}(x)=1] > 2/3$. The contradiction shows that $\mathrm{Prob}[M^{B\oplus H}(x)=1] > 2/3$ if and only if $(*)$ is true.

Let us be given a subset U of $\mathbf{B}^n$ having at most $(9k+3)k$ elements. The values of any oracle $C \in O(B|U)$ on words of length less than n_{i+1} can be identified by means of polynomial amount of information and the value $\mathrm{Prob}[M^{C\oplus H}(x)=1]$ can be computed within polynomial space. Thus given U we can decide within polynomial space if $(*)$ is true. Therefore given U we can decide in polynomial time using the oracle H if $(*)$ is true.

Machine $N_1^{B\oplus H}$ on input x works as follows. Query the value of B on all the words of length at most n_{i-1}. Then guess a set $U \subset \mathbf{B}^n$ having $(9k+3)k$ elements and accept if $(*)$ is true. □

Theorem 5 *There exists an oracle A such that $\mathbf{NP}^A$-sets are separable, $\mathbf{BPP}^A \ne \mathbf{P}^A$ and $\mathbf{R}^A = \mathbf{P}^A$.*

Proof. Let $V = V(Z(\mathbf{BPP}))$. To ensure separability of $\mathbf{NP}^A$-sets we need the following analog of Lemma 1.2.

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

Lemma 5.1 Let $\langle N_0, N_1\rangle$ be a pair of NP-machines being correct on $B \oplus H$ for any $B \in V \cap \Gamma$. Then there exists a good machine P that for any $B \in V \cap \Gamma$ on input x with oracle B finds an $l \in \{0,1\}$ for which $N_l^{B\oplus H}(x) = 0$

Proof. This lemma can be proven just as Lemma 4.1. The only difference is that we have to take x so large that $(1/3)2^n > 2p(|x|)$ (and not $2^n > 2p(|x|)$ as in that proof). □

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

We need the following analog of of Lemma 1.2.

Lemma 5.2 Let M be a probabilistic polynomial-time oracle machine being correct on oracle $B \oplus H$ for any $B \in V \cap \Gamma$. Then there exists a good machine P that with any oracle $B \in V \cap \Gamma$ recognizes the same language as M does with oracle $B \oplus H$.

Proof. By Lemma 5.1 it suffices to construct a pair $\langle N_0, N_1\rangle$ of NP-machines such that

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

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

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

Thus, we have to construct an NP-machine N_0 satisfying this requirement. Let x be an input to N_0 and let B be in $V \cap \Gamma$. Let $n = n_i$ be defined in usual way. Assume that M on input x cannot query oracle values on words of length $\geq n_{i+1}$. Let $k = \mathrm{poly}(|x|)$ be the maximal number of queries to B which machine M can make during the work on input x. Let $Q(B)$ denote the set of all the oracles in V agreeing with B on all the words of length different from n.

Denote by $w_C(y)$ the probability of the event "$M^{C\oplus H}(x)$ on input x queries '$C(y) =?$'". Let $W = \{y \in \mathbf{B}^n \mid w_B(y) > 1/4k\}$. Note that $|W| < 4k^2$.

Claim. $\mathrm{Prob}[M^{C\oplus H}(x) = 1] = 0$ for any $C \in Q(B)$ agreeing with B on W.

Proof of the claim. Assume the contrary. Let C be a counterexample. Then for at least one random string, $M^{C\oplus H}(x) = 1$. Denote by U the set of all the $y \in \mathbf{B}^n$ such that the query '$C(y) =?$' is made during the computation of $M^{C\oplus H}$ on x for that random string. Obviously, $|U| \leq k$. Let D be the oracle agreeing with C on U and with B on remaining words. If n is large enough, then D is in Γ. If D belonged to V we would obtain a contradiction: we know that probability of the event "$M^{D\oplus H}(x) = 1$" is positive, hence, this probability would be greater than $1/2$. Therefore, by Lemma 2.2, $\sum_{D(y)\neq B(y)} w_B(y)$ would be greater than

1/2. On the other hand, $\{y \mid D(y) \neq B(y)\} \subset U \setminus W$, consequently,

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

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

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

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

As $(1/3)2^n \leq \#_0(D|\mathbf{B}^n)$, there exists a set $T \subset \mathbf{B}^n$ having exactly $|U|$ elements, not intersecting with $U \cup W$ and such that $D(y) = 0$ for any $y \in T$. Take such T and change the value of D on all the words in T. Now we have $\#_1(D|\mathbf{B}^n) > (2/3)2^n$, therefore D is in V. As $D(y) = C(y)$ for any $y \in U$, we have $\mathrm{Prob}[M^{D\oplus H}(x) = 1] > 0$. Therefore, $\mathrm{Prob}[M^{D\oplus H}(x) = 1] > 1/2$.

Recall that $\mathrm{Prob}[M^{B\oplus H}(x) = 1] = 0$. Hence, by Lemma 2.2,

$$\sum_{D(y)\neq B(y)} w_B(y) > 1/2.$$

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

The rest is as in the proof of Lemma 4.2.

For a set $U \subset \mathbf{B}^n$ let $O(B|U)$ denote the set of all oracles C agreeing with B on all the words in U and on all the words of length different from n and such that the number of words in $\mathbf{B}^n \setminus U$ on which C is equal to zero is at most k. Assume that $|x|$ is so large that $4k^2 + k < 2^{n-1}$. Then $O(B|U) \subset Q(B)$ for any U having at most $4k^2$ elements.

Let us prove that $\mathrm{Prob}[M^{B\oplus H}(x) = 1] = 0$ if and only if

(∗) there is U such that $|U| \leq 4k^2$ and $\mathrm{Prob}[M^{C\oplus H}(x) = 1] = 0$ for any $C \in O(B|U)$.

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

Assume that $\mathrm{Prob}[M^{B\oplus H}(x) = 1] > 1/2$ but there is U such that $|U| \leq 4k^2$ and $\mathrm{Prob}[M^{C\oplus H}(x) = 1] = 0$ for any $C \in O(B|U)$. Take such U. Let us fix any random string for which $M^{B\oplus H}(x) = 1$ and denote by R the set of $y \in \mathbf{B}^n$ such that the query '$B(y) =?$' was made during the computation of $M^{B\oplus H}$ on input x for that string. Let D denote the oracle in $Q(B)$ agreeing with B on $U \cup R$ and being equal to 1 on all the words in $\mathbf{B}^n \setminus (U \cup R)$. Then $\mathrm{Prob}[M^{D\oplus H}(x) = 1] > 0$ since D agrees with B on R. On the other hand D

Table 2: The sign "+" in a line of the table indicates that the analog of Lemma 1.2 is true for the corresponding universum. The sign "–" in a line of the table indicates that the analog of Lemma 1.1 is true for the corresponding universum. The letter "o" means that the proof was omitted (because we do not need the corresponding assertion).

	NP-sep.	o-NP-sep.	NP ∩ Co-NP	BPP	R
$V(Z(\leq 1))$	−	+	+	+	+
$V(Z(=1))$	−	−	−	+	+
$V(Z(\geq 1))$	+	−	+	+	+
$V(Z(\mathbf{BPP}))$	+	− (o)	+	−	+
$V(Z(\mathbf{R}))$	+	− (o)	+	−	−
$V(\mathbf{F})$	+ (o)	− (o)	+ (o)	+ (o)	+ (o)

is in $O(B|U)$ hence $\mathrm{Prob}[M^{D\oplus H}(x) = 1] = 0$. The contradiction shows that $\mathrm{Prob}[M^{B\oplus H}(x) = 1] = 0$ if and only if $(*)$ is true.

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

Thus the theorem is proved. □

The above proven facts on the five standard universums are shown on the Table 2. The sixth line in the table contains unproven facts about the universum $V(\mathbf{F})$ (recall that $\mathbf{F}$ denotes the set of all functions from $\bigcup_{n\in\mathbf{S}} \mathbf{B}^{\mathbf{B}^n}$ into $\mathbf{B}$). This information is presented for the sake of completeness.

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

Theorem 6 *There exists an oracle A such that $\mathbf{NP}^A$-sets and $\mathbf{Co\text{-}NP}^A$-sets are inseparable, $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A = \mathbf{P}^A$ and $\mathbf{BPP}^A = \mathbf{P}^A$ (11th line in the table).*

Proof. Let $V = V(Z(\leq 1) + Z(\geq 1))$. Let us prove first the analog of Lemma 1.2 for NP ∩ Co-NP-machines.

Lemma 6.1 Let $\langle N_0, N_1\rangle$ be a pair of NP-machines such that the languages accepted by $N_0^{B\oplus H}$ and $N_1^{B\oplus H}$ are complementary for any $B \in V \cap \Gamma$. Then there exists a good machine P that with any oracle $B \in V \cap \Gamma$ accepts the same language as N_0 does with oracle $B \oplus H$.

Proof. Let Γ, N_0 and N_1 satisfy the conditions of the lemma.

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

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

for any $\alpha \in \mathbf{F}_n$. In this case we consider $\langle N_0, N_1 \rangle$ as a pair defining a problem of separation of Co-NP-sets and reason as in the proof of Lemma 1.2.

If i is odd, then by definition

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

for any $\alpha \in \mathbf{F}_n$. In this case we consider $\langle N_0, N_1 \rangle$ as a pair defining a problem of separation of NP-sets and reason as in the proof of Lemma 4.1. $\square$

The analog of Lemma 1.2 for BPP-machines can be proved similarly. $\square$

To prove the next Theorems 7–12 we do not need any new ideas. Therefore we shall only present the bases used in their proofs.

Theorem 7 *There exists an oracle A such that $\mathbf{NP}^A$-sets are separable, $\mathbf{Co\text{-}NP}^A$-sets are inseparable, $\mathbf{BPP}^A \neq \mathbf{P}^A$, $\mathbf{R}^A = \mathbf{P}^A$ (6th line in the table).*

Proof. Take the base $Z = Z(\mathbf{BPP}) + Z(\geq 1)$. $\square$

Theorem 8 *There exists an oracle A such that $\mathbf{NP}^A$-sets are separable, $\mathbf{Co\text{-}NP}^A$-sets are inseparable and $\mathbf{P}^A \neq \mathbf{R}^A$ (7th line in the table).*

Proof. Take the base $Z = Z(\mathbf{R}) + Z(\geq 1)$. $\square$

Theorem 9 *There exists an oracle A such that $\mathbf{NP}^A$-sets are inseparable, $\mathbf{Co\text{-}NP}^A$-sets are inseparable, $\mathbf{BPP}^A \neq \mathbf{P}^A$, $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A = \mathbf{P}^A$ and $\mathbf{R}^A = \mathbf{P}^A$ (12th line in the table).*

Proof. Take the base $Z = Z(\leq 1) + Z(\geq 1) + Z(\mathbf{BPP})$. $\square$

Theorem 10 *There exists an oracle A such that $\mathbf{NP}^A$-sets are inseparable, $\mathbf{Co\text{-}NP}^A$-sets are inseparable, $\mathbf{R}^A \neq \mathbf{P}^A$ and $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A = \mathbf{P}^A$ (13th line in the table).*

Proof. Take the base $Z = Z(\leq 1) + Z(\geq 1) + Z(\mathbf{R})$. $\square$

Theorem 11 *There exists an oracle A such that $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A \neq \mathbf{P}^A$, $\mathbf{BPP}^A \neq \mathbf{P}^A$ and $\mathbf{R}^A = \mathbf{P}^A$ (15th line in the table).*

Proof. Take the base $Z = Z(=1) + Z(\mathbf{BPP})$. $\square$

Theorem 12 *There exists an oracle A such that $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A \neq \mathbf{P}^A$, $\mathbf{R}^A \neq \mathbf{P}^A$ and $\mathbf{R}^A \cap \mathbf{Co\text{-}R}^A = \mathbf{P}^A$ (16th line in the table).*

Proof. Take the base $Z = Z(=1) + Z(\mathbf{R})$. $\square$

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

Theorem 13 *There exists an oracle A such that $\mathbf{P}^A \neq \mathbf{NP}^A$, $\mathbf{NP}^A$-sets are separable, $\mathbf{Co\text{-}NP}^A$-sets are separable and $\mathbf{BPP}^A = \mathbf{P}^A$ (second line in the table).*

Proof. We use a diagonal construction as in the proof Theorem 1 but instead of a chain of intervals we construct a chain

$$\Gamma_0 \cap V_0 \supset \Gamma_1 \cap V_1 \supset \dots,$$

where Γ_i's are intervals and V_i's are subsets of Ω. All universums V_i's are taken from the family $\{V(Z^j) \mid j \geq 1\}$, where $Z^j = \bigcap_{n \in \mathbf{S}} \{\alpha \in \mathbf{F}_n \mid \#_1 \alpha \leq n/j\}$. Since all the sets in this family are closed in Cantor's topology, the intersection $\bigcap_{i=0}^{\infty} \Gamma_i \cap V_i$ is non-empty.

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

We do not present the whole diagonal construction but instead present only specific points. The steps on which we satisfy the requirement $L^B \notin \mathbf{P}^{B \oplus H}$ are made as usual: on those steps we do not need to change current universum.

Current universum is changed on steps on which we satisfy the requirement of separability of NP-sets. We have now to use the following analog of Lemma 1.2.

Lemma 13.1 Assume that N_0, N_1 are NP-machines such that the languages accepted by $N_0^{B \oplus H}$ and $N_1^{B \oplus H}$ are disjoint for any $B \in \Gamma \cap V(Z^j)$. Then there exists a good machine separating those languages for any $B \in \Gamma \cap V(Z^{2j})$.

Proof. Let the conditions of the lemma be true. The good machine separating those languages works on input x as follows. Let $n = n_i$ be defined as usually. An *l-certificate* ($l = 0, 1$) is a function of the form $C \mid Query_{N_l}^C(c, C \oplus H) \cap \mathbf{B}^n)$, where C is an oracle in $\Gamma \cap V(Z^{2j})$ agreeing with B on words of length different from n, $N_l^{C \oplus H}(x) = 1$ and c is an accepting computation of $N_l^{C \oplus H}$ on input x.

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

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

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

Further we can reason just as in the proof of Lemma 4.1. $\square$

Lemma 13.2 Assume that N_0, N_1 are NP-machines such that the languages accepted by $N_0^{B \oplus H}$ and $N_1^{B \oplus H}$ span $\mathbf{B}^*$ for all $B \in \Gamma \cap V(Z^j)$. Then there exists a good machine that on input x with any oracle $B \in \Gamma \cap V(Z^j)$ finds an $l \in \{0, 1\}$ such that $N_l^{B \oplus H}(x) = 1$.

Proof. The good machine that finds l works on input x as follows. Start with the oracle C that is equal to B on words of length different from n and to zero on remaining words. Find an l such that $N_l^{C\oplus H}(x) = 1$ (such l does exist since C is in $V(Z^j) \cap \Gamma$). Then we either discover that $N_l^{B\oplus H}(x) = N_l^{C\oplus H}(x)$ or find a $u \in \mathbf{B}^n$ such that $B(u) = 1$. In the latter case include u in C and repeat the process.

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

We need also the analog of Lemma 1.2 for BPP-machines:

Lemma 13.3 Let M be a BPP-machine being correct on $B \oplus H$ for any $B \in V(Z^j) \cap \Gamma$. Then there exists a good machine P such that P^B recognizes the same language as $M^{B\oplus H}$ does for any $B \in V(Z^j) \cap \Gamma$.

Proof. Let the conditions of the lemma be true. We have to construct a good machine that with any oracle $B \in V(Z^j) \cap \Gamma$ recognizes the same language as $M^{B\oplus H}$ does. Let that machine work as follows. Let x be the input. Let n be defined as in Lemma 1.2. Let the oracle C be equal to B on words of length different from n and to zero on remaining words. Let $U = \{u \in \mathbf{B}^n \mid w_C(u) > 1/3k\}$, where k stands for the polynomial upper bound for the number of queries made by M on input x and $w_C(u)$ denotes the probability of event "$M^{C\oplus H}$ on input x queries '$C(u) =?$'". Query '$B(u) =?$' for all $u \in U$. If there exists no $u \in U$ such that $B(u) = 1$ then by Lemma 2.2

$$\left|\mathrm{Prob}[M^{B\oplus H}(x) = 1] - \mathrm{Prob}[M^{C\oplus H}(x) = 1]\right| \leq \sum_{y:C(y)\neq D(y)} w_C(y) \leq k \cdot \frac{1}{3k} = 1/3.$$

Therefore, in this case $M^{B\oplus H}$ accepts x iff $M^{C\oplus H}$ accepts x.

Otherwise include in C all those $u \in U$ which are in B and repeat the process.

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

Let us present one more application of the universum method consisting in the new proof of a known theorem.

Theorem 14 (Hartmanis and Immerman [HI 86]). *There exists an oracle A such that $\mathbf{P}^A \neq \mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A \neq \mathbf{NP}^A$ and the class $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A$ has an m-complete language.*

Proof. It is sufficient to construct an oracle A such that $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A \neq \mathbf{P}^A$, $\mathbf{NP}^A \not\subset \mathbf{Co\text{-}NP}^A$ and the class $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A$ has an m-complete language. The oracle A will have the form $B \oplus H$, where H is a **PSPACE**-complete set. Thus, we have to construct the oracle B.

Take the universum

$$V = \{A \in \Omega \mid \#_1(A|\mathbf{B}^n) = 1 \text{ for any even } n \text{ and}$$
$$\#_1(A|\mathbf{B}^n) \leq 1 \text{ for any odd } n\}.$$

The language in $\mathbf{NP}^A \setminus \mathbf{Co\text{-}NP}^A$ will be

$$L_1^B = \{1^n \mid n \text{ is odd and } \exists u \in \mathbf{B}^n \; B(u) = 1\}.$$

The language in $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A \setminus \mathbf{P}^A$ will be

$$L_2^B = \{1^n \mid n \text{ is even and } \exists u \in \mathbf{B}^{n-1} B(1u) = 1\}.$$

Obviously, $L_1^B \in \mathbf{NP}^A$ and $L_2^B \in \mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A$ for any $B \in V$ (recall that $A = B \oplus H$). Thus, we have to construct an oracle $B \in V$ such that

1. $L_1^B \notin \mathbf{Co\text{-}NP}^A$;
2. $L_2^B \notin \mathbf{P}^A$;
3. $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A$ has a complete language.

To this end let us enumerate all the polynomial-time deterministic and nondeterministic oracle machines and all the pairs of nondeterministic polynomial-time oracle machines. In usual way we construct a chain

$$\Gamma_0 \supset \Gamma_1 \supset \Gamma_2 \supset \Gamma_3 \supset \ldots$$

of intervals such that any Γ_i intersects with V and the following holds. If $i = 3k$, then kth nondeterministic machine does not accept the language $\{0,1\}^* \setminus L_1^B$ for any $B \in \Gamma_i \cap V$. If $i = 3k+1$ then kth deterministic machine does not recognize the set L_2^B for any $B \in \Gamma_i \cap V$. And if $i = 3k+2$ then either the languages accepted by the nondeterministic machines in kth pair are complementary for any $B \in \Gamma_i \cap V$ or those languages are not complementary for any $B \in \Gamma_i \cap V$.

Take any oracle B in $\bigcap_{i=1}^{\infty} \Gamma_i \cap V$. The assertions 1 and 2 are true. It remains to prove that $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A$ has a complete language. Denote by N_j the jth nondeterministic polynomial-time Turing machine and by $p_j(|x|)$ a polynomial restricting its running time.

For $C \in \Omega$ denote by C^n the word of length $2^{n+1} - 1$ encoding C's value on words of length at most n in lexicographic order.

Let us note that a pair $\langle N_j, N_k \rangle$ of NP-machines defines a language in $\mathbf{NP}^{B \oplus H} \cap \mathbf{Co\text{-}NP}^{B \oplus H}$ iff $N_j(x, B \oplus H) + N_k(x, B \oplus H) = 1$ for any x. As a complete language we take the following language:

$$\begin{aligned} L^B = \{\langle j, k, B^n, x, 0^{p_j(|x|)+p_k(|x|)} \rangle \mid j, k, n \in \mathbf{N}, \; & N_j(x, B \oplus H) = 1 \\ \text{and } N_j(x, C \oplus H) + N_k(x, C \oplus H) = 1 & \text{ for any } C \in V \cap \Gamma(B \mid \mathbf{B}^{\leq n})\}. \end{aligned}$$

Let us prove that L^B is in $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A$. To this end let us prove that L^B is in $\mathbf{NP}^A$ (the remaining part $L^B \in \mathbf{Co\text{-}NP}^A$ can be proved entirely similar).

Let us construct first a nondeterministic polynomial space oracle machine that accepts L^B and makes polynomial number of queries.

Let w be an input word. Decide first whether w has the form

$$\langle j, k, D^n, x, 0^{p_j(|x|)+p_k(|x|)}\rangle$$

for some $D \in V$ and some j, k, n. Then decide whether $B^n = D^n$ and whether $N_j(x, C \oplus H) + N_k(x, C \oplus H) = 1$ for all $C \in V$ such that $C^n = D^n$ (this can be done within polynomial space since both values $N_j(x, C \oplus H)$ and $N_k(x, C \oplus H)$ depend only on value of C on words of length at most $p_j(|x|) + p_k(|x|)$, therefore all the needed information about C can be written within polynomial space). If this is not the case, then reject. Otherwise run N_j on input x with oracle $B \oplus H$ and accept if $N_j(x, B \oplus H) = 1$.

As in the proof of Lemma 1.3 we can convert the constructed nondeterministic polynomial-space machine into a nondeterministic polynomial-time machine with oracle H.

Thus, it remains to prove that L^B is complete in $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A$. Let a language L be in $\mathbf{NP}^A \cap \mathbf{Co\text{-}NP}^A$. Let $\langle N_j, N_k\rangle$ be a pair of nondeterministic polynomial-time oracle machines such that $L(x) = N_j(x, B \oplus H) = 1 - N_k(x, B \oplus H)$ for any x. The construction of the oracle ensures that there exists n such that $N_j(x, C \oplus H) + N_k(x, C \oplus H) = 1$ for any $C \in V \cap \Gamma(B \mid \mathbf{B}^{\leq n})$. Let us fix such an n. The mapping $x \mapsto \langle j, k, B^n, x, 0^{p_j(|x|)+p_k(|x|)}\rangle$ reduces L to L^B. $\square$

Remark In a similar way we could prove all the previous theorems in a stronger form: we could add the assertion that all the involved classes have complete problems.

5 General theorems

In this section we formalize the method applied in the previous section. All the theorems are presented here without proofs. The proofs may be found in [MV 94].

5.1 Basic definitions

We deal with decision problems as well as with separation problems. It is convenient to consider a decision problem as a particular case of a separation problem.

Definition 2 A *separation problem* is a function P from $\mathbf{B}^*$ into $\{0, 1, *\}$ (the meaning of this definition is as follows: we have to separate the set $\{x \mid P(x) = 0\}$ from the set $\{x \mid P(x) = 1\}$). A language is a separation problem L such that $L(x) \neq *$ for all $x \in \mathbf{B}^*$.

For any probabilistic polynomial-time oracle machine M consider the map-

ping

$$M(x,A) = \begin{cases} 1, & \text{if } \mathrm{Prob}[M^A(x)=1] > 2/3; \\ 0, & \text{if } \mathrm{Prob}[M^A(x)=1] < 1/3; \\ \#, & \text{otherwise.} \end{cases} \tag{1}$$

In the sequel we do not distinguish the machine M and the mapping $\langle x, A\rangle \mapsto M(x,A)$. Call any mapping from $\mathbf{B}^* \times \Omega$ into $\{0,1,\#\}$ an *oracle machine*.

Likewise, consider for every pair $N = \langle N_0, N_1\rangle$ of polynomial-time nondeterministic machines the mapping

$$N(x,A) = \begin{cases} 1, & \text{if } N_1^A(x)=1,\ N_0^A(x)=0; \\ 0, & \text{if } N_1^A(x)=0,\ N_0^A(x)=1; \\ *, & \text{if } N_0^A(x)=N_1^A(x)=0; \\ \#, & \text{if } N_0^A(x)=N_1^A(x)=1. \end{cases} \tag{2}$$

Definition 3 A *description* is a mapping from the set $\mathbf{B}^* \times \Omega$ into the set $\{0,1,\#,*\}$. A description D is called *correct on* an oracle A if $D(x,A) \neq \#$ for all $x \in \mathbf{B}^*$.

For example, (1) and (2) are descriptions, and (1) is an oracle machine.

Definition 4 For a description D and for an oracle A denote by D^A the mapping $x \mapsto D(x,A)$. Given a class $\mathcal{D}$ of descriptions and an oracle A denote by $\mathcal{D}^A$ the set $\{D^A \mid D \in \mathcal{D}$ and D is correct on $A\}$.

Obviously, if M is an oracle machine being correct on A, then M^A is a language.

Denote by BPP the set of all descriptions having the form (1), where M is a probabilistic polynomial-time oracle machine. (We use bold face letters for classes of languages, e.g., **BPP**, and roman letters for classes of descriptions, thus, BPP $\neq$ **BPP**, however $\mathrm{BPP}^A = \mathbf{BPP}^A$ for any A).

Denote by NP-separation the class of all descriptions of the form (2), where N_0 and N_1 are polynomial-time nondeterministic oracle machines. Likewise one can define Co-NP-separation, as well as classes of machines P, NP, R, R$\cap$Co-R, NP $\cap$ Co-NP, PSPACE.

Say that a separation problem P_1 is *easier* than a separation problem P_2 ($P_1 \leq P_2$ in symbols) if $P_1(x) \neq *$ implies $P_2(x) = P_1(x)$. In other words, $P_1(x) \leq P_2(x)$ for all $x \in \mathbf{B}^*$, where $\leq$ denotes the partial ordering on the set $\{0,1,*\}$ defined by inequalities $* \leq 0$, $* \leq 1$. Given classes K_1 and K_2 of separation problems we write $K_1 \leq K_2$ if for any $P_1 \in K_1$ there exists $P_2 \in K_2$ such that $P_1 \leq P_2$. Obviously, if K_1 is a class of languages, then $K_1 \leq K_2$ means the same as $K_1 \subset K_2$. For example, $\mathrm{BPP}^A \leq \mathrm{P}^A$ means the same as $\mathbf{BPP}^A \subset \mathbf{P}^A$ and $(\text{NP-separation})^A \leq \mathrm{P}^A$ means that $\mathbf{NP}^A$-sets are separable.

5.2 The most general scheme of the universum method

The universum method can be applied for proving theorems of the following form. Let

$$\mathcal{K}_1, \ldots, \mathcal{K}_n,\ \mathcal{L}_1, \ldots, \mathcal{L}_n,\ \mathcal{M}_1, \ldots, \mathcal{M}_m,\ \mathcal{N}_1, \ldots, \mathcal{N}_m$$

be classes of descriptions. We want to prove that there exists an oracle A such that

$$\mathcal{K}_i^A \not\leq \mathcal{L}_i^A \text{ for } i = 1, \ldots, n \text{ and}$$
$$\mathcal{M}_j^A \leq \mathcal{N}_j^A \text{ for } j = 1, \ldots, m.$$

Consider for simplicity of notation the case $m = n = 1$.

Let us call a *universum* any nonempty subset V of Ω.

Definition 5 A *superuniversum* is any countable family $\mathcal{V}$ of universums having a largest universum up to inclusion and such that the following two assertions hold:

1. For any $V \in \mathcal{V}$ and for any interval Γ intersecting with V there exists $V' \in \mathcal{V}$ such that $V' \subset V \cap \Gamma$.

2. For any countable chain $V_1 \supset V_2 \supset V_3 \supset \ldots$ of elements of $\mathcal{V}$ the intersection $\bigcap_{i=1}^{\infty} V_i$ is non-empty.

The reader can see that in all the applications presented in this paper, the elements of $\mathcal{V}$ have the form $V \cap \Gamma$, where Γ is an interval and V is closed in Cantor's topology, and therefore are closed too (recall that Cantor's topology is the topology the base of which is the set of intervals). This implies condition 2 because Ω is compact in Cantor's topology.

For example, if V is closed in Cantor's topology, then the family

$$\mathcal{V}(V) = \{V \cap \Gamma \mid \Gamma \text{ is an interval intersecting with } V\}$$

is a superuniversum. Such universums were used in the proofs of Theorems 1–12, 14. The single exception was Theorem 13.

The largest universum in $\mathcal{V}$ is denoted by $\rfloor(\mathcal{V})$.

Thus, we wish to prove that there exists an oracle A for which

$$\mathcal{K}^A \not\leq \mathcal{L}^A,$$
$$\mathcal{M}^A \leq \mathcal{N}^A,$$

where $\mathcal{K}$, $\mathcal{L}$, $\mathcal{M}$, $\mathcal{N}$ are classes of descriptions. The method can be applied only if $\mathcal{K}$, $\mathcal{L}$, $\mathcal{M}$, and $\mathcal{N}$ are countable and consist of the so called polynomial-local descriptions.

Definition 6 A description D is called *polynomial-local* if the following holds. There exists a polynomial $p(n)$ such that for all $x \in \mathbf{B}^*$ and all $A, B \in \Omega$ if $A|\mathbf{B}^{\leq p(|x|)} = B|\mathbf{B}^{\leq p(|x|)}$, then $D(x, A) = D(x, B)$.

The first general theorem holds for any countable classes of polynomial-local descriptions. Obviously, all the classes considered in the previous section consist of polynomial-local descriptions.

Let H be an oracle and $\mathcal{D}$ be a class of description.

Notation 1 Denote by $\mathcal{D}_H$ the class of descriptions $\{\langle x, A\rangle \mapsto D(x, A \oplus H) \mid D \in \mathcal{D}\}$.

For example, BPP_H is the class of descriptions of the form (1), where M is a polynomial-time probabilistic oracle machine having an extra oracle H. And generally, if $\mathcal{K}$ is a class of machines of certain type, then $\mathcal{K}_H$ is the class of machines of that type having the extra oracle H.

Assume that for a superuniversum $\mathcal{V}$ and for an oracle H the following two assertions are true:

(a) There is a description K in $\mathcal{K}_H$ that is correct on any oracle in $\mathrm{l}(\mathcal{V})$ and such that there are no $L \in \mathcal{L}_H$ and $V \in \mathcal{V}$ such that $K(x, A) \leq L(x, A)$ for any x and any $A \in V$;

(b) for any $V \in \mathcal{V}$ and any $M \in \mathcal{M}_H$ being correct on any oracle in V there exist $N \in \mathcal{N}_H$ and $V' \in \mathcal{V}$ such that $V' \subset V$ and $M(x, A) \leq N(x, A)$ for any x and any $A \in V'$.

We claim that in this case there exists an oracle A such that $\mathcal{K}^A \not\leq \mathcal{L}^A$ and $\mathcal{M}^A \leq \mathcal{N}^A$.

Theorem 15 *Let $\mathcal{K}, \mathcal{L}, \mathcal{M}$, and $\mathcal{N}$ be countable classes of polynomial-local descriptions, H be an oracle and $\mathcal{V}$ be a superuniversum such that (a) and (b) are true. Then there exists an oracle A such that $\mathcal{K}^A \not\leq \mathcal{L}^A$ and $\mathcal{M}^A \leq \mathcal{N}^A$.*

5.2.1 A generalization of Theorem 15

A natural question is if the above theorem can be generalized to involve other assertions on relativized classes, for example, "$\mathcal{K}^A$ has a complete problem". The answer is affirmative.

The general approach is as follows. Suppose we wish to prove that there exists an oracle A such that an assertion $\varphi(A)$ on complexity classes is true. Suppose that the assertion $\varphi(Y)$ can be expressed by a closed formula in a first order language having atomic formulae of the form $P(x_1, \ldots, x_n, Y)$, where each variable x_i ranges over a countable set (possibly, different for different variables). (When we say that a formula is closed we do not consider Y as a variable.) For example, the assertion $\mathcal{K}^Y \not\leq \mathcal{L}^Y \wedge \mathcal{M}^Y \leq \mathcal{N}^Y$ can be expressed by the following

closed formula:

$$\exists K \in \mathcal{K}(\forall x \in \mathbf{B}^* K(x,Y) \neq \# \wedge \neg\exists L \in \mathcal{L} \forall x \in \mathbf{B}^* K(x,Y) \leq L(x,Y))$$
$$\wedge\neg\exists M \in \mathcal{M}(\forall x \in \mathbf{B}^* M(x,Y) \neq \# \wedge \neg\exists N \in \mathcal{N} \forall x \in \mathbf{B}^* M(x,Y) \leq N(x,Y))$$

having atomic formulae "$K(x,Y) = \#$", $K \in \mathcal{K}$, "$M(x,Y) = \#$", $M \in \mathcal{M}$, "$K(x,Y) \leq L(x,Y)$", $K \in \mathcal{K}$, $L \in \mathcal{L}$, "$M(x,Y) \leq N(x,Y)$", $M \in \mathcal{M}$, $N \in \mathcal{N}$.

Let $\mathcal{V}$ be a family of non-empty subsets of Ω, satisfying the Condition 2 in the definition of superuniversum. Call a set $W \subset \Omega$ *pseudo open* if for any V in $\mathcal{V}$ intersecting with W there exists V' in $\mathcal{V}$ such that $V' \subset V \cap W$. (Note that we do not require $\mathcal{V}$ to be a base of a topology.) Let us enrich our language by constants attached to all the elements from the domains of all the variables. Assume that the following condition is true:

(**) for any atomic formula $P(x_1, \ldots, x_n, Y)$ for any values $a_1, \ldots, a_n$ of x_1, $\ldots$, x_n the set $\{A \in \Omega \mid P(a_1, \ldots, a_n, A)\}$ is pseudo open.

Note that Condition 1 in the definition of superuniversum implies that all the sets $\{A \in \Omega \mid P(x,A) = \#\}$, $\{A \in \Omega \mid P(x,A) \neq \#\}$, $\{A \in \Omega \mid P(x,A) \leq Q(x,A)\}$, and $\{A \in \Omega \mid P(x,A) \not\leq Q(x,A)\}$ are pseudo open for any $x \in \mathbf{B}^*$ and any polynomial-local descriptions P and Q.

Let us define the forcing relation $V \vdash \varphi(Y)$, where $\varphi(Y)$ is a closed formula of the enriched language. We use induction on the number of symbols in $\varphi(Y)$. Assume that only connectivities $\wedge$ and $\neg$ and only the quantifier $\exists$ are used in formulae.

1. If $\varphi(Y)$ is atomic, then $V \vdash \varphi(Y)$ if $\varphi(A)$ is true for all $A \in V$.
2. $V \vdash \varphi(Y) \wedge \psi(Y)$ if $V \vdash \varphi(Y)$ and $V \vdash \psi(Y)$.
3. $V \vdash \exists z \in Z \varphi(Y,z)$ if there exists $b \in Z$ such that $V \vdash \varphi(Y,b)$.
4. $V \vdash \neg\varphi(Y)$ if $V' \nvdash \varphi(Y)$ for all $V' \subset V$.

Obviously, the forcing relation is monotone, that is, if $V \vdash \varphi(Y)$ and $V' \subset V$, then $V' \vdash \varphi(Y)$.

Lemma 6 *For any closed atomic formula $\varphi(Y)$, $V \vdash \neg\varphi(Y)$ iff the formula $\neg\varphi(A)$ is true for all $A \in V$.*

Proof. This easily follows from (**).

We claim that if $\mathcal{V}$ satisfies the Condition 1 in the definition of superuniversum, then the assertion (a) means that $\rfloor(\mathcal{V}) \vdash \mathcal{K}^Y \not\leq \mathcal{L}^Y$ and the assertion (b) means that $\rfloor(\mathcal{V}) \vdash \mathcal{M}^Y \leq \mathcal{N}^Y$, thus (a)$\wedge$(b) means that $\rfloor(\mathcal{V}) \vdash (\mathcal{K}^Y \not\leq \mathcal{L}^Y \wedge \mathcal{M}^Y \leq \mathcal{N}^Y)$. Indeed, by above lemma, Condition 1 implies that for all polynomial-local descriptions P and Q, for all $x \in \mathbf{B}^*$, both sets

$\{A \in \Omega \mid P(x,A) = \#\}$ and $\{A \in \Omega \mid \neg P(x,A) \le Q(x,A)\}$ are pseudo open and therefore $V \vdash \forall x \in \mathbf{B}^*\ P(x,Y) \ne \#$ iff P is correct on any oracle in V and $V \vdash \forall x \in \mathbf{B}^* P(x,Y) \le Q(x,Y)$ iff $P(x,A) \le Q(x,A)$ for any $A \in V$ and any x. Therefore, $\rfloor(V) \vdash \mathcal{K}^Y \not\le \mathcal{L}^Y$ means that there exists $K \in \mathcal{K}$ being correct on any oracle in $\rfloor(\mathcal{V})$ such that there exist no V and $L \in \mathcal{L}$ such that $K(x,A) \le L(x,A)$ for any $A \in V$ and any x. Likewise the assertion $\rfloor(V) \vdash \mathcal{M}^Y \le \mathcal{N}^Y$ means that for all $M \in \mathcal{M}$ and all V, M is incorrect on some oracle in V or there exist $V' \subset V$ and $N \in \mathcal{N}$ such that $M(x,A) \le N(x,A)$ for any $A \in V'$ and any x.

The following theorem generalizes Theorem 15.

Theorem 16 *Let $\eta(Y)$ be an assertion such that $V_0 \vdash \eta(Y)$ for some V_0 in $\mathcal{V}$. Then there exists A such that $\eta(A)$ is true.*

5.3 Non-uniformity

Thus, if we want to prove that there exists an oracle A such that $\mathcal{K}^A \not\le \mathcal{L}^A$ and $\mathcal{M}^A \le \mathcal{N}^A$, then we have to find a superuniversum $\mathcal{V}$ and an oracle H such that (a) and (b) are true. In this form the method is universal. Indeed, if there exists an oracle A such that $\mathcal{K}^A \not\le \mathcal{L}^A$ and $\mathcal{M}^A \le \mathcal{N}^A$ then both (a) and (b) hold for $\mathcal{V} = \{\{A\}\}$, $H = \emptyset$ or for $\mathcal{V} = \{\{\emptyset\}\}$, $H = A$.

Now we describe the very universum method. It does not use any notion of computability. It is not universal as we will see in Section 6.

Given a class $\mathcal{D}$ of descriptions define the nonuniform counterpart of the class $\mathcal{D}$ as follows.

Definition 7 The nonuniform counterpart of a class $\mathcal{D}$ of descriptions is the class

$$\text{n.u.}\mathcal{D} = \bigcup_{C \in \Omega} \mathcal{D}_C.$$

For particular classes of descriptions we can give equivalent definitions using no notion of computability. Let us do this for the classes considered in the previous section. To do this we have to define the notion of a Boolean decision tree and the notion of a branching program. More exactly, we define families of decision trees and families of branching programs; for the sake of brevity we omit the word "family".

A *Boolean decision tree* (or simply decision tree) is a pair

$$T = \langle \mathit{question}, \mathit{result} \rangle,$$

where *question* is a function from $\mathbf{B}^* \times \mathbf{B}^*$ into $\mathbf{B}^* \cup \{\$\}$, *result* is a function from $\mathbf{B}^* \times \mathbf{B}^*$ into $\{0,1,\$\}$. The decision tree T computes the description $\langle x, A\rangle \mapsto T(x,A)$ defined as follows: $T(x,A)$ is equal to the result returned by the program shown on Figure 3. (We assume that this program always halt.)

Call a decision tree *polynomial* if

1) there exists a polynomial p such that $|\mathit{question}(x,w)| \le p(|x|,|w|)$ for any

Figure 3:

```
begin
   w := Λ;
   while result(x, w) = $
      do
         y := question(x, w);
         b := A(y);
         w := wb;
      od
   return result(x, w);
end.
```

$x, w \in \mathbf{B}^*$ and any $A \in \Omega$ and
2) the program shown on Figure 3 halts after executing poly($|x|$) loops; it is important that in this case the program makes only poly($|x|$) queries to A.

It must be stressed that we do not require any computability of functions *question* and *result*. Obviously, if both functions *question* and *result* are polynomial-time computable, then the description $T(x, A)$ can be computed by a polynomial-time oracle machine.

We claim that a description D is in n.u.P iff D can be computed by a polynomial decision tree. This is quite easy: let D be in n.u.P, say $D(x, A) = M(x, A \oplus C)$, where M is a polynomial-time oracle machine and C is an oracle. Then let *question*(x, w) be the question to oracle A made by M after getting answers $w(1)$, $w(2)$, ..., $w(|w|)$ to previous questions made to A and let *result*(x, w) be the value returned by M after getting answers $w(1)$, $w(2)$, ..., $w(|w|)$ to questions made to A (if the specified values are undefined, then we consider them to be equal to \$). Obviously, the defined decision tree $T = \langle$*question*, *result*$\rangle$ is polynomial and $T(x, A) = M(x, A \oplus C) = D(x, A)$ for all x, A. Conversely, let a description D be computable by a polynomial decision tree, say $D(x, A) = T(x, A)$, where $T = \langle$*question*, *result*$\rangle$. Then take as C any oracle relative to which both functions *question* and *result* are computable in polynomial time. The program shown on Figure 3 defines then a P_C-machine computing $D(x, A)$.

It is easy to prove that a description D belongs to n.u.NP iff there exists a polynomial $p(n)$ and a description $D' \in$ n.u.P such that

$$D(x, A) = 1 \iff \exists y \in \mathbf{B}^{p(|x|)} D'(\langle x, y\rangle, A) = 1.$$

Likewise one can characterize the classes n.u.BPP, n.u.R, n.u.(NP$\cap$Co-NP) = n.u.NP $\cap$ n.u.Co-NP, n.u.(R $\cap$ Co-R) = n.u.R $\cap$ n.u.Co-R.

Figure 4:

```
begin
   s := initial_state(x);
   while result(s) = $
      do
         y := question(s);
         b := A(y);
         s := next_state(s, b);
      od
   return result(s)
end.
```

The nonuniform counterpart n.u.PSPACE of the class PSPACE can be defined as follows. A *branching program* is a quadruple

$$P = \langle initial\_state, next\_state, question, result \rangle,$$

where $initial\_state\colon \mathbf{B}^* \to \mathbf{B}^*$, $next\_state\colon \mathbf{B}^* \times \mathbf{B} \to \mathbf{B}^*$, $question\colon \mathbf{B}^* \to \mathbf{B}^* \cup \{\$\}$, $result\colon \mathbf{B}^* \to \{0, 1, \$\}$. The value $P(x, A)$ returned by the branching program P on input $\langle x, A \rangle$ is computed by the program shown on Figure 4. Call a branching program *polynomial-space* if there exists a polynomial p such that the length of word s does not exceed $p(|x|)$ in all the steps of execution of that program. It is easy to prove that

> $D \in$ n.u.PSPACE $\Leftrightarrow$ there exists a polynomial-space branching program P computing D.

Now we are able to present the second general theorem which is the essence of the universum method.

Suppose there exists a superuniversum $\mathcal{V}$ such that the following two assertions hold:

(a′) There is a description $K \in$ n.u.$\mathcal{K}$ that is correct on any oracle in $1(\mathcal{V})$ and such that there are no $L \in$ n.u.$\mathcal{L}$ and $V \in \mathcal{V}$ such that $K(x, A) \leq L(x, A)$ for any x and any $A \in V$;

(b′) for any $V \in \mathcal{V}$ and any description $M \in$ n.u.$\mathcal{M}$ being correct on any oracle in V there exist $N \in$ n.u.$\mathcal{N}$ and $V' \in \mathcal{V}$ such that $V' \subset V$ and $M(x, A) \leq N(x, A)$ for any x and any $A \in V'$.

Note that (a′) and (b′) are obtained respectively from (a) and (b) by replacing uniform classes relativized by H by the corresponding nonuniform classes.

The following theorem states that (a′) and (b′) imply the existence of an oracle H for which (a) and (b) are true, and hence the existence of an oracle A such that $\mathcal{K}^A \not\leq \mathcal{L}^A$ and $\mathcal{M}^A \leq \mathcal{N}^A$.

To prove that theorem we need some extra restrictions on classes $\mathcal{K}$, $\mathcal{L}$, $\mathcal{M}$ and $\mathcal{N}$. Let us formulate those restrictions. We would mention that they are rather cumbersome and therefore it may be better for the reader to skip them and to read the following theorem assuming that the classes $\mathcal{K}$, $\mathcal{L}$, $\mathcal{M}$ and $\mathcal{N}$ are some of the classes considered in the previous section.

Call a class $\mathcal{D}$ of descriptions *a polynomial complexity class* if

1. $\mathcal{D}$ consists of polynomial-local descriptions,
2. $\mathcal{D}$ includes the class P, and
3. there exist a larger class $\mathcal{D}'$ of descriptions, an enumeration D_0, D_1, D_2, ... of that class, and a function $c : \mathbf{N} \times \mathbf{B}^* \times \Omega \to \mathbf{N}$ (called complexity function) such that the following two conditions hold:
   (a) $\mathcal{D} = \{D_i \mid i \in \mathbf{N},\ c(i, x, A) \leq p(|x|)$ for some polynomial p and all $x, A\}$, and
   (b) for all polynomial-time oracle machines Q, R and S outputting respectively an integer, a binary word and an element of $\mathbf{B}$ there exists m such that

$$D_m(x, A) = D_{Q(x,A)}(R(x, A), S_x^A)$$
$$\text{and } c(m, x, A) \leq \text{poly}\Big(Q(x, A) + |x| + c(Q(x, A), R(x, A), S_x^A)\Big),$$

   where S_x^A stands for the oracle $\{y \in \mathbf{B}^* \mid S(x, y, A) = 1\}$. In particular, if D_i is in $\mathcal{D}$, then the description $\langle x, A\rangle \mapsto D_i(R(x, A), S_x^A)$ is in $\mathcal{D}$, too.

It is easy to verify that all the classes considered in the previous section are polynomial complexity classes. In the next theorem we assume the classes $\mathcal{K}$, $\mathcal{L}$, $\mathcal{M}$ and $\mathcal{N}$ to be polynomial complexity classes.

Theorem 17 *If a superuniversum $\mathcal{V}$ satisfies (a′) and (b′) then $\mathcal{V}$ satisfies (a) and (b) for some oracle H.*

6 When the universum method cannot be used

We say that the universum method can be applied to prove that there exists an oracle A such that $\mathcal{K}^A \not\leq \mathcal{L}^A$ and $\mathcal{M}^A \leq \mathcal{N}^A$ if there exists a superuniversum $\mathcal{V}$ such that (a′) and (b′) are true. In this section we present two theorems that cannot be proven by the universum method. We omit their proofs, which can be found in [MV 94].

Theorem 18 [Ko 89] *There exists an oracle A such that*

$$\mathbf{P}^A = \mathbf{NP}^A \neq \mathbf{PSPACE}^A.$$

Theorem 19 [R 78] *There exists an oracle A such that*

$$\mathbf{P}^A \neq \mathbf{R}^A = \mathbf{PSPACE}^A.$$

In fact, we can prove that the following corollaries of Theorems 18 and 19 cannot be proven by the universum method.

Corollary 20 *There exists an oracle A such that*

$$\mathbf{NP}^A = \mathbf{Co\text{-}NP}^A \neq \mathbf{PSPACE}^A$$

Corollary 21 *There exists an oracle A such that*

$$\mathbf{P}^A \neq \mathbf{PSPACE}^A \text{ and } \mathbf{Co\text{-}NP}^A \subset \mathbf{R}^A.$$

Both Theorems 18 and 19 can be proven by a method that may be called "coding method" because, in those proofs, some difficult-to-compute information is encoded via oracle values, to ensure the truth of positive assertion ($\mathcal{M}^A \leq \mathcal{N}^A$). To prove Theorem 18, one needs a lower bound by [H 86, Y 85] on complexity of computation of PARITY function by means of AND,OR-circuits of bounded depth, which has a rather complicated proof. Theorem 19 was proved in the paper [R 78] (in fact, the weaker assertion that $\mathbf{P} \neq \mathbf{R} = \mathbf{NP}$ under some oracle was proved there but the proof is good also for our case).

Theorem 22 *The universum method cannot be applied to prove Corollary 20, i.e., there exists no superuniversum $\mathcal{V}$ such that (a') and (b') are true for $\mathcal{K}$ = PSPACE, $\mathcal{L}$ = NP and $\mathcal{M}$ = Co-NP, $\mathcal{N}$ = NP.*

Theorem 23 *Corollary 21 cannot be proven by the universum method, that is, there exists no superuniversum $\mathcal{V}$ such that (a') and (b') hold for $\mathcal{K}$ = PSPACE, $\mathcal{L}$ = P and $\mathcal{M}$ = Co-NP, $\mathcal{N}$ = R.*

Acknowledgments

The authors would like to thank Alexander Razborov and Alexander Shen for helpful comments and anonymous referee for many corrections.

References

[BGS 75] T. Baker, J. Gill and R. Solovay. "Relativization of P=?NP Question", *SIAM Journal on Computing*, 1975, vol. 4, No. 4, pp. 431–442.

[IN 88] R. Impagliazzo and M. Naor. "Decision trees and downward closures", *Third Conference on Structure in Complexity Theory*, 1988, pp. 29–38.

[BI 87] M. Blum and R. Impagliazzo. "General Oracle and Oracle Classes", *28th Annual IEEE Symposium on Foundation of Computer Science*, 1987, pp. 118–126.

[FFKL 93] S. Fenner, L. Fortnow, S. A. Kurtz and L. Li. " An oracle builder's toolkit", *Eight Annual Conference on Structure in Complexity Theory*, May 1993, pp. 120–131.

[FR 94] L. Fortnow and J. Rogers. "Separability and One-way Functions", Manuscript, 1994.

[GS 88] J. Grollman and A. L. Selman. "Complexity measures for public-key cryptosystems", *SIAM Journal on Computing,* 1988, vol. 17, No. 2, pp. 309–335.

[H 86] J. Hastad. "Almost optimal lower bounds for small depth circuits", in: S. Micali, ed., *Advances in Computer Research,?* vol. 5: Randomness and Computation (JAI Press, Greenwich, CT); see also *Computational Limitations for Small Depth Circuits* (MIT Press, Cambridge, MA, 1986).

[HI 86] J. Hartmanis and N. Immerman. "On complete problems for **NP** ∩ **Co-NP**", *International Colloquium on Automata, Languages and Programming* 1985, pp. 250–259. Springer-Verlag *Lecture Notes in Computer Science,* vol. 194, 1985.

[Ko 89] Ker-I Ko. "Relativized polynomial-time hierarchies having exactly k levels", *SIAM Journal on Computing*, 1989, vol. 18, No. 2, pp. 392–408.

[L 83] C. Lautemann. "BPP and the polynomial hierarchy", *Information Processing Letters*, 1983, vol. 17, No. 4, pp. 215–217.

[MV 94] An. A. Muchnik and N. K. Vereshchagin. "A General Method to Construct Oracles Realizing Given Relationships between Complexity Classes", Technical Report 500, University of Rochester, 1994.

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

[R 78] C. Rackoff. "Relativized Questions Involving Probabilistic Algorithms", *10th Annual ACM Symposium on Theory of Computing,* 1978, pp. 338–342. *Journal of ACM* **29** (1982), 261–268.

[S 82] M. Sipser. "On Relativizations and the Existence of Complete Sets", *International Colloquium on Automata, Languages and Programming*, 1982. *Lecture Notes in Computer Science*, 1982, vol. 140, pp. 523-531.

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

[Y 85] A. C. Yao. "Separating the polynomial time hierarchy by oracles", *26th Annual IEEE Symposium on Foundations of Computer Science*, 1985, pp. 1–10.