Electronic Colloquium on Computational Complexity, Report No. 50 (2014)

On the probabilistic closure of the loose unambiguous hierarchy

Edward A. Hirsch* Dmitry Sokolov†

March 21, 2014

Abstract

Unambiguous hierarchies [NR93, LR94, NR98] are defined similarly to the polynomial hierarchy; however, all witnesses must be unique. These hierarchies have subtle differences in the mode of using oracles. We consider a "loose" unambiguous hierarchy $\mathbf{prUH}_\bullet$ with relaxed definition of oracle access to promise problems. Namely, we allow to make queries that miss the promise set; however, the oracle answer in this case can be arbitrary (a similar definition of oracle access has been used in [CR08]).

We prove that the first part of Toda's theorem $\mathbf{PH} \subseteq \mathbf{BP} \cdot \oplus\mathbf{P} \subseteq \mathbf{P}^{\mathbf{PP}}$ can be rectified to $\mathbf{PH} = \mathbf{BP} \cdot \mathbf{prUH}_\bullet$, that is, the closure of our hierarchy under Schöning's $\mathbf{BP}$ operator *equals* the polynomial hierarchy. It is easily seen that $\mathbf{BP} \cdot \mathbf{prUH}_\bullet \subseteq \mathbf{BP} \cdot \oplus\mathbf{P}$.

The proof follows the same lines as Toda's proof, so the main contribution of the present note is a new definition.

1 Introduction

Around 1990, there was a burst of results about interactive protocols [GMR89, Bab85, GS86, BOGKW88, BM88, LFKN92, BFL91, Sha90].

In the same time, Seinosuke Toda proved that $\mathbf{PH} \subseteq \mathbf{BP} \cdot \oplus\mathbf{P} \subseteq \mathbf{P}^{\mathbf{PP}}$ [Tod91]. The first part of his result can be viewed as an Arthur-Merlin game (recall that $\mathbf{AM} = \mathbf{BP} \cdot \mathbf{NP}$); however, Merlin must have an *odd* number of correct proofs. One can describe its proof as follows. We depart from a relativized version of Valiant-Vazirani's lemma and turn the polynomial hierarchy, level by level, into a multi-round Arthur-Merlin game where Merlin always has a *unique* witness. Then, this multi-round game is collapsed to just two rounds by a technique somewhat similar to the reduction of the number of rounds in Arthur-Merlin

*Steklov Institute of Mathematics at St.Petersburg, Russian Academy of Sciences, 27 Fontanka, 191023 St.Petersburg, Russia. Web: `http://logic.pdmi.ras.ru/~hirsch/`

†Steklov Institute of Mathematics at St.Petersburg, Russian Academy of Sciences, 27 Fontanka, 191023 St.Petersburg, Russia. Web: `http://logic.pdmi.ras.ru/~sokolov/`

ISSN 1433-8092

proofs ($\mathbf{AM}(k) = \mathbf{AM}(2)$) [BM88]: the probability of error is reduced and this allows to exchange neighbouring Arthur and Merlin's turns. However, it seems like to make these ideas work one needs to argue about classes of computations that are closed under the complement (since $\exists$ and $\forall$ quantifiers alternate in the polynomial hierarchy) and under majority (to reduce the probability of error). Toda overcame these obstacles by generalizing nondeterministic computations with unique witnesses to computations with an odd number of witnesses. This nice solution, however, led to the intermediate class $\mathbf{BP} \cdot \oplus\mathbf{P}$, which was not known to belong to the polynomial hierarchy, and was actually wider than needed.

In this paper we rectify the first part of Toda's theorem by replacing computations with an odd number of witnesses by unambiguous computations. However, simply requiring unique witnesses does not work. To the best of our knowledge, two notions of unambiguous hierarchies (constant-round games with unique strategies) were studied to the date: a hierarchy $\mathbf{UH}$ [NR93, NR98][1] of unambiguous computations with oracle access to languages ($\mathbf{UP}^{\mathbf{UP}^{\cdots^{\mathbf{UP}}}}$, the computation needs to be unambiguous only for the correct oracle) and a hierarchy $\mathcal{UH}$ [LR94, NR98] with *guarded* oracle access to promise problems[2] (that is, the next level of the hierarchy is obtained by adding an oracle access to the promise version of $\mathbf{UP}$, but queries outside the promise set are prohibited). Both hierarchies are contained in the unambiguous alternating polynomial-time class $\mathbf{UAP}$ [CGRS04] and thus in $\mathbf{SPP}$ [NR98] (hence $\mathbf{PP}$ and $\oplus\mathbf{P}$). Obviously they are also contained in $\mathbf{PH}$; however, replacing $\oplus\mathbf{P}$ by $\mathbf{UH}$ or $\mathcal{UH}$ does not work: Valiant-Vazirani's reduction $\mathbf{NP} \subseteq \mathbf{RP}^{\mathbf{promiseUP}}$ (in what follows, we abbreviate **promise** by **pr**) sometimes outputs an instance that has more than one solution and it is unclear how to avoid querying the oracle for such an instance (which is prohibited in $\mathbf{UH}$ or $\mathcal{UH}$).

We therefore relax the definition of the unambiguous hierarchy allowing to query the oracle outside its promise set. However, the computation must return a correct answer for *all* possible answers of the oracle to such queries. We call this a *loose access* to the oracle. (A similar notion was used by Chakaravarthy and Roy [CR08] for querying $\mathbf{prMA}$ and $\mathbf{prAM}$ by deterministic computations, and it is also implicitly used for probabilistic computations querying $\mathbf{prUP}$ when one formulates Valiant-Vazirani's lemma as $\mathbf{NP} \subseteq \mathbf{RP}^{\mathbf{prUP}}$.) The resulting hierarchy $\mathbf{prUH}_\bullet$ contains the two hierarchies $\mathbf{UH}$ and $\mathcal{UH}$ and is still contained in $\mathbf{PH}$. We prove that $\mathbf{PH} \subseteq \mathbf{BP} \cdot \mathbf{prUH}_\bullet$ (the proof goes along the same lines as Toda's theorem; however, we have to use oracles instead of Schöning's dot-operators all the way until the very end). Since $\mathbf{BP} \cdot \mathbf{prUH}_\bullet \subseteq \mathbf{BP} \cdot \oplus\mathbf{P}$, this is a strengthening of the first part of Toda's theorem. Moreover, our result is actually an *equality*; thus, we give a natural characterization of $\mathbf{PH}$ as a probabilistic closure of unambiguous computations.

Spakowski and Tripathi [ST09] asked[3] whether $\mathbf{UH}$ and $\mathcal{UH}$ collapse simultaneously with $\mathbf{PH}$. Since our result is proved level-by-level, it implies that a collapse of $\mathbf{prUH}_\bullet$ to the i-th level collapses $\mathbf{PH}$ to the $(i+2)$-th level. This, however, leaves open the question whether

[1] They attribute the initiation of this study to Hemachandra.

[2] This is similar to smart reductions used in [GS88] and was apparently suggested in the context of unambiguous computations in [CHV92a, CHV92b]).

[3] They attribute this question to [LR94]; however, we did not find it there.

a collapse of **UH** or $\mathcal{UH}$ implies a collapse of $\mathbf{prUH}_{\bullet}$ (and **PH**).

In what follows, we give definitions and prove our main theorem and its consequences. We conclude with a big list of further directions.

2 Definitions

Promise problems. A *language* is a subset of $\{0,1\}^*$, and a *promise problem* is a pair (L, A), where L is a language, and $A \subseteq \{0,1\}^*$ is a promise set. To solve a promise problem, we need to solve only its instances belonging to A.

For a class of languages $\mathcal{C}$, we consider the class of promise problems $\mathbf{pr}\mathcal{C}$ (slightly abusing the notation): namely, we consider the definition of $\mathcal{C}$ and replace all references to "every input" by references to "every input in A", where A is a promise set.

For example, $(L, A) \in \mathbf{prBPP} \iff$ there is a polynomial-time probabilistic machine M such that $\forall x \in A\ \Pr\{M(x) = L(x)\} \geq 3/4$.

Note that if a class has a semantic requirement (such as bounded error or witness uniqueness), the machine needs to satisfy it only on the promise set. Also note that nevertheless if machines in the original class stop in polynomial time, we can w.l.o.g. assume that the machines in the new class still stop in polynomial time even outside the promise (if the computational model allows to add a polynomial alarm clock).

However, if a class $\mathcal{C}$ of languages has syntactic requirements only (that is, the corresponding machines can be recursively enumerated), the corresponding promise class essentially equals $\mathcal{C}$, i.e., $\mathbf{pr}\mathcal{C} = \{(L, A) \mid L \in \mathcal{C},\ A \subseteq \{0,1\}^*\}$.

When considering a class $\mathcal{D}$ of promise problems, we assume it closed downwards w.r.t. the promise set, i.e., if $(L, A) \in \mathcal{D}$ and $B \subseteq A$, then $(L, B) \in \mathcal{D}$.

Loose oracle access. We define *loose* oracle access to a promise problem so that the oracle returns a correct answer if a query is in the promise set and returns an arbitrary answer otherwise.

For example, $L \in \mathbf{BPP}^{(O,A)} \iff$ there is a probabilistic polynomial-time oracle machine $M^\bullet$ that decides the membership in L correctly with probability at least $3/4$ irrespectively of the answers returned by the oracle on queries that do not belong to A. (In particular, the oracle can return different answers for the same query outside A.) A more formal definition follows.

Definition 1. $L \in \mathbf{BPP}^{(O,A)}$ iff there is a probabilistic polynomial-time oracle machine $M^\bullet$ that uses $r(n)$ random bits such that for every input x of length n, there is a set R of random strings of length $r(n)$ such that $|R| \geq \frac{3}{4}2^{r(n)}$ and for every string $h \in R$ and for every language L' that agrees with L on the promise set A, $M^{L'}(x, h) = L(x)$ (where $M^\bullet$ is considered as a deterministic machine receiving the input x and the random string h).

We will use the notion of loose access similarly not just for $\mathbf{BPP}^\bullet$, but for other oracle machine types as well. Throughout this paper, whenever we talk about oracle access to promise problems, we mean the "loose" definition by default.

Loose unambiguous hierarchy. We define the loose unambiguous hierarchy as follows. (To avoid possible confusion, we define only the promise version.)

- $\mathbf{prU\Sigma}_{\bullet 1} = \mathbf{prUP}$,
- $\mathbf{prU\Sigma}_{\bullet i+1} = \mathbf{prUP}^{\mathbf{prU\Sigma}_{\bullet i}}$ (with loose oracle access),
- $\mathbf{prUH}_{\bullet} = \bigcup_i \mathbf{prU\Sigma}_{\bullet i}$.

Trivially, the unambiguous hierarchies considered in [NR93, LR94, NR98] are level-by-level contained in the levels of $\mathbf{prUH}_{\bullet}$.

Proposition 1. For any class of languages $\mathcal{C}$, $\mathcal{C}^{\mathbf{prUH}_{\bullet}} \subseteq \mathcal{C}^{\oplus\mathbf{P}}$.

Proof. For any language A, queries to $L \in \mathbf{prUP}^A$ can be answered by a $\oplus\mathbf{P}^A$ oracle (consider the machine corresponding to L and treat it as an $\oplus\mathbf{P}$ machine; its answers outside the promise will be arbitrary, but it does not harm as loose access assumes that any answers will do). The statement follows by gradual top-down replacement of the oracle in $\mathbf{prU\Sigma}_{\bullet i+1} = \mathbf{prUP}^{\mathbf{prUP}^{\cdots\mathbf{prUP}}}$ starting from the highest level oracle $\mathbf{prUP}$ and collapsing $\oplus\mathbf{P}^{\oplus\mathbf{P}}$ to $\oplus\mathbf{P}$ [PZ83], i.e., $\mathcal{M}^{\mathbf{prUP}^{\mathbf{prUP}}} \subseteq \mathcal{M}^{\mathbf{prUP}^{\oplus\mathbf{P}}} \subseteq \mathcal{M}^{\oplus\mathbf{P}^{\oplus\mathbf{P}}} = \mathcal{M}^{\oplus\mathbf{P}} \subseteq \ldots \subseteq \mathcal{C}^{\oplus\mathbf{P}}$.

Note that machines underlying the class $\mathcal{C}$ do not matter since all oracle queries made by them on which their answer depends are answered correctly. $\square$

Schöning's BP· operator. Uwe Schöning [Sch89] introduced the following dot-operator **BP**· in order to consider a probabilistic version of any complexity class.

For any class of languages $\mathcal{C}$, the class $\mathbf{BP}\cdot\mathcal{C}$ is the class of languages L such that there exist $C \in \mathcal{C}$, $\epsilon > 0$ and a polynomial p such that

$$\forall x \in \{0,1\}^* \quad \Pr_{y\in\{0,1\}^{p(|x|)}}\{x \in L \iff (x,y) \in C\} > \frac{1}{2} + \epsilon,$$

or, put another way,

$$\begin{aligned}\forall x \in \{0,1\}^* \quad x \in L \Rightarrow & \Pr_{y\in\{0,1\}^{p(|x|)}}\{(x,y) \in C\} > \frac{1}{2} + \epsilon, \text{ and} \\ x \notin L \Rightarrow & \Pr_{y\in\{0,1\}^{p(|x|)}}\{(x,y) \in \overline{C}\} > \frac{1}{2} + \epsilon.\end{aligned}$$

Later other similar operators were introduced. The original proof of Toda's theorem goes in terms of these operators and concludes with $\mathbf{BP}\cdot\oplus\mathbf{P}$. Our proof (as well as folklore versions of the proof of Toda's theorem) goes in terms of oracle classes; however, the final result can be formulated in terms of the **BP**· operator as well. In order to be able to reformulate it, we need to define a promise version of the **BP**· operator with loose access to the inner promise problem.

For any class of promise problems $\mathcal{D}$, the class $\mathbf{BP} \cdot \mathcal{D}$ is the class of languages L such that there exist $D = (C, A) \in \mathcal{D}$, $\epsilon > 0$ and a polynomial p such that

$$\begin{aligned} \forall x \in \{0,1\}^* \quad x \in L \Rightarrow & \Pr_{y \in \{0,1\}^{p(|x|)}} \{(x,y) \in C \cap A\} > \frac{1}{2} + \epsilon, \text{ and} \\ x \notin L \Rightarrow & \Pr_{y \in \{0,1\}^{p(|x|)}} \{(x,y) \in \overline{C} \cap A\} > \frac{1}{2} + \epsilon. \end{aligned}$$

The following proposition is well-known and easy to see. We include its proof for completeness and to make sure it works for loose oracle access.

Proposition 2.
1. For a class of languages $\mathcal{C}$, $\mathbf{BPP}^{\mathcal{C}} = \mathbf{BP} \cdot \mathbf{P}^{\mathcal{C}}$.
2. For a class of promise problems $\mathcal{D}$, $\mathbf{BPP}^{\mathcal{D}} = \mathbf{BP} \cdot \mathbf{prP}^{\mathcal{D}}$.

Proof. 1. Inclusion $\mathbf{BP} \cdot \mathbf{P}^{\mathcal{C}} \subseteq \mathbf{BPP}^{\mathcal{C}}$ is trivial.

Consider $L \in \mathbf{BPP}^{\mathcal{C}}$. Let M^C be an oracle polynomial-time Turing machine for L. Consider a new language $L' = \{(x,r) \mid M_r^C(x) = 1\}$, where M_r^C is the answer of M^C for the particular string r of random bits.

Clearly, $L' \in \mathbf{P}^{\mathcal{C}}$. Also $\Pr_r[L(x) = L'(x,r)] > \frac{2}{3}$, hence $L \in \mathbf{BP} \cdot \mathbf{P}^{\mathcal{C}}$.

2. We use the same strategy for promise classes. Inclusion $\mathbf{BP} \cdot \mathbf{prP}^{\mathcal{D}} \subseteq \mathbf{BPP}^{\mathcal{D}}$ is trivial.

Let us write $N^{(C,B)}(x) = 1$ if machine N returns 1 for any possible answers returned by the oracle for queries outside B, $N^{(C,B)}(x) = 0$ if the answer is always 0, and $N^{(C,B)} = \perp$ if the answer depends on the oracle answers for queries outside B. Note that this notation makes sense even for deterministic machines.

Consider $L \in \mathbf{BPP}^{\mathcal{D}}$. Let M^D be an oracle polynomial-time Turing machine for L with loose access to the promise problem D. Consider a new promise problem $(L', A) = (\{(x,r) \mid M_r^D(x) = 1\}, \{(x,r) \mid M_r^D(x) = L(x)\})$

Clearly, $(L', A) \in \mathbf{prP}^{\mathcal{D}}$. Also $\Pr_r[L(x) = L'(x,r)] > \frac{2}{3}$, hence $L \in \mathbf{BP} \cdot \mathbf{prP}^{\mathcal{D}}$. $\square$

Proposition 3. $\mathbf{BP} \cdot \mathbf{prUH}_{\bullet} \subseteq \mathbf{BP} \cdot \oplus\mathbf{P}$.

Proof. Note that $\mathbf{BPP}^{\oplus\mathbf{P}} = \mathbf{BP} \cdot \mathbf{P}^{\oplus\mathbf{P}}$ by Proposition 2. Since $\oplus\mathbf{P} = \mathbf{P}^{\oplus\mathbf{P}} = \oplus\mathbf{P}^{\oplus\mathbf{P}}$ [PZ83], $\mathbf{BP} \cdot \oplus\mathbf{P} = \mathbf{BPP}^{\oplus\mathbf{P}}$. On the other hand, by Proposition 1 $\mathbf{BPP}^{\mathbf{prUH}_{\bullet}} \subseteq \mathbf{BPP}^{\oplus\mathbf{P}}$. It remains to check that $\mathbf{BP} \cdot \mathbf{prUH}_{\bullet} \subseteq \mathbf{BPP}^{\mathbf{prUH}_{\bullet}}$, that is, that our definitions of loose access for the $\mathbf{BP}\cdot$ operator and for oracle access match each other. Indeed, if on input x the $\mathbf{BPP}$ machine picks a random string, queries the oracle for (x,r) and returns its answer, the definition of $\mathbf{BP} \cdot \mathbf{prUH}_{\bullet}$ guarantees that in case $x \in L$ the proportion of strings r that yield the positive answer is at least $\frac{1}{2} + \epsilon$. Simlarly, for $x \notin L$ the probability to get the negative answer is at least $\frac{1}{2} + \epsilon$. The probability of success is then amplified to 3/4 by repetition and taking majority. $\square$

3 Proofs

In order to prove the result, we need a relativized version of Valiant-Vazirani lemma. (Since its proof hashes witnesses of the nondeterministic machine without accessing the computation itself, it clearly relativizes. The relativized $\oplus\mathbf{P}$ version of this lemma was implicitly used by [Tod91] and explicitly mentioned, for example, in [For09]).

Lemma 1 (Valiant, Vazirani [VV86]; Toda [Tod91])**.** $\mathbf{NP}^{\mathcal{C}} \subseteq \mathbf{BPP}^{\mathbf{prUP}^{\mathcal{C}}}$.

The following two lemmas generalize the corresponding lemmas in [Tod91]. Their proofs go along the same lines.

Lemma 2. $\mathbf{BPP}^{\mathbf{prBPP}^{\mathcal{C}}} = \mathbf{BPP}^{\mathcal{C}}$, where $\mathcal{C}$ can be either a class of languages or a class of promise problems.

Proof. Consider the corresponding oracle machine $M^\bullet$ making oracle queries to the oracle $O \in \mathbf{prBPP}^{\mathcal{C}}$. We can assume w.l.o.g. that the error probability of both probabilistic machines is exponentially small, say, 2^{-n} where n is the input length. In order to simulate the oracle O we just run the corresponding machine as a subroutine. The overall error of the new algorithm is the error of M^O plus $O(n^k \cdot 2^{-n})$, where $O(n^k)$ bounds the running time (hence, the number of queries) of $M^\bullet$. Note that since promise misses do not harm $M^\bullet$, the won't harm the new algorithm either (they are counted in the error probability of $M^\bullet$). $\square$

Lemma 3. $\mathbf{prUP}^{\mathbf{prBPP}^{\mathcal{C}}} \subseteq \mathbf{prBPP}^{\mathbf{prUP}^{\mathcal{C}}}$, where $\mathcal{C}$ can be either a class of languages or a class of promise problems.

Proof. Let $(L, A) \in \mathbf{prUP}^{\mathbf{prBPP}^{\mathcal{C}}}$. Consider the corresponding nondeterministic oracle machine $M^\bullet$ making oracle queries to an oracle $(O, B) \in \mathbf{prBPP}^{\mathcal{C}}$. Assume that $M^\bullet$ stops in time $p(n)$ (in particular, makes at most $p(n)$ queries of length at most $p(n)$ each), where n is the input length, and w.l.o.g $p(n) \geq n + 1$. The promise problem (O, B) is decided by a probabilistic polynomial-time machine Q^C (where $C \in \mathcal{C}$) that has error probability at most $2^{-p(n)^2}$ for every query of length at most $p(n)$ in its promise set B. Let $r(n)$ be polynomial bounding the running time of O on queries of length at most $p(n)$ (in particular, $r(n)$ bounds the number of random bits). Consider the set of random strings R_n of length $r(n)$ that lead to the correct answer of Q^C on every input in $\{0,1\}^{\leq p(n)} \cap B$. Note that $|R_n|/2^{r(n)} \geq 1 - \sum\limits_{i=1}^{p(n)} 2^i 2^{-p(n)^2} \geq 1 - 2^{-p(n)+1}$.

On input x, the new probabilistic oracle machine simply picks a random string ρ of length $r(n)$ and makes a query (x, ρ) to the promise problem $\bigcup\limits_{i=1}^{\infty} (L_i \times \{0,1\}^{r(i)}, A_i \times R_i)$, where $L_i = L \cap \{0,1\}^i$ and $A_i = A \cap \{0,1\}^i$, accepted by the following $\mathbf{UP}^{\mathcal{C}}$ machine. This machine $N^\bullet$ behaves similarly to $M^\bullet$. However, instead of querying M's oracle O (to which it does not have access) M uses the oracle C and employs Q^C as a subroutine using ρ as its random string (the same random string for each query). If $\rho \in R_n$, then all possible queries to O are answered correctly (in particular, all queries in all branches of the nondeterministic computation of N^C), and the computation protocol of N^C in this case is exactly the same as the protocol of $M^{(O,B)}$. The probability to choose such a random string is at least $1 - 2^{-p(n)+1} \geq 1 - 2^{-n}$. $\square$

We are now ready to prove the main result.

Theorem 1. $\Sigma_i, \Pi_i \subseteq \mathbf{BPP}^{\mathbf{prU\Sigma_{\bullet}}_i}$.

Proof. We prove this statement by induction. Indeed, $\Sigma_i = \mathbf{NP}^{\Sigma_{i-1}} \subseteq \mathbf{NP}^{\mathbf{BPP}^{\mathbf{prU\Sigma_{\bullet}}_{i-1}}}$ by the induction hypothesis. Then by Lemma 1 $\Sigma_i \subseteq \mathbf{BPP}^{\mathbf{prUP}^{\mathbf{BPP}^{\mathbf{prU\Sigma_{\bullet}}_{i-1}}}}$. Lemma 3 puts the latter class into $\mathbf{BPP}^{\mathbf{prBPP}^{\mathbf{prUP}^{\mathbf{prU\Sigma_{\bullet}}_{i-1}}}} = \mathbf{BPP}^{\mathbf{prBPP}^{\mathbf{prU\Sigma_{\bullet}}_i}}$. Then Lemma 2 collapses it to $\mathbf{BPP}^{\mathbf{prU\Sigma_{\bullet}}_i}$. The induction base is given by Lemma 1 for $\mathcal{C} = \{\emptyset\}$.

Since $\mathbf{BPP}^{\mathcal{C}}$ is closed under complement, the statement for Π_i also follows. $\square$

Corollary 1. $\mathbf{PH} = \mathbf{BP} \cdot \mathbf{prUH_{\bullet}}$. Moreover, a collapse of $\mathbf{prUH_{\bullet}}$ to the i-th level implies a collapse of $\mathbf{PH}$ to the $(i+2)$-th level, and a collapse of $\mathbf{PH}$ to the i-th level implies a collapse of $\mathbf{prUH_{\bullet}}$ to the same level.

Proof. By the relativized version of Gács–Sipser–Lautemann's theorem $\mathbf{BPP}^{\mathbf{prU\Sigma_{\bullet}}_i} \subseteq \Sigma_2^{\mathbf{prU\Sigma_{\bullet}}_i}$. Then $\Sigma_2^{\mathbf{prU\Sigma_{\bullet}}_i} \subseteq \Sigma_{i+2}$, because querying $\mathbf{prUP}^{\bullet}$ can be replaced by querying $\mathbf{NP}^{\bullet}$. Thus $\mathbf{BP} \cdot \mathbf{prU\Sigma_{\bullet}}_i \subseteq \Sigma_{i+2}$ and $\mathbf{BP} \cdot \mathbf{prUH_{\bullet}} \subseteq \mathbf{PH}$.

On the other hand, Theorem 1 and Proposition 2 imply $\Sigma_i \subseteq \mathbf{BP} \cdot \mathbf{U\Sigma_{\bullet}}_{i+1}$ and thus $\mathbf{PH} \subseteq \mathbf{BP} \cdot \mathbf{prUH_{\bullet}}$.

If $\mathbf{prUH_{\bullet}}$ collapses to the i-th level, then $\mathbf{PH} \subseteq \mathbf{BP} \cdot \mathbf{prUH_{\bullet}} = \mathbf{BP} \cdot \mathbf{prU\Sigma_{\bullet}}_i \subseteq \Sigma_{i+2}$. $\square$

Then the following corollary (proved by Toda [Tod91]) is immediate (see Propositions 1 and 2).

Corollary 2. $\mathbf{PH}$ is contained in $\mathbf{BP} \cdot \oplus\mathbf{P}$.

Remark 1. Note that one can consider $\mathbf{BP}\cdot$ classes as an analogue of $\mathbf{AM} = \mathbf{BP} \cdot \mathbf{NP}$. For example, Toda's theorem provides Arthur-Merlin protocols with an odd number of correct proofs. For protocols it suffices for the innermost machine to provide correct answers (and satisfy the requirements of the class) only for a substantial number of "useful" queries; we can ignore queries that appear with small total probability. Valiant-Vazirani's construction can be considered as an Arthur-Merlin protocol where in the positive case Merlin has a unique correct answer with high probability; however, in case of a bad luck Merlin may have zero or many correct answers. Theorem 1 and Corollary 1 can be considered in similar terms.

4 Open questions

Given the present rectification of the first part of Toda's theorem (actually, an equality $\mathbf{PH} = \mathbf{BP} \cdot \mathbf{prUH_{\bullet}}$), it is natural to ask about the second part. With new formulation in hand, can we do better than $\mathbf{P}^{\mathbf{PP}}$ as the upper bound for $\mathbf{PH}$?

Similarly to **PH** and other versions of the unambiguous hierarchy, it is natural to ask what class comprises "more-than-constant" levels of it, i.e., what is the analogue of the unambiguous alternative time **UAP** for $\mathbf{prUH}_\bullet$?

A shot in the same direction would be a full classification of alternating machines that have $\exists$, $\exists!$, $\forall$, $\forall!$, **BP** and other interesting types of states for both bounded and unbounded alternation. This classification would put Toda's theorem, $\mathbf{AM} = \mathbf{AM}(k)$, $\mathbf{IP} = \mathbf{PSPACE}$ and other results in a common framework.

Even if we cannot provide an analogue of **UAP**, what is the smallest known class containing $\mathbf{prUH}_\bullet$? All we know is $\mathbf{prUH}_\bullet \subseteq \mathbf{pr}\oplus\mathbf{P}$; can we put $\mathbf{prUH}_\bullet$ in **prSPP**? If it is not the case, then even the question $\mathbf{prUH}_\bullet \subseteq^? \mathbf{prPP}$ remains open, and the containment in Wagner's $\nabla\mathbf{P}$ class [Wag] or its analogue is also open (in both cases, one can only hope for the corresponding level of the counting hierarchy and the similarly built $\nabla\mathbf{P}$-hierarchy, respectively).

The relation of $\mathbf{prUH}_\bullet$ to other versions of the unambiguous hierarchy remains unclear. In particularly, while we resolve the question of [ST09] affirmatively for $\mathbf{prUH}_\bullet$ (yes, it collapses simultaneously with **PH**), the question remains unresolved for other unambiguous hierarchies.

The last, but still very important question, is the smallest class for which we can prove fixed-polynomial circuit lower bounds. To the best of our knowledge the current progress is limited to **prMA** [San09] and $\mathbf{O_2}$ (the input-oblivious version of the symmetric second level class $\mathbf{S_2}$) [CR06], but even though these classes are contained in $\mathbf{prZPP^{NP}}$ and $\mathbf{ZPP^{NP}} \subseteq \mathbf{BPP^{NP}} = \mathbf{BPP^{prUP}}$, respectively, the question of proving such bounds for the "Valiant-Vazirani" class $\mathbf{RP^{prUP}}$ (and even $\mathbf{prRP^{prUP}}$) remains open.

Acknowledgement

The authors are indebted to Alexander Knop who participated in numerous discussions on the matter. The authors are also grateful to Dmitry Itsykson for multiple comments.

The second author is partially supported by the president grant MK-2813.2014.1 for young scientists.

References

[Bab85] László Babai. Trading group theory for randomness. In Robert Sedgewick, editor, *STOC*, pages 421–429. ACM, 1985.

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

[BM88] László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988.

[BOGKW88] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In Janos Simon, editor, *STOC*, pages 113–131. ACM, 1988.

[CGRS04] Marcel Crâsmaru, Christian Glaßer, Kenneth W. Regan, and Samik Sengupta. A protocol for serializing unique strategies. In Jirí Fiala, Václav Koubek, and Jan Kratochvíl, editors, *MFCS*, volume 3153 of *Lecture Notes in Computer Science*, pages 660–672. Springer, 2004.

[CHV92a] Jin-Yi Cai, Lane A. Hemachandra, and Jozef Vyskoč. Promise problems and access to unambiguous computation. In Ivan M. Havel and Václav Koubek, editors, *MFCS*, volume 629 of *Lecture Notes in Computer Science*, pages 162–171. Springer, 1992.

[CHV92b] Jin-Yi Cai, Lane A. Hemachandra, and Jozef Vyskoč. Promise problems and guarded access to unambiguous computation. In Klaus Ambos-Spies, Steven Homer, and Uwe Schöning, editors, *Complexity Theory: Current Research*, pages 101–146. Cambridge University Press, 1992.

[CR06] Venkatesan T. Chakaravarthy and Sambuddha Roy. Oblivious symmetric alternation. In Bruno Durand and Wolfgang Thomas, editors, *STACS*, volume 3884 of *Lecture Notes in Computer Science*, pages 230–241. Springer, 2006.

[CR08] Venkatesan T. Chakaravarthy and Sambuddha Roy. Finding irrefutable certificates for $\mathrm{S}_2{}^{\mathrm{p}}$ via Arthur and Merlin. In Susanne Albers and Pascal Weil, editors, *STACS*, volume 1 of *LIPIcs*, pages 157–168. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2008.

[For09] Lance Fortnow. A simple proof of Toda's theorem. *Theory of Computing*, 5(1):135–140, 2009.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[GS86] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In Juris Hartmanis, editor, *STOC*, pages 59–68. ACM, 1986.

[GS88] Joachim Grollmann and Alan L. Selman. Complexity measures for public-key cryptosystems. *SIAM J. Comput.*, 17(2):309–335, 1988.

[LFKN92] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[LR94] Klaus-Jörn Lange and Peter Rossmanith. Unambiguous polynomial hierarchies and exponential size. In *Structure in Complexity Theory Conference*, pages 106–115. IEEE Computer Society, 1994.

[NR93] Rolf Niedermeier and Peter Rossmanith. Extended locally definable acceptance types (extended abstract). In Patrice Enjalbert, Alain Finkel, and Klaus W. Wagner, editors, *STACS*, volume 665 of *Lecture Notes in Computer Science*, pages 473–483. Springer, 1993.

[NR98] Rolf Niedermeier and Peter Rossmanith. Unambiguous computations and locally definable acceptance types. *Theor. Comput. Sci.*, 194(1-2):137–161, 1998.

[PZ83] Christos H. Papadimitriou and Stathis Zachos. Two remarks on the power of counting. In Armin B. Cremers and Hans-Peter Kriegel, editors, *Theoretical Computer Science*, volume 145 of *Lecture Notes in Computer Science*, pages 269–276. Springer, 1983.

[San09] Rahul Santhanam. Circuit lower bounds for Merlin–Arthur classes. *SIAM J. Comput.*, 39(3):1038–1061, 2009.

[Sch89] Uwe Schöning. Probabilistic complexity classes and lowness. *J. Comput. Syst. Sci.*, 39(1):84–100, 1989.

[Sha90] Adi Shamir. IP=PSPACE. In *FOCS*, pages 11–15. IEEE Computer Society, 1990.

[ST09] Holger Spakowski and Rahul Tripathi. Hierarchical unambiguity. *SIAM J. Comput.*, 38(5):2079–2112, 2009.

[Tod91] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991.

[VV86] Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theor. Comput. Sci.*, 47(3):85–93, 1986.

[Wag] Klaus W. Wagner. On weak alternation versus strong counting. Personal communication, 2013. A preliminary version appeared as *Alternating machines using partially defined "AND" and "OR"*. Technical Report 39, Institut fur Informatik, Universität Würzburg, January 1992.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de