Hierarchical Unambiguity

Holger Spakowski[1]* and Rahul Tripathi[2]

[1] Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, 40225 Düsseldorf, Germany (Email: spakowsk@cs.uni-duesseldorf.de)

[2] Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA (Email: tripathi@cse.usf.edu)

Abstract. We develop techniques to investigate relativized hierarchical unambiguous computation. We apply our techniques to push forward some known constructs involving relativized unambiguity based complexity classes (UP and $\mathrm{Promise\text{-}UP}$) to new constructs involving arbitrary levels of the relativized unambiguous polynomial hierarchy (UPH). Our techniques are developed on constraints imposed by hierarchical assembly of *unambiguous* nondeterministic polynomial-time Turing machines, and so our techniques differ substantially, in applicability and in nature, from standard techniques (such as the switching lemma [Hås87]), which are known to play roles in carrying out similar generalizations.

Aside from achieving these generalizations, we resolve a question posed by Cai, Hemachandra, and Vyskoč [CHV93] on an issue related to nonadaptive Turing access to UP and adaptive smart Turing access to $\mathrm{Promise\text{-}UP}$.

1 Introduction

Baker, Gill, and Solovay in their seminal paper [BGS75] introduced the concept of relativization in complexity theory, and showed that the primitive levels of the polynomial hierarchy, i.e. P and NP, separate in some relativized world. Baker and Selman [BS79] made progress in extending this relativized separation—$\mathrm{P} \neq \mathrm{NP}$ in some relativized world—to the next levels of the polynomial hierarchy: They proved that there is a relativized world where $\Sigma_2^p \neq \Pi_2^p$, and so $\Sigma_2^p \neq \Sigma_3^p$ relative to the same world. However, Baker and Selman [BS79] observed that their proof techniques do not apply in achieving relativized separations at higher levels of the polynomial hierarchy because of certain constraints in their counting argument. Thus, it required the development of entirely different proof techniques for separating all the levels of the relativized polynomial hierarchy. The landmark paper by Furst, Saxe, and Sipser [FSS84] established the close connection between the relativization of the polynomial hierarchy and lower bounds for small depth circuits computing certain functions. Techniques for proving such lower bounds were developed in a series of papers [FSS84,Sip83,Yao85,Hås87], which were motivated by questions about the relativized structure of the polynomial hierarchy. Yao [Yao85] finally succeeded in separating the levels of the relativized polynomial hierarchy by applying these new techniques. Håstad [Hås87] gave the most refined presentation of these techniques via the *switching lemma*. Even to date, Håstad's switching

* Supported in part by the DFG under grants RO 1202/9-1 and RO 1202/9-3.

lemma [Hås87] is used as an indispensable tool to separate relativized hierarchies, composed of classes stacked one on top of another. (See, for instance, [Hås87,Ko89,BU98,ST] where the switching lemma is used as a strong tool in proving the feasibility of oracle constructions.) A major contribution of our paper lies in demonstrating that certain known oracle constructions involving the primitive levels of the unambiguous polynomial hierarchy (UPH) and the promise unambiguous polynomial hierarchy ($\mathcal{UPH}$), i.e. UP and $\mathrm{P}_s^{\mathrm{Promise\text{-}UP}}$, respectively, can be extended to oracle constructions that involve arbitrary higher levels of UPH, purely by counting arguments alone. In fact, it seems implausible to achieve these extensions by well-known techniques from circuit complexity (e.g., the switching lemma [Hås87] and the polynomial method surveyed in [Bei93,Reg97]).

The class UP is the unambiguous version of NP. UP has proved to be useful in studying worst-case one-to-one one-way functions [Ko85,GS88] and some closure properties of $\#\mathrm{P}$ [OH93]. Lange and Rossmanith [LR94] generalized the notion of unambiguity to higher levels of the polynomial hierarchy. They introduced the following unambiguity based hierarchies: AUPH, UPH, and $\mathcal{UPH}$. It is known that $\mathrm{AUPH} \subseteq \mathrm{UPH} \subseteq \mathcal{UPH} \subseteq \mathrm{UAP}$ [LR94,CGRS04], where UAP (unambiguous alternating polynomial-time) is the analog of UP for alternating polynomial-time Turing machines. These hierarchies received renewed interests in some recent papers (see, for instance, [ACRW04,CGRS04,ST,GT05]). Spakowski and Tripathi [ST], developing on circuit complexity-theoretic proof techniques of Sheu and Long [SL96], and of Ko [Ko89], obtained results on the relativized structure of these hierarchies. Spakowski and Tripathi [ST] proved that there is a relativized world where these hierarchies are infinite. They also proved that for each $k \geq 2$, there is a relativized world where these hierarchies collapse so that they have exactly k distinct levels and their k'th levels collapse to PSPACE. The present paper supplements this investigation with a focus on the structure of the unambiguous polynomial hierarchy.

1.1 Results

We prove a combinatorial lemma (Lemma 11) and demonstrate its usefulness in generalizing known relativization results involving classes such as UP and Promise-UP to new relativization results that involve arbitrary levels of the unambiguous polynomial hierarchy (UPH).

In Subsection 4.1, we use Lemma 11 to show that certain inclusion relationships between bounded ambiguity classes ($\mathrm{UP}_{O(1)}$ and FewP) and the levels of the unambiguous polynomial hierarchy (UPH) do not relativize. Theorem 13 of this subsection subsumes an oracle result of Beigel [Bei89] for any constant $k \geq 1$ and Theorem 16 generalizes a result of Cai, Hemachandra, and Vyskoč [CHV93] from the case of $k = 2$ to the case of any arbitrary $k \geq 2$; the parameter k is a part of these theorems.

Subsection 4.2 studies the issue of simulating nonadaptive access to $\mathrm{U}\Sigma_h^p$, the h'th level of the unambiguous polynomial hierarchy, by adaptive access to $\mathrm{U}\Sigma_h^p$. Theorem 18 of this subsection generalizes a result of Cai, Hemachandra, and Vyskoč [CHV92] from the case of $h = 1$ to the case of any arbitrary $h \geq 1$; the parameter h is a part of the theorem. Lemma 11 is used as a key tool in proving Theorem 18.

We improve upon Theorem 18 of Subsection 4.2 in Subsection 4.3. There are compelling reasons for the transition from Subsection 4.2 to Subsection 4.3, which we elaborate in Subsection 4.3. Theorem 20 in that subsection not only resolves a question posed by Cai, Hemachandra, and Vyskoč [CHV93], but also generalizes one of their results. In particular, Theorem 20 holds for any total, polynomial-time computable and polynomially bounded function $k(\cdot)$ and arbitrary $h \geq 1$, while a similar result of Cai, Hemachandra, and Vyskoč [CHV93] holds only for any arbitrary *constant* k and $h = 1$; the parameters k and h are parts of Theorem 20. Lemma 11 is one of the ingredients in the proof of this theorem.

Subsection 4.4 investigates the complimentary issue of simulating adaptive access to $\mathrm{U}\Sigma_h^p$ by nonadaptive access to $\mathrm{U}\Sigma_h^p$. Theorem 21 of this subsection generalizes a result of Cai, Hemachandra, and Vyskoč [CHV93] from the case of $h = 1$ to the case of any arbitrary constant $h \geq 1$. Again, Lemma 11 is useful in making this generalization possible.

In Subsection 4.5, we study the notion of one-sided helping introduced by Ko [Ko87]. Theorem 22 of this subsection generalizes and improves one of the results of Cai, Hemachandra, and Vyskoč [CHV93].

Finally, in Section 5 we consider the possibility of imposing more stringent restriction in the statement of Lemma 11. The investigation in this subsection leads to a generic collapse of $\mathrm{U}\Sigma_k^p$ to P, for each $k \geq 1$, under the assumption $\mathrm{P} = \mathrm{NP}$. This generalizes a result of Blum and Impagliazzo [BI87] from the case of $k = 1$ to the case of any arbitrary $k \geq 1$.

Due to the space limit, all proofs are omitted. They will appear in the full version of the paper.

2 Preliminaries

2.1 Notations

Let $\mathbb{N}^+$ denote the set of positive integers. Σ denotes the alphabet $\{0, 1\}$. For every oracle NPTM N, oracle A, and string $x \in \Sigma^*$, we use the shorthand $N^A(x)$ for "the computation tree of N with oracle A on input x." We fix a standard, polynomial-time computable and invertible, one-to-one, multiarity pairing function $\langle ., \dots, . \rangle$ throughout the paper. Let $\circ$ denote the composition operator on functions. For any polynomial $p(.)$ and integer $i \geq 1$, let $(p\circ)^i(\cdot)$ denote $p \circ p \circ \cdots \circ p(\cdot)$, i.e. the polynomial obtained by i compositions of p.

For any complexity class $\mathcal{C}$ and for any natural notion of polynomial-time reducibility r (e.g., $r \in \{m, dtt, tt, k\text{-}tt, T, k\text{-}T, b\}$), let $R_r^p(\mathcal{C})$ denote the closure of $\mathcal{C}$ under r. That is, $R_r^p(\mathcal{C}) =_{df} \{L \mid (\exists L' \in \mathcal{C})[L \leq_r^p L']\}$. We refer the reader to any standard textbook in complexity theory (e.g. [BC93,HO02,Pap94]) for complexity classes and reductions not defined in this paper.

We introduce a notion called a $\Sigma_k(A)$-system. This notion is useful for concisely representing the computation of a stack of oracle NPTMs.

Definition 1. *1. For any $k \in \mathbb{N}^+$ and $A \subseteq \Sigma^*$, we call a tuple $[A; N_1, N_2, \ldots, N_k]$, where A is an oracle and $N_1, N_2, \ldots, N_k$ are oracle nondeterministic Turing machines, a $\Sigma_k(A)$-system. The* computation of a $\Sigma_k(A)$-system $[A; N_1, N_2, \ldots, N_k]$ *on input x, denoted by $[A; N_1, N_2, \ldots, N_k](x)$, is defined as follows:*

- *For $k = 1$, $[A; N_1](x) =_{df} N_1^A(x)$, and*
- *for $k > 1$, $[A; N_1, N_2, \ldots, N_k](x) =_{df} N_1^{L(N_2^{\cdot^{\cdot^{\cdot^{L(N_k^A)}}}})}(x)$.*

2. The language accepted by a $\Sigma_k(A)$-system*, denoted by $L[A; N_1, N_2, \ldots, N_k]$, is defined inductively as follows:*

$$L[A; N_1, N_2, \ldots, N_k] =_{df} \begin{cases} L(N_1^A) & \text{if } k = 1, \text{ and} \\ L(N_1^{L[A; N_2, N_3, \ldots, N_k]}) & \text{if } k > 1. \end{cases}$$

We capture the notion of unambiguity in $\Sigma_k(A)$-systems in the following definition.

Definition 2. *1. We say that a $\Sigma_k(A)$-system $[A; N_1, N_2, \ldots, N_k]$ is* unambiguous *if for every $1 \leq i \leq k$ and for every $x \in \Sigma^*$, $[A; N_i, N_{i+1}, \ldots, N_k](x)$ has at most one accepting path.*

2. For any $\Sigma_k(A)$-system $[A; N_1, N_2, \ldots, N_k]$, we define

$$L_{\text{unambiguous}}[A; N_1, N_2, \ldots, N_k] = \begin{cases} L[A; N_1, N_2, \ldots, N_k] & \text{if } [A; N_1, N_2, \ldots, N_k] \text{ is} \\ & \text{unambiguous}, \\ \text{undefined} & \text{otherwise}. \end{cases}$$

Roughly speaking, a property of an oracle machine is called *robust* if the machine retains that property with respect to every oracle. Below we define the property of robust unambiguity for a $\Sigma_k(A)$-system.

Definition 3. *We say that a $\Sigma_k(A)$-system $[A; N_1, N_2, \ldots, N_k]$ is* robustly unambiguous *if for every set B, the $\Sigma_k(A \oplus B)$-system $[A \oplus B; N_1, N_2, \ldots, N_k]$ is unambiguous.*

2.2 Promise Problems and Smart Reductions

Even, Selman, and Yacobi [ESY84] introduced and studied the notion of promise problems. Promise problems are generalizations of decision problems in that the set of Yes-instances and the set of No-instances must partition the set of all instances in a decision problem, whereas this is not necessarily the case with promise problems. Over the years, the notion of promise problems has proved to be useful in complexity theory. (See [Gol05] for a nice survey on some such applications of promise problems.)

Definition 4 (Based on [Gol05]; cf. [ESY84]). *A promise problem $\Pi = (\Pi_{\text{yes}}, \Pi_{\text{no}})$ is defined in terms of disjoint sets Π_{yes}, $\Pi_{\text{no}} \subseteq \Sigma^*$. The set Π_{yes} is called the set of Yes-instances, the set Π_{no} is called the set of No-instances, and the set $\Pi_{\text{yes}} \cup \Pi_{\text{no}}$ is called the promise set.*

Definition 5. *A set L polynomial-time* smart *Turing reduces to a promise problem $\Pi = (\Pi_{\text{yes}}, \Pi_{\text{no}})$, denoted by $L \leq_{s,T}^{p} \Pi$ or $L \in \mathrm{P}_s^{\Pi}$, if there is a deterministic polynomial-time Turing machine M such that for all $x \in \Sigma^*$,*

1. $x \in L \Longleftrightarrow M^{\Pi}(x)$ *accepts, and*
2. *if* $M^{\Pi}(x)$ *asks a query* y *to* Π*, then* $y \in \Pi_{\text{yes}} \cup \Pi_{\text{no}}$.

If on all input $x \in \Sigma^*$*, the querying machine* M *asks at most* k *queries, for some constant* $k \in \mathbb{N}^+$*, then we say that* L *polynomial-time* smart k*-Turing reduces to* Π *and write* $L \leq^p_{s,k\text{-}T} \Pi$ *or* $L \in \mathrm{P}_s^{\Pi[k]}$.

The following definitions are standard.

Definition 6. *Let* Π *be any promise problem.* $R^p_{s,T}(\Pi)$ *is the class of all sets* L *such that* $L \leq^p_{s,T} \Pi$*; for all* $k \in \mathbb{N}^+$*,* $R^p_{s,k\text{-}T}(\Pi)$ *is the class of all sets* L *such that* $L \leq^p_{s,k\text{-}T} \Pi$*;* $R^p_{s,b}(\Pi)$ *is the class of all sets* L *for which there exists some* $k \in \mathbb{N}^+$ *such that* $L \leq^p_{s,k\text{-}T} \Pi$.

Definition 7. *For any class of promise problems* $\mathcal{C}$ *and any reduction* $r \in \{T, k\text{-}T, b\}$*, we define* $R^p_{s,r}(\mathcal{C}) =_{df} \bigcup_{\Pi \in \mathcal{C}} R^p_{s,r}(\Pi)$.

We will study the computational power of smart Turing reductions to a particular class of promise problems, namely the class Promise-UP, which is defined as follows.

Definition 8. Promise-UP *is the class of all promise problems* $\Pi = (\Pi_{\text{yes}}, \Pi_{\text{no}})$ *for which there exists a nondeterministic polynomial-time Turing machine* N *such that for all* $x \in \Sigma^*$*,* $x \in \Pi_{\text{yes}} \Longleftrightarrow \#\mathrm{acc}_N(x) = 1,$ *and* $x \in \Pi_{\text{no}} \Longleftrightarrow \#\mathrm{acc}_N(x) = 0.$

The class $\mathrm{P}_s^{\text{Promise-UP}}$ of sets that polynomial-time smart Turing reduce to Promise-UP is a prominent class that behaves remarkably differently than the related class P^{UP}. While $\mathrm{P}_s^{\text{Promise-UP}}$ is known to contain the class FewP and the graph isomorphism problem [AK02], similar results for the case of P^{UP} are unknown.[3]

2.3 Unambiguity Based Hierarchies

Niedermeier and Rossmanith [NR98] observed that the notion of unambiguity in NPTMs can be generalized in three, perhaps distinct, ways to define unambiguity based hierarchies.

Definition 9 (Unambiguity Based Hierarchies [LR94,NR98]).

1. *The* alternating unambiguous polynomial hierarchy $\mathrm{AUPH} =_{df} \bigcup_{k \geq 0} \mathrm{AU}\Sigma^p_k = \bigcup_{k \geq 0} \mathrm{AU}\Pi^p_k$*, where* $\mathrm{AU}\Sigma^p_0 = \mathrm{AU}\Pi^p_0 =_{df} \mathrm{P}$ *and for every* $k \geq 1$*,* $\mathrm{AU}\Sigma^p_k = \exists! \cdot \mathrm{AU}\Pi^p_{k-1}$ *and* $\mathrm{AU}\Pi^p_k =_{df} \forall! \cdot \mathrm{AU}\Sigma^p_{k-1}$. [4]

[3] Arvind and Kurur [AK02] showed that the graph isomorphism problem (GI) belongs to the class SPP. Crasmaru et al. [CGRS04] observed that the proof of classifying GI into SPP, as given by Arvind and Kurur [AK02], actually yields a somewhat improved classification for GI. Their observation was that the graph isomorphism problem in fact belongs to $R^p_{s,T}(\text{Promise-UP})$, a subclass of SPP.

[4] For any arbitrary class $\mathcal{C}$, $\exists! \cdot \mathcal{C}$ is the class of all sets L for which there exists a polynomial $p(\cdot)$ and a set $L' \in \mathcal{C}$ such that for all $x \in \Sigma^*$, if $x \in L$ then there exists a unique $y \in \Sigma^{p(|x|)}$ such that $\langle x, y \rangle \in L'$, and if $x \notin L$ then for all $y \in \Sigma^{p(|x|)}$, $\langle x, y \rangle \notin L'$. Likewise, $\forall! \cdot \mathcal{C}$ is the class of all sets L for which there exists a polynomial $p(\cdot)$ and a set $L' \in \mathcal{C}$ such that for all $x \in \Sigma^*$, if $x \in L$ then for all $y \in \Sigma^{p(|x|)}$, $\langle x, y \rangle \in L'$, and if $x \notin L$ then there exists a unique $y \in \Sigma^{p(|x|)}$ such that $\langle x, y \rangle \notin L'$.

2. *The* unambiguous polynomial hierarchy *is* $\mathrm{UPH} =_{df} \bigcup_{k\geq 0} \mathrm{U}\Sigma_k^p$*, where* $\mathrm{U}\Sigma_0^p =_{df} \mathrm{P}$ *and for every* $k \geq 1$, $\mathrm{U}\Sigma_k^p =_{df} \mathrm{UP}^{\mathrm{U}\Sigma_{k-1}^p}$. *For each* $k \geq 0$*, the class* $\mathrm{U}\Pi_k^p =_{df} \mathrm{coU}\Sigma_k^p$.
3. *The* promise unambiguous polynomial hierarchy *is* $\mathcal{UPH} =_{df} \bigcup_{k\geq 0} \mathcal{U}\Sigma_k^p$*, where* $\mathcal{U}\Sigma_0^p =_{df} \mathrm{P}$, $\mathcal{U}\Sigma_1^p =_{df} \mathrm{UP}$*, and for every* $k \geq 2$, $\mathcal{U}\Sigma_k^p$ *is the class of all sets* $L \in \Sigma_k^p$ *such that for some oracle* NPTMs $N_1, N_2, \ldots, N_k$, $L = L(N_1^{L(N_2^{\cdot^{\cdot^{\cdot L(N_k)}}})})$*, and for every* $x \in \Sigma^*$ *and for every* $1 \leq i \leq k-1$, $N_1^{L(N_2^{\cdot^{\cdot^{\cdot L(N_k)}}})}(x)$ *has at most one accepting path and if* N_i *asks a query* w *to its oracle* $L(N_{i+1}^{\cdot^{\cdot^{\cdot L(N_k)}}})$ *during the computation of* $N_1^{\cdot^{\cdot^{\cdot L(N_k)}}}(x)$*, then* $N_{i+1}^{\cdot^{\cdot^{\cdot L(N_k)}}}(w)$ *has at most one accepting path. For each* $k \geq 0$*, the class* $\mathcal{U}\Pi_k^p =_{df} \mathrm{co}\mathcal{U}\Sigma_k^p$.

The following relationships among these complexity classes and other important classes are known.

Theorem 10. *1. For all* $k \geq 0$, $\mathrm{AU}\Sigma_k^p \subseteq \mathrm{U}\Sigma_k^p \subseteq \mathcal{U}\Sigma_k^p \subseteq \Sigma_k^p$ [LR94].
2. *For all* $k \geq 1$, $\mathrm{UP}_{\leq k} \subseteq \mathrm{AU}\Sigma_k^p \subseteq \mathrm{U}\Sigma_k^p \subseteq \mathcal{U}\Sigma_k^p \subseteq \mathrm{UAP} \subseteq \mathrm{SPP}$ ([LR94] + [NR98] + [CGRS04]).

3 Main Lemma

Our main lemma is Lemma 11, which we will use throughout this paper for generalizing known oracle constructions involving classes such as UP and Promise-UP to new oracle constructions involving arbitrary levels of the UPH. Roughly, Lemma 11 states the computational limitations of a $\Sigma_k(\mathcal{O})$-system, for any arbitrary $k \geq 1$, under certain weak conditions.

Lemma 11. *Fix a* $\Sigma_k(\mathcal{O})$*-system* $[\mathcal{O}; N_1, N_2, \ldots, N_k]$*, a string* $x \in \Sigma^*$*, and a set* $U \subseteq \Sigma^*$ *such that* $\mathcal{O} \cap U = \emptyset$*. Let* $r(.)$ *be a polynomial that bounds the running time of each of the machines* $N_1, N_2, \ldots, N_k$*. Then the following holds:*

1. *Suppose* $[\mathcal{O}; N_1, N_2, \ldots, N_k](x)$ *accepts and for every* $A \subseteq U$ *with* $||A|| \leq k$, $[\mathcal{O} \cup A; N_1, N_2, \ldots, N_k]$ *is unambiguous. Let*
$$C = \{\alpha \in U \mid [\mathcal{O} \cup \{\alpha\}; N_1, N_2, \ldots, N_k](x) \text{ rejects}\}.$$
Then $||C|| \leq 5^k \cdot \prod_{i=1}^{k} (r\circ)^i(|x|)$.
2. *Suppose* $[\mathcal{O}; N_1, N_2, \ldots, N_k](x)$ *rejects and for every* $A \subseteq U$ *with* $||A|| \leq k+1$, $[\mathcal{O} \cup A; N_1, N_2, \ldots, N_k]$ *is unambiguous. Let*
$$C = \{\alpha \in U \mid [\mathcal{O} \cup \{\alpha\}; N_1, N_2, \ldots, N_k](x) \text{ accepts}\}.$$
Then $||C|| \leq 5^k \cdot \prod_{i=1}^{k} (r\circ)^i(|x|)$.

Any oracle machine can be interpreted as a function mapping a set of strings to another set of strings as follows: A machine N maps any set $\mathcal{O}$ to the set $L(N^{\mathcal{O}})$. Therefore it makes sense to consider the function $\mathcal{L} : 2^{\Sigma^*} \to 2^{\Sigma^*}$ defined by a $\Sigma_k(\cdot)$-system $[\cdot; N_1, N_2, \ldots, N_k]$. (That is, define $\mathcal{L}$ so that for every $\mathcal{O} \subseteq \Sigma^*$, $\mathcal{L}(\mathcal{O}) =_{df} L[\mathcal{O}; N_1, N_2, \ldots, N_k]$.) We introduce a convenient notion called "(h, t)-ambiguity" for (partial) functions such as the ones defined by $\Sigma_k(\cdot)$-systems.

Definition 12. *For any $h \in \mathbb{N}^+$ and polynomial $t(\cdot)$, we call a partial function $\mathcal{L} : 2^{\Sigma^*} \to 2^{\Sigma^*}$* (h, t)-ambiguous *if for every $\mathcal{O}, U \subseteq \Sigma^*$ with $\mathcal{O} \cap U = \emptyset$, one of the following is true:*

1. *For some $A \subseteq U$ with $||A|| \leq h$, $\mathcal{L}(\mathcal{O} \cup A)$ is undefined, or*
2. *for every $w \in \Sigma^*$,*

$$||\{\alpha \in U \mid w \in \mathcal{L}(\mathcal{O} \cup \{\alpha\}) \iff w \notin \mathcal{L}(\mathcal{O})\}|| \leq t(|w|).$$

The machine N_1 in a $\Sigma_k(\mathcal{O})$-system $[\mathcal{O}; N_1, N_2, \ldots, N_k]$ has oracle access to the set $L[\mathcal{O}; N_2, N_3, \ldots, N_k]$. In many of our proofs, we first apply Lemma 11 to prove that under certain conditions the $\Sigma_{k-1}(\cdot)$-subsystem $[\cdot; N_2, N_3, \ldots, N_k]$ defines a (k, t)-ambiguous function $\mathcal{L}'$, where t is some polynomial and for any $\mathcal{O} \subseteq \Sigma^*$, $\mathcal{L}'(\mathcal{O})$ is defined to be $L[\mathcal{O}; N_2, N_3, \ldots, N_k]$. Then we can assume that the machine N_1 has oracle access to the set $\mathcal{L}(\mathcal{O})$, where $\mathcal{L}$ can be any arbitrary (k, t)-ambiguous function, rather than to the set $L[\mathcal{O}; N_2, N_3, \ldots, N_k]$. This works because the (k, t)-ambiguity of the function $\mathcal{L}'$ defined by the $\Sigma_{k-1}(\cdot)$-subsystem $[\cdot; N_2, N_3, \ldots, N_k]$ is the only property of $\mathcal{L}'$ that is needed in the proofs. This approach greatly simplifies our proof arguments since we will not need to deal with stacks of oracle NPTMs.

4 Applications

4.1 Comparing Bounded Ambiguity Classes with the Levels of UPH

We compare nondeterministic polynomial-time complexity classes ($\mathrm{UP}_{O(1)}$ and FewP), which are based on Turing machines having restrictions on the number of accepting paths, with levels of the unambiguous polynomial hierarchy (UPH). It is known that $\mathrm{UP}_{\leq k} \subseteq \mathrm{U}\Sigma_k^p$ in all relativized worlds. Theorem 13 shows the optimality of this inclusion with respect to relativizable proof techniques. Beigel [Bei89] constructed an oracle relative to which $\mathrm{UP}_{k(n)+1} \not\subseteq \mathrm{UP}_{k(n)}$, for every polynomial $k(n) \geq 2$. Theorem 13 subsumes this oracle result of Beigel [Bei89] for any constant k.

By a slight modification of the oracle construction in Theorem 13, we can show that the second level of the promise unambiguous hierarchy $\mathcal{U}\Sigma_2^p$ is not contained in the unambiguous polynomial hierarchy UPH. Results on relativized separations of levels of some unambiguity based hierarchy from another hierarchy have been investigated earlier. Rossmanith (see [NR98]) gave a relativized separation of $\mathrm{AU}\Sigma_k^p$ from $\mathrm{U}\Sigma_k^p$, for any $k \geq 2$. Spakowski and Tripathi [ST] constructed an oracle relative to which $\mathrm{AU}\Sigma_k^p \not\subseteq \Pi_k^p$, for any $k \geq 1$. Our relativized separation of $\mathcal{U}\Sigma_2^p$ from UPH does not seem to be implied from these previous results in any obvious way.

Theorem 13. $(\forall k \geq 1)(\exists \mathcal{A})[\mathrm{UP}^{\mathcal{A}}_{\leq k+1} \not\subseteq \mathrm{U}\Sigma^{p,\mathcal{A}}_k]$.

A straightforward adaptation of the proof technique in Theorem 13 allows to separate the second level, $\mathcal{U}\Sigma^p_2$, of the promise unambiguous polynomial hierarchy from the unambiguous polynomial hierarchy, UPH, in some relativized world. We obtain this relativized separation via Theorem 14, where a subclass, namely $\mathrm{FewP}^{\mathcal{A}}$, of $\mathcal{U}\Sigma^{p,\mathcal{A}}_2$ is separated from $\mathrm{UPH}^{\mathcal{A}}$.

Theorem 14. $(\exists \mathcal{A})\left[\mathrm{FewP}^{\mathcal{A}} \not\subseteq \mathrm{UPH}^{\mathcal{A}}\right]$.

Corollary 15. *There is a relativized world where $\mathcal{U}\Sigma^p_2$ is not contained in* UPH.

Cai, Hemachandra, and Vyskoč [CHV93] proved that smart 2-Turing access to $\mathrm{Promise\text{-}UP}$ cannot be subsumed by $\mathrm{coNP}^{\mathrm{UP}} \cup \mathrm{NP}^{\mathrm{UP}}$ in some relativized world. As a consequence, they showed that there is a relativized world where smart bounded adaptive reductions to $\mathrm{Promise\text{-}UP}$ and smart nonadaptive reductions to $\mathrm{Promise\text{-}UP}$ are nonequivalent, a characteristic that stands in contrast with the cases of UP and NP. (Both UP and NP are known to have equivalence between bounded adaptive reductions and nonadaptive reductions in all relativized worlds (see [CHV93,Wag90].) We generalize their result in Theorem 16, where we prove that smart k-Turing access to $\mathrm{Promise\text{-}UP}$ cannot be relativizably contained in $\mathrm{coNP}^{\mathrm{U}\Sigma^{p,A}_{k-1}} \cup \mathrm{NP}^{\mathrm{U}\Sigma^{p,A}_{k-1}}$, for any $k \geq 2$.

Theorem 16. $(\forall k \geq 2)(\exists \mathcal{A})\left[R^p_{s,k\text{-}T}(\mathrm{Promise\text{-}UP}^{\mathcal{A}}) \not\subseteq \mathrm{coNP}^{\mathrm{U}\Sigma^{p,A}_{k-1}} \cup \mathrm{NP}^{\mathrm{U}\Sigma^{p,A}_{k-1}}\right]$.

4.2 Simulating Nonadaptive Access by Adaptive Access (Non-promise Case)

It is known that adaptive Turing access to NP is exponentially more powerful compared to nonadaptive Turing access to NP. That is, $R^p_{(2^k-1)\text{-}tt}(\mathrm{NP}) \subseteq R^p_{k\text{-}T}(\mathrm{NP})$ [Bei91] and this inclusion relativizes. However, for the case of unambiguous nondeterministic computation such a relationship between nonadaptive access and adaptive access is not known. Cai, Hemachandra, and Vyskoč [CHV92] showed that even proving the superiority of adaptive Turing access over nonadaptive Turing access with one single query more might be nontrivial for unambiguous nondeterministic computation:

Theorem 17 ([CHV92]). *For any total, polynomial-time computable and polynomially bounded function $k(\cdot)$, there exists an oracle $\mathcal{A}$ such that*

$$R^p_{(k(n)+1)\text{-}tt}(\mathrm{UP}^{\mathcal{A}}) \not\subseteq R^{p,\mathcal{A}}_{k(n)\text{-}T}(\mathrm{UP}^{\mathcal{A}}).$$

In the next theorem, we generalize this result to the higher levels of the unambiguous polynomial hierarchy UPH.

Theorem 18. *For any total, polynomial-time computable and polynomially bounded function $k(\cdot)$, and $h \in \mathbb{N}^+$, there exists an oracle $\mathcal{A}$ such that*

$$\mathrm{R}^p_{(k(n)+1)\text{-}dtt}(\mathrm{UP}^{\mathcal{A}}_{\leq h}) \not\subseteq \mathrm{R}^{p,\mathcal{A}}_{k(n)\text{-}T}(\mathrm{U}\Sigma^{p,\mathcal{A}}_h),$$

and hence $\mathrm{R}^p_{(k(n)+1)\text{-}dtt}(\mathrm{U}\Sigma^{p,\mathcal{A}}_h) \not\subseteq \mathrm{R}^{p,\mathcal{A}}_{k(n)\text{-}T}(\mathrm{U}\Sigma^{p,\mathcal{A}}_h)$.

4.3 Simulating Nonadaptive Access by Adaptive Access (Promise Case)

Cai, Hemachandra, and Vyskoč [CHV93] proved the following partial improvement of their Theorem 17.

Theorem 19 ([CHV93]). *For any constant k, there exists an oracle $\mathcal{A}$ such that*

$$R^p_{(k+1)\text{-}tt}(\mathrm{UP}^{\mathcal{A}}) \not\subseteq R^{p,\mathcal{A}}_{s,k\text{-}T}(\text{Promise-UP}^{\mathcal{A}}).$$

Note that we have replaced "UP" by "Promise-UP" on the righthand side of the non-inclusion relation of Theorem 17. This is a significant improvement for the following reason. The computational powers of $R^p_b(\mathrm{UP})$ and $R^p_{s,b}(\text{Promise-UP})$ (the bounded Turing closure of UP and the bounded smart Turing closure of Promise-UP, respectively) are known to be remarkably different in certain relativized worlds. While it is easy to show that $\mathrm{UP}_{\leq k}$ is robustly (i.e., for every oracle) contained in $\mathrm{P}^{\text{Promise-UP}}_{s,k\text{-}T}$ for any $k \geq 1$, we have shown in Theorem 13 that for no $k \geq 2$, $\mathrm{UP}_{\leq k}$ is robustly contained in P^{UP}. Therefore, it is not immediately clear whether this improvement is impossible, i.e. whether $R^p_{(k+1)\text{-}tt}(\mathrm{UP}) \subseteq R^p_{s,k\text{-}T}(\text{Promise-UP})$ holds relative to all oracles.

However, Cai, Hemachandra, and Vyskoč [CHV93] could achieve this improvement only by paying a heavy price. In their own words:

> In our earlier version dealing with UP^A, the constant k can be replaced by any arbitrary polynomial-time computable function $f(n)$ with polynomially bounded value. It remains open whether the claim of the current strong version of Theorem 3.1 can be similarly generalized to non-constant access.

We resolve this open question. We show that Theorem 19 holds with constant k replaced by any total, polynomial-time computable and polynomially bounded function $k(\cdot)$. This result is subsumed as the special case $h = 1$ of our main result, Theorem 20, of this subsection.

Theorem 20. *For any total, polynomial-time computable and polynomially bounded function $k(\cdot)$, and $h \in \mathbb{N}^+$, there exists an oracle $\mathcal{A}$ such that*

$$\mathrm{R}^p_{(k(n)+1)\text{-}dtt}(\mathrm{UP}^{\mathcal{A}}_{\leq h}) \not\subseteq \mathrm{R}^{p,\mathcal{A}}_{s,k(n)\text{-}T}(\text{Promise-UP}^{\mathrm{U}\Sigma^{p,\mathcal{A}}_{h-1}}),$$

and hence $\mathrm{R}^p_{(k(n)+1)\text{-}dtt}(\mathrm{U}\Sigma^{p,\mathcal{A}}_h) \not\subseteq \mathrm{R}^{p,\mathcal{A}}_{s,k(n)\text{-}T}(\text{Promise-UP}^{\mathrm{U}\Sigma^{p,\mathcal{A}}_{h-1}})$.

Theorem 20 is furthermore a generalization of Theorem 19 to higher levels of the unambiguous polynomial hierarchy.

4.4 Simulating Adaptive Access by Nonadaptive Access

Sections 4.2 and 4.3 studied the limitations of simulating nonadaptive queries to $\mathrm{UP}_{\leq h}$ by adaptive queries to $\mathrm{U}\Sigma^p_h$ in relativized settings. This section complements these investigations. In particular, Theorem 21 of this section shows that in a certain relativized world, it is impossible to simulate adaptive k-Turing access to $\mathrm{UP}_{\leq h}$ by nonadaptive

(2^k-2)-tt access to $\mathrm{U}\Sigma_h^p$. This also implies optimality of robustly (i.e., for every oracle) simulating adaptive k-Turing accesses by nonadaptive (2^k-1)-tt accesses to classes such as $\mathrm{UP}_{\leq h}$ and $\mathrm{U}\Sigma_h^p$, since for any class $\mathcal{C}$, we can trivially, via brute-force method, simulate adaptive k-Turing reduction to the class by nonadaptive (2^k-1)-tt reduction to the same class.

Theorem 21 generalizes a result of Cai, Hemachandra, and Vyskoč [CHV93] from the case of $h=1$ to the case of arbitrary constant $h \geq 1$.

Theorem 21. *For any constants $k, h \in \mathbb{N}^+$, there exists an oracle $\mathcal{A}$ such that*

$$R^p_{k\text{-}T}(\mathrm{UP}^{\mathcal{A}}_{\leq h}) \not\subseteq R^{p,\mathcal{A}}_{(2^k-2)\text{-}tt}(\mathrm{NP}^{\mathrm{U}\Sigma^{p,\mathcal{A}}_{h-1}}),$$

and hence $R^p_{k\text{-}T}(\mathrm{UP}^{\mathcal{A}}_{\leq h}) \not\subseteq R^{p,\mathcal{A}}_{(2^k-2)\text{-}tt}(\mathrm{U}\Sigma^{p,\mathcal{A}}_h)$.

4.5 Fault-tolerant Access

Ko [Ko87] introduced the notion of *one-sided helping* by a set A in the computation of a set B. A set A is said to provide *one-sided help* to a set B if there is a deterministic oracle Turing machine M computing B and a polynomial $p(\cdot)$ such that (a) on any input $x \in B$, $M^A(x)$ accepts in time $p(|x|)$, and (b) for all inputs y and for all oracles C, $M^C(y)$ accepts (though perhaps $M^C(y)$ may take a longer time than $p(|y|)$) if and only if $y \in B$. Since the machine M, accepting the set B, is capable of answering correctly on faulty oracles, i.e. oracles C different from the oracle A that provides one-sided help to B, the oracle access mechanism is termed fault-tolerant (see [CHV93]). Ko [Ko87] defined $\mathrm{P}_{1\text{-help}}(A)$ to be the class of all sets B that can be one-sided helped by A.

We generalize and improve the relativized separation of $\mathrm{P}_{1\text{-help}}(\mathrm{UP})$ from UP by Cai, Hemachandra, and Vyskoč [CHV93] in Theorem 22.

Theorem 22. *For all $h \geq 1$, there exists an oracle $\mathcal{A}$ such that*

$$\mathrm{P}_{1\text{-help}}(\mathrm{UP}^{\mathcal{A}}_{\leq h}) \not\subseteq R^{p,\mathcal{A}}_{s,b}(\text{Promise-UP}^{\mathrm{U}\Sigma^{p,\mathcal{A}}_{h-1}}).$$

5 Robust Unambiguity

So far we looked at several applications of Lemma 11 in constructing relativized worlds involving arbitrary levels of the unambiguous polynomial hierarchy. Lemma 11, in essence, shows the computational limitations of a $\Sigma_k(A)$-system under certain weak restrictions. What if we impose a more stringent restriction on a $\Sigma_k(A)$-system? This question is relevant to our next investigation.

We study the power of robustly unambiguous $\Sigma_k(A)$-system in Theorem 23. (Recall from Section 2, a $\Sigma_k(A)$-system $[A; N_1, N_2, \ldots, N_k]$ is robustly unambiguous if for every oracle B, $[A \oplus B; N_1, N_2, \ldots, N_k]$ is unambiguous.) Theorem 23 illustrates the following fact: A robustly unambiguous $\Sigma_k(A)$-system is so weak that given any oracle set B and input x, the hierarchical nondeterministic polynomial-time oracle access to B in $[A \oplus B; N_1, N_2, \ldots, N_k](x)$ can be stripped down and turned into a deterministic polynomial-time oracle access (to B) without changing the acceptance behavior of the $\Sigma_k(A \oplus B)$-system on input x. As a corollary, we obtain a generic collapse of $\mathrm{U}\Sigma_k^p$ to P, for each $k \geq 1$, assuming $\mathrm{P} = \mathrm{NP}$.

Theorem 23. *For all $A \subseteq \Sigma^*$ and $k \geq 1$, if the $\Sigma_k(A)$-system $[A; N_1, N_2, \ldots, N_k]$ is robustly unambiguous, then for every $B \subseteq \Sigma^*$,*

$$L[A \oplus B; N_1, N_2, \ldots, N_k] \in \mathrm{P}^{\Sigma_k^{p,A} \oplus B}.$$

Corollary 24. *If $\mathrm{P} = \mathrm{NP}$, then relative to a (Cohen) generic G, $\mathrm{P} = \mathrm{U}\Sigma_k^p$ for all $k \geq 1$.*

The last corollary generalizes a result of Blum and Impagliazzo: If $\mathrm{P} = \mathrm{NP}$, then relative to a (Cohen) generic G, $\mathrm{P}^G = \mathrm{UP}^G$ [BI87]. Fortnow and Yamakami [FY96] demonstrated that similar collapses relative to any (Cohen) generic G do not occur at higher levels of the polynomial hierarchy. They proved that for each $k \geq 2$, there exists a tally set in $\mathrm{UP}^{\Sigma_{k-1}^{p,G},G} \cap \Pi_k^{p,G}$ but not in $\mathrm{P}^{\Sigma_{k-1}^{p,G},G}$. Thus Corollary 24 contrasts with this generic separation by Fortnow and Yamakami.

Acknowledgment We thank Lane Hemaspaandra and Jörg Rothe for their constant encouragement and support.

References

[ACRW04] S. Aida, M. Crâsmaru, K. Regan, and O. Watanabe. Games with uniqueness properties. *Theory of Computing Systems*, 37(1):29–47, 2004.

[AK02] V. Arvind and P. Kurur. Graph isomorphism is in SPP. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, pages 743–750, Los Alamitos, November 2002. IEEE Computer Society.

[BC93] D. Bovet and P. Crescenzi. *Introduction to the Theory of Complexity*. Prentice Hall, 1993.

[Bei89] R. Beigel. On the relativized power of additional accepting paths. In *Proceedings of the 4th Structure in Complexity Theory Conference*, pages 216–224. IEEE Computer Society Press, June 1989.

[Bei91] R. Beigel. Bounded queries to SAT and the boolean hierarchy. *Theoretical Computer Science*, 84(2):199–223, 1991.

[Bei93] R. Beigel. The polynomial method in circuit complexity. In *Proceedings of the 8th Structure in Complexity Theory Conference*, pages 82–95, San Diego, CA, USA, May 1993. IEEE Computer Society Press.

[BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P=?NP question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

[BI87] M. Blum and R. Impagliazzo. Generic oracles and oracle classes. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pages 118–126, October 1987.

[BS79] T. Baker and A. Selman. A second step toward the polynomial hierarchy. *Theoretical Computer Science*, 8:177–187, 1979.

[BU98] C. Berg and S. Ulfberg. A lower bound for perceptrons and an oracle separation of the $\mathrm{PP}^{\mathrm{PH}}$ hierarchy. *Journal of Computer and System Sciences*, 56(3):263–271, 1998.

[CGRS04] M. Crâsmaru, C. Glaßer, K. Regan, and S. Sengupta. A protocol for serializing unique strategies. In *Proceedings of the 29th International Symposium on Mathematical Foundations of Computer Science*. Springer-Verlag *Lecture Notes in Computer Science #3153*, August 2004.

[CHV92] J. Cai, L. Hemachandra, and J. Vyskoč. Promise problems and access to unambiguous computation. In *Proceedings of the 17th Symposium on Mathematical Foundations of Computer Science*, pages 162–171. Springer-Verlag *Lecture Notes in Computer Science #629*, August 1992.

[CHV93] J. Cai, L. Hemachandra, and J. Vyskoč. Promises and fault-tolerant database access. In K. Ambos-Spies, S. Homer, and U. Schöning, editors, *Complexity Theory*, pages 101–146. Cambridge University Press, 1993.

[ESY84] S. Even, A. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2):159–173, 1984.

[FSS84] M. Furst, J. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17:13–27, 1984.

[FY96] L. Fortnow and T. Yamakami. Generic separations. *Journal of Computer and System Sciences*, 52(1):191–197, February 1996.

[Gol05] O. Goldreich. On promise problems. Technical report TR05–018, Electronic Colloquium on Computational Complexity (ECCC), 2005.

[GS88] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. *SIAM Journal on Computing*, 17(2):309–335, 1988.

[GT05] C. Glaßer and S. Travers. Machines that can output empty words. Technical report TR05–147, Electronic Colloquium on Computational Complexity (ECCC), 2005.

[Hås87] J. Håstad. *Computational Limitations of Small-Depth Circuits*. MIT Press, 1987.

[HO02] L. Hemaspaandra and M. Ogihara. *The Complexity Theory Companion*. Springer, 2002.

[Ko85] K. Ko. On some natural complete operators. *Theoretical Computer Science*, 37(1):1–30, 1985.

[Ko87] K. Ko. On helping by robust oracle machines. *Theoretical Computer Science*, 52:15–36, 1987.

[Ko89] K. Ko. Relativized polynomial time hierarchies having exactly k levels. *SIAM Journal on Computing*, 18(2):392–408, 1989.

[LR94] K.-J. Lange and P. Rossmanith. Unambiguous polynomial hierarchies and exponential size. In *Proceedings of the 9th Structure in Complexity Theory Conference*, pages 106–115. IEEE Computer Society Press, June/July 1994.

[NR98] R. Niedermeier and P. Rossmanith. Unambiguous computations and locally definable acceptance types. *Theoretical Computer Science*, 194(1–2):137–161, 1998.

[OH93] M. Ogiwara and L. Hemachandra. A complexity theory for feasible closure properties. *Journal of Computer and System Sciences*, 46(3):295–325, 1993.

[Pap94] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Reg97] K. Regan. Polynomials and combinatorial definitions of languages. In L. Hemaspaandra and A. Selman, editors, *Complexity Theory Retrospective II*, pages 261–293. Springer-Verlag, 1997.

[Sip83] M. Sipser. Borel sets and circuit complexity. In *Proceedings of the 15th ACM Symposium on Theory of Computing*, pages 61–69. ACM Press, 1983.

[SL96] M. Sheu and T. Long. UP and the low and high hierarchies: A relativized separation. *Mathematical Systems Theory*, 29(5):423–449, 1996.

[ST] H. Spakowski and R. Tripathi. On the power of unambiguity in alternating machines. *Theory of Computing Systems*. To appear.

[Wag90] K. Wagner. Bounded query classes. *SIAM Journal on Computing*, 19(5):833–846, 1990.

[Yao85] A. Yao. Separating the polynomial-time hierarchy by oracles. In *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science*, pages 1–10, 1985.