RELATIVIZED HELPING OPERATORS

PATRIZIO CINTIOLI

Dipartimento di Matematica e Informatica, Università di Camerino,
Via Madonna delle Carceri, 62032 Camerino, Italy
patrizio.cintioli@unicam.it

(Received 19 June 2005)

Abstract: Schöning and Ko respectively introduced the concepts of *helping* and *one-side-helping*, and then defined new operators, $\mathbf{P}_{\mathrm{help}}(\cdot)$ and $\mathbf{P}_{1-\mathrm{help}}(\cdot)$, acting on classes of sets $\mathcal{C}$ and returning classes of sets $\mathbf{P}_{\mathrm{help}}(\mathcal{C})$ and $\mathbf{P}_{1-\mathrm{help}}(\mathcal{C})$. A number of results have been obtained on this subject, principally devoted to understanding how wide the $\mathbf{P}_{\mathrm{help}}(\mathcal{C})$ and $\mathbf{P}_{1-\mathrm{help}}(\mathcal{C})$ classes are. For example, it seems that the $\mathbf{P}_{\mathrm{help}}(\cdot)$ operator contracts $\mathbf{NP}\cap co\mathbf{NP}$, while the $\mathbf{P}_{1-\mathrm{help}}(\cdot)$ operator enlarges $\mathbf{UP}$. To better understand the relative power of $\mathbf{P}_{1-\mathrm{help}}(\cdot)$ versus $\mathbf{P}_{\mathrm{help}}(\cdot)$ we propose to search, for every relativizable class $\mathcal{D}$ containing $\mathbf{P}$, the largest relativizable class $\mathcal{C}$ containing $\mathbf{P}$ such that for every oracle B $\mathbf{P}^B_{\mathrm{help}}(\mathcal{C}^B)\subseteq\mathbf{P}^B_{1-\mathrm{help}}(\mathcal{D}^B)$. In [1] it has been observed that $\mathbf{P}_{\mathrm{help}}(\mathbf{UP}\cap co\mathbf{UP})=\mathbf{P}_{1-\mathrm{help}}(\mathbf{UP}\cap co\mathbf{UP})$, and this is true in any relativized world. In this paper we consider the case of $\mathcal{D}=\mathbf{UP}\cap co\mathbf{UP}$ and demonstrate the existence of an oracle A for which $\mathbf{P}^A_{\mathrm{help}}(\mathbf{UP}^A_2\cap co\mathbf{UP}^A_2)$ is not contained in $\mathbf{P}^A_{1-\mathrm{help}}(\mathbf{UP}^A\cap co\mathbf{UP}^A)$. We also prove that for every integer $k\geq 2$ there exists an oracle A such that $\mathbf{P}^A_{\mathrm{help}}(\mathbf{UP}^A_k\cap \mathrm{co}\mathbf{UP}^A_k)\not\subseteq\mathbf{UP}^A_k$.

Keywords: oracle Turing machines, structural complexity, relativized separations, helping

1. Robust machines

The notion of *a robust algorithm* was defined about twenty years ago in [2] by Uwe Schöning. He was mainly interested in questions of *problem solving* and the man-machine interaction. Let us describe in detail the setting he had in mind.

Let us assume that a machine has to solve a computational problem and can rely on the assistance of a human operator. Thus, during computation the machine can ask the human operator, who answers according to his/her knowledge, experience, intuition, *etc.* Of course, this may help the machine, but it would be desiderable for the machine to preserve some independence from such assistance and be able to provide the right output even when the human operator is wrong, or a new human operator, with different knowledge and intuition, is asked. At the same time, it is reasonable to expect that right answers and reliable experts speed up computation and thus improve the machine's efficiency.

The formal approach proposed by Uwe Schöning to handle these settings was that of *robust Turing machines* [2]. Let us recall this notion, dealing with oracle Turing machines, *i.e.* Turing machines M which can query an "oracle" (meaning a set A of words). Let us assume for the sake of simplicity that the following hold:

i) the queries the machine raises to oracle A are of membership nature, *i.e.* they have the form of "is y in A?" where y is a generic word on the alphabet of A;
ii) the problems handed by the machine are decisional, *i.e.* their outputs are "yes" or "no"; we also assume that every computation of the machine has an end.

Let M be a machine as described in ii) and let Σ be the alphabet of M. Then, $L(M)$ denotes the language of the words y in Σ^* for which M outputs "yes". If M is an oracle Turing machine and A is a given oracle, then M^A is the M machine with the additional licence of querying oracle A.

Definition 1. i) *A deterministic oracle Turing machine M is called robust if for every oracle $A \subseteq \Sigma^*$, $L(M^\emptyset) = L(M^A)$.*
ii) *An oracle A is called a helper for a robust machine M if M^A runs in polynomial time. In this case we also say that language $L(M^\emptyset) = L(M^A)$ is helped by A.*

A weaker notion is that of *one-side helper*.

Definition 2. *A set A is called a one-side helper for a robust machine M if there exists a polynomial p such that for every $x \in L(M^\emptyset)$ $M^A(x)$ runs in time $p(|x|)$ where $|x|$ is the length of x.*

2. Helping Operators

Definition 3. *If A is a set, then a language L is in $\mathbf{P}_{\mathrm{help}}(A)$ if and only if there exists a robust Turing machine M with helper A such that $L = L(M^\emptyset)$; if $\mathcal{C}$ is a class of sets, $\mathbf{P}_{\mathrm{help}}(\mathcal{C}) := \bigcup_{A\in\mathcal{C}} \mathbf{P}_{\mathrm{help}}(A)$.*

Definition 4. *If A is a set, then a language L is in $\mathbf{P}_{1-\mathrm{help}}(A)$ if and only if there exists a robust machine M one-side helped by A such that $L = L(M^\emptyset)$; if $\mathcal{C}$ is a family of sets, $\mathbf{P}_{1-\mathrm{help}}(\mathcal{C}) := \bigcup_{A\in\mathcal{C}} \mathbf{P}_{1-\mathrm{help}}(A)$.*

Hence $\mathbf{P}_{\mathrm{help}}(\cdot)$ and $\mathbf{P}_{1-\mathrm{help}}(\cdot)$ can be viewed as operators acting on classes of sets $\mathcal{C}$ and returning classes of sets $\mathbf{P}_{\mathrm{help}}(\mathcal{C})$ and $\mathbf{P}_{1-\mathrm{help}}(\mathcal{C})$. Schöning and Ko have established the following basic characterizations:

Theorem 1. (*Schöning* [2]) $\mathbf{P}_{\mathrm{help}}(\mathbf{NP}) = \mathbf{NP} \cap co\mathbf{NP}$.

Theorem 2. (*Ko* [3]) $\mathbf{P}_{1-\mathrm{help}}(\mathbf{NP}) = \mathbf{NP}$.

Actually, the proofs of these two theorems apply to any relativized world and show that for every oracle A $\mathbf{P}^A_{\mathrm{help}}(\mathbf{NP}^A) = \mathbf{NP}^A \cap co\mathbf{NP}^A$. Indeed, this relativized framework is our main interest throughout this paper, previously described in detail in [1]. But, just to explain the A exponent in the last statement, let us recall that $\mathbf{NP}^A$ is the class of languages accepted by non deterministic polynomial-time oracle Turing machines with oracle A, while $co\mathbf{NP}^A$ is the class of languages $\overline{L}$ with $L \in \mathbf{NP}^A$. Introducing $\mathbf{P}^A_{\mathrm{help}}(\cdot)$ and $\mathbf{P}^A_{1-\mathrm{help}}(\cdot)$ is more laborious and refers to machines M with two oracles, A and B, to the $L(M^{A,B})$ languages they decide and in particular to A-robust machines M (meaning that, for every additional B oracle, $L(M^{A,\emptyset}) = L(M^{A,B})$). Then, for every relativizable class $\mathcal{C}$, $\mathbf{P}^A_{\mathrm{help}}(\mathcal{C}^A)$ is the class of languages accepted by A-robust machines helped by oracles in $\mathcal{C}^A$, and $\mathbf{P}^A_{1-\mathrm{help}}(\mathcal{C}^A)$ is defined similarly, referring to one-side helpers. As has been said, [1] provides more detail. $\mathbf{P}_{\rm help}(\cdot)$ and $\mathbf{P}_{1-{\rm help}}(\cdot)$ have been intensively studied in several papers, including [1–7]. For instance, one can see that both of them increase with respect to inclusion, which means that for any two relativizable classes $\mathcal{C}$ and $\mathcal{D}$, and for every oracle A, $\mathcal{C}^A \subseteq \mathcal{D}^A$ implies $\mathbf{P}^A_{\rm help}(\mathcal{C}^A) \subseteq \mathbf{P}^A_{\rm help}(\mathcal{D}^A)$ and $\mathbf{P}^A_{1-{\rm help}}(\mathcal{C}^A) \subseteq \mathbf{P}^A_{1-{\rm help}}(\mathcal{D}^A)$. By the way, this is just what the Turing operator $\mathbf{P}(\cdot)$ does. However, $\mathbf{P}_{\rm help}(\cdot)$ and $\mathbf{P}_{1-{\rm help}}(\cdot)$ sometimes exhibit peculiar behaviour, different from that of $\mathbf{P}(\cdot)$ or other operators. For instance, $\mathbf{P}(\cdot)$ satisfies the *non-contraction* property, that is, for every relativizable class $\mathcal{C}$, $\mathcal{C}^A \subseteq \mathbf{P}^A(\mathcal{C}^A)$ for all oracles A, while the helping operators do not exhibit this property. Indeed, it is almost possible to show that they exhibit the opposite property, *i.e.* the following condition: Whenever a relativizable class $\mathcal{C}$ is closed under $\leq^p_T$, then $\mathbf{P}^A_{\rm help}(\mathcal{C}^A) \subseteq \mathcal{C}^A$ and $\mathbf{P}^A_{1-{\rm help}}(\mathcal{C}^A) \subseteq \mathcal{C}^A$ for every oracle A (see [3]). Furthermore, in some cases and for certain oracles, the inclusion is proper. However, for an arbitrary relativizable class $\mathcal{C}$ the relationship between $\mathcal{C}$ and $\mathbf{P}_{\rm help}(\mathcal{C})$, or $\mathbf{P}_{1-{\rm help}}(\mathcal{C})$ is not uniform, and yet to be completely understood. Let us offer a list of examples likely to summarize all the possible known behaviours of $\mathbf{P}_{\rm help}(\cdot)$ and $\mathbf{P}_{1-{\rm help}}(\cdot)$.

Examples for $\mathbf{P}_{\rm help}(\cdot)$:

1. (contraction) if $\mathcal{C} = \mathbf{NP} \cap co\mathbf{NP}$, then for any A $\mathbf{P}^A_{\rm help}(\mathcal{C}^A) \subseteq \mathcal{C}^A$, and for some A this inclusion is proper (see [7]);
2. (equality) if $\mathcal{C} = \mathbf{UP} \cap co\mathbf{UP}$, then for any A $\mathbf{P}^A_{\rm help}(\mathcal{C}^A) = \mathcal{C}^A$; actually $\mathbf{UP} \cap co\mathbf{UP}$ is the only class known so far with this property, apart from the trivial case of $\mathcal{C} = \mathbf{P}$;
3. (incomparability) if $\mathcal{C} = \mathbf{UP}$, then there exist A, B such that $\mathbf{P}^A_{\rm help}(\mathcal{C}^A) \not\subseteq \mathcal{C}^A$ (see [7]) and $\mathbf{P}^B_{\rm help}(\mathcal{C}^B) \not\supseteq \mathcal{C}^B$ (take B such that $\mathbf{NP}^B \cap co\mathbf{NP}^B \not\supseteq \mathbf{UP}^B$).

Examples for $\mathbf{P}_{1-{\rm help}}(\cdot)$:

1. (contraction) if $\mathcal{C} = \mathbf{NP} \cap co\mathbf{NP}$, then for any A $\mathbf{P}^A_{1-{\rm help}}(\mathcal{C}^A) \subseteq \mathcal{C}^A$, and for some A this inclusion is proper (see [7]);
2. (equality) if $\mathcal{C} = \mathbf{UP} \cap co\mathbf{UP}$, then for any A $\mathbf{P}^A_{1-{\rm help}}(\mathcal{C}^A) = \mathcal{C}^A$ (see [1]); this is also true when $\mathcal{C} = \mathbf{NP}$, and of course in the trivial case of $\mathcal{C} = \mathbf{P}$;
3. (enlargement) if $\mathcal{C} = \mathbf{UP}$, then for every A $\mathbf{P}^A_{1-{\rm help}}(\mathcal{C}^A) \supseteq \mathcal{C}^A$ [3] and for some B $\mathbf{P}^B_{\rm help}(\mathcal{C}^B) \supset \mathcal{C}^B$ [4];
4. (incomparability) if $\mathcal{C} = \mathbf{UP}_k \cap co\mathbf{UP}_k$ with $k \geq 2$, then there exists A such that $\mathbf{P}^A_{1-{\rm help}}(\mathcal{C}^A) \not\supseteq \mathcal{C}^A$ [1] and there exists B such that $\mathbf{P}^B_{1-{\rm help}}(\mathcal{C}^B) \not\subseteq \mathcal{C}^B$ (see Theorem 5 and Corollary 2 of this paper).

It should be noted that a complete characterization of the $\mathcal{C}$ classes of fixed behaviour (contraction, incomparability, *etc.*) with respect to $\mathbf{P}_{\rm help}(\cdot)$ or $\mathbf{P}_{1-{\rm help}}(\cdot)$ is often lacking. It should also be noted that the previous list does not contain any "enlargement" example for $\mathbf{P}_{\rm help}(\cdot)$; indeed, no class satisfying this property is known at the present time.

The starting point of this paper is the following simple observation: the examples listed above show that the $\mathbf{P}_{\rm help}(\cdot)$ operator contracts $\mathbf{NP} \cap co\mathbf{NP}$ with respect to some oracles, and that $\mathbf{P}_{1-{\rm help}}(\cdot)$ enlarges $\mathbf{UP}$ with respect to some oracles. So, it might be the case that the $\mathbf{P}_{\rm help}(\cdot)$ operator really contracts the $\mathbf{NP} \cap co\mathbf{NP}$ class and the $\mathbf{P}_{1-{\rm help}}(\cdot)$ operator really enlarges the $\mathbf{UP}$ class. Hence, we could ask if

$\mathbf{P}^A_{\mathrm{help}}(\mathbf{NP}^A \cap co\mathbf{NP}^A)$ is contained in $\mathbf{P}^A_{1-\mathrm{help}}(\mathbf{UP}^A)$ for every A. This leads to a more general line of research, aiming at understanding the relative power of $\mathbf{P}_{1-\mathrm{help}}(\cdot)$ versus $\mathbf{P}_{\mathrm{help}}(\cdot)$. In detail, given a fixed relativizable class $\mathcal{D}$ containing $\mathbf{P}$, we ask which is the largest relativizable class $\mathcal{C}$ such that for every A $\mathbf{P}^A_{\mathrm{help}}(\mathcal{C}^A) \subseteq \mathbf{P}^A_{1-\mathrm{help}}(\mathcal{D}^A)$. Looking for such class $\mathcal{C}$ makes sense since, for any two relativizable classes $\mathcal{C}_1$ and $\mathcal{C}_2$ and for any oracle A, $\mathbf{P}^A_{\mathrm{help}}(\mathcal{C}^A_1) \cup \mathbf{P}^A_{\mathrm{help}}(\mathcal{C}^A_2) = \mathbf{P}^A_{\mathrm{help}}(\mathcal{C}^A_1 \cup \mathcal{C}^A_2)$. To formulate the problem more precisely , let us restrict ourselves to a family $\mathcal{F}$ of classes defined as follows. $\mathcal{F}$ is the smallest family containing the basic counting classes $\{\mathbf{UP}_k\}_{k \geq 1}$, $\mathbf{NP}$, and such that for any classes $\mathcal{C}_1$ and $\mathcal{C}_2$ in $\mathcal{F}$, $\mathcal{C}_1 \cup \mathcal{C}_2$, $\mathcal{C}_1 \cap \mathcal{C}_2$ and $co\mathcal{C}_1$ are also in $\mathcal{F}$. This choice is not restrictive, and the family of classes can be extended or altered by allowing other operators or starting from different basic classes. The $(\mathcal{F}, \subseteq)$ structure is partially ordered with maximum element $\mathbf{NP} \cup co\mathbf{NP}$ and minimum element $\mathbf{UP} \cap co\mathbf{UP}$. Of course, we assume that $\mathbf{P} \neq \mathbf{UP} \neq \mathbf{UP}_2 \neq \cdots \neq \mathbf{NP}$. In fact, if $\mathbf{P} = \mathbf{NP}$ then $\mathcal{F}$ is trivialized and reduced to a unique point. However, it should be borne in mind that for some class $\mathcal{D}$ in $\mathcal{F}$ our problem may not have a solution $\mathcal{C}$ in $\mathcal{F}$. Nevertheless, when $\mathcal{D} = \mathbf{NP}$, then the largest $\mathcal{C} \in \mathcal{F}$ class solving our problem is $\mathcal{C} = \mathbf{NP} \cup co\mathbf{NP}$, as $\mathbf{NP}^A \cap co\mathbf{NP}^A = \mathbf{P}^A_{\mathrm{help}}(\mathbf{NP}^A \cup co\mathbf{NP}^A) \subseteq \mathbf{P}^A_{1-\mathrm{help}}(\mathbf{NP}^A) = \mathbf{NP}^A$, for every oracle A (see [2, 3]). In this paper we continue this investigation for the case when $\mathcal{D} = \mathbf{UP} \cap co\mathbf{UP}$ and show that, for a suitable oracle A, $\mathbf{P}^A_{\mathrm{help}}(\mathbf{UP}^A_2 \cap co\mathbf{UP}^A_2)$ is not contained in $\mathbf{P}^A_{1-\mathrm{help}}(\mathbf{UP}^A \cap co\mathbf{UP}^A)$. We leave open the question of existence of an oracle B for which $\mathbf{P}^B_{\mathrm{help}}(\mathbf{UP}^B_2 \cap co\mathbf{UP}^B)$ is not contained in $\mathbf{P}^B_{1-\mathrm{help}}(\mathbf{UP}^B \cap co\mathbf{UP}^B)$, but observe that, if such oracle B exists, then our problem for $\mathcal{D} = \mathbf{UP} \cap co\mathbf{UP}$ is solved by $\mathcal{C} = \mathbf{UP} \cap co\mathbf{UP}$ (see Section 5).

In section 3 notation is introduced and some preliminaries are provided. Section 4 contains the main results of our paper. Section 5 is devoted to concluding remarks. The reader's familiarity with basic concepts of the complexity theory and formal languages has been assumed (for example, see [8–10]).

3. Notation and preliminaries

For the sake of simplicity, we refer to the usual binary alphabet $\{0,1\}$ with an additional technical symbol $\sharp$.

Let Σ denote the $\{0,1,\sharp\}$ alphabet formed in this way. Let Σ^* denote the set of finite words on Σ and, for every positive integer n, let Σ^n be the set of words of length n in Σ^*. For $x \in \Sigma^*$:

- $|x|$ denotes the length of x,
- for every $1 \leq i \leq |x|$, $(x)_i$ is the ith symbol of x,
- for $a \in \{0,1,\sharp\}$, $|x|_a$ is the number of occurrences of symbol a in x,
- for every positive integer n and for every $a \in \{0,1,\sharp\}$, a^n is the word of exactly n repetitions of symbol a.

For $u,v \in \Sigma^*$, we say that u is a prefix of the word uv and v is a suffix of the word uv.

As has been mentioned, if $\mathcal{C}_1$ and $\mathcal{C}_2$ are two classes of languages, then $\mathcal{C}_1 \cup \mathcal{C}_2 = \{L : L \in \mathcal{C}_1 \text{ or } L \in \mathcal{C}_2\}$, $\mathcal{C}_1 \cap \mathcal{C}_2 = \{L : L \in \mathcal{C}_1 \text{ and } L \in \mathcal{C}_2\}$, $co\mathcal{C} = \{\overline{L} : L \in \mathcal{C}\}$. It should be noted that this complement operator behaves in a manner different from that of the analogous set-theoretic operator. For example, it is easy to see that for $\mathcal{C}_1,\mathcal{C}_2$ classes of languages $co(\mathcal{C}_1 \cup \mathcal{C}_2) = co\mathcal{C}_1 \cup co\mathcal{C}_2$ and $co(\mathcal{C}_1 \cap \mathcal{C}_2) = co\mathcal{C}_1 \cap co\mathcal{C}_2$.

An acceptor or transducer machine M is *polynomial-time* if there exists a polynomial p such that for every $x \in \Sigma^*$, M on input x halts in at most $p(|x|)$ steps.

Definition 5. [11] *For any set B and for every integer $k \geq 1$, a language L is in $\mathbf{UP}_k^B$ if there exists a non deterministic polynomial-time oracle acceptor N such that, for every finite string x, if $x \in L$ then $N^B(x)$ has at most k accepting paths.*

Definition 6.

1. *A function $\tau : \Sigma^* \to \Sigma^*$ is polynomially bit-computable with oracle E if there exist two functions $f : \Sigma^* \times \mathbb{N} \to \Sigma$ and $g : \Sigma^* \to \Sigma$, both polynomial-time computable with oracle E, such that, for every string x, $\tau(x) = f(x,1)f(x,2)\cdots f(x,g(x))$.*
2. *Let (A,B) be a pair of disjoint languages. For any oracle E, we denote by $\mathrm{C}(A,B)^E$ the class of all the languages L for which there exists a function τ polynomially bit-computable with oracle E such that, for every string x,*

$$x \in L \Leftrightarrow \tau(x) \in A \quad \text{and} \quad x \notin L \Leftrightarrow \tau(x) \in B.$$

[12, 13] characterize most complexity classes between **P** and **PSPACE** in terms of pairs of languages. For instance, let $A_{\mathrm{UP}_2 \cap \mathrm{coUP}_2} = \{x \in \{1,\sharp\}^* : 1 \leq |x|_1 \leq 2\}$ and $B_{\mathrm{UP}_2 \cap \mathrm{coUP}_2} = \{x \in \{0,\sharp\}^* : 1 \leq |x|_0 \leq 2\}$, then it is possible to check that $\mathrm{C}(A_{\mathrm{UP}_2 \cap \mathrm{coUP}_2}, B_{\mathrm{UP}_2 \cap \mathrm{coUP}_2})^E = \mathbf{UP}_2^E \cap co\mathbf{UP}_2^E$ for all oracles E. More generally, for every $k \geq 2$, let $A_{\mathrm{UP}_k \cap \mathrm{coUP}_k} = \{x \in \{\sharp,1\}^* : 1 \leq |x|_1 \leq k\}$ and $B_{\mathrm{UP}_k \cap \mathrm{coUP}_k} = \{x \in \{\sharp,0\}^* : 1 \leq |x|_0 \leq k\}$. Then the $(A_{\mathrm{UP}_k \cap \mathrm{coUP}_k}, B_{\mathrm{UP}_k \cap \mathrm{coUP}_k})$ pair characterizes the $\mathbf{UP}_k \cap \mathrm{co}\mathbf{UP}_k$ class, in the sense that $C(A_{\mathrm{UP}_k \cap \mathrm{coUP}_k}, B_{\mathrm{UP}_k \cap \mathrm{coUP}_k})^E = \mathbf{UP}_k^E \cap \mathrm{co}\mathbf{UP}_k^E$ for every oracle E.

Let us also mention the following characterization of the **UP** class via a suitable pair of languages $(A_{\mathrm{UP}}, B_{\mathrm{UP}})$ [12, 13]: For every oracle E, $\mathbf{UP}^E = C(A_{\mathrm{UP}}, B_{\mathrm{UP}})^E$ where $A_{\mathrm{UP}} := \{x \in \{0,1\}^* : |x|_1 = 1\}$ and $B_{\mathrm{UP}} := \{0\}^*$.

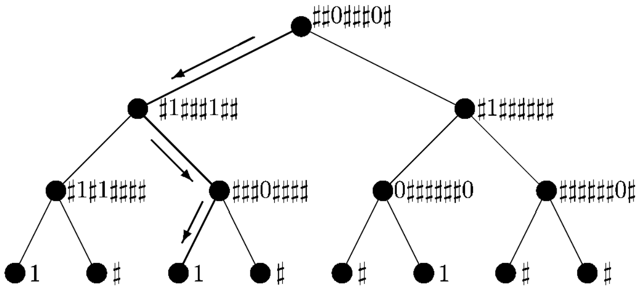
Let us also at this stage introduce languages $A_{\mathrm{UP}_k} := \{x \in \{0,1\}^* : 1 \leq |x|_1 \leq k\}$ and $B_{\mathrm{UP}} := \{0\}^*$ and recall that their pair $(A_{\mathrm{UP}_k}, B_{\mathrm{UP}_k})$ characterizes $\mathbf{UP}_k$.

[12, 13] also provide a basic tool for obtaining relativized separations between complexity classes defined by pairs of languages. This tool relies on a new type of reducibility between pairs of languages, called polylogarithmic-time bit-reduction.

Definition 7. *A pair of languages (A,B) is polylogarithmic-time bit-reducible to a pair (A',B'), in short $(A,B) \leq_m^{pl} (A',B')$, iff there exist two polylogarithmic-time computable*[1] *functions $f : \Sigma^* \times \mathbb{N} \to \Sigma$ and $g : \Sigma^* \to \Sigma$ such that, for every string x,*

$$x \in A \ \Leftrightarrow f(x,1)f(x,2)\cdots f(x,g(x)) \in A'$$

1. A function is *polylogarithmic-time computable* if it can be computed in polylogarithmic time by a deterministic Turing machine with an additional tape on which the machine can write down an index i and then receive the i^{th} symbol of the input string (if i is greater than the length of the input string the machine receives a fixed special symbol).



♯♯0♯♯♯0♯ ♯1♯♯♯1♯♯ ♯1♯♯♯♯♯♯ ♯1♯1♯♯♯♯ ♯♯♯0♯♯♯♯ 0♯♯♯♯♯♯0 ♯♯♯♯♯♯0♯ 1♯1♯♯1♯♯

Figure 1. A helping tree of depth 3 and its encoding; the correct path is that labelled with arrows

and

$$x \in B \Leftrightarrow f(x,1)f(x,2)\cdots f(x,g(x)) \in B'.$$

Theorem 3. ([12, 13]) *Let (A,B) and (A',B') be two pairs of languages. Then there exists an oracle E for which $\mathrm{C}(A,B)^E \not\subseteq \mathrm{C}(A',B')^E$ if and only if $(A,B) \not\leq_m^{pl} (A',B')$.*

Let us single out a pair of languages $(A_{\mathrm{PhU}_2}, B_{\mathrm{PhU}_2})$ characterizing $\mathbf{P}_{\mathrm{help}}(\mathbf{UP}_2 \cap co\mathbf{UP}_2)$ *i.e.* satisfies $\mathrm{C}(A_{\mathrm{PhU}_2}, B_{\mathrm{PhU}_2})^E = \mathbf{P}_{help}^E(\mathbf{UP}_2^E \cap co\mathbf{UP}_2^E)$ for all oracles E. Let t be a complete and ordered binary tree whose inner nodes are labelled by strings in $A_{\mathrm{UP}_2 \cap \mathrm{coUP}_2} \cup B_{\mathrm{UP}_2 \cap \mathrm{coUP}_2}$, all of the same length $2^{(\mathrm{depth\,of}\,t)}$, and whose leaves are labelled by symbols in $\{0,1,\sharp\}$. We call such a tree t a *helping tree*. For any helping tree t there is a special path, that we call the *correct path* of t, defined as follows: starting from the root, at any node n we go to its left son if the label of n is in B_{UP_2} and we go to its right son if the label is in A_{UP_2}. Any helping tree t can be encoded by a string $x = z_1 z_2 \cdots z_{2^h-1} y_1 y_2 \cdots y_{2^h}$ where h is the depth of t, strings z_i are the labels of the inner nodes (all of length 2^h), symbols y_i are the labels of the leaves, and all the labels are concatenated by the order of a breadth first search (see Figure 1). At this point, we can define the $(A_{\mathrm{PhU}_2}, B_{\mathrm{PhU}_2})$ pair in the following way: $A_{\mathrm{PhU}_2} = \{x \mid x$ encodes a helping tree t whose leaf labels are in $\{1,\sharp\}$ and the label of the leaf of the correct path of t is $1\}$ and $B_{\mathrm{PhU}_2} = \{x \mid x$ encodes a helping tree t whose leaf labels are in $\{0,\sharp\}$ and the label of the leaf of the correct path of t is $0\}$. Then, for every oracle E, $\mathrm{C}(A_{\mathrm{PhU}_2}, B_{\mathrm{PhU}_2})^E = \mathbf{P}_{\mathrm{help}}^E(\mathbf{UP}_2^E \cap co\mathbf{UP}_2^E)$.

4. Main result

Now we can state the main result of our paper.

Theorem 4. *There exists an oracle E such that $\mathbf{P}_{\mathrm{help}}^E(\mathbf{UP}_2^E \cap co\mathbf{UP}_2^E)$ is not contained in $\mathbf{UP}^E$.*

Proof. As $\mathbf{UP}^E = C(A_{\mathrm{UP}}, B_{\mathrm{UP}})^E$ and $\mathbf{P}_{\mathrm{help}}^E(\mathbf{UP}_2^E \cap co\mathbf{UP}_2^E) = C(A_{\mathrm{PhU}_2}, B_{\mathrm{PhU}_2})^E$ for every E, it suffices to show that $C(A_{\mathrm{UP}}, B_{\mathrm{UP}})^E$ does not include $C(A_{\mathrm{PhU}_2}, B_{\mathrm{PhU}_2})^E$ for some E, equivalently that $(A_{\mathrm{PhU}_2}, B_{\mathrm{PhU}_2}) \not\leq_m^{pl} (A_{\mathrm{UP}}, B_{\mathrm{UP}})$ (due to Theorem 3). So, we have to exclude the existence of two polylogarithmic-time computable functions $f:\{0,1,\sharp\}^*\times\mathbb{N}\to\{0,1,\sharp\}$ and $g:\{0,1,\sharp\}^*\to\mathbb{N}$ such that, $\forall x\in\{0,1,\sharp\}^*$:

$$x\in A_{PhU_2}\Leftrightarrow f(x,1)f(x,2)\cdots f(x,g(x))\in A_{UP}$$

and

$$x\in B_{PhU_2}\Leftrightarrow f(x,1)f(x,2)\cdots f(x,g(x))\in B_{UP}.$$

Assume towards a contradiction that these functions f and g exist. Let R and T be two polylogarithmic-time Turing transducers computing f and g, respectively. The running time of R and T is bounded by a given polynomial q, which means that for every $x\in\{0,1,\sharp\}^*$ and every $i\le T(x)$, both $T(x)$ and $R(x,i)$ perform at most $q(\log|x|)$ steps. Let $n=2^{2d}$ for some positive integer d be such that $\frac{\sqrt{n}}{2q(\log n)}>q(\log n)+1$. From now on, we put the following, for every string x of length $n=2^{2d}$:

- $\sigma(x):=R(x,1)R(x,2)\cdots R(x,T(x))$,
- $nodes(x):=$ the prefix of x of length $2^{2d}-2^d$,
- $leaves(x):=$ the suffix of x of length 2^d.

For example, if x encodes the helping tree in Figure 1, then $d=3$, $nodes(x)=\sharp\sharp0\sharp\sharp\sharp0\sharp\sharp1\sharp\sharp\sharp1\sharp\sharp\sharp1\sharp\sharp\sharp\sharp\sharp\sharp\sharp1\sharp1\sharp\sharp\sharp\sharp\sharp\sharp\sharp0\sharp\sharp\sharp\sharp0\sharp\sharp\sharp\sharp\sharp\sharp0\sharp\sharp\sharp\sharp\sharp\sharp0\sharp$ and $leaves(x)=1\sharp1\sharp\sharp1\sharp\sharp$. We claim that there exists a word $\overline{x}$, $|\overline{x}|=n$, with $\overline{x}\in A_{\mathrm{PhU}_2}$ and $R(\overline{x},1)R(\overline{x},2)\cdots R(\overline{x},T(\overline{x}))\notin A_{\mathrm{UP}}$ (which contradicts the former assumption on f and g).

Given any word $x\in\{0,1,\sharp\}^*$, a *position* of x is any integer i with $1\le i\le|x|$. Before starting with the construction of word $\overline{x}$, we make a preliminary hypothesis on the Turing transducer R:

"for every string $x\in A_{\mathrm{PhU}_2}$ of length n and for every $i\le T(x)$,
if $R(x,i)=1$ then $R(x,i)$ reads at least a symbol '1' in $leaves(x)$".

In fact, if $R(x,i)=1$ but $R(x,i)$ does not read any symbol '1' in $leaves(x)$, then consider the string $\tilde{x}$ with $|\tilde{x}|=|x|$, $nodes(\tilde{x})=nodes(x)$ and $leaves(\tilde{x})=\sharp^{m-1}0\sharp^{\sqrt{n}-m}$ for some position m of x which is not read by $R(x,i)$: $\tilde{x}$ is in B_{PhU_2} but $\sigma(\tilde{x})$ is not in B_{UP}, since $R(\tilde{x},i)=1$, which contradicts the latter assumption on f and g. Let r be the least positive integer greater than or equal to $\frac{\sqrt{n}}{2q(\log n)}$; we obtain word $\overline{x}$ by a procedure of r stages, each containing in its turn r substages.

Begin procedure

Stage 1. Let us put:

- $Q_0^1:=\{s\in\mathbb{N}:T(\sharp^n)$ reads the position $s\le n$ of $\sharp^n\}$,
- $p_1:=$ the least positive integer such that $n-\sqrt{n}+p_1\notin Q_0^1$,
- $b_1b_2\cdots b_d:=$ the sequence in $\{0,1\}^d$ associated with the correct path leading to the leaf p_1.

Substage 1.1. For every i, $1\le i\le m=2^d-1$, let $s_i^{(1)}$ be the minimal positive integer $\le 2^d$ such that $2^d(i-1)+s_i^{(1)}\notin Q_0^1$. Take $x_{1.1}\in\{0,1,\sharp\}^n$ such that $leaves(x_{1.1})=\sharp^{p_1-1}1\sharp^{\sqrt{n}-p_1}$ and $nodes(x_{1.1})=z_1^{(1)}z_2^{(1)}\cdots z_m^{(1)}$, where for every $1\le i\le m$

$$z_i^{(1)}=\begin{cases}\sharp^{s_i^{(1)}-1}b_j\sharp^{\sqrt{n}-s_i^{(1)}} & \text{if node } i \text{ is the } j^{\text{th}} \text{ node}\\ & \text{of the correct path leading to leaf } p_1,\\ \sharp^{\sqrt{n}} & \text{otherwise.}\end{cases}$$

The word $x_{1.1}$ is in A_{PhU_2}, so there exists $i_1\ge1$ such that $R(x_{1.1},i_1)=1$.

- $Q_1^1 := Q_0^1 \cup \{s : 1 \le s \le n$ and s is a position of $x_{1.1}$ read by either $R(x_{1.1}, i_1)$ or $R(\sharp^n, i_1)\}$,
- $I_1 := \{i_1\}$.

  Go to the next substage.

Substage $1.k$, $1 < k \le r$. For every i, $1 \le i \le m = 2^d - 1$, let $s_i^{(k)}$ be the minimal positive integer $\le 2^d$ such that $2^d(i-1) + s_i^{(k)} \notin Q_{k-1}^1$. Take $x_{1.k} \in \{0,1,\sharp\}^n$ such that $leaves(x_{1.k}) = \sharp^{p_1 - 1} 1 \sharp^{\sqrt{n} - p_1}$ and $nodes(x_{1.k}) = z_1^{(k)} z_2^{(k)} \cdots z_m^{(k)}$, where for every $1 \le i \le m$

$$z_i^{(k)} = \begin{cases} \sharp^{s_i^{(k)} - 1} b_j \sharp^{\sqrt{n} - s_i^{(k)}} & \text{if node } i \text{ is the } j^{\text{th}} \text{ node of the correct path leading to leaf } p_1, \\ \sharp^{\sqrt{n}} & \text{otherwise.} \end{cases}$$

The word $x_{1.k}$ is in A_{PhU_2}, so there exists $i_k \ge 1$ such that $R(x_{1.k}, i_k) = 1$. Put

- $Q_k^1 := Q_{k-1}^1 \cup \{s : 1 \le s \le n$ and s is a position of $x_{1.k}$ read by either $R(x_{1.k}, i_k)$ or $R(\sharp^n, i_k)\}$,
- $I_k := I_{k-1} \cup \{i_k\}$.

  Go to the next substage.

  Let $I_r := \{i_1, i_2, \ldots, i_r\}$ be the set of all the indices obtained at the end of substage $1.r$.

Claim 1. *There exists an index $j_1 \in I_r$ such that $R(x_{1.j_1}, j_1)$ outputs 1 without reading positions of $nodes(x_{1.j_1})$ containing a symbol in $\{0,1\}$.*

The proof of this claim is provided in the Appendix of the paper.

Stage $h > 1$, $h \le r$. Let

- $j_1, j_2, \ldots, j_{h-1}$ be such that, for $t = 1, 2, \ldots, h-1$, $R(x_{1.j_t}, j_t)$ outputs 1 without reading any positions of $nodes(x_{1.j_t})$ containing a symbol in $\{0,1\}$.

  Put

- $Q_{-1}^h := Q_r^{h-1}$,
- $p_h =$ the least positive integer such that $n - \sqrt{n} + p_h \notin Q_{-1}^h$,
- $b_1 b_2 \cdots b_t =$ the sequence of symbols in $\{0,1\}$ associated to the correct path leading to the leaf p_h.

Substage $h.1$. For every i, $1 \le i \le m = 2^d - 1$, let $s_i^{(1)}$ be the minimal positive integer $\le 2^d$ such that $2^d(i-1) + s_i^{(1)} \notin Q_{-1}^h$. Take $x_{h.1} \in \{0,1,\sharp\}^n$ such that $leaves(x_{h.1}) = \sharp^{p_h - 1} 1 \sharp^{\sqrt{n} - p_h}$ and $nodes(x_{h.1}) = z_1^{(1)} z_2^{(1)} \cdots z_m^{(1)}$, with $m = 2^d - 1$, where for every $1 \le i \le m$

$$z_i^{(1)} = \begin{cases} \sharp^{s_i^{(1)} - 1} b_j \sharp^{\sqrt{n} - s_i^{(1)}} & \text{if node } i \text{ is the } j^{\text{th}} \text{ node of the correct path leading to leaf } p_h, \\ \sharp^{\sqrt{n}} & \text{otherwise.} \end{cases}$$

The word $x_{h.1}$ is in A_{PhU_2}, so there exists $i_1 \ge 1$ such that $R(x_{h.1}, i_1) = 1$. Let us pose

- $Q_1^h := Q_{-1}^h \cup \{s : 1 \le s \le n$ and s is a position of $x_{h.1}$ read by either $R(x_{h.1}, i_1)$ or $R(\sharp^n, i_1)\}$,
- $I_1 := \{i_1\}$.

Go to the next substage.

Substage $h.k$, $1<k\leq r$. For every i, $1\leq i\leq m=2^d-1$, let $s_i^{(k)}$ be the minimal positive integer $\leq 2^d$ such that $2^d(i-1)+s_i^{(k)}\notin Q_{k-1}^h$. Take $x_{h.k}\in\{0,1,\sharp\}^n$ such that $leaves(x_{h.k})=\sharp^{p_h-1}1\sharp^{\sqrt{n}-p_h}$ and $nodes(x_{h.k})=z_1^{(k)}z_2^{(k)}\cdots z_m^{(k)}$, with $m=2^d-1$, where for every $1\leq i\leq m$

$$z_i^{(k)}=\begin{cases}\sharp^{s_i^{(k)}-1}b_j\sharp^{\sqrt{n}-s_i^{(k)}} & \text{if node } i \text{ is the } j^{\text{th}} \text{ node of the correct path leading to leaf } p_h,\\ \sharp^{\sqrt{n}} & \text{otherwise.}\end{cases}$$

The word $x_{h.k}$ is in A_{PhU_2}, so there exists $i_k\geq 1$ such that $R(x_{h.k},i_k)=1$. Put

- $Q_k^h:=Q_{k-1}^h\cup\{s:1\leq s\leq n$ and s is a position of $x_{h.k}$ read by either $R(x_{h.k},i_k)$ or $R(\sharp^n,i_k)\}$,
- $I_k:=I_{k-1}\cup\{i_k\}$.

Go to the next substage.

Claim 2. *There exists an index $j_h\in I_r:=\{i_1,i_2,\ldots,i_r\}$ such that $R(x_{h.j_h},i_{j_h})$ outputs 1 without reading positions of $nodes(x_{h.j_h})$ containing a symbol in $\{0,1\}$.*

(Claim 2 can be proven in a way similar to Claim 1. We omit its proof).

Go to the next stage.

End Procedure

Let $j_1,j_2,\ldots,j_r$ be the indices obtained at the end of stage r, and let $p_1<p_2<\cdots<p_r$ be the positions of $leaves(x_{1.j_1})$, $leaves(x_{2.j_2}),\cdots,leaves(x_{r.j_r})$ read by $R(x_{1.j_1},j_1),R(x_{2.j_2},j_2),\cdots,R(x_{r.j_r},j_r)$, respectively. It should be noted that indices $j_1,j_2,\ldots,j_r$ are pairwise distinct, because for every $s=1,2,\ldots,r$, the computation $R(x_{s.j_s},j_s)$ reads the position p_s of $leaves(x_{s.j_s})$, which is not read by $R(x_{h.j_h},j_h)$ for $h=1,2,\ldots,s-1$.

Let us consider the $R(x_{r.j_r},j_r)$ computation. This computation performs at most $q(\log n)$ steps, and since $r>q(\log n)+1$ there exists an s in $\{1,2,\ldots,r-1\}$ such that the $R(x_{r.j_r},j_r)$ computation does not read the position p_s of $leaves(x_{r.j_r})$, which is read by $R(x_{s.j_s},j_s)$.

Let $\overline{x}$ be the word of length n built as follows:

- $leaves(\overline{x})=\sharp^{p_s-1}1\sharp^{p_r-p_s-1}1\sharp^{\sqrt{n}-p_r}$, and
- $nodes(\overline{x})$ encodes a correct path $c_1c_2\ldots c_d\in\{0,1\}^d$ leading to leaf p_s, where each symbol c_i is in a position of $nodes(\overline{x})$ which is read neither by $R(x_{s.j_s},j_s)$ nor by $R(x_{r.j_r},j_r)$.

It follows that $R(\overline{x},j_r)=1$ and $R(\overline{x},j_s)=1$, with $\overline{x}\in A_{PhU_2}$, so $|\sigma(\overline{x})|_1\geq 2$, that is $\sigma(\overline{x})\notin A_{UP}$.

■

Corollary 1. *There exists an oracle A such that $\mathbf{P}^A_{\mathrm{help}}(\mathbf{UP}_2^A\cap co\mathbf{UP}_2^A)$ is not contained in $\mathbf{P}^A_{1-\mathrm{help}}(\mathbf{UP}^A\cap co\mathbf{UP}^A)$.*

Proof. For every oracle E $\mathbf{P}^E_{1-\mathrm{help}}(\mathbf{UP}^E\cap co\mathbf{UP}^E)=\mathbf{UP}^E\cap co\mathbf{UP}^E\subseteq\mathbf{UP}^E$. So, if A is the oracle of the theorem above, then $\mathbf{P}^A_{\mathrm{help}}(\mathbf{UP}_2^A\cap co\mathbf{UP}_2^A)\not\subseteq\mathbf{P}^A_{1-\mathrm{help}}(\mathbf{UP}^A\cap co\mathbf{UP}^A)$. ■

We can now show the result stated in Section 2 (see Example 4 concerning $\mathbf{P}_{1-\mathrm{help}}(\cdot)$).

Theorem 5. *For every integer* $k \geq 2$ *there exists an oracle* A *such that* $\mathbf{P}^A_{\mathrm{help}}(\mathbf{UP}^A_k \cap \mathrm{co}\mathbf{UP}^A_k) \not\subseteq \mathbf{UP}^A_k$.

Proof. (*Sketch*). The proof is similar to that of Theorem 4, with minor changes. First of all, one builds a pair of languages $(A_{\mathrm{PhU}_k}, B_{\mathrm{PhU}_k})$ characterizing $\mathbf{P}_{\mathrm{help}}(\mathbf{UP}_k \cap \mathrm{co}\mathbf{UP}_k)$. For $k \geq 2$ the words in A_{PhU_k} and in B_{PhU_k} can be described as those in A_{PhU_2} and in B_{PhU_2}, respectively, but the inner nodes are labelled by words in $A_{\mathrm{UP}_k \cap \mathrm{coUP}_k} \cup B_{\mathrm{UP}_k \cap \mathrm{coUP}_k}$, where $A_{\mathrm{UP}_k \cap \mathrm{coUP}_k} = \{x \in \{\sharp, 1\}^* : 1 \leq |x|_1 \leq k\}$ and $B_{\mathrm{UP}_k \cap \mathrm{coUP}_k} = \{x \in \{\sharp, 0\}^* : 1 \leq |x|_0 \leq k\}$. As has been mentioned, the $(A_{\mathrm{UP}_k \cap \mathrm{coUP}_k}, B_{\mathrm{UP}_k \cap \mathrm{coUP}_k})$ pair characterizes the $\mathbf{UP}_k \cap \mathrm{co}\mathbf{UP}_k$ class. Moreover, let us take $(A_{\mathrm{UP}_k}, B_{\mathrm{UP}_k})$ the pair of languages and recall that the $(A_{\mathrm{UP}_k}, B_{\mathrm{UP}_k})$ pair characterizes $\mathbf{UP}_k$. The next step is to prove that $(A_{\mathrm{PhU}_k}, B_{\mathrm{PhU}_k}) \not\leq^{pl}_m (A_{\mathrm{UP}_k}, B_{\mathrm{UP}_k})$. For this purpose, let us choose $n = 2^{2d}$ such that $\frac{\sqrt{n}}{2q(\log n)} > kq(\log n) + k$. We then define a procedure as in Theorem 4, but with $r > kq(\log n) + k$ stages, each containing in its turn r substages, and so obtain $k+1$ indices $i_1, i_2, \ldots, i_{k+1}$ and a word $\overline{x}$ in A_{PhU_k} such that $R(\overline{x}, i_1) = R(\overline{x}, i_2) = \cdots = R(\overline{x}, i_{k+1}) = 1$. Then $|R(\overline{x}, i_1) R(\overline{x}, i_2) \cdots R(\overline{x}, i_{k+1})| > k$, hence $R(\overline{x}, i_1) R(\overline{x}, i_2) \cdots R(\overline{x}, i_{k+1})$ is not in A_{UP_k}. ■

Corollary 2. *For every integer* $k \geq 2$ *there exists an oracle* A *such that* $\mathbf{P}^A_{\mathrm{help}}(\mathbf{UP}^A_k \cap \mathrm{co}\mathbf{UP}^A_k) \not\subseteq \mathbf{UP}^A_k \cap \mathrm{co}\mathbf{UP}^A_k$.

5. Concluding remarks

We do not know if our separation result solves the question introduced at the end of section 1 when $\mathcal{D} = \mathbf{UP} \cap co\mathbf{UP}$. Let us discuss this point now.

Let $\mathcal{C}$ denote the largest class for which $\mathbf{P}_{\mathrm{help}}(\mathcal{C}) \subseteq \mathbf{P}_{1-\mathrm{help}}(\mathbf{UP} \cap co\mathbf{UP})$ in every relativized world, if any.

Case 1. There exists an oracle A for which $\mathbf{P}^A_{\mathrm{help}}(\mathbf{UP}^A_2 \cap co\mathbf{UP}^A) \not\subseteq \mathbf{P}^A_{1-\mathrm{help}}(\mathbf{UP}^A \cap co\mathbf{UP}^A)$. In this case $\mathcal{C} = \mathbf{UP} \cap co\mathbf{UP}$. In fact, $\mathbf{P}^A_{\mathrm{help}}(\mathbf{UP}^A_2 \cap co\mathbf{UP}^A) \not\subseteq \mathbf{P}^A_{1-\mathrm{help}}(\mathbf{UP}^A \cap co\mathbf{UP}^A)$ implies $\mathbf{P}^A_{\mathrm{help}}(\mathbf{UP}^A \cap co\mathbf{UP}^A_2) \not\subseteq \mathbf{P}^A_{1-\mathrm{help}}(\mathbf{UP}^A \cap co\mathbf{UP}^A)$. In fact $co(\mathbf{UP}_2 \cap co\mathbf{UP}) = \mathbf{UP} \cap co\mathbf{UP}_2$ and $\mathbf{P}^X_{\mathrm{help}}(\mathcal{E}^X) = \mathbf{P}^X_{\mathrm{help}}(co\mathcal{E}^X)$ for every relativizable class $\mathcal{E}$ and for every oracle X. Furthermore, the unique class in $\mathcal{F}$ included in both $\mathbf{UP}_2 \cap co\mathbf{UP}$ and $\mathbf{UP} \cap co\mathbf{UP}_2$ is $(\mathbf{UP}_2 \cap co\mathbf{UP}) \cap (\mathbf{UP} \cap co\mathbf{UP}_2) = \mathbf{UP} \cap co\mathbf{UP}$.

Case 2. For some $k \geq 2$ $\mathbf{P}^A_{\mathrm{help}}(\mathbf{UP}^A_k \cap co\mathbf{UP}^A) \subseteq \mathbf{P}^A_{1-\mathrm{help}}(\mathbf{UP}^A \cap co\mathbf{UP}^A)$ for every oracle A and there exists an oracle B for which $\mathbf{P}^B_{\mathrm{help}}(\mathbf{UP}^B_{k+1} \cap co\mathbf{UP}^B) \not\subseteq \mathbf{P}^B_{1-\mathrm{help}}(\mathbf{UP}^B \cap co\mathbf{UP}^B)$. Reasoning as in Case 1, $\mathcal{C} = (\mathbf{UP}_k \cup co\mathbf{UP}) \cup (\mathbf{UP} \cap co\mathbf{UP}_k)$.

Case 3. For every integer $k \geq 1$ and every oracle A it holds that $\mathbf{P}^A_{\mathrm{help}}(\mathbf{UP}^A_k \cap co\mathbf{UP}^A) \subseteq \mathbf{P}^A_{1-\mathrm{help}}(\mathbf{UP}^A \cap co\mathbf{UP}^A)$ and there exists an oracle B for which $\mathbf{P}^B_{\mathrm{help}}(\mathbf{NP}^B \cap co\mathbf{UP}^A) \not\subseteq \mathbf{P}^B_{1-\mathrm{help}}(\mathbf{UP}^B \cap co\mathbf{UP}^B)$. In this case no class in $\mathcal{F}$ works as $\mathcal{C}$.

Case 4. For every oracle A it holds that $\mathbf{P}^A_{\mathrm{help}}(\mathbf{NP}^A \cap co\mathbf{UP}) \subseteq \mathbf{P}^A_{1-\mathrm{help}}(\mathbf{UP}^A \cap co\mathbf{UP}^A)$. In this case $\mathcal{C} = (\mathbf{NP} \cap co\mathbf{UP}) \cup (\mathbf{UP} \cap co\mathbf{NP})$.

Appendix

Proof of Claim 1.

Let us first prove that there are two indices $i_u, i_v \in \{i_1, i_2, \ldots, i_r\}$ with $u \neq v$ and $i_u = i_v$. It is enough to prove that $|\{i_1, i_2, \ldots, i_r\}| \leq q(\log n) + 1$, since $r > q(\log n) + 1$. Let us suppose towards a contradiction that $|\{i_1, i_2, \ldots, i_r\}| > q(\log n) + 1$ and consider $R(x_{1.r}, i_r)$. This computation reads at most $q(\log n)$ positions of $x_{1.r}$, so there is at least an index $i_k \in \{i_1, i_2, \ldots, i_r\}$, $i_k \neq i_r$, such that $R(x_{1.r}, i_r)$ does not read any positions of $nodes(x_{1.k})$ read by $R(x_{1.k}, i_k)$. At the same time, the procedure ensures that $R(x_{1.k}, i_k)$ does not read the positions of $nodes(x_{1.r})$ read by $R(x_{1.r}, i_r)$. So, let us construct $\tilde{x} \in \{0, 1, \sharp\}^*$ such that

- $leaves(\tilde{x}) = \sharp^{p_1 - 1} 1 \sharp^{\sqrt{n} - p_1}$ and
- $nodes(\tilde{x})$ is the word of length $2^{2d} - 2^d$ such that for every $1 \leq i \leq 2^{2d} - 2^d$,
  - $(nodes(\tilde{x}))_i = 1$ if $(nodes(x_k))_i = 1$ or $(nodes(x_r))_i = 1$,
  - $(nodes(\tilde{x}))_i = 0$ if $(nodes(x_k))_i = 0$ or $(nodes(x_r))_i = 0$,
  - $(nodes(\tilde{x}))_i = \sharp$ if $(nodes(x_k))_i = \sharp$ and $(nodes(x_r))_i = \sharp$.

It should be noted that no index i, $i \leq 2^{2d} - 2^d$, satisfies $(nodes(x_k))_i = 0$ and $(nodes(x_r))_i = 1$ or vice versa. Then $\tilde{x} \in A_{\mathrm{PhU}_2}$ but $R(\tilde{x}, i_k) = 1$ and $R(\tilde{x}, i_r) = 1$ with $i_k \neq i_r$, that is $|\sigma(\tilde{x})|_1 \geq 2$. This contradicts the fact that the $(A_{\mathrm{PhU}_2}, B_{\mathrm{PhU}_2})$ pair is polylogarithmic-time bit-reducible to the $(A_{\mathrm{UP}}, B_{\mathrm{UP}})$ pair.

So $|\{i_1, i_2, \ldots, i_r\}| \leq q(\log n)$, which implies that there are two integers $u < v$ with $i_u = i_v$. Let $m = 2^d - 1$. We know that $R(x_{1.u}, i_u)$ does not read positions $s_1^{(v)}, 2^d + s_2^{(v)}, 2^d \cdot 2 + s_3^{(v)}, \ldots, 2^d(m-1) + s_m^{(v)}$ of $nodes(x_{1.v})$. But $i_u = i_v$, so $R(x_{1.v}, i_v)$ does not read positions $s_1^{(v)}, 2^d + s_2^{(v)}, 2^d \cdot 2 + s_3^{(v)}, \ldots, 2^d(m-1) + s_m^{(v)}$ of $nodes(x_{1.v})$ either, hence the claim follows provided we put $j_1 := i_v$, where $x_{1.j_1} = x_{1.v}$. ■

References

[1] Cintioli P and Silvestri R 1997 *Inf. Proc. Let.* **61** 189

[2] Schöning U 1985 *Theor. Comp. Sci.* **40** 57

[3] Ko K 1987 *Theor. Comp. Sci.* **52** 15

[4] Cai J, Hemachandra L A and Vyskŏc J 1993 *Complexity Theory* (Ambos-Spies K, Homer S and Schöning U, Eds.), Cambridge University Press, pp. 101–146

[5] Hemachandra L 1993 *Proc. 20th Int. Colloquium on Automata, Languages and Programming; Lecture Notes in Computer Science* **700** 189

[6] Ogihara M 1995 *Inf. Proc. Lett.* **54** 41

[7] Cintioli P and Silvestri R 1997 *Theor. Comp. Systems* **30** 165

[8] Bovet D P and Crescenzi P 1994 *Introduction to the Theory of Complexity*, Prentice-Hall, Englewood Cliffs, NJ

[9] Balcázar J L, Díaz J and Gabarró J 1988 *Structural Complexity I*, vol. 1, Springer Verlag

[10] Johnson D S 1990 *Handbook of Theoretical Computer Science* (van Leeuwen J, Ed.) **A**, pp. 67–161

[11] Beigel R, Hemachandra L and Wechsung G 1989 *Proc. 4th IEEE Structure in Complexity Theory Conference*, University of Oregon, Eugene, Oregon, USA, pp. 225–227

[12] Bovet D P, Crescenzi P and Silvestri R 1992 *Theor. Comp. Sci.* **104** 263

[13] Vereshchagin N K 1994 *Russian Academy of Sciences. Izvestiya Mathematics (AMS)* **42** 261