Theoretical Informatics and Applications
Theoret. Informatics Appl. **35** (2001) 367–377

THE HELPING HIERARCHY

Patrizio Cintioli[1] and Riccardo Silvestri[2]

Abstract. Schöning [14] introduced a notion of helping and suggested the study of the class $\mathrm{P_{help}}(\mathcal{C})$ of the languages that can be helped by oracles in a given class $\mathcal{C}$. Later, Ko [12], in order to study the connections between helping and "witness searching", introduced the notion of self-helping for languages. We extend this notion to classes of languages and show that there exists a self-helping class that we call SH which contains all the self-helping classes. We introduce the Helping hierarchy whose levels are obtained applying a constant number of times the operator $\mathrm{P_{help}}(\cdot)$ to the set of all the languages. We show that the Helping hierarchy collapses to the k-th level if and only if SH is equal to the k-th level. We give characterizations of all the levels and use these to construct a relativized world in which the Helping hierarchy is infinite.

Mathematics Subject Classification. 68Q15, 68Q05, 03D15.

1. Introduction

Schöning [14] proposed a notion of oracle set helping the computation of a language. He introduced the basic concept of *robust machine*, that is, a deterministic oracle Turing machine that always recognizes the same language, independent of the oracle that is used. The oracle is only for possibly speeding up the computation. Precisely, a language L is said to be recognized in polynomial time with the help of an oracle E if there is a robust machine M recognizing L such that M with oracle E runs in polynomial time. The first basic result obtained by Schöning states that the class $\mathrm{P_{help}}$ of languages recognized in polynomial time with the help of some oracle is equal to $\mathrm{NP} \cap \text{co-NP}$. This result ruled out the most interesting possibility: the existence of important problems (*e.g.* NP-complete problems) that can be helped by some oracle. However, it raised the natural question of characterizing the class

[1] Dipartimento di Matematica e Fisica, Università di Camerino, Via Madonna delle Carceri, 62032 Camerino (MC), Italy; e-mail: `cintioli@campus.unicam.it`

[2] Dipartimento di Scienze dell'Informazione, Università di Roma "La Sapienza", Via Salaria 113, 00198 Roma, Italy; e-mail: `silvestri@dsi.uniroma1.it`

$\mathrm{P_{help}}(\mathcal{C})$ of languages recognized in polynomial time with the help of some oracle in $\mathcal{C}$, where $\mathcal{C}$ is a given class of oracle languages. Along this direction, Schöning [14] showed that $\mathrm{P_{help}(NP)} = \mathrm{NP} \cap \text{co-NP}$ and that robust machines helped by oracles in BPP recognize languages in ZPP, that is $\mathrm{P_{help}(BPP)} \subseteq \mathrm{ZPP}$. Later, Ko [12] proved that $\mathrm{P_{help}(UP \cap co\text{-}UP)} = \mathrm{UP \cap co\text{-}UP}$, and more recently Ogihara [13] showed that, for every prime power k, $\mathrm{P_{help}(MOD}_k\mathrm{P)} = \mathrm{MODZ}_k\mathrm{P} \cap \text{co-}\mathrm{MODZ}_k\mathrm{P}$. However, besides the classes NP, UP$\cap$co-UP, and $\mathrm{MOD}_k\mathrm{P}$, it is not known the exact characterization of $\mathrm{P_{help}}(\mathcal{C})$ for any other class $\mathcal{C}$. Whether $\mathrm{P_{help}(BPP)} = \mathrm{ZPP}$ or not (see the survey [11]) is an open question. It was shown in [8] the existence of a relativized world in which $\mathrm{ZPP} \not\subseteq \mathrm{P_{help}(BPP)}$. Similarly, it is not known whether $\mathrm{P_{help}(UP)} = \mathrm{UP \cap co\text{-}UP}$, and in [8] a relativized world was constructed in which $\mathrm{P_{help}(UP)} \not\subseteq \mathrm{Few}$ (see also [7]).

As pointed out by Ko, if L is recognized by a robust machine M with the help of an oracle H, then M has to find, through queries to H, a witness for x, for each instance x (for a formalization see [2, 3]). In order to investigate these connections, Ko introduced the concept of self-helping: a language L is a *self-helper* if $L \in \mathrm{P_{help}}(\{L\})$. Ko studied the relationship between self-helping and the notion of self-reducibility, showing that if a language L and its complement are both disjunctive self-reducible then L is a self-helper. Recently, Arvind [1] proved that self-helping implies Ptime self-witnessing (see [10]). We extend the concept of self-helping to classes: a class of languages $\mathcal{C}$ is a *self-helping class* if $\mathcal{C} \subseteq \mathrm{P_{help}}(\mathcal{C})$. A nice feature of this extended concept is that there exists a self-helping class that we call SH which contains all the self-helping classes (see Sect. 2). Obviously, $\mathrm{SH} \subseteq \mathrm{NP}\cap\text{co-NP}$ and, since UP$\cap$co-UP is a self-helping class, $\mathrm{UP \cap co\text{-}UP} \subseteq \mathrm{SH}$. The question SH =?NP $\cap$ co-NP is equivalent to ask whether or not the ability of deciding languages that admit witnesses (*i.e.* languages in NP $\cap$ co-NP) can help "witness searching". We introduce the Helping hierarchy whose k-th level is obtained by applying $k + 1$ times the operator $\mathrm{P_{help}}(\cdot)$ to the set of all the languages. It results that the level zero is equal to NP $\cap$ co-NP and, unlike common hierarchies, the Helping hierarchy is downward, that is, each level contains the next one. It turns out that SH is included in the intersection of all the levels of the hierarchy and the hierarchy collapses to level k if and only if SH is equal to level k. In Section 3 we give a characterization of each level of the Helping hierarchy. Then, in Section 4, we use these characterizations to obtain our main result, that is the construction of a relativized world in which the Helping hierarchy is infinite. This means that proving the collapse of the Helping hierarchy (in particular, the collapse to level zero, that is, SH = NP $\cap$ co-NP), if possible at all, is very hard. In fact standard techniques, that relativize like direct simulation, cannot be used.

2. The helping hierarchy

We assume familiarity with standard complexity theory notations and complexity classes (*e.g.* see [4–6]).

Our definition of the notion of helping is formally different from the usual one [14] so that it focus on the language which is helped rather than on the language that helps. Of course, the notion in itself remains equal to the usual one.

Definition 2.1.

1. A *robust machine* is a deterministic oracle Turing machine M such that, for every oracle E, $M^E(x)$ halts for all inputs x and $L(M^E) = L(M^\emptyset)$.
2. A language L is *recognized in polynomial time with the help of oracle H* if there exists a robust machine M and a polynomial p such that $L(M^H) = L$ and $M^H(x)$ halts in $p(|x|)$ steps for all inputs x.
3. For each class of languages $\mathcal{C}$, let $\mathrm{P}_{\mathrm{help}}(\mathcal{C})$ be the class of languages recognized in polynomial time with the help of some oracle in $\mathcal{C}$. In particular let $\mathrm{P}_{\mathrm{help}}$ denote $\mathrm{P}_{\mathrm{help}}(2^{\{0,1\}^*})$, where $2^{\{0,1\}^*}$ denotes the powerset of $\{0,1\}^*$.

Ko [12] introduced the concept of self-helping for languages and we extend it to classes of languages.

Definition 2.2.

1. A language L is a *self-helper* if $L \in \mathrm{P}_{\mathrm{help}}(\{L\})$.
2. A class of languages $\mathcal{C}$ is a *self-helping class* if $\mathcal{C} \subseteq \mathrm{P}_{\mathrm{help}}(\mathcal{C})$.

For instance, from a result in [12] it follows that $\mathrm{UP} \cap \text{co-UP}$ is a self-helping class. Moreover, denoting by SELFHELPER the class of all the self-helpers, it is easy to see that SELFHELPER and $\mathrm{P}_{\mathrm{help}}(\mathrm{SELFHELPER})$ are self-helping classes.

Let SH be the union of all the self-helping classes. It is not hard to see that $\mathrm{P}_{\mathrm{help}}(\mathrm{SH}) = \mathrm{SH}$. Thus, SH is the largest self-helping class and the following inclusions hold

$$\mathrm{UP} \cap \text{co-UP} \subseteq \mathrm{P}_{\mathrm{help}}(\mathrm{SELFHELPER}) \subseteq \mathrm{SH} \subseteq \mathrm{NP} \cap \text{co-NP},$$

where the first inclusion derives from a result in [12]. We believe that a very interesting question is whether or not $\mathrm{SH} = \mathrm{NP} \cap \text{co-NP}$. In fact, if $\mathrm{SH} \neq \mathrm{NP} \cap \text{co-NP}$ then there exist languages that admit witnesses for which no language that admits witnesses can help the "witness searching" for such languages. Given a language L, a witness for an instance x is a kind of short (or polynomial length-bounded) proof for the assertion "$x \in L$" or for its negation, which can be verified in polynomial time. We call assertions like the above ones, short-proof assertions. Informally, $\mathrm{SH} \neq \mathrm{NP} \cap \text{co-NP}$ implies the existence of short-proof assertions for which finding their proofs cannot be helped by the ability of deciding the truth of any short-proof assertion. In order to shed light on the question $\mathrm{SH} =?\mathrm{NP} \cap \text{co-NP}$ we introduce the Helping hierarchy.

Definition 2.3. Let $\mathrm{P}^0_{\mathrm{help}} := \mathrm{P}_{\mathrm{help}}(2^{\{0,1\}^*})$ and, for every $k \geq 1$, $\mathrm{P}^k_{\mathrm{help}} := \mathrm{P}_{\mathrm{help}}(\mathrm{P}^{k-1}_{\mathrm{help}})$. For each $k \geq 0$, we say that $\mathrm{P}^k_{\mathrm{help}}$ is the kth level of the *Helping hierarchy.* Moreover, Let $\mathrm{P}^\omega_{\mathrm{help}} := \bigcap_{k \geq 0} \mathrm{P}^k_{\mathrm{help}}$.

Observe that, from the basic result of Schöning [14], it follows that $\mathrm{P}^0_{\mathrm{help}} = \mathrm{NP} \cap \text{co-NP}$. The next result lists some basic properties of the Helping hierarchy, one of these says that the hierarchy is downward, unlike common hierarchies that are upward.

Proposition 2.4.

1. $\mathrm{P}^0_{\mathrm{help}} \supseteq \mathrm{P}^1_{\mathrm{help}} \supseteq \mathrm{P}^2_{\mathrm{help}} \supseteq \cdots \supseteq \mathrm{P}^{\omega}_{\mathrm{help}}$.
2. *For any* $k \geq 0$, $\quad \mathrm{P}^k_{\mathrm{help}} = \mathrm{P}^{k+1}_{\mathrm{help}} \quad \Longrightarrow \quad \mathrm{P}^k_{\mathrm{help}} = \mathrm{P}^{k+1}_{\mathrm{help}} = \mathrm{P}^{k+2}_{\mathrm{help}} = \cdots = \mathrm{P}^{\omega}_{\mathrm{help}}$.

The connection between SH and the Helping hierarchy is stated by the following proposition:

Proposition 2.5.

1. $\mathrm{SH} \subseteq \mathrm{P}^{\omega}_{\mathrm{help}}$.
2. *For any* $k \geq 0$, $\quad \mathrm{P}^k_{\mathrm{help}} = \mathrm{P}^{k+1}_{\mathrm{help}} \quad \Longleftrightarrow \quad \mathrm{P}^k_{\mathrm{help}} = \mathrm{SH}$.

In Section 4 we show the existence of a relativized world in which the Helping hierarchy is infinite. This means that proving the collapse of the Helping hierarchy (in particular, the collapse to level zero, that is, SH = NP ∩ co-NP), if possible at all, is very hard. In fact standard techniques, like direct simulation, relativize and thus they cannot be used.

3. Characterizations

Firstly, we observe that each of the notions defined in the previous section admits a natural relativized version. The next theorem gives a characterization for each level of the Helping hierarchy. These characterizations will be very useful to prove our separation result.

To state the theorem we need some notations. Let $\leq_{\mathrm{lex}}$ denote the lexicographic order relation for strings in $\{0,1\}^*$. Let $\langle\,\cdot,\,\cdot\rangle_2$ denote a pairing function over finite strings in $\{0,1\}^*$ with the standard nice computability, and invertibility properties. For every $k \geq 1$, let $\langle\, y_1,\, y_2,\, \ldots\,,\, y_k\rangle$ denote

$$\langle\, k,\, \langle\, y_1,\, \langle\, y_2,\, \langle\, \ldots\,,\, \langle\, y_{k-1},\, y_k\rangle_2\, \ldots\rangle_2\rangle_2\rangle_2\rangle_2$$

(in particular, $\langle\, y_1\rangle$ denote $\langle\, 1, y_1\rangle_2$). For every k and n we define

$$\mathcal{T}^n_k := \{\langle\, y_1, y_2, \ldots\,, y_i\rangle \mid 1 \leq i \leq k,\ y_j \in \{0,1\}^n,\ j = 1, 2, \ldots\,, i\}$$

and $\mathcal{T}_k := \bigcup_{n\geq 1} \mathcal{T}^n_k$. Define the following order relation in $\mathcal{T}_k$: for every $\langle y_1, y_2, \ldots\,, y_i\rangle$, $\langle y'_1, y'_2, \ldots\,, y'_j\rangle \in \mathcal{T}_k$,

$$\begin{aligned}\langle\, y_1, y_2, \ldots\,, y_i\rangle \sqsubseteq \langle\, y'_1, y'_2, \ldots\,, y'_j\rangle \quad \Longleftrightarrow \quad & i \leq j \ \wedge\ y_1 = y'_1 \ \wedge \\ & y_2 = y'_2 \ \wedge\ \cdots\ \wedge\ y_i = y'_i.\end{aligned}$$

For every k and every $u = \langle y_1, y_2, \ldots\,, y_i\rangle \in \mathcal{T}_k$ let $\mathsf{rk}(u) = i$. For every $u \in \mathcal{T}_k$, if $\mathsf{rk}(u) \geq 2$ we denote by $\mathsf{pre}(u)$ the unique element of $\mathcal{T}_k$ such that $\mathsf{pre}(u) \sqsubset u$ and for which there is no $v \in \mathcal{T}_k$ such that $\mathsf{pre}(u) \sqsubset v \sqsubset u$. If $\mathsf{rk}(u) = 1$ we set $\mathsf{pre}(u)$ equal to the special symbol $\perp$. The next theorem can be viewed as an extension of a similar characterization given in [12, 14]:

Theorem 3.1. *Let $k \geq 0$ and let E be any oracle. A language L belongs to $\mathrm{P}^{k,E}_{\mathrm{help}}$ if and only if there exists a $\{0,1,\sharp\}$-valued function $R \in \mathrm{FP}^E$ and a polynomial p such that, for any $x \in \{0,1\}^*$,*

1. $(\forall u \in \mathcal{T}^{p(|x|)}_{k+1})[R(x,u) \in \{L(x), \sharp\}]$;
2. $(\exists w \in \mathcal{T}^{p(|x|)}_{k+1})[\mathsf{rk}(w) = k+1 \ \wedge\ R(x,w) = L(x)]$;
3. $(\forall u, v \in \mathcal{T}^{p(|x|)}_{k+1})[(\mathsf{rk}(u) = \mathsf{rk}(v) \wedge R(x,u) = R(x,v) = L(x)) \implies \mathsf{pre}(u) = \mathsf{pre}(v)]$;
4. $(\forall u, v \in \mathcal{T}^{p(|x|)}_{k+1})[(R(x,u) = L(x) \ \wedge\ v \sqsubseteq u) \implies R(x,v) = L(x)]$.

Before proving the theorem we briefly explain the meaning of this characterization. Consider a language L that admits witnesses. For any instance x let $W(x)$ be the set of witnesses for x. Since we are interested in "witness searching", we ask whether also the language of witnesses admits witnesses, that is, whether a language of the kind $W[L] = \{\langle x, u\rangle \mid u \text{ is a prefix of some } y \in W'(x)\}$ admits witnesses, where $W'(x)$ is some nonempty subset of $W(x)$. For instance, any language L in NP∩co-NP admits witnesses but, in general, we do not know whether some $W[L]$ admits witnesses. This lead us to a refinement of the notion of the existence of witnesses. We say that a language L admits *witnesses of depth* 1 if L admits witnesses and for $k > 1$ we say that L admits *witnesses of depth k* if some $W[L]$ admits witnesses of depth $k-1$. It is quite easy to see that any language in SH admits witnesses of any depth. Moreover, the conditions (1–4) of the theorem formalize the fact that any language in $\mathrm{P}^k_{\mathrm{help}}$ admits witnesses of depth $k+1$. In fact, if $L \in \mathrm{P}^k_{\mathrm{help}}$ and R is a function that satisfies the conditions of the theorem w.r.t. L, then we can set the witnesses for L as $W(x) = \{\langle y_1\rangle \mid R(x, \langle y_1\rangle) = L(x)\}$. To see that there is also a $W[L]$ that admits witnesses it suffices to consider $W'(x) = \{\langle y_1\rangle \mid \exists y_2\ R(x, \langle y_1, y_2\rangle) = L(x)\}$. For such a $W[L]$ the witnesses are given by $W(\langle x, u\rangle) = \{\langle y_1, y_2\rangle \mid u \text{ is a prefix of } \langle y_1\rangle \text{ and } R(x, \langle y_1, y_2\rangle) = L(x)\}$. If $k \geq 2$ also some $W[W[L]]$ admits witnesses as can be seen setting $W'(\langle x, u\rangle) = \{\langle y_1, y_2\rangle \mid u \text{ is a prefix of } \langle y_1\rangle \text{ and } \exists y_3\ R(x, \langle y_1, y_2, y_3\rangle) = L(x)\}$. Thus, for such a $W[W[L]]$ the witnesses are $W(\langle\langle x, u\rangle, v\rangle) = \{\langle y_1, y_2, y_3\rangle \mid u \text{ is a prefix of } \langle y_1\rangle$ and v is a prefix of $\langle y_1, y_2\rangle$ and $R(x, \langle y_1, y_2, y_3\rangle) = L(x)\}$. And so on for $k \geq 3$.

Proof of Theorem 3.1 Since the proof is essentially independent of the oracle E, we only prove the case with $E = \emptyset$. The proof is by induction on k. For $k = 0$, conditions (3) and (4) are always satisfied. Since $\mathrm{P}^0_{\mathrm{help}} = \mathrm{NP} \cap \text{co-NP}$, it is easy to see that for any language $L \in \mathrm{P}^0_{\mathrm{help}}$ there exist R and p satisfying conditions (1) and (2). On the other hand, if R and p satisfy those conditions w.r.t. a language L, then it easily follows that $L \in \mathrm{P}^0_{\mathrm{help}}$.

Now, suppose that the thesis holds for $k-1$ and prove it for k. Let L be any language in $\mathrm{P}^k_{\mathrm{help}}$. Since from Definition 2.3 $\mathrm{P}^k_{\mathrm{help}} = \mathrm{P}_{\mathrm{help}}(\mathrm{P}^{k-1}_{\mathrm{help}})$, there exist a robust machine M and a language $H \in \mathrm{P}^{k-1}_{\mathrm{help}}$ such that M with the help of

H recognizes L in polynomial time. We modify machine M so that, for every oracle X and every input x, $M^X(x)$ always halts within polynomially many steps in the length of x, its output always belongs to $\{L(x), \sharp\}$, and $M^H(x)$ outputs $L(x)$. Moreover, w.l.o.g., we assume that M, on every input x and for every oracle, makes exactly $q(|x|)$ queries and all such queries have length $q(|x|)$, for some polynomial q[3]. For any x and any sequence $b_1 \cdots b_{q(|x|)} \in \{0,1\}^{q(|x|)}$ of possible oracle answers, considering the computation of M on input x in which the ith query is answered according to the value of b_i, we denote by $M^{b_1 \cdots b_{q(|x|)}}(x)$ and by $a_h(x, b_1 \cdots b_{q(|x|)})$ (for $h = 1, \ldots, q(|x|)$), respectively, the output and the hth query of that computation.

Since $H \in \mathrm{P}_{\mathrm{help}}^{k-1}$, by the induction hypothesis, there exists a $\{0,1,\sharp\}$-valued function $S \in \mathrm{FP}$ and a polynomial r that satisfy conditions (1–4) w.r.t. k and language H. We prove that there exists a function R and a polynomial p that satisfy those conditions w.r.t. $k+1$ and L. Define polynomial p as $p(n) = q(n)r(q(n))$. In order to define R, we need to introduce the following predicate: for every x and for every $\langle y_1, \ldots, y_j \rangle \in \mathcal{T}_{k+1}^{p(|x|)}$, let $check(x, \langle y_1, \ldots, y_j \rangle)$ be equal to

$$\begin{aligned} & y_1 = b_1 \cdots b_{q(|x|)} 0^{p(|x|)-q(|x|)} \wedge (\forall 2 \le i \le j\ \exists z_{1,i-1}, z_{2,i-1}, \ldots, z_{q(|x|),i-1} : \\ & y_i = z_{1,i-1} z_{2,i-1} \cdots z_{q(|x|),i-1} \wedge |z_{1,i-1}| = |z_{2,i-1}| = \cdots = |z_{q(|x|),i-1}|) \wedge \\ & (\forall 1 \le h \le q(|x|)\ \ S(a_h(x, b_1 \cdots b_{q(|x|)}), \langle z_{h,1}, \ldots, z_{h,j-1} \rangle) = b_h). \end{aligned}$$

Define R as follows

$$R(x, \langle y_1, \ldots, y_j \rangle) := \begin{cases} 1 & \text{if } y_1 = b_1 \cdots b_{q(|x|)} 0^{p(|x|)-q(|x|)} \text{ and} \\ & check(x, \langle y_1, \ldots, y_j \rangle) \text{ and } M^{b_1 \cdots b_{p(|x|)}}(x) = 1 \\ 0 & \text{if } y_1 = b_1 \cdots b_{q(|x|)} 0^{p(|x|)-q(|x|)} \text{ and} \\ & check(x, \langle y_1, \ldots, y_j \rangle) \text{ and } M^{b_1 \cdots b_{p(|x|)}}(x) = 0 \\ \sharp & \text{otherwise.} \end{cases}$$

Using the fact that S satisfies conditions (1–4) w.r.t. k and H, it is quite easy to verify that R satisfies conditions (1–4) w.r.t. $k+1$ and L.

It remains to prove that if for a language there exists a function and a polynomial that satisfy conditions (1–4) then that language belongs to $\mathrm{P}_{\mathrm{help}}^{k}$. Let L be a language for which there exists a $\{0,1,\sharp\}$-valued function R and a polynomial p that satisfy conditions (1–4) w.r.t. $k+1$ and L. W.l.o.g. we suppose that, for all n, $p(n) \ge n$ and $p(n) < p(n+1)$. For every x and every y define $[x, y] := x0^{p(|x|)-|x|}y$. Let $D := \{[x,y] \mid x, y \in \{0,1\}^* \text{ with } |y| = p(|x|)\}$. It is clear that D is decidable in polynomial time, that for every $w \in D$ there is only one pair (x, y) such that $w = [x, y]$, and that this pair can be obtained from w in polynomial time. Consider

[3]If this is not the case we can replace H with $H' := \{\langle z, y \rangle \mid z \in H\}$ (H' still belongs to $\mathrm{P}_{\mathrm{help}}^{k-1}$, since this class is closed downward w.r.t. the polynomial-time many-one reducibility) and we can modify M in that every query z is replaced by a query of the type $\langle z, 0^m \rangle$ for some suitable integer m.

the following language:

$$H \;:=\; \{w \mid \exists; x,y:\ w=[x,y]\ \wedge\ (\exists\langle y_1,\dots,y_{k+1}\rangle \in \mathcal{T}_{k+1}^{p(|x|)})[y \le_{\mathrm{lex}} y_1\ \wedge\ R(x,\langle y_1,\dots,y_{k+1}\rangle)\in\{0,1\}]\}\cdot$$

Since R satisfies conditions (1)-(4) w.r.t. L, it is easy to verify that $L \in \mathrm{P}_{\mathrm{help}}(H)$. To complete the proof it is enough to show that $H \in \mathrm{P}_{\mathrm{help}}^{k-1}$. By the inductive hypothesis, for showing $H \in \mathrm{P}_{\mathrm{help}}^{k-1}$ it suffices to find a function S and a polynomial q that satisfy conditions (1–4) w.r.t. k and H. Let $q(n) := n$ and define function S as follows: for every w and for every $\langle z_1,\dots,z_j\rangle \in \mathcal{T}_k^{q(|w|)}$,

$$S(w,\langle z_1,\dots,z_j\rangle) := \begin{cases} 1 & \text{if } \exists x,y:\ w=[x,y] \text{ and } \forall 1\le i\le j\ z_i = y_i y_{i+1} \text{ with} \\ & |y_i| = |y_{i+1}| = p(|x|)\text{and } y \le_{\mathrm{lex}} y_1 \text{ and} \\ & R(x,\langle y_1,\dots,y_{j+1}\rangle)\in\{0,1\} \\ 0 & \text{if either } w\notin D \text{ and for } i=1,\dots,j,\ z_i = 0^{|w|} \\ & \text{or } \exists x,y:\ w=[x,y] \text{ and } \forall 1\le i\le j\ z_i = y_i y_{i+1} \text{ with} \\ & |y_i| = |y_{i+1}| = p(|x|)\text{and } y \not\le_{\mathrm{lex}} y_1 \text{ and} \\ & R(x,\langle y_1,\dots,y_{j+1}\rangle)\in\{0,1\} \\ \sharp & \text{otherwise.} \end{cases}$$

It is not hard to see that S and q satisfy condition (1)-(4) w.r.t. k and H. □

4. Relativized separations

The next theorem shows that for every $k \ge 0$, there is an oracle that separates $\mathrm{P}_{\mathrm{help}}^{k}$ from $\mathrm{P}_{\mathrm{help}}^{k+1}$. From this it is routine to obtain an oracle for which all the levels of the Helping hierarchy are separated, thus showing the existence of a relativized world in which the Helping hierarchy is infinite.

Theorem 4.1. *For any $k \ge 0$ there exists an oracle H such that $\mathrm{P}_{\mathrm{help}}^{k,H} \not\supseteq \mathrm{P}_{\mathrm{help}}^{k+1,H}$.*

Proof. Our oracle will be a function from $\mathcal{T}_{k+1}$ to $\{0,1,\sharp\}$. For every $n \in \mathbb{N}$, $b \in \{0,1\}$ $u,v \in \mathcal{T}_{k+1}^{n}$ and for every $E : \mathcal{T}_{k+1} \to \{0,1,\sharp\}$ we denote by $E_n^b[u,v]$ the oracle function defined as follows

$$(\forall w \in \mathcal{T}_{k+1}) \quad E_n^b[u,v](w) := \begin{cases} b & \text{if } w \in \mathcal{T}_{k+1}^{n} \text{ and } w \sqsubseteq u \text{ or } w \sqsubseteq v \\ \sharp & \text{if } w \in \mathcal{T}_{k+1}^{n} \text{ and } w \not\sqsubseteq u \text{ and } w \not\sqsubseteq v \\ E(w) & \text{otherwise.} \end{cases}$$

If $u = v$, we write $E_n^b[u]$ in place of $E_n^b[u,u]$. For any oracle E define a language $L(E)$ as follows

$$L(E) := \{0^n \mid (\exists u \in \mathcal{T}_{k+1}^{n})[\mathsf{rk}(u) = k+1 \wedge E(u) = 1]\}\cdot$$

The oracle H will be constructed by a direct diagonalization. Let $\{(R_i,p_i)\}$ be an enumeration of all the pairs (R,p) in which R is a polynomial-time oracle transducer and p is a polynomial. For every i let q_i be a polynomial such that the running time of $R_i^E(x,u)$ is bounded by $q_i(|x|)$, for all oracles E, strings x, and $u \in \mathcal{T}_{k+2}^{p_i(|x|)}$. Thus, a language L belongs to $\mathrm{P}_{\mathrm{help}}^{k+1,E}$ if and only if there exists a pair (R_i, p_i) of the above enumeration such that R_i^E and p_i satisfy, for every input x, conditions (1–4) of Theorem 3.1 w.r.t. $k+2$ and L. We need the following concept: for any n, we say that a pair (u,v) is an *n-branch* if $u,v \in \mathcal{T}_{k+1}^n$, $\mathsf{rk}(u) = k+1$, and either $v = u$ or $\mathsf{pre}(v) \sqsubseteq u$, and $v \not\sqsubseteq u$. We will construct our oracle H so that for every $n \in \mathbb{N}$ there is an n-branch (u,v) and $b \in \{0,1\}$ such that $H = H_n^b[u,v]$. It is not hard to see that for such an H it results $L(H) \in \mathrm{P}_{\mathrm{help}}^{k,H}$.

Begin Construction

Stage 0**:** let H_0 be the oracle such that for every $n \in \mathbb{N}$, for $1 \le m \le k+1$, $H_0(\underbrace{\langle 0^n, \dots, 0^n \rangle}_{m}) := 1$, and $H_0(w) := \sharp$ elsewhere. Set $l(0) := 0$.

Stage i**:** let H_{i-1} be the oracle so far constructed. For the sake of convenience, we omit the subscript $i-1$ and call it simply H. Define n to be the least integer such that $q_i(n) < \frac{2^n-1}{2}$ and $n > q_{i-1}(l(i-1))$. Find an n-branch (u,v) and $b \in \{0,1\}$ such that $R_i^{H_n^b[u,v]}$ and p_i do not satisfy, on input 0^n, conditions (1–4) of Theorem 3.1 w.r.t. $k+2$ and $L(H_n^b[u,v])$. Set $H_i := H_n^b[u,v]$ and $l(i) := n$.

End Construction

Let $H := \lim_i H_i$. It is clear that the limit exists since for any u there is a value $c \in \{0,1,\sharp\}$ such that $H_j(u) = c$ for almost all j. In order to show that at any stage it is possible to find a suitable pair (u,v) we need an easy combinatorial lemma on directed graphs. Let $G = (V,A)$ be a directed graph; given a vertex $r \in V$, let $d^+(r) := |\{(r,s) \mid (r,s) \in A\}|$. We say that G is a digraph of *positive degree d* if $d = max\{d^+(r) \mid r \in V\}$.

Lemma 4.2. *Let $G = (V,A)$ be a directed graph of positive degree d. If $d < \frac{|V|-1}{2}$, then there exist two distinct vertices r,s such that $(r,s),(s,r) \notin A$.*

Proof. The number of all the unordered pairs of vertices is $\frac{|V|(|V|-1)}{2}$; moreover, it holds that

$$|A| = \sum_{r \in V} d^+(r) \le |V|d < \frac{|V|(|V|-1)}{2}\cdot$$

This means that there are two distinct vertices r,s such that $(r,s),(s,r) \notin A$. $\square$

The next lemma is the core of the proof:

Lemma 4.3. *Let (R_i,p_i), H, and n be as at stage i of the construction. Then there exists an n-branch (u,v) and $b \in \{0,1\}$ such that $R_i^{H_n^b[u,v]}$ and p_i do not satisfy, on input 0^n, conditions (1–4) of Theorem 3.1 w.r.t. $k+2$ and $L(H_n^b[u,v])$.*

Proof. For the sake of convenience, call R_i, p_i, and q_i, respectively R, p, and q. Suppose by the way of contradiction that, for all n-branches (u,v) and all $b \in \{0,1\}$, $R^{H_n^b[u,v]}$ and p satisfy, on input 0^n, conditions (1–4) of Theorem 3.1

w.r.t. $k+2$ and $L(H_n^b[u,v])$. From this assumption, it derives that, for any $u \in \mathcal{T}_{k+1}^n$ of rank $k+1$, there exist $z^{k+1}(u)$ and $z^{k+2}(u)$ in $\mathcal{T}_{k+2}^{p(n)}$ of rank $k+1$ and $k+2$, respectively, such that $z^{k+1} \sqsubset z^{k+2}$, $R^{H_n^1[u]}(0^n, z^{k+1}(u)) = 1$, and $R^{H_n^1[u]}(0^n, z^{k+2}(u)) = 1$. We need some notations. For an oracle machine M, an oracle E, and input x, let $Q(M^E(x))$ denote the set of oracle queries made by the computation $M^E(x)$. We say that a set $B \subseteq \mathcal{T}_{k+1}^n$ is a *level-set* of rank h if $B = \{u \mid \mathsf{rk}(u) = h \;\wedge\; \mathsf{pre}(u) = v\}$ for some $v \in \mathcal{T}_{k+1}^n \cup \{\perp\}$. W.l.o.g. we assume that for every oracle E, and all $u, v \in \mathcal{T}_{k+2}^{p(n)}$ with $v \sqsubseteq u$, it holds $Q(R^E(0^n, v)) \subseteq Q(R^E(0^n, u))$.

We firstly prove the case $k = 0$. Define the digraph $G := (\mathcal{T}_1^n, A)$ where $A := \{(v, w) \mid w \in Q(R^{H_n^1[v]}(0^n, z^2(v)))\}$. Note that G has positive degree at most $q(n)$ and $|\mathcal{T}_1^n| = 2^n$. Since $q(n) < \frac{2^n-1}{2}$, from Lemma 4.2, there exist $r, s \in \mathcal{T}_1^n$ such that $(r, s), (s, r) \notin A$. This implies that $R^{H_n^1[r,s]}(0^n, z^2(r)) = R^{H_n^1[r]}(0^n, z^2(r)) = 1$ and $R^{H_n^1[r,s]}(0^n, z^2(s)) = R^{H_n^1[s]}(0^n, z^2(s)) = 1$. Since R satisfies condition (3) of Theorem 3.1 w.r.t. $L(H_n^1[r,s])$, it must be the case that $z^1(r) = z^1(s)$. It follows that $s \notin Q(R^{H_n^1[s]}(0^n, z^1(s)))$, and thus $R^{H_n^0[s]}(0^n, z^1(s)) = 1$, which violates condition (1) w.r.t. $L(H_n^0[s])$.

For $k \geq 1$, we proceed in a similar but more involved manner. For every h with $1 \leq h \leq k$ and for every level-set B of rank h define

$$\begin{aligned} \mathrm{C}(B) \;&:=\; \{r \in B \mid \exists t : r \sqsubset t \;\wedge\; \mathsf{rk}(t) = k+1 \;\wedge \\ &\quad\; \{u \mid r \sqsubset u \sqsubseteq t\} \cap Q(R^{H_n^1[t]}(0^n, z^{k+2}(t))) = \emptyset\}. \end{aligned}$$

Claim 4.4. Let h be such that $1 \leq h \leq k$ and let B be a level-set of rank h. If $\mathrm{C}(B) = B$ then there exist $s, t_s \in \mathcal{T}_{k+1}^n$ such that $s \sqsubset t_s$, $s \in B$, $\mathsf{rk}(t_s) = k+1$, and $\{u \mid s \sqsubseteq u \sqsubseteq t_s\} \cap Q(R^{H_n^1[t_s]}(0^n, z^{k+1}(t_s))) = \emptyset$.

Proof. Suppose that $\mathrm{C}(B) = B$. Then, for any $r \in B$ there exists t_r such that $r \sqsubset t_r$, $\mathsf{rk}(t_r) = k+1$, and $\{u \mid r \sqsubset u \sqsubseteq t_r\} \cap Q(R^{H_n^1[t_r]}(0^n, z^{k+2}(t_r))) = \emptyset$. Consider the directed graph whose set of vertices is B and the set of edges is $A := \{(w, v) \mid w, v \in B \text{ and } \{u \mid v \sqsubseteq u \sqsubseteq t_v\} \cap Q(R^{H_n^1[t_w]}(0^n, z^{k+2}(t_w))) \neq \emptyset\}$. This graph has positive degree at most $q(n)$ and $|B| = 2^n$. Since $q(n) < \frac{2^n-1}{2}$, by Lemma 4.2 there exist $r, s \in B$ such that $(r, s), (s, r) \notin A$. This implies that $R^{H_n^1[r,t_s]}(0^n, z^{k+2}(t_s)) = R^{H_n^1[t_s]}(0^n, z^{k+2}(t_s)) = 1$ and $R^{H_n^1[r,t_s]}(0^n, z^{k+2}(t_r)) = R^{H_n^1[t_r]}(0^n, z^{k+2}(t_r)) = 1$ (recall that $\{u \mid r \sqsubset u \sqsubseteq t_r\} \cap Q(R^{H_n^1[t_r]}(0^n, z^{k+2}(t_r))) = \emptyset$). Since R satisfies, on input 0^n, condition (3) of Theorem 3.1 w.r.t. $L(H_n^1[r, t_s])$, it must be the case that $z^{k+1}(t_s) = z^{k+1}(t_r)$. From this and the fact that $\{u \mid s \sqsubseteq u \sqsubseteq t_s\} \cap Q(R^{H_n^1[t_r]}(0^n, z^{k+1}(t_r))) = \emptyset$ it follows that $\{u \mid s \sqsubseteq u \sqsubseteq t_s\} \cap Q(R^{H_n^1[t_r]}(0^n, z^{k+1}(t_s))) = \emptyset$. Moreover $\{u \mid r \sqsubseteq u \sqsubseteq t_r\} \cap Q(R^{H_n^1[t_s]}(0^n, z^{k+1}(t_s))) = \emptyset$, hence, if $v \in Q(R^{H_n^1[t_r]}(0^n, z^{k+1}(t_s))) \cap Q(R^{H_n^1[t_s]}(0^n, z^{k+1}(t_s)))$ then $v \notin \{u \mid s \sqsubseteq u \sqsubseteq t_s\} \cup \{u \mid r \sqsubseteq u \sqsubseteq t_r\}$, and thus $H_n^1[t_s](v) = H_n^1[t_r](v)$. It follows that $Q(R^{H_n^1[t_s]}(0^n, z^{k+1}(t_s))) = Q(R^{H_n^1[t_r]}(0^n, z^{k+1}(t_s)))$. Hence, $\{u \mid s \sqsubseteq u \sqsubseteq t_s\} \cap Q(R^{H_n^1[t_s]}(0^n, z^{k+1}(t_s))) = \emptyset$. $\square$

Claim 4.5. For any h with $1 \leq h \leq k$ there exists a level-set B of rank h such that $|\mathrm{C}(B)| < |B|$.

Proof The proof is by induction on h.

Basic step: Let $h := 1$ and let B be a level-set of rank 1. Suppose that $\mathrm{C}(B) = B$. From Claim 4.4 it derives that there exist $s, t_s \in \mathcal{T}^n_{k+1}$ such that $s \sqsubset t_s$, $s \in B$, $\mathsf{rk}(t_s) = k+1$ and $\{u \mid s \sqsubseteq u \sqsubseteq t_s\} \cap Q(R^{H^1_n[t_s]}(0^n, z^{k+1}(t_s))) = \emptyset$. Note that, since $\mathsf{rk}(s) = 1$, the set $\{u \mid s \sqsubseteq u \sqsubseteq t_s\}$ is equal to $\{u \mid u \sqsubseteq t_s\}$, that is, the set of all the elements u such that $H^1_n[t_s](u) = 1$. Hence, since $\{u \mid u \sqsubseteq t_s\} \cap Q(R^{H^1_n[t_s]}(0^n, z^{k+1}(t_s))) = \emptyset$ and $R^{H^1_n[t_s]}(0^n, z^{k+1}(t_s)) = 1$, it holds that $R^{H^0_n[t_s]}(0^n, z^{k+1}(t_s)) = 1$, which violates condition (1) w.r.t. $L(H^0_n[t_s])$.

Inductive step: Assume that the thesis is true for $h < k$ and let us prove it for $h+1$. From the inductive hypothesis it follows that a level-set B of rank h exists such that $|\mathrm{C}(B)| < |B|$. Then there is at least an element $v \in B - \mathrm{C}(B)$. Let $P(v) := \{u \mid \mathsf{pre}(u) = v\}$. Clearly, $P(v)$ is a level-set of rank $h+1$. Suppose that $\mathrm{C}(P(v)) = P(v)$. Then, by Claim 4.4 there exist $s, t_s \in \mathcal{T}^n_{k+1}$ such that $s \sqsubset t_s$, $s \in P(v)$, $\mathsf{rk}(t_s) = k+1$, and $\{u \mid s \sqsubseteq u \sqsubseteq t_s\} \cap Q(R^{H^1_n[t_s]}(0^n, z^{k+1}(t_s))) = \emptyset$. Note that, since $s \in P(v)$, it holds that $\mathsf{pre}(s) = v$, thus $\{u \mid s \sqsubseteq u \sqsubseteq t_s\} = \{u \mid v \sqsubset u \sqsubseteq t_s\}$. This implies that $v \in \mathrm{C}(B)$, a contradiction. $\square$

From Claim 4.5 there exists a level-set B of rank k such that $|\mathrm{C}(B)| < |B|$. Let $r \in B - \mathrm{C}(B)$. Then, for any t such that $\mathsf{pre}(t) = r$ it holds that $t \in Q(R^{H^1_n[t]}(0^n, z^{k+1}(t)))$ (note that $\{u \mid r \sqsubset u \sqsubseteq t\} = \{t\}$). Define a digraph $G = (V, A)$ where

$$V := \{t \mid \mathsf{pre}(t) = r\}, \quad A := \{(t, s) \mid s \in Q(R^{H^1_n[t]}(0^n, z^{k+2}(t)))\}.$$

This digraph has positive degree at most $q(n)$ and $|V| = 2^n$. Since $q(n) < \frac{2^n - 1}{2}$, by Lemma 4.2 there exist $s, t \in V$ such that $(s, t), (t, s) \notin A$. At this point we proceed by proving that $z^{k+1}(s) = z^{k+1}(t)$. Since

1. $s \notin Q(R^{H^1_n[t]}(0^n, z^{k+2}(t)))$ and $t \notin Q(R^{H^1_n[s]}(0^n, z^{k+2}(s)))$;
2. $\mathsf{rk}(s) = \mathsf{rk}(t) = k+1$;
3. $\mathsf{pre}(s) = \mathsf{pre}(t)$;

it must be the case that $R^{H^1_n[s,t]}(0^n, z^{k+2}(s)) = R^{H^1_n[s]}(0^n, z^{k+2}(s)) = 1$ and $R^{H^1_n[s,t]}(0^n, z^{k+2}(t)) = R^{H^1_n[t]}(0^n, z^{k+2}(t)) = 1$. This together with the fact that R satisfies, on input 0^n, condition (3) w.r.t. $L(H^1_n[s,t])$, imply that $z^{k+1}(s) = z^{k+1}(t)$. Now, note that if $u \in Q(R^{H^1_n[s]}(0^n, z^{k+1}(s))) \cap Q(R^{H^1_n[s,t]}(0^n, z^{k+1}(s)))$ then $u \neq t$, and thus $H^1_n[s](u) = H^1_n[s,t](u)$. This implies that

$$Q(R^{H^1_n[s]}(0^n, z^{k+1}(s))) = Q(R^{H^1_n[s,t]}(0^n, z^{k+1}(s))).$$

Symmetrically, it can be seen that $Q(R^{H^1_n[t]}(0^n, z^{k+1}(t))) = Q(R^{H^1_n[s,t]}(0^n, z^{k+1}(t)))$. Hence, it holds that $t \notin Q(R^{H^1_n[s]}(0^n, z^{k+1}(s))) = Q(R^{H^1_n[s,t]}(0^n, z^{k+1}(s))) =$

$Q(R^{H_n^1[s,t]}(0^n, z^{k+1}(t)))=Q(R^{H_n^1[t]}(0^n, z^{k+1}(t)))$, that is, $t \notin Q(R^{H_n^1[t]}(0^n, z^{k+1}(t)))$ yielding a contradiction. □

Acknnowledgements. We are grateful to Lane Hemaspaandra for pointing out the paper [13] to us. We want also to thank Luca Trevisan for drawing our attention to the paper [1]. We thank Sergio De Agostino for useful comments. We thanks two anonymous referees for suggestions which improved the paper.

References

[1] V. Arvind, A Note on the Self-Witnessing Property of Computational Problems, *Proc. 2nd Annual International Conference on Computing and Combinatorics (COCOON'96).* Springer-Verlag, *Lecture Notes in Comput. Sci.* **1090** (1996) 241-249.

[2] J.L. Balcázar, Self-reducibility, *Proc. 4th Symposium on Theoretical Aspects of Computer Science.* Springer-Verlag, *Lecture Notes in Comput. Sci.* **247** (1987) 136-147.

[3] J.L. Balcázar, Self-reducibility structures and solutions of NP problems. *Rev. Mat. Complut.* **2** (1989) 175-184.

[4] D.P. Bovet and P. Crescenzi, *Introduction to the Theory of Complexity.* Prentice-Hall (1994).

[5] J.L. Balcázar, J. Díaz and J. Gabarró, *Structural Complexity I*, Vol. 1. Springer-Verlag (1988).

[6] J.L. Balcázar, J. Díaz and J. Gabarró, *Structural Complexity II*, Vol. 2. Springer-Verlag (1990).

[7] J. Cai, L. Hemachandra and J. Viskoč, Promises Problems and Access to Unambiguous Computation, *Proc. 17th Symposium on Mathematical Foundations of Computer Science.* Springer-Verlag, *Lecture Notes in Comput. Sci.* **629** (1992) 162-171.

[8] P. Cintioli and R. Silvestri, Helping by Unambiguous Computation and Probabilistic Computation. *Theory Comput. Syst.* **30** (1997) 165-180.

[9] P. Cintioli and R. Silvestri, Revisiting a Result of Ko. *Inform. Process. Lett.* **61** (1997) 189-194.

[10] M. Fellows and N. Koblitz, Self-witnessing polynomial time complexity and prima factorization, in *Proc. 6th Structure in Complexity Theory Conference* (1992) 107-110.

[11] L. Hemachandra, Fault-Tolerance and Complexity, in *Proc. 20th International Colloquium on Automata, Languages, and Programming.* Springer-Verlag, *Lecture Notes in Comput. Sci.* (1993).

[12] K. Ko, On Helping by Robust Oracle Machines. *Theoret. Comput. Sci.* **52** (1987) 15-36.

[13] M. Ogihara, On Helping by Parity-like Languages. *Inform. Process. Lett.* **54** (1995) 41-43.

[14] U. Schöning, Robust Algorithms: A Different Approach to oracles. *Theoret. Comput. Sci.* **40** (1985) 57-66.

Communicated by J.L. Balcazar.
Received June, 2000. Accepted November, 2001.