ISSN 1342-2812

Research Reports on Mathematical and Computing Sciences

Stringent Relativization

Jin-Yi Cai and Osamu Watanabe

Oct. 2006, C–227

Department of
Mathematical and
Computing Sciences
Tokyo Institute of Technology

SERIES C: Computer Science

Stringent Relativization*
— A New Approach for Studying Complexity Classes —

Jin-Yi Cai†Computer Sci. Dept.
Univ. of Wisconsin, Madison
WI 53706, USA
(jyc@cs.wisc.edu)

Osamu Watanabe‡
Dept. of Math. and Comp. Sci.
Tokyo Inst. of Technology
Tokyo 152-8552, Japan
watanabe@is.titech.ac.jp

Research Report C-227

Abstract

A new notion of relativization—stringent relativization—has been proposed recently [CW06] for discussing collapsing relations of complexity classes, with which we hope to open a new approach for studying complexity classes. Starting with the motivation of this notion, we discuss the meaning and implication of collapsing relations under the stringent relativization.

1 Introduction to Stringent Relativization

The notion of relativization was introduced to demonstrate the difficulty of proving certain relations among complexity classes. For example, regarding the most famous complexity relation, that of P and NP, Baker, Gill and Solovay [BGS75] showed that we can relativize it in both ways; that is, there exist two oracles A and B such that $\mathrm{P}^A = \mathrm{NP}^A$ (the *collapsing*) holds and $\mathrm{P}^B \neq \mathrm{NP}^B$ (the *separation*) holds. This pair of relativized results means that the relation between P and NP cannot be solved by any argument that is also applicable to relativized classes.

Here we focus on relativized collapsing relations. Cai and Watanabe [CW03, CW04, CW06] recently raised some questions on the stringency of the relativization model used for discussing collapsing relations, and they proposed an alternative relativization model, which requires us to make more stringent relativized comparisons. Note that relativization and its related notions have been used as tools for analyzing complexity classes; see, e.g., [BDG89, HO02]. We think that our stringent relativization also provides a new approach for studying complexity classes. (**Remark:** There have been some minor changes in the definition of "stringent relativization" in those three papers [CW03, CW04, CW06]. In this note we follow the definition and notations introduced/used in [CW06].)

*This work was supported in part by a NSF/JSPS grant U.S.-Japan Collaborative Research NSF SBE-INT 9726724.

†Supported in part by a NSF grant CCR-0511679.

‡Supported in part by a Grant-in-Aid for Scientific Research on Priority Areas "New Horizons in Computing" 2004-2007.

1.1 Motivation and Definition

Consider the relation between P and NP. For this relation, perhaps the most straightforward proof of a relativized collapse is to use any standard PSPACE-complete set K as an oracle. Since $\mathrm{NP}^K \subseteq \mathrm{PSPACE}$, the result of any NP^K-computation on a given instance x can be obtained by asking *one query* to the oracle K; furthermore, the query is easy to compute (i.e., polynomial-time computable) from x. This proves that $\mathrm{NP}^K \subseteq \mathrm{P}^K$. While this argument is valid, we feel that it is based on a model of computation which is not stringent enough.

To clarify our point, consider the above argument more carefully. Let K be the canonical PSPACE-complete set,

$$K \;=\; \{\, (M, x, 0^s) \;:\; M \text{ accepts } x \text{ by using } s \text{ space} \,\},$$

where M, x, and 0^s are respectively the description of a deterministic Turing machine (with no oracle access), a string in $\{0,1\}^*$, and a sequence of s 0's. We assume that $(M, x, 0^s)$ is encoded as a string in $\{0,1\}^*$ in some reasonable way.

We show that $\mathrm{P}^K = \mathrm{NP}^K$ by demonstrating that the result of any execution of any NP-query machine relative to K can be obtained by asking one simple query to K. Consider any polynomial-time nondeterministic query Turing machine Q_0 and an input x. Note that the computation of Q_0^K can be simulated by some PSPACE-machine M_0; in particular, $Q_0^K(x)$ can be simulated by $M_0(x)$ using $s_0(|x|)$ space. Thus, the result of $Q_0^K(x)$ is determined by asking a query $(M_0, x, 0^{s_0(|x|)})$ to K. Now the construction of a P-query machine Q_1 simulating Q_0^K is easy; Q_1 simply asks the query $(M_0, x, 0^{s_0(|x|)})$ to K, and then outputs its answer. In this way, for a given NP-query machine, we can define a P-query machine simulating it with queries to K.

Notice here that a query made by Q_1 is longer than queries asked in the simulated Q_0^K computation. This seems unavoidable so long as the oracle K is used. That is, queries made by simulating machine Q_1 are longer than those asked by simulated machine Q_0. Intuitively, this relativized collapse is shown by allowing simulating machines more access to an oracle than simulated machines.

Similar to this case, most of the known relativized collapsing results are proved by using such asymmetric access to an oracle. One can argue that this asymmetry is within a polynomial factor; but it nonetheless denies access to certain segments of the oracle to the simulated machine while it affords such access to the simulating machine. We think that a better comparison can be made of the underlying computational powers if we can introduce a relativization model where such asymmetry is naturally avoided.

Our solution is simple. We introduce some length bound $\ell(n)$ and require that both simulated and simulating machines query only strings of length $\leq \ell(n)$ on length n input. In this way, we restrict both machines to access to the same portion of a given oracle set. Precisely, the notion of stringent relativized class is defined as follows.

Definition 1. Fix one length bound function $\ell(n)$. We assume that the domain of inputs and queries is $\{0,1\}^*$.

(1) A query machine Q is *$\ell(n)$-query length bounded* if for any string x of length n, the machine, during the execution on input x, makes only queries of length at most $\ell(n)$.

(2) For any complexity class $\mathcal{C}$, let $\mathcal{Q}$ be a class of $\ell(n)$-query length bounded query machines corresponding to $\mathcal{C}$. For any set G, $\mathcal{C}^{\langle G \rangle}$ is the class of sets recognized by some Q in $\mathcal{Q}$ relative to G.

Remark: We can use any reasonable function for the length bound $\ell(n)$; but for technical reasons, we need to assume $\ell(n) > n$. For the sake of simplicity, we fix $\ell(n) = 3n$ throughout this note.

Now for any two complexity classes $\mathcal{C}_1$ and $\mathcal{C}_2$, we say that $\mathcal{C}_1$ *is collapsed to* $\mathcal{C}_2$ *under the stringent relativization* if $\mathcal{C}_1^{\langle G\rangle} \subseteq \mathcal{C}_2^{\langle G\rangle}$ for some oracle set G.

The stringent relativization can also be regarded as a nonuniform computation model, where a machine can access an exponentially long advice string. More specifically, we consider machines that have *random access* to a given advice string of size $L(n) = 2^{\ell(n)}$ $(= 2^{3n})$ for length n inputs. For any advice function s from n to $\{0,1\}^{L(n)}$, we use $\mathrm{P}/_{\mathrm{r}}\langle s\rangle$ (and similarly for other complexity classes) to denote the class of sets recognized by polynomial-time machines with random access to the advice string $s(0)s(1)\cdots s(n)$ during their computation on length n inputs. Then it is easy to see that for any oracle G, we have an advice function s such that $\mathrm{P}^{\langle G\rangle} = \mathrm{P}/_{\mathrm{r}}\langle s\rangle$, and vice versa.

1.2 Open Problems (1)

One may ask whether it is still possible to prove that P = NP relative to K under the stringent relativization, by using a more sophisticated reduction. Or instead of using the canonical PSPACE-complete set K, one can use some other well designed PSPACE-complete set to prove that P = NP under the stringent relativization. Or one may make use of some complete set in some much higher class $\mathcal{C}$ than PSPACE to prove the stringent relativized collapse between P and NP. While we conjecture that there is no such possibility, we have not been able to give any evidence supporting our conjecture. It would be interesting if one can derive some unlikely consequence from the assumption that $\mathrm{P}^{\langle K'\rangle} = \mathrm{NP}^{\langle K'\rangle}$ for some *complete set* K' of some higher complexity class.

2 Collapse under the Stringent Relativization

We describe a standard method for proving a stringent relativized collapse, and survey major collapsing results we have been able to prove so far.

Suppose that we want to construct some oracle set such that $\mathcal{C}_1^{\langle G\rangle} \subseteq \mathcal{C}_2^{\langle G\rangle}$. Let $\mathcal{Q}_1$ be the class (more precisely, an enumeration) of $\ell(n)$-query length bounded query machines corresponding to the class $\mathcal{C}_1$. Formally speaking, we need to consider all machines $Q_1, Q_2, \ldots$ in $\mathcal{Q}_1$ and design G so that all of them are simulated by somewhat weaker machines for the class $\mathcal{C}_2$ with the help of G. But usually it is enough to consider the simulation of one given machine Q_{i_0} in $\mathcal{Q}_1$; once we establish a way to construct an oracle for simulating Q_{i_0}, it is rather a standard argument to define G that works for all Q_i's in $\mathcal{Q}_1$. Also we usually need to consider only one given input lengths n and we may assume that the part $G^{<\ell(n)}$ has been already constructed; generalizing the construction for all input length is again easy. Thus, we fix one query machine $Q \in \mathcal{Q}_1$ and one input length n. Let $N = 2^n$ and $L = 2^{\ell(n)}$ $(= 2^{3n})$, where N is the number of all length n input strings. We consider enumerations of input strings in $\{0,1\}^n$ and query strings in $\{0,1\}^{\ell(n)}$ w.r.t. some appropriate ordering; let $w_1, \ldots, w_N$ and $q_1, \ldots, q_L$ be, respectively, this enumeration of $\{0,1\}^n$ and $\{0,1\}^{\ell(n)}$.

Note that for each w_i, the computation of $Q(w_i)$ is determined by the choice of an oracle set G, more specifically, by the part $G^{\leq \ell(n)}$. Let $x_1, \ldots, x_L$ be Boolean variables specifying the membership of $q_1, \ldots, q_L$ in G; that is, $x_j = 1$ if and only if q_j is chosen as an element of G.

Then the result of $Q^G(w_i)$ is a function f_{w_i} over these Boolean variables $x_1, \ldots, x_L$. Hence, our goal is to determine assignments to $x_1, \ldots, x_L$ so that the following system of equations hold.

$$\begin{array}{rcl} f_{w_1}(x_1, \ldots, x_L) & = & g_{w_1}(x_1, \ldots, x_L), \\ f_{w_2}(x_1, \ldots, x_L) & = & g_{w_2}(x_1, \ldots, x_L), \\ & \vdots & \\ f_{w_N}(x_1, \ldots, x_L) & = & g_{w_N}(x_1, \ldots, x_L). \end{array} \tag{1}$$

Here g is some relatively simple function corresponding to some $\mathcal{C}_2$-machine. For example, for constructing an oracle collapsing NP to P, each g_{w_i} corresponds to the execution of some P-machine on input w_i. Thus, g_{w_i} can see only poly(n) variables $x_{j_1}, \ldots, x_{j_K}$ of $x_1, \ldots, x_L$, whereas f_{w_i} has access to all $x_1, \ldots, x_L$.

As an example of this method, let us see the proof of the following stringent collapse. This is the same as the standard oracle construction [Wil83]; in fact, it is essentially the only example that the standard oracle construction can be used for proving a stringent collapse.

Theorem 1. There exists an oracle G such that $\mathrm{NP}^{\langle G \rangle} \subseteq (\mathrm{P/poly})^{\langle G \rangle}$.

Proof Outline. Consider any NP-query machine Q and any sufficiently large input length n. We assume that the oracle G has been constructed up to length $< \ell(n)$, and our task is to determine the part $G^{=\ell(n)}$, in other words, the assignments to the above mentioned Boolean variables $x_1, \ldots, x_L$. The idea is to first "freeze" (see below) the result of Q^G on every w_i by fixing assignments to some poly$(n)N$ variables of $x_1, \ldots, x_L$, which we call "freezing assignments". Then the other variables can be assigned arbitrarily without changing $Q^G(w_1), \ldots, Q^G(w_N)$. Since L $(= 2^{3n}) \gg$ poly$(n)N$ $(=$ poly$(n)2^n)$, there exists some j_0 such that all N variables $x_{j_0+1}, x_{j_0+2}, \ldots, x_{j_0+N}$ have not been used for the "freezing assignments". Thus, we use these variables to encode $Q^G(w_1), \ldots, Q^G(w_N)$; then one can get the result $Q^G(w_i)$ by simply asking q_{j_0+i} to G; in other words, the system of equations (1) holds by using a function g so that $g_{w_i}(x_1, \ldots, x_L) = x_{j_0+i}$, which can be implemented as a P-computation using an advice information for j_0.

To freeze the result of Q^G on, say, w_1, we ask whether there exists an extension of (so far constructed) G so that Q^G accepts w_1. If such an extension exists, then choose some accepting path of $Q^G(w_1)$, and fix the membership of queries to $G^{=\ell(n)}$ asked on the path, i.e., fix assignments to corresponding variables of $x_1, \ldots, x_L$, thereby "freezing" the result of $Q^G(w_1)$. (On the other hand, if no such extension exists, we do nothing; w_1 is rejected anyway no matter how G is extended.) We do this process for inputs $w_2, w_3, \ldots, w_N$, in turn. Note that at most some poly(n) assignments are made for each input; thus, as a total, all computations are frozen by assigning at most poly$(n)N$ variables. □

There are two ways to strengthen the collapse of the above theorem; consider either higher classes than NP, or lower classes than P/poly. Stringent collapsing results we have obtained so far belong to such extensions.

First consider classes higher than NP, in particular, classes in the polynomial-time hierarchy and PH itself vs. P/poly. Recall P/poly = SIZE[poly], where SIZE$[s(n)]$ denotes the class of languages recognized by $O(s(n))$-size circuits, and SIZE[poly] denotes $\bigcup_{k>0} \mathrm{SIZE}[n^k]$. With this circuit class, we point out that the above proof of Theorem 1 in fact shows that $\mathrm{NP}^{\langle G \rangle} \subseteq \mathrm{SIZE}^{\langle G \rangle}[n]$ for some G. But due to the real separation [Kan82], we cannot have, e.g.,

$\Sigma_2^{\mathrm{P},\langle \mathrm{X}\rangle} \subseteq \mathrm{SIZE}^{\langle X\rangle}[p(n)]$ for any fixed polynomial p. Thus, the above proof outline cannot be used (at least in a straightforward way) to prove $\mathrm{PH}^{\langle H\rangle} \subseteq \mathrm{SIZE}^{\langle H\rangle}[\mathrm{poly}]$ for some H. Yet we have been able to prove it [CW03] by following the above mentioned method. The key idea is to use "random partial assignments" [FSS81, Ajt83, Yao85, Cai86, Hås86]; we can show that the value of functions f_{w_i}'s corresponding to any PH-computation can be "frozen" by using random partial assignments to $x_1, \ldots, x_L$, while keeping enough unassigned variables. We then "derandomized" it by [NW94] thereby obtaining a way to encode the results so that a simple function g corresponding to some P/poly-computation can decode it.

Next consider collapses to P, lower than the class P/poly. In [CW06], we could improve the argument of [CW03] and show an oracle collapsing PH to P. The idea of using "random partial assignments" is the same; a new idea is to use the parity of some polynomially many variables $x_{j_1}, \ldots, x_{j_K}$, where the set of variables can be determined in polynomial-time from a given input w_i. That is, we define g to be $g_{w_i}(x_1, \ldots, x_L) = x_{j_1} \oplus x_{j_2} \oplus \cdots \oplus x_{j_K}$. Then using the fact that each f_{w_i} becomes a low degree polynomial after some appropriate random partial assignment and using the algebraic technique of Razborov and Smolensky [Raz87, Smo87], we can show some assignment exists satisfying our target equations (1). By a similar argument, we can also show the collapse of $\oplus$P to P. Altogether we have the following collapsing result.

Theorem 2. For any prime p, there exists an oracle set J such that $(\mathrm{Mod}_p^{\mathrm{PH}})^{\langle J\rangle} = \mathrm{P}^{\langle J\rangle}$.

2.1 Open Problems (2)

We can consider various collapsing relations proved by the standard relativization. Among them the most typical one is the relation between P and PSPACE. Showing the stringent collapse of PSPACE to P is an interesting open question and it has, as we will see below, an important implication to real complexity classes. A problem with a similar implication and maybe a more tractable one is to show

$$\mathrm{DSPACE}[n]^{\langle G\rangle} \subseteq (\mathrm{P/poly})^{\langle G\rangle}, \tag{2}$$

by some oracle set G.

3 Some Discussions

We think that our stringent relativization notion provides us with a new approach for studying complexity classes and their relationships. Below we discuss this from two points of view.

3.1 A High Level Point of View: New Relativization Framework

As we have seen above, there is some asymmetry of oracle access in the standard relativization. One may claim that this asymmetry is indeed natural; a machine with a larger resource bound should be able to access the part of the oracle that a machine with a smaller resource bound cannot access. While this claim makes sense, the asymmetry sometimes gives too easy a collapsing result, not reflecting the hardness of proving the real separation, especially for fine comparisons w.r.t. specific resource bounds.

Consider, for example, the relation between $\mathrm{NTIME}[n^2]$ and $\mathrm{DTIME}[n^3]$. We conjecture that $\mathrm{NTIME}[n^2] \not\subseteq \mathrm{DTIME}[n^3]$, and we expect this to be a difficult assertion to prove. To

justify the difficulty of proving this conjecture, we need a collapsing "relativized" result showing NTIME[n^2] $\subseteq$ DTIME[n^3]. But by the standard relativization framework, such collapse is shown by the following trivial argument: As before, we consider the simulation of a given NTIME[n^2]-query machine Q_0 by some DTIME[n^3]-query machine Q_1. Then for this simulation, we only have to encode, for each input length n, the results of Q_0's computations on all length n inputs, in the range of length n^3 part of the oracle set, which can be read by the DTIME[n^3]-query machine Q_1. This encoding is possible because Q_0 cannot make queries of length n^3. We think that such a trivial argument does not help much in our understanding as to why is it hard to prove NTIME[n^2] $\not\subseteq$ DTIME[n^3].

Cast in this light, then, what would be a reasonable relativization model? By relativization, we would like to show a "parallel world", where a relation against our conjecture such as NTIME[n^2] $\subseteq$ DTIME[n^3] indeed holds. Such a parallel world is introduced by considering a "non-standard model" of computation obtained by extending a set of primitives of computation. That is, we assume that some new set of primitives, where each primitive is computable at unit computational cost. Clearly, adding some finite set of primitives does not make any difference. Thus, we need an *unbounded* number of primitives; one reasonable framework is to extend primitives by some *a priori* fixed family consisting of an infinite number of black box Boolean predicates $\{B_k\}_{k\geq 0}$, where each B_k is a predicate defined on $\{0,1\}^k$. Of course, these new primitives must be used under a certain restriction, and a natural approach is to limit their usage in terms of the resource bound of the computation examined. This approach is taken for the standard relativization model; but this approach in general has the asymmetry problem that we pointed out earlier. For our stringent relativization model, a different restriction is introduced. We use some fixed function $\ell(n)$ on input size n, and we require the same finite subset $\{B_k\}_{0\leq k\leq \ell(n)}$ of predicates be used by both simulating and simulated machines at any given input length n. Note that the query length bound $\ell(n)$ must be smaller than the resource bounds of both simulating and simulated machines, and for investigating "polynomially bounded" complexity classes, we propose to use $\ell(n) = cn$ for the query length bound.

The stringent relativization provides an intermediate world between the one given by the standard relativization and the real world. Showing collapsing results in the stringent model gets harder than in the standard relativization model (and it is still much easier (or indeed possible) than in the real world); due to this, we expect that better understanding on the relation between target complexity classes can be obtained. For example, based on the argument proving $\mathrm{PH}^{\langle J\rangle} \subseteq \mathrm{P}^{\langle J\rangle}$, we can prove that $\mathrm{NTIME}^{\langle J\rangle}[n^2] \subseteq \mathrm{DTIME}^{\langle J\rangle}[n^k]$ for some constant $k > 2$. Here recall the key method explained in Section 2. Although some NTIME[n^2]-machine can make all $2^{\ell(n)}$ queries on its nondeterministic computation on every input of length n, one can make any such nondeterministic computation useless by choosing answers to these queries (i.e., the oracle $J^{=\ell(n)}$) appropriately, thereby deriving some relatively simple DTIME[n^k]-simulation. We think that such an observation would give us some insight into the nondeterministic computation.

3.2 A Technical Perspective: New Approach

Relativization has been also used to study the real world computation. Some stringent collapses may have some implications for the structure of real complexity classes. In fact, the following very nice observation due to Lance Fortnow [For05] shows the technical importance of the stringent P vs. PSPACE relation.

Theorem 3. The stringent collapse of PSPACE to P implies

$$\mathrm{P} \not\subseteq \mathrm{DSPACE}[\mathrm{polylog}(n)].$$

Proof Outline. We consider the contrapositive; that is, we assume that $\mathrm{P} \subseteq \mathrm{DSPACE}[\mathrm{polylog}(n)]$ and derive $\mathrm{PSPACE}^{\langle G\rangle} \not\subseteq \mathrm{P}^{\langle G\rangle}$ for any oracle set G.

First note that there exists some deterministic machine diagonalizing all P-machines and running in time $2^{O(n)}$. It is easy to modify it to a $2^{O(n)}$-time stringent query machine Q that diagonalizes all polynomial-time stringent query machines. That is, for any oracle set G, we have $L(Q^{\langle G\rangle}) \notin \mathrm{P}^{\langle G\rangle}$. Consider the following set.

$$L_0 \;=\; \{\; (x, a_1, a_2, \ldots, a_{\widetilde{L}}) \,|\, \widetilde{L} = 2^{3|x|+1} - 1, \text{ and } Q^{[a_1,\ldots,a_{\widetilde{L}}]} \text{ accepts } x \;\}.$$

Here by $[a_1, \ldots, a_{\widetilde{L}}]$ (where each $a_i \in \{0,1\}$), we mean a set $A \subseteq \{0,1\}^{\leq 3n}$ with a characteristic function χ_A satisfying $\chi_A(\lambda) = a_1$, $\chi_A(0) = a_2$, $\chi_A(1) = a_3$, …, $\chi_A(1^{3n}) = a_{\widetilde{L}}$. Then we have $L_0 \in \mathrm{P}$, and by the assumption, it holds that $L_0 \in \mathrm{DSPACE}[\mathrm{polylog}(n)]$; that is, we have some deterministic polylog-space machine recognizing L_0. It is then easy to see that this polylog-space computation (on a given input of length n) can be simulated by some deterministic polynomial-space stringent query machine using the oracle $[a_1, a_2, \ldots, a_{\widetilde{L}}]$. Therefore, we have $L(Q^{\langle G\rangle}) \in \mathrm{PSPACE}^{\langle G\rangle}$ for any oracle G, thereby concluding $\mathrm{PSPACE}^{\langle G\rangle} \not\subseteq \mathrm{P}^{\langle G\rangle}$. □

Note that the above argument can be modified to show that

$$\exists G\, [\; \mathrm{DSPACE}^{\langle G\rangle}[n] \subseteq \mathrm{P}^{\langle G\rangle} \;] \implies \mathrm{DL} \subsetneq \mathrm{P}.$$

where $\mathrm{DL} = \mathrm{DSPACE}[\log n]$. Also note that we can diagonalize all polynomial-size circuits by some Δ_3^{E}-computation, thereby showing $\Delta_3^{\mathrm{E}} \not\subseteq \mathrm{P/poly}$ [CW03], where Δ_3^{E} denotes the $2^{O(n)}$-version of the class Δ_3^{P}. Thus, by almost the same argument, we can show the following relation.

$$\exists G\, [\; \mathrm{DSPACE}^{\langle G\rangle}[n] \subseteq (\mathrm{P/poly})^{\langle G\rangle} \;] \implies \mathrm{DL} \subsetneq \Delta_3^{\mathrm{P}}. \tag{3}$$

Therefore, the solution to our open question (2) implies this real separation.

3.3 Open Problems (3)

Recall that the separation from P/poly can be made by a subclass of Δ_3^{E}; for example, we can show that $\mathrm{S}_2^{\mathrm{E}} \not\subseteq \mathrm{P/poly}$, where $\mathrm{S}_2^{\mathrm{E}} \subseteq \Sigma_2^{\mathrm{E}} \cap \Pi_2^{\mathrm{E}}$ [Cai01]. Thus, one may expect a stronger consequence than (3), namely, $\mathrm{DL} \subsetneq \mathrm{S}_2^{\mathrm{P}}$. But the proof of the separation lacks certain constructivity in a technical sense (see [CW03]), and therefore it cannot be used for Fortnow's proof above. Our last open question is to improve Fortnow's proof and show the following relation.

$$\exists G\, [\; \mathrm{DSPACE}^{\langle G\rangle}[n] \subseteq (\mathrm{P/poly})^{\langle G\rangle} \;] \implies \mathrm{DL} \subsetneq \mathrm{S}_2^{\mathrm{P}}.$$

Acknowledgments

We would like to thank Lane Hemaspaandra for giving us the opportunity to present this article in SIGACT News and for various comments on the stringent relativization notion. We also thank Lance Fortnow for generously allowing us to present his unpublished observation in this article.

and the second author is also supported in part by a Grant-in-Aid for Scientific Research on Priority Areas "New Horizon in Computing" 2004–2007.

References

[Ajt83] M. Ajtai, Σ_1^1-formulae on finite structures, *Ann. Pure Applied Logic* 24, 1–48, 1983.

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

[BDG89] J. Balcázar, J. Díaz, and J. Gabarró, *Structural Complexity I & II*, Springer, 1989 and 1990.

[Cai86] J-Y. Cai, With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy, in *Proc. 18th ACM Sympos. on Theory of Comput.* (STOC'89), 21–29, 1986. (The final version appeared in *J. Comp. Syst. Sci.* 38(1), 68–85, 1989.)

[Cai01] J-Y. Cai, $S_2^P \subseteq ZPP^{NP}$, in *Proc. 42th IEEE Symposium on Foundations of Computer Science* (FOCS'01), IEEE, 620–628, 2001.

[CW03] J-Y. Cai and O. Watanabe, On proving circuit lower bounds against the polynomial-time hierarchy: positive and negative results, *Proc. 9th International Computing and Combinatorics Conference* (COCOON'03), LNCS 2697, 202–211, 2003. (The journal version appeared in *SIAM J. Comput.* 33(4), 989–1009, 2004.)

[CW04] J-Y. Cai and O. Watanabe, Relativized collapsing between BPP and PH under stringent oracle access, *Inform. Process. Lett.* 90, 147–154, 2004.

[CW06] J-Y. Cai and O. Watanabe, Random access to advice strings and collapsing results, *Algorithmica*, to appear.

[For05] L. Fortnow, personal communication, 2005.

[FSS81] M. Furst, J. Saxe, and M. Sipser, Parity, circuits, and the polynomial time hierarchy, in *Proc. 22nd IEEE Symposium on Foundations of Computer Science* (FOCS'81), IEEE, 260–270, 1981.

[Hås86] J. Håstad, Almost optimal lower bounds for small depth circuits, in *Proc. 18th ACM Symposium on Theory of Computing* (STOC'86), ACM, 6–20, 1986.

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

[Kan82] R. Kannan, Circuit-size lower bounds and non-reducibility to sparse sets, *Information and Control* 55, 40–56, 1982.

[NW94] N. Nisan and A. Wigderson, Hardness vs randomness, *J. Comput. Syst. Sci.* 49, 149–167, 1994.

[Raz87] A. Razborov, Lower bounds on the size of bounded depth networks over a complete basis with logical addition, *Mathematical Notes of the Academy of Sciences of the USSR* 41, 333–338, 1987.

[Smo87] R. Smolensky, Algebraic methods in the theory of lower bounds for Boolean circuit complexity, in *Proc. 19th ACM Symposium on Theory of Computing* (STOC'87), ACM, 77-82, 1987.

[Wil83] C.B. Wilson, Relativized circuit complexity, in *Proc. 24th IEEE Symposium on Foundations of Computer Science* (FOCS'83), IEEE, 329–334, 1983.

[Yao85] A.C. Yao, Separating the polynomial-time hierarchy by oracles, in *Proc. 26th IEEE Symposium on Foundations of Computer Science* (FOCS'85), IEEE, 1-10, 1985.