Information Processing Letters 115 (2015) 725–730

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

On the probabilistic closure of the loose unambiguous hierarchy

Edward A. Hirsch *, Dmitry Sokolov

Steklov Institute of Mathematics at St.Petersburg, Russian Academy of Sciences, 27 Fontanka, 191023 St.Petersburg, Russia

ARTICLE INFO

Article history:
Received 19 April 2014
Received in revised form 24 March 2015
Accepted 22 April 2015
Available online 28 April 2015
Communicated by A. Muscholl

Keywords:
Computational complexity
Randomized algorithms
Unambiguous computations
Toda's theorem

ABSTRACT

Unambiguous hierarchies [1–3] 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 [4]).

In this short note 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 strengthened to $\mathbf{PH} = \mathbf{BP} \cdot \mathbf{prUH}_\bullet$, that is, the closure of our hierarchy under Schöning's **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 that allows to characterize **PH** as a probabilistic closure of unambiguous computations.

1. Introduction

Around 1990, there was a burst of results about interactive protocols [5–12].

In the same time, Seinosuke Toda proved that $\mathbf{PH} \subseteq \mathbf{BP} \cdot \oplus\mathbf{P} \subseteq \mathbf{P}^{\mathbf{PP}}$ [13]. The first part of his result can be viewed as an Arthur–Merlin game (recall that $\mathbf{AM} = \mathbf{BP} \cdot \mathbf{NP}$; cf. also [14]); however, Merlin must have an *odd* number of correct proofs. One can describe the proof of this part 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 has *unique* witnesses. 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 proofs ($\mathbf{AM}(k) = \mathbf{AM}(2)$) [9]: the probability of error is reduced and this allows to exchange neighbouring Merlin and Arthur'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 strengthen 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

* Corresponding author.
E-mail address: hirsch@pdmi.ras.ru (E.A. Hirsch).

http://dx.doi.org/10.1016/j.ipl.2015.04.010
0020-0190/

(constant-round games with unique strategies) were studied to the date: a hierarchy **UH** [1,3][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 $\mathcal{UH}$ [2,3] 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 **UP**, but queries outside the promise set are prohibited). Both hierarchies are contained in the unambiguous alternating polynomial-time class **UAP** [18] and thus in **SPP** [3] (hence in **PP** and $\oplus\mathbf{P}$). Obviously these hierarchies are also contained in **PH**; however, replacing $\oplus\mathbf{P}$ by **UH** or $\mathcal{UH}$ in Toda's theorem 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 **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 [4] for querying **prMA** and **prAM** by deterministic computations, and it is also implicitly used for probabilistic computations querying **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 **UH** and $\mathcal{UH}$ and is still contained in **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 **PH** as a probabilistic closure of unambiguous computations.

Spakowski and Tripathi [19] asked[3] whether **UH** and $\mathcal{UH}$ collapse simultaneously with **PH**. Since our result is proved level by level, it implies that a collapse of $\mathbf{prUH}_\bullet$ to the i-th level collapses **PH** to the $(i+2)$-th level. This, however, leaves open the question whether 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, one can formally define **prBPP** and **prUP** as follows.

Definition 1. $(L, A) \in \mathbf{prBPP} \iff$ there is a polynomial-time probabilistic machine M such that $\forall x \in A \ \Pr\{M(x) = L(x)\} \geq 3/4$.

Definition 2. $(L, A) \in \mathbf{prUP} \iff$ there is a polynomial-time nondeterministic machine M such that $\forall x \in A$ the machine M has at most one accepting path on x, and such $x \in A$ belongs to L iff there is such an accepting path.

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 set (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 is 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.

The notion is absolutely clear for $\mathbf{P}^{(O,A)}$, that is, for polynomial-time deterministic oracle Turing machines. It can be applied also to other computational devices. 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.

We will use the notion of loose access similarly not just for bounded-error probabilistic oracle Turing machines ($\mathbf{BPP}^\bullet$), 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. In order to avoid misunderstanding, we include more formal definition for the two main classes of computations used in this paper.

Definition 3. Let (O, A) be a promise problem. A language $L \in \mathbf{BPP}^{(O,A)}$ 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 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 O on the promise set A, $M^{L'}(x, h) = L(x)$ (where $M^\bullet$ is considered

[1] The authors of [1,3] attribute the initiation of this study to Hemachandra.

[2] This is similar to smart reductions used in [15] and was apparently suggested in the context of unambiguous computations in [16,17].

[3] They attribute this question to [2]; however, we did not find it there.