Random Access to Advice Strings and Collapsing Results

Jin-Yi Cai * Osamu Watanabe †

Abstract

We propose a model of computation where a Turing machine is given random access to an *advice string*. With random access, an advice string of exponential length becomes meaningful for polynomially bounded complexity classes. We compare the power of complexity classes under this model. It gives a more stringent notion than the usual model of computation with relativization. Under this model of random access, we prove that there exist advice strings such that the Polynomial-time Hierarchy PH and Parity Polynomial-time ⊕P all collapse to P. Our main proof technique uses the decision tree lower bounds for constant depth circuits [Yao85, Cai86, Hås86], and the algebraic machinery of Razborov and Smolensky [Raz87, Smo87].

1 Introduction

In computational complexity theory, we cannot separate between many complexity classes. It is generally believed that these separation results are very hard to prove. Among the supporting evidence for such a pessimistic belief, people frequently cite the collapsing results under *relativization*, especially for complexity classes defined in non-randomized terms, such as P, NP, Σ_d^{p}, ⊕P, PSPACE, etc.

Consider, for example, the most famous P vs. NP conjecture. 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. Intuitively, for each oracle set X, the relative computation model allowing oracle queries to X provides a "relativized complexity world" where all computation is the same as our real world except that one can use some special set of instructions, i.e., queries to the oracle set X. It is said that most of known proofs can be *relativized*; that is, they are applicable in such relativized worlds. Therefore, having the above oracles A and B means that these proof techniques can not resolve the P vs. NP conjecture. For P vs. NP or PSPACE, perhaps the most straightforward proof of a relativized collapse is $\mathrm{P}^{\mathrm{QBF}} = \mathrm{NP}^{\mathrm{QBF}} = \mathrm{PSPACE}^{\mathrm{QBF}}$.

However, we feel that this argument is basd on a model of computation which is not stringent enough. This is especially true for most of the relativized collapsing results. More precisely, relativized collapsing results are often proved by allowing stronger usage of an oracle to a simulating machine than to a simulated machine.

*Computer Sciences Department, University of Wisconsin, Madison, WI 53706, USA. Research supported in part by NSF grants CCR-0208013 and CCR-0196197. Email: jyc@cs.wisc.edu

†Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Meguro-ku Ookayama, Tokyo 152-8552, Japan. Email: watanabe@is.titech.ac.jp

Consider two complexity classes $\mathcal{C}_1$ (such as P) and $\mathcal{C}_2$ (such as NP or PSPACE). Let $\{\mathcal{M}_i\}$ be an enumeration representing the class $\mathcal{C}_2$, and let $\mathcal{M}$ be an arbitrary machine from this enumeration. A typical proof for a relativized collapsing result is to code the computation of $\mathcal{M}$ for inputs of length n, in the oracle, in such a way that another machine $\mathcal{M}'$ representing $\mathcal{C}_1$ can recover the results. In order not to "interfere" with computations of $\mathcal{M}$ at length n, these results are coded at locations *beyond* what $\mathcal{M}$ can access at input of length n, and $\mathcal{M}'$ is allowed a running time and oracle access greater than that of $\mathcal{M}$. This encoding is sometimes explicitly carried out, sometimes implicitly done such as with the proof of $\mathrm{P}^{\mathrm{QBF}} = \mathrm{PSPACE}^{\mathrm{QBF}}$. In terms of the simulation by the $\mathrm{P}^{\mathrm{QBF}}$ machine $\mathcal{M}'$ simulating the $\mathrm{PSPACE}^{\mathrm{QBF}}$ computation $\mathcal{M}$ on an input x, $\mathcal{M}'$ will access an oracle location polynomially longer than where the corresponding access $\mathcal{M}$ makes. That is, $\mathcal{M}'$ is given more powerful oracle access than $\mathcal{M}$. 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 affords such access to the simulating machine. Moreover, if one actually relativizes the proofs of the few separation results such as the hierarchy theorems, one observes that this asymmetry is *not* present in the relativized proof.

In order to rectify this problem we propose a model of computation that is more stringent than the usual relativization computation. This turns out to be equivalent to a generalization of the notion of advice strings proposed by Karp and Lipton [KL80]. Intuitively, any relativized result can be regarded as a comparison between complexity classes under a certain nonuniform setting provided by an (infinite) advice, namely an oracle. Here we generalize the advice string formulation of Karp and Lipton by allowing random access to the advice string, so that advice strings longer than polynomial length become meaningful for polynomial time bounded computations. Then we compare complexity classes, given such nonuniform advice strings. That is, we compare two machines $\mathcal{M}_1$ and $\mathcal{M}_2$ (from complexity classes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively) that have random access to the same advice string s_n given by an advice function, for their computation of any input of length n. Both machines will have complexity bounds that allow access to any bit of the advice string. This way we compare them on the same footing. Note that, since the advice string has a length accessible to both $\mathcal{M}_1$ and $\mathcal{M}_2$, we cannot in general "preserve" the computation of one and let it be read by another, as in the usual relativization model.

Our main results in this paper show that both parity polynomial-time $\oplus$P and the polynomial-time hierarchy PH collapse to P for some exponential-size advice strings. More precisely, for P and $\oplus$P (resp., P and PH), we show some advice function giving an advice string of length $2^{(1+\epsilon)n}$ for each input length n, with which $\oplus$P (resp., PH) collapses to P. We use decision tree lower bounds for constant depth circuits [Yao85, Cai86, Hås86] and the algebraic machinery of Razborov and Smolensky [Raz87, Smo87]. It is open whether one can collapse PSPACE and P under this notion of random access to advice.

Results of this type are mainly of value in delineating the limit of our ability to settle some outstanding questions on complexity classes. Our model of random access to advice strings provides a more stringent model than the usual relativization model, and therefore it provides a more stringent perspective on the "provability" question. The open status of a collapse of PSPACE to P under random access to advice is particularly interesting in view of a result of Kozen [Ko78]: If PSPACE $\neq$ P, then there exists a proof of this fact by diagonalization.

2 Random Access to Advice Strings

Recall the definition of $\mathcal{C}$/poly by Karp and Lipton [KL80]. We generalize this notion by allowing the underlying machines to have random access to an advice string. Let us fix any "length function" ℓ from $\mathbf{N}$ to $\mathbf{N}$. A function $s : n \mapsto \{0,1\}^{\ell(n)}$ is called an *advice function of size* $\ell(n)$. Given any advice function s of size $\ell(n)$, we say a language L is in the class $\mathcal{C}/s$ *via random access to advice* if there is some machine $\mathcal{M}$ representing the class $\mathcal{C}$, such that $x \in L$ iff $\mathcal{M}(x; s(|x|))$ accepts, where we denote the computation $\mathcal{M}$ on x with random access to $s(|x|)$ by $\mathcal{M}(x; s(|x|))$. (The notion of *random access* is the usual one: A machine $\mathcal{M}$ can write down an index to a bit of $s(|x|)$ on a special tape and then it gets that bit in unit time.) We denote this language as $L(\mathcal{M}; s)$. Clearly, if a time bound being considered is larger than the advice size, then the random accessibility is not necessary, and this notion is the same as the one by Karp and Lipton. (In the following, all complexity bounds and length functions are time and space constructible as appropriate. Furthermore, we assume that $\log(\ell(n))$ is polynomially bounded, which is reasonable for comparing with polynomial-time classes even if we allow random access to an advice string.)

Let s be any advice function. We say *collapsing occurs w.r.t.* s (write as $\mathcal{C}_1/s \subseteq \mathcal{C}_2/s$) if for every machine $\mathcal{M}_1$ representing $\mathcal{C}_1$, there is a machine $\mathcal{M}_2$ representing $\mathcal{C}_2$, such that $L(\mathcal{M}_1; s) = L(\mathcal{M}_2; s)$. We say *two classes are equal w.r.t.* s (write as $\mathcal{C}_1/s = \mathcal{C}_2/s$) if both $\mathcal{C}_1/s \subseteq \mathcal{C}_2/s$ and $\mathcal{C}_2/s \subseteq \mathcal{C}_1/s$. On the other hand, we say *separation occurs w.r.t.* s (write as $\mathcal{C}_1/s \not\subseteq \mathcal{C}_2/s$) if there exists some machine $\mathcal{M}_1$ representing $\mathcal{C}_1$ such that $L(\mathcal{M}_1; s) \neq L(\mathcal{M}_2; s)$ for any machine $\mathcal{M}_2$ representing $\mathcal{C}_2$.

Then our main results can be stated as follows.

Theorem 1. For any length bound $\ell(n) \geq 2^{(1+\delta)n}$, where $\delta > 0$ is any positive constant, there exists an advice function s of advice size $\ell(n)$ such that $\oplus\mathrm{P}/s \ = \ \mathrm{P}/s$.

Remark. The same result is provable for the relationship between P and Mod_p class, for any prime p. Also $\delta > 0$ can be improved. We only need $\ell(n)/2^n$ to be superpolynomial.

Theorem 2. For any length bound $\ell(n) \geq 2^{(1+\delta)n}$, where $\delta > 0$ is any positive constant, there exists an advice function s of advice size $\ell(n)$ such that $\mathrm{PH}/s \ = \ \mathrm{P}/s$.

3 Class P vs. Class $\oplus$P

In this section we consider the relation between P and $\oplus$P and prove Theorem 1. The proof techniques will be extended in the next section to prove Theorem 2.

To simplify the presentation we will consider only $\log(\ell(n)) = (1+\delta)n$. It is easy to extend the following proof to any $\ell(n)$ with $\log(\ell(n)) \geq (1+\delta)n$.

Proof of Theorem 1. Let $\mathcal{M}_1, \mathcal{M}_2, \ldots$ be a standard enumeration of all $\oplus$P machines. Our goal is to construct an advice function s with $s(n) \in \{0,1\}^{\ell(n)}$, with which the computation of every $\mathcal{M}_i(x; s(|x|))$ can be simulated by some P computation with the common advice $s(|x|)$. Let us fix any $\oplus$P machine $\mathcal{M}$ and any input length n, and discuss how to design $s(n)$ so that some P machine can simulate $\mathcal{M}$ on $\{0,1\}^n$ with advice $s(n)$. It would be easy later to "paste" together a single $s(n)$ for all machines to be considered at length n. (Only finitely many need to be dealt with at any finite length n. We will omit this detail.)

Let $m = n^{O(1)}$ be the maximum number of accesses to the advice string made by $\mathcal{M}$ on any nondeterministic path on any input of length n. We assume that n is sufficiently large.

Let $L = 2^{(1+\delta)n}$. We will consider the advice string $s(n)$ of length L as being indexed by a binary string of length $I = (1+\delta)n$.

For any $x \in \{0,1\}^n$, define S_x to be some subset of $\{0,1\}^I$ of size $\approx nm$. We want $\{S_x\}_{x\in\{0,1\}^n}$ to be a family of pair-wise disjoint subsets of $\{0,1\}^I$. For example, for $s = \lceil \log nm \rceil$, we can define

$$S_x \;=\; \{\, xu0^{I-(n+s)} \mid u \in \{0,1\}^s \,\}.$$

Each string in $\bigcup_{x\in\{0,1\}^n} S_x$ is the index of a bit in $s(n)$. We assign Boolean variables for these bits, and denote the set of these Boolean variables as Z. Let $M = |Z|$; note that $M \leq 2nm2^n \ll 2^I$. Let us name the Boolean variables in Z as $z_1, z_2, \ldots, z_M$.

Assign arbitrarily the bit values for all bits in $s(n)$ other than those in Z. Then, for any input $x \in \{0,1\}^n$, $\mathcal{M}(x; s(n))$ is completely determined by the values of z_i. That is, $\mathcal{M}(x; s(n))$ is a function on Boolean variables $z_1, \ldots, z_M$. Furthermore, since $\mathcal{M}(x; s(n))$ is a parity computation asking at most m queries on each nondeterministic path, we may consider $\mathcal{M}(x; s(n))$ as a parity (or its negation) of (at most $\sum_{i=0}^{m} 2^i \binom{M}{i}$ many) conjunctions of at most m literals from $z_1, \ldots, z_M$. Thus, $\mathcal{M}(x; s(n))$ is expressed by a polynomial $f_x(z_1, \ldots, z_M)$ of degree $\leq m$ with integer coefficients mod 2. Note that $f_x(z_1, \ldots, z_M)$ is multilinear, because we may assume that each bit is not queried more than once on each nondeterministic path.

Now we would like to assign $z_1, \ldots, z_M$ so that the following system of equations $(*1)$ holds (under the mod 2 computation) for $\{0,1\}^n = \{x_1, \ldots, x_N\}$ (where $N = 2^n$).

$$(*1)\quad \begin{cases} f_{x_1}(z_1, \ldots, z_M) &=& 1 - \prod_{z_j \in S_{x_1}} z_j, \\ & \vdots & \\ f_{x_N}(z_1, \ldots, z_M) &=& 1 - \prod_{z_j \in S_{x_N}} z_j. \end{cases}$$

If this assignment is feasible (i.e., the advice string $s(n)$ is constructed satisfying $(*1)$), then for any $x \in \{0,1\}^n$, one simply needs to check the membership of elements of S_x; $\mathcal{M}(x; s(n))$ can then be computed as $1 - \prod_{z_j \in S_x} z_j$ in polynomial time.

Suppose, for a contradiction, that this is impossible to achieve. Then, since for every 0 or 1 value of $z_1, \ldots, z_M$, each f_x takes a 0 or 1 value, it follows that for every assignment to the $z_1, \ldots, z_M$, there exists some $x \in \{0,1\}^n$ such that

$$f_x(z_1, \ldots, z_M) \;=\; \prod_{z_j \in S_x} z_j.$$

Thus, for all 0,1-assignments to $z_1, \ldots, z_M$, we have

$$\prod_{1\leq i\leq N} \left[\prod_{z_j \in S_{x_i}} z_j - f_{x_i}(z_1, \ldots, z_M) \right] \;=\; 0.$$

Then it follows from Fact 1 stated below that modulo the ideal $J = (z_1^2 - z_1, \dots, z_M^2 - z_M)$, the left hand side expression is identical to 0. In other words, we have the identity

$$\prod_{1 \le i \le N} \prod_{z_j \in S_{x_i}} z_j \;=\; L(z_1, \dots, z_M),$$

in the ring $\mathbf{Z}_2[z_1, \dots, z_M]/J$, where L is a polynomial of degree at most $(N-1)2^s + m$. On the other hand, the degree of the lefthand side of the above equality is $N2^s$, which is larger than $(N-1)2^s + m$. A contradiction. $\Box$

Fact 1. For any prime p, let $F(x_1, ..., x_n)$ be a polynomial evaluated to 0 modulo p on all 0,1-assignments to $x_1, ..., x_n$. Then modulo the ideal $J = (x_1^2 - x_1, ..., x_n^2 - x_n)$, i.e., in the ring $\mathbf{Z}_p[x_1, ..., x_n]/J$, $F(x_1, ..., x_n)$ is identical to 0.

4 Class P vs. Class PH

We now show that there exists an advice function of advice size $2^{(1+\delta)n}$, such that the class PH collapses to P with random access to the advice strings given by the advice function. The modification in the proof from $2^{(1+\delta)n}$ to larger $\ell(n)$ is obvious. For simplicity of presentation we will assume $\ell(n) = 2^{(1+\delta)n}$ in what follows. We prove the following result for a fixed level $\Sigma_d^{\rm p}$; the construction for the advice string for PH follows since PH is a countable union of classes $\Sigma_d^{\rm p}$, $d \ge 0$.

Theorem 3. For any constant $d \ge 0$, and constant $\delta > 0$, let $\ell(n) = 2^{(1+\delta)n}$; then there exists an advice function s of advice size $\ell(n)$ such that $\Sigma_d^{\rm p}/s \;=\; {\rm P}/s$.

Before stating our proof in detail, we explain its outline and some background. We begin by recalling the decision tree version of the Switching Lemma.

Some notions and notations first. For any Boolean function f over variables $x_1, \dots, x_n$, a *random restriction* ρ is a random function that assigns each x_i either 0, 1, or $*$, with probability $\Pr[\rho(x_i) = *] = p$ (for some specified parameter p) and $\Pr[\rho(x_i) = 0] = \Pr[\rho(x_i) = 1] = (1-p)/2$, for each i independently. Assigning $*$ means to leave it as a variable. Let $f\mid_\rho$ denote a function obtained by this random restriction.

The decision tree complexity of a Boolean function f, denoted by $\mathrm{DC}(f)$, is the smallest depth of a Boolean decision tree computing the function. It can be shown easily that if $\mathrm{DC}(f) \le t$, then f can be expressed both as an AND of OR's as well as an OR of AND's, with bottom fan-in at most t. Moreover, what is crucial for our argument is the following property: If $\mathrm{DC}(f) \le t$, then f can be expressed as a polynomial on the variables, with integer coefficients and with degree at most t. In fact this polynomial always evaluates to 0 or 1, for any 0-1 assignments to its variables.

Superpolynomial lower bounds for constant depth circuits were first proved by Furst, Saxe and Sipser [FSS81], and by Ajtai [Ajt83]. Exponential lower bounds of the form $2^{n^{\Omega(1/d)}}$ for depth d circuits were first proved by Yao [Yao85] in a breakthrough result. Yao's bound was further improved by Håstad [Hås86] to $2^{\frac{1}{10}n^{\frac{1}{d-1}}}$, and his proof has become the standard proof. Independently, Yao's work was improved upon in another direction. Cai [Cai86] investigated whether constant depth circuits of size $2^{n^{\Omega(1/d)}}$ must err on an asymptotically 50 % of inputs against parity. To attack this problem, the decision tree point of view was first introduced in [Cai86]. This approach in terms of inapproximability has been found most fruitful in the beautiful work of Nisan and Wigderson [Nis91, NW94] on pseudorandom generators.

Adapting Håstad's proof to the decision tree model, one can prove the following.

Lemma 1. For any depth $d+1$ Boolean circuit C on $z_1, \ldots, z_L$, with bottom fan-in at most t,

$$\Pr[\ \mathrm{DC}(C\ |_\rho) \geq t\ ]\ \leq\ \frac{\mathrm{size}(C)}{2^t},$$

where ρ is a random restriction with the parameter $p = \Pr[z_i = *] = 1/(10t)^d$.

We now explain our construction. Fix any Σ_d^{p} machine $\mathcal{M}$ and any sufficiently large input length n. We want to construct $s(n)$, such that the computation $\mathcal{M}(x; s(n))$ can be simulated by a polynomial-time deterministic machine, for all x of length n. Constructing the advice function s for the simulation of *all* Σ_d^{p} machines can be done as before for $\oplus$P and is omitted here.

Thus, from now on, we are concerned with the simulation of $\mathcal{M}$ on 2^n inputs of length n. Let m be an integer bounding $\mathcal{M}$'s running time on inputs of length n, where $m = O(n^k)$ for some $k \geq 0$. Let $I = (1+\delta)n$ and $L = 2^I$. Let $z_1, z_2, \ldots, z_L$ be Boolean variables denoting the bits in $s(n)$. Let Z denote the set of all Boolean variables $z_1, \ldots, z_L$. With a slight abuse of notation we will also let Z denote a set of corresponding indeterminants.

For any input string $x \in \{0,1\}^n$, consider the computation of $\mathcal{M}(x; s(n))$. The computation $\mathcal{M}(x; s(n))$ is a function from the Boolean variables $z_1, \ldots, z_L$ to $\{0,1\}$. Furthermore, since $\mathcal{M}$ is a Σ_d^{p} machine, by a standard interpretation (see [FSS81]) of the Σ_d^{p} query computation, we may regard $\mathcal{M}(x; s(n))$ as a depth $d+1$ circuit on input variables $z_1, \ldots, z_L$, of size at most $m2^m$ and bottom fan-in at most m.

Our first step is to assign a random restriction ρ to $z_1, \ldots, z_L$ of an appropriate probability $p_0 = \Pr[\,z_i = *\,]$. By Lemma 1, with high probability the circuit is reduced to small depth decision trees with depth $t = 2m$. In fact, by choosing p_0 appropriately, we can even show that with high probability, a random restriction converts *all* circuits for all 2^n input strings to depth t decision trees.

Then these small depth decision trees can be expressed by low degree (i.e., degree $2m$) polynomials with integer coefficients. That is, after the random restriction, each computation $\mathcal{M}(x; s(n))$ is expressed as a degree $2m$ polynomial p_x. We have arrived at a similar situation to the parity computation. We will use a similar technique to attack this. However the exact approach in the $\oplus$P case does not work.

In the $\oplus$P case the function encoded is essentially the AND function $\bigwedge z_j$. This will not survive the random restriction. Instead we will try to encode the parity on a suitable subset, one for each x. Our encoding is implemented as follows. For each $x \in \{0,1\}^n$, we define a segment $S_x \subset \{0,1\}^I$ of enough size, roughly speaking, $20m/p_0$, which is polynomial in n. These segments are chosen so that the family $\{S_x\}_{x\in\{0,1\}^n}$ is pair-wise disjoint. As in the proof of previous section, we would like to use the assignment of variables in S_x to encode the result of $\mathcal{M}(x; s(n))$. Here notice that the random restriction ρ has already assigned values to some of the variables in S_x. But since (i) $|S_x| = 20m/p_0$, and (ii) variables remain unassigned with probability p_0, we can prove that with high probability, *all* segments S_x have at least $3m$ unassigned variables after the random restriction. We use these unassigned variables for encoding.

Thus, there exists a random restriction satisfying the following.

(a) Each computation $\mathcal{M}(x; s(n))$ is reduced to a decision tree T_x of depth at most $2m$.

(b) Each segment S_x has at least $3m$ unassigned variables, i.e., assigned $*$ by the restriction.

Fix ρ_0 to be one such restriction. Denote by Z_0 the set of variables in $\bigcup_{x\in\{0,1\}^n} S_x$ that are assigned $*$ by ρ_0, and rename variables so that $Z_0 = \{z_1, \ldots, z_M\}$ and $Z - Z_0 = \{z_{M+1}, \ldots, z_L\}$.

The restriction ρ_0 may assign $*$ to some variables in $Z - Z_0$, we now assign all such variables to 0. Then as explained above, the result of each computation of $\mathcal{M}(x; s(n))$ is expressed as a degree $2m$ polynomial $p_x(z_1, \ldots, z_M)$ over the integers $\mathbf{Z}$. For each x, we try to equate $p_x(z_1, \ldots, z_M)$ to the parity of S_x, i.e., $\oplus_{z_i\in S_x} z_i$. (Note that S_x contains variables not in Z_0 $= \{z_1, \ldots, z_M\}$ whose values are already fixed. By the term $\oplus_{z_i\in S_x} z_i$, we mean the parity of all variables in S_x including such variables.) In other words, we wish to choose an assignment to $z_1, \ldots, z_M$ so that the following system of equations $(*2)$ holds for $\{0,1\}^n = \{x_1, \ldots, x_N\}$, where $N = 2^n$.

$$(*2)\ \left\{\begin{array}{lcl} p_{x_1}(z_1, \ldots, z_M) & = & \oplus_{z_j\in S_{x_1}} z_j, \\ & \vdots & \\ p_{x_N}(z_1, \ldots, z_M) & = & \oplus_{z_j\in S_{x_N}} z_j. \end{array}\right.$$

Using a trick of exchanging 0,1 values by $1, -1$ values, and reason about dimensions over a finite field $\mathbf{Z}_3$, we can give an argument similar to the one in the previous section, and show that it is indeed possible to find such an assignment. Then the result follows.

Now we specify the parameters and the conditions explained above, and describe our proof precisely.

We focus on the simulation of some Σ_d^{p} machine $\mathcal{M}(x; s(n))$ on N $(= 2^n)$ inputs of length n for sufficiently large n. Let $m = O(n^k)$ be an integer bounding $\mathcal{M}$'s running time on length n inputs, and let $I = (1 + \delta)n$ and $L = 2^I$. We regard the computation of $\mathcal{M}(x; s(n))$ as a function over Boolean variables $z_1, \ldots, z_L$, where each z_i is the boolean variable for a bit in $s(n)$. Furthermore, we may consider $\mathcal{M}(x; s(n))$ as a circuit C_x of depth $\le d + 1$, size $\le m2^m$, and bottom fan-in $\le m$.

As explained above, we consider a random restriction to the variables $z_1, \ldots, z_L$, with $p_0 = 1/(20m)^d$ being the probability $\Pr[\, z_i = *\,]$. For each $x \in \{0,1\}^n$, the segment S_x is defined by $S_x = \{\, xu0^{\ell-n-n_0} \,:\, u \in \{0,1\}^{n_0}\,\}$, where $n_0 = \lceil \log_2 20m/p_0 \rceil = \lceil (d+1) \log_2 20m \rceil$. Clearly, any S_x and $S_{x'}$, for $x \ne x'$, are disjoint, and $|S_x|$ is of size larger than $20m/p_0$ but still polynomial in n.

We want some random restriction ρ, such that it satisfies the following two conditions.

(a) For every $x \in \{0,1\}^n$, the circuit C_x is reduced to a depth $t = 2m$ decision tree.

(b) For every $x \in \{0,1\}^n$, the segment S_x has at least $3m$ unassigned variables.

By using Lemma 1 and Chernoff's bound (see, e.g., Corollary A.1.14 of [AS00]), it is easy to show the following claim:

Claim 1. Under our choice of parameters, the probability that a random restriction ρ satisfies both (a) and (b) is not zero.

Hence, there exists some random restriction satisfying both (a) and (b).

Consider one of the restrictions ρ_0 satisfying both (a) and (b). We define $s(n)$ based on this ρ_0; that is, we will assign a bit in $s(n)$ to 0 or 1 according to ρ_0. We will assign those variable assigned $*$ by ρ_0 later. Let Z_* be the set of variables assigned $*$ by ρ_0. From condition (b) it follows that each S_x has at least $3m$ variables in Z_*. For each S_x, we pick lexicographically the first $3m$ such variables, and define Z_0 to be the set of those variables, over all x. Note that Z_0 has exactly $3mN$ variables because all S_x's are disjoint. By renaming variables, we assume that $Z_0 = \{z_1, \ldots, z_M\}$, where $M = 3mN$. We assign 0 to all variables in $Z_* - Z_0$; thus, Z_0 is the set of remaining unassigned variables.

From condition (a), the computation $\mathcal{M}(x; s(n))$ for each $x \in \{0,1\}^n$ is represented as a depth $2m$ decision tree T_x on $z_1, \ldots, z_M$. Then we can express T_x as a low degree polynomial p_x in the following way. For the trivial decision tree of depth 0 (where no variable is accessed at all), the value is a constant 0 or 1. Inductively, suppose in the decision tree T, the first branch is on the variable z_i, and depending on its value, its left subtree is T_0 for $z_i = 0$, and its right subtree is T_1 for $z_i = 1$. Then we see immediately that the polynomial $p = (1 - z_i)p_0 + z_i p_1$ evaluates to the truth value of T, where p_0 and p_1 are the polynomials that correspond to the subtrees T_0 and T_1 respectively. In this way, we can define the polynomial p_x computing the value of T_x. Note here that the degree of p is at most $1 + \max\{\deg p_0, \deg p_1\}$. In particular, we have $\deg p_x \le 2m$ for each decision tree T_x.

For these polynomials p_x, $x \in \{0,1\}^n$, we show below that there exists an 0,1-assignment to variables in Z_0 satisfying ($*2$) above. We complete ρ_0 by using one of such assignments, and define $s(n)$ accordingly. Then one can compute the value of $\mathcal{M}(x; s(n))$, for each $x \in \{0,1\}^n$, by asking queries on all the bits indexed in S_x and taking the parity of the answers. Since the size of S_x is polynomially bounded in n, this is a P computation with random access to $s(n)$.

Now the remaining task is to prove that ($*2$) has a solution. Let us first transform ($*2$) to a system of equations in $\mathbf{Z}_3$. Note that the polynomials p_x, though defined over the integers $\mathbf{Z}$, only evaluate to the values 0 or 1 when each z_i takes either 0 or 1. This fact is verified inductively by looking at the above decomposition $p = (1 - z_i)p_0 + z_i p_1$. Furthermore, this property is invariant even if the polynomials are evaluated modulo q, for any prime q. Thus, we may argue these polynomials under the modulo q computation, for any prime q. In particular, we consider the polynomials under the modulo 3 computation, i.e., over the finite field $\mathbf{Z}_3$.

Then by a linear transformation, we can change the representation of 0 and 1 by $+1$ and -1 respectively; that is, 0 is represented by $+1$ and 1 by -1. More specifically, for each polynomial p_x, we replace z_i by $z'_i = 1 + z_i$, and express $p'_x = 1 + p_x$ as polynomials in z'_i's. Note that when $z_i = 0$ and 1 respectively, $z'_i = 1$ and -1 respectively, and similarly for p_x and p'_x. On the other hand, the parity is now expressed by simply a product. (In the following we will rewrite z_i for z'_i and p_x for p'_x.) Thus, the system of equations ($*2$) is transformed into the following system of equations in $\mathbf{Z}_3$.

$$(*3)\quad \left\{ \begin{array}{lllll} p_{x_1}(z_1, \ldots, z_M) & = & \prod_{z_j \in S_{x_1}} z_j & = & \alpha_{x_1} \cdot \prod_{z_j \in Z_0 \cap S_{x_1}} z_j \\ & & \vdots & & \\ p_{x_N}(z_1, \ldots, z_M) & = & \prod_{z_j \in S_{x_N}} z_j & = & \alpha_{x_N} \cdot \prod_{z_j \in Z_0 \cap S_{x_N}} z_j. \end{array} \right.$$

Where each $\alpha_x \in \{-1, +1\}$ denotes the product of all determinate variables $z_i \in S_x - Z_0$.

We claim that there is at least one assignment to ± 1 for all $z_i \in Z_0$ satisfying $(*3)$. Suppose, for a contradiction, that there is no such assignment. Then, since for every ± 1 values of $z_1, \ldots, z_M$, each p_x takes a ± 1 value, it follows that for every $+1, -1$-assignment $(a_1, \ldots, a_M)$ to the z_i's, there must be at least one x such that

$$p_x(a_1, \ldots, a_M) \; = \; -\alpha_x \cdot \prod_{z_i \in Z_0 \cap S_x} a_i.$$

Thus, we have

$$\prod_{1 \le i \le N} \left[\alpha_{x_i} \prod_{z_j \in Z_0 \cap S_{x_i}} z_j + p_{x_i}(z_1, \ldots, z_M) \right] \; = \; 0,$$

for all $+1, -1$-assignments to $z_1, \ldots, z_M$. Then it follows that the lefthand side expression is identical to 0 modulo the ideal $I = (z^2 - 1 : z \in Z_0)$. In other words, we have the identity

$$\prod_{1 \le i \le N} \prod_{z_j \in Z_0 \cap S_{x_i}} z_j \; = \; L(z_1, \ldots, z_M)$$

in the ring $\mathbf{Z}_3[z_1, \ldots, z_M]/I$, where L is a multilinear polynomial of degree at most $3m(N-1) + 2m$. On the other hand, the lefthand side is multilinear and its degree is $3mN$, which is larger than $3m(N-1) + 2m$. A contradiction.

This completes the proof of Theorem 3, and hence Theorem 2. With some more work one can show

Theorem 4. For any prime p, and for any length bound $\ell(n) \ge 2^{(1+\delta)n}$, where $\delta > 0$ is any positive constant, there exists an advice function s of advice size $\ell(n)$ such that $\mathrm{Mod}_p^{\mathrm{PH}}/s \; = \; \mathrm{P}/s$.

5 Relations to the Conventional Relativized Results

Our model of random access to advice can be viewed as a restricted type of relativization. Here we explain the position of our results and proofs in the context of conventional relativization results.

First it should be noted that most relativized separation results are proved in a stringent way; that is, the proofs of such results can be modified easily for proving the same separation w.r.t. some advice function of some exponential (or super-polynomial) advice size. Most typically we can prove the following relation.

Proposition 5. For any super-polynomial length bound $\ell(n)$, there exists an advice function s of advice size $\ell(n)$ such that $\mathrm{NP}/s \; \not\subseteq \; \mathrm{P}/s$.

Since our nonuniform notion is a generalization of the standard nonuniform model by Karp and Lipton, there are immediate implications for our nonuniform comparison from some of the results for the standard nonuniform model. For example, it has been known [Kan82] that $\mathrm{PH} \not\subseteq \mathrm{P}/p(n)$ for any fixed polynomial $p(n)$. Since $\mathrm{PH} \subseteq \mathrm{PH}/s$ for any advice function s, the following fact is immediate from this result. This fact justifies the consideration of at least super-polynomial advice size for obtaining a nonuniform collapsing result for P and PH.

Proposition 6. For any polynomially bounded advice $\ell(n)$, there is no advice function s of advice size $\ell(n)$ for which $\mathrm{PH}/s \subseteq \mathrm{P}/s$.

While most relativized collapsing results are proved in a non stringent way, there are some relativized collapsing proofs in the literature that also yield nonuniform collapsing results in our context. For example, the following result is provable by a well-known technique.

Proposition 7. For any length bound $\ell(n) \geq 2^{(2+\delta)n}$, for any positive constant $\delta > 0$, there exists an advice function s of advice size $\ell(n)$ such that $\mathrm{NP}/s \subseteq (\mathrm{P/poly})/s$.

We omit the proof here.

A similar argument in fact proves NEXP $\subseteq$ P/poly in the standard relativization model [He86]. This is because for any given NEXP machine $\mathcal{M}$ running in time $2^{p(n)}$, we can consider query strings of length $3p(n)$; since $2^{n+p(n)} < 2^{2p(n)}$, we still have enough room in $\{0,1\}^{3p(n)}$ to encode the results of $\mathcal{M}$ on all length n inputs. Some circuit of size $cp(n)$ for some sufficiently large $c > 0$ can retrieve this encoded information. On the other hand, this argument does not work in our context because the advice size cannot be bounded even exponentially. It should be also remarked here that a higher collapse is not immediate from a lower one in our context; for example, the relatively simple proof of $\mathrm{NP}/s \subseteq (\mathrm{P/poly})/s$ for some advice s of some exponential advice size bound does not give a proof of $\mathrm{PH}/s' \subseteq (\mathrm{P/poly})/s'$ for some s' of some exponential advice size bound. This latter result is indeed true, first proved by the authors using complicated arguments based on Nisan-Wigderson pseudorandom generators. The results of the present paper give a simplified proof of a stronger result.

We believe that this model of random access to advice strings is an interesting model, which poses challenging problems. It is sufficiently different from the conventional relativization model for specific problems. Previous known proofs of relativized collapsing results do not, in general, imply the corresponding collapsing results in this model of random access to advice. Claims to the contrary should be first verified against the open problem of PSPACE vs. P.

References

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

[AS00] N. Alon and J. Spencer, *The Probabilistic Method*, John Wiley & Sons, Inc., 2000.

[BGS75] T. Baker, J. Gill, and R. Solovay, Relativizatons 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.*, 21–29, 1986. (The final version appeared in *J. Comp. Syst. Sci.* 38(1), 68–85, 1989.)

[DK00] D. Du and K. Ko, *Theory of Computational Complexity*, John Wiley & Sons, Inc., 2000.

[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.

[He86] H. Heller, On relativized exponential and probabilistic complexity classes, *Information and Control* 71(3), 231–243, 1986.

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

[KL80] R. Karp and R. Lipton, Some connections between nonuniform and uniform complexity classes, in *Proc. 12th ACM Symposium on Theory of Computing* (STOC'80), ACM, 302–309, 1980. (An extended version appeared as: Turing machines that take advice, in *L'Enseignement Mathématique* (2nd series) 28, 191–209, 1982.)

[Ko78] D. Kozen, Indexing of subrecursive classes, in *Proc. 10th ACM Symposium on Theory of Computing* (STOC'78), ACM, 287–295, 1978. (The final version appeared in *Theoretical Computer Science* 11, 277–301, 1980.)

[Nis91] N. Nisan, Pseudorandom bits for constant depth circuits, *Combinatorica* 11(1), 63–70, 1991.

[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.

[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.