On Proving Circuit Lower Bounds Against PH and Some Related Lower Bounds for Constant Depth Circuits

Jin-Yi Cai [1] Osamu Watanabe [2]

[1]Computer Sciences Department, University of Wisconsin, Madison, WI 53706, USA. Research supported in part by NSF grants CCR-9634665. Email: jyc@cs.wisc.edu

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

Abstract

We consider the problem of proving circuit lower bounds against the polynomial-time hierarchy. We give both positive and negative results. For the positive side, for any fixed integer $k > 0$, we give an explicit Σ_2^p language, acceptable by a Σ_2^p-machine with running time $O(n^{k^2+k})$, that requires circuit size $> n^k$. This provides a constructive version of an existence theorem of Kannan [Kan82]. Our main theorem is on the negative side. We give evidence that it is infeasible to give relativizable proofs that any single language in the polynomial-time hierarchy requires super polynomial circuit size. Our proof techniques are based on the decision tree version of the Switching Lemma for constant depth circuits and Nisan-Wigderson pseudorandom generator. We also take this opportunity to publish some unpublshed older results of the first author on constant depth circuits, both straight lower bounds and inapproximability results based on decision tree type Switching Lemmas.

1 Introduction

It is a most basic open problem in Theoretical Computer Science to give circuit lower bounds for various complexity classes. The class P has polynomial size circuits. It is also widely believed that NP does not share this property, i.e., that some specific set such as SAT in NP requires super polynomial circuit size. While this remains the most concrete approach to the NP vs. P problem, we can't even prove this for any fixed $k > 1$ that any set $L \in \mathrm{NP}$ requires circuit size $> n^k$.

If we relax the restriction from NP to the second level of the Polynomial-time Hierarchy Σ_2^{p}, R. Kannan [Kan82] did prove that for any fixed polynomial n^k, there is some set L in Σ_2^{p} which requires circuit size $> n^k$. Kannan in fact proved the existence theorem for some set in $\Sigma_2^{\mathrm{p}} \cap \Pi_2^{\mathrm{p}}$. This result has been improved by Köbler and Watanabe [KW98] who showed, based on the technique developed in [BCGKT], that such a set exists in $\mathrm{ZPP}^{\mathrm{NP}}$. More recently, the work in [Cai01] implies that a yet lower class $\mathrm{S}_2^{\mathrm{p}}$ contains such a set. (See [BFT98, MVW99] for related topics.)

However, Kannan's proof for Σ_2^{p}, and all the subsequent improvements mentioned above, are not "constructive" in the sense that it does not identify a single Σ_2^{p} machine whose language requires circuit size $> n^k$. At the top level, all these proofs are of the following type: *Either* SAT does not have n^k size circuit, in which case we are done, *or* SAT has n^k size circuit, then we can define some other set, which by the existence of the hypothetical circuit for SAT can be shown in Σ_2^{p}, and it requires circuit size $> n^k$. Constructively, Kannan gave a set in $\Sigma_4^{\mathrm{p}} \cap \Pi_4^{\mathrm{p}}$. In [MVW99] a set in Δ_3^{p} was constructively given. We improve this to Σ_2^{p}.

Theorem 1 *For any integer $k > 0$, we can construct a Σ_2^{p}-machine with $O(n^{k^2} \log^{k+1} n)$ running time that accepts a set with no n^k size circuits.*

A similar constructive proof is still open for the stronger statements, i.e., the existence of a set with n^k circuit size lower bound in $\Sigma_2^{\mathrm{p}} \cap \Pi_2^{\mathrm{p}}$ (resp., $\mathrm{ZPP}^{\mathrm{NP}}$, and $\mathrm{S}_2^{\mathrm{p}}$).

Our main result in this paper deals with the difficulty in proving super polynomial circuit size lower bound for any set in the Polynomial-time Hierarchy, PH. While it is possible to prove lower bound above any fixed polynomial, at least for some sets in Σ_2^{p}, the real challenge is to prove super polynomial circuit size lower bound for a single language. Not only have we not been able to do this for any set in NP, but also no super polynomial lower bound is known for any set in PH. In this paper we prove that it is infeasible to give relativizable super polynomial lower bound for any set in the Polynomial-time Hierarchy.

For our relativized argument, we propose a new computation model that gives us more "stringent" relativized results. Relativization results can be generally classified as either separation or collapsing/containment results. The implication of a relativized separation result is that an unconditional collapse is unlikely to be proved. Similarly a relativized collapsing result implies that an unconditional separation is hard to prove. We deal with relativized collapsing results here since we are interested in demonstrating the difficulty of proving unconditional circuit lower bound for PH.

By surveying existing relativized collapsing results, we came to realize an asymmetry often present. In almost all of these relativized collapsing results the proof is achieved by allowing stronger access of oracles to the simulating computation than the simulated computation. For example, in the usual proof of $\mathrm{P}^A = \mathrm{NP}^A$ or $\mathrm{P}^A = \mathrm{PSPACE}^A$, we encode QBF in the oracle. In terms of the simulation by the $\mathrm{P}^{\mathrm{QBF}}$ machine $\mathcal{M}$ simulating an $\mathrm{NP}^{\mathrm{QBF}}$ or $\mathrm{PSPACE}^{\mathrm{QBF}}$ 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, P^A machines are given more powerful oracle access. 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. In our present study of specific polynomial bounds such as n^k of either circuit size or running time, this arbitrary polynomial stretch in oracle access is not acceptable.

The following example will make this point clear. Hopcroft, Paul, and Valiant [HPV77] proved that any machine with time complexity t can be simulated by some machine with space complexity $s = \frac{t}{\log t}$ that is slightly less than t. In this situation where we deal with specific bounds like t, to allow the simulating machine oracle access beyond what's allowed for the simulated machine will cause unacceptable consequences. We can argue as follows: For any time complexity t, let $s = \frac{t}{\log t}$ be the space complexity in the simulation. Take an intermediate bound s' that is slightly more than s but slightly less than t (all asymptotically). Then we can construct an oracle X such that $\mathrm{DSPACE}^X(s') \subseteq \mathrm{DTIME}^X(t)$ holds, with the device of "unequal access". The idea is to encode all $\mathrm{DSPACE}^X(s')$ languages (up to a given length n) in X at length $t(n)$. Note that any s'-space machine have access only up to length $s'(n) < t(n)$, the encoding part is beyond where the s'-space machine can look. On the other hand, if the Hopcroft-Paul-Valiant simulation were to be relativizable with this "unequal access" to the oracles, then we have $\mathrm{DTIME}^X(t) \subseteq \mathrm{DSPACE}^X(s)$. Thus, $\mathrm{DSPACE}^X(s') \subseteq \mathrm{DTIME}^X(t) \subseteq \mathrm{DSPACE}^X(s)$, violating the relativization of space hierarchy theorem.

We study in this paper some specific circuit size lower bound w.r.t. some specific, say n^k, time bound. Then, as the above observation suggests, we must adopt the following more "stringent" oracle computation model. In this more "stringent" oracle access model, we require that, for any n, the simulating machine or circuit does not access the oracle beyond where the simulated machine or circuit can access, on any input of length n. We consider circuits consisting of standard AND, OR, and NOT gates and oracle query gates. An oracle query gate takes m input bits $z = b_1 b_2 \dots b_m$, and has output $\chi_{[z \in X]}$, i.e., it outputs 1 or 0 depending on whether $z \in X$ or otherwise. The proviso of "stringent access to an oracle" is as follows: To show that, at length n, a circuit C_n computes the language of machine $\mathcal{M}$ with running time n^k, we allow the circuit only access to those strings that can be accessed by the simulated machine $\mathcal{M}$, namely all strings of length at most n^k.

In general, we call a relativized (collapsing) result *stringent* if it is proved under an oracle computation model with this stringent oracle access. While this model is absolutely needed for specific time/size bound such as n^k, we believe that in fact the model is meaningful in a more general setting. (For more general situations, it might be better to consider a more robust notion of "stringent relativization". We leave this issue and more general investigations of "stringent relativization" for our future work.)

From a more general perspective, the main utility of relativization is to show the inadequacy of certain proof techniques. Certainly the more "stringent" a requirement we place on the type of relativization, the stronger the result will be, and perhaps it says more about the infeasibility of certain techniques. Imagine there are three possible claims of proving a certain lower bound, such as Σ_2^{p} requires superpolynomial circuit size:

1. A proof totally specific to a specific set in Σ_2^{p}, and uses specific properties of the circuit combinatorics.
2. A proof for a general Σ_2^{p} machine and uses properties of the circuit, but the proof can be carried through if we allowed the machine and circuit to access any but the same segment of any oracle.
3. A proof more general, for a general Σ_2^{p} machine and circuit, and the proof can go through even if we allowed the machine and circuit to access different segments of the oracle.

Any relativization to the contrary says nothing about the first possibility. A relativization to the contrary with stringent access model rules out possibility 2 and 3. If we did not have the "stringent access requirement", then we can only rule out 3, but not 2.

In this paper we focus on specific time/size bound. In the stringent oracle access model, we prove that for any alternating oracle TM $\mathcal{M}$ with running time $O(n^k)$, there is an oracle X and a polynomial size circuit family accepting it. Therefore we rule out possibility 2 and 3 above.

Theorem 2 (Main Theorem) *For any integer $d > 0$ and any real $k > 1$, let $\mathcal{M}$ be an oracle Σ_d^{p}-machine with running time $O(n^k)$. Then we have an oracle X and a family of Boolean circuits $\{C_n\}_{n\geq 0}$ that accepts $L(\mathcal{M}^X)$ under a stringent access to the oracle X. For all sufficiently large n, the size of C_n is bounded by n^{cdk}, for some universal constant $c > 0$.*

This result shows that for proving n^k circuit size lower bound in Σ_d^{p} by a relativizable technique, we need at least $\Omega(n^{k/cd})$ running time. In particular, it is infeasible to give a relativizable proof of super polynomial circuit size lower bound for any set in PH.

Our proof technique for the main theorem is based on the decision tree version of the Switching Lemma for constant depth circuits and Nisan-Wigderson pseudorandom generator.

As these results crucially depend on lower bounds of constant depth circuits, we take this opportunity to publish some unpublshed older results of the first author on constant depth circuits, as it would fit the theme. These include both straight lower bounds and inapproximability results based on decision tree type Switching Lemmas. We give some better constants in the exponents than previously published lower bounds.

2 Proof of Theorem 1

R. Kannan [Kan82] proved that for any fixed polynomial n^k, there is some set L in $\Sigma_2^{\mathrm{p}} \cap \Pi_2^{\mathrm{p}}$ with circuit size $> n^k$. However, in terms of explicit construction, he only gave a set in $\Sigma_4^{\mathrm{p}} \cap \Pi_4^{\mathrm{p}}$. An improvement to Δ_3^{p} was stated in [MVW99].

In this section we give a constructive proof of Kannan's theorem for Σ_2^{p}.

For any $n \geq 0$, a binary sequence χ of length $l \leq 2^n$ is called a *partial characteristic sequence*, which will specify the membership of lexicographically the first l strings of $\{0,1\}^n$. We denote this subset of $\{0,1\}^n$ by $L(\chi)$. We say that χ is consistent with a circuit C with n input gates, iff $\forall i$, $1 \leq i \leq l$, $C(x_i)$ outputs the ith bit of χ, where x_i is the ith string of $\{0,1\}^n$.

We can encode every circuit C of size $\leq s$ as a string u of length $\mathrm{len}(s)$, where $\mathrm{len}(s)$ is defined as $\mathrm{len}(s) = c_{\mathrm{circ}}\lfloor s \log s \rfloor$ with some constant c_{circ}. We may consider every u with $|u| = \mathrm{len}(s)$ encodes some circuit of size $\leq s$; if a string u is not a proper code or the encoded circuit has size $> s$, we assume that this u encodes the constant 0 circuit. The following lemma is immediate by counting.

Lemma 3 *For any $s > 1$, there exists a partial characteristic sequence of length $l = \mathrm{len}(s) + 1$ that is not consistent with any circuit of size $\leq s$.*

Our goal is to define a set L that has no n^k size circuit but that is recognized by some explicitly defined Σ_2^{p} machine. Our construction follows essentially the same outline as the one given in [MVW99], which in turn uses ideas given in Kannan's original proof. The further improvement is mainly an ever more efficient use of alternation.

For a given n, denote by $\ell = \mathrm{len}(n^k) + 1$, and we try to construct a partial characteristic sequence χ of length ℓ that is consistent with *no* circuit of size $\leq n^k$. We will introduce an auxiliary set PreCIRC that is in NP. With this PreCIRC, some Σ_2^{p} machine can *uniquely* determine the desired characteristic sequence χ_{non} (on its accepting path). We would like to define our set L (partially) consistent with this sequence χ_{non}. But Σ_2^{p} computation using some auxiliary NP set cannot be implemented, in general, by any Σ_2^{p} machine. Suppose here that PreCIRC has n^k size circuits; then some Σ_2^{p} machine can guess such circuits, verify them, and use them for computing χ_{non} and recognizing strings according to χ_{non}. What if there are no such circuits for PreCIRC? We will define L so that one part of L is consistent with PreCIRC (while the other part is consistent with χ_{non} if PreCIRC is computable by some n^k size circuits). If PreCIRC has no n^k size circuit, then the part of L that is consistent with PreCIRC can guarantee the desired hardness of L.

Now we describe our construction in detail. Denote by $\ell = \mathrm{len}(n^k) + 1$. By "$v \succ u$" we mean that u is a prefix of v. To compute the "hard" characteristic sequence χ_{non}, we want to determine, for a given pair of a partial characteristic sequence χ and a string u, whether u can be extended to some description v of a circuit that is consistent with χ. The set PreCIRC is defined for this task. More precisely, for any $n > 0$, and for any strings χ of length ℓ and u of length $\leq \mathrm{len}(n^k)$, we define PreCIRC as follows.

$$\begin{aligned}&1^n 0 \chi u 0 1^{\mathrm{len}(n^k)-|u|} \in \mathrm{PreCIRC}\\ &\quad\Leftrightarrow\ (\exists v \succ u)\ [\ |v| = \mathrm{len}(n^k), \text{ and the circuit encoded by } v \text{ is consistent with } \chi\ ].\end{aligned}$$

Strings of any other form are not contained in PreCIRC. For simplifying our notation, we will simply write (χ, u) for $1^n 0 \chi u 0 1^{\mathrm{len}(n^k)-|u|}$. Since n determines ℓ, and the length of χ is ℓ, χ and u are uniquely determined from $0^n 1 \chi u 1 0^{\mathrm{len}(n^k)-|u|}$. The length of (χ, u) is $\widetilde{n} = n + 2\ell + 1$. Note that $\widetilde{n}$ is $O(n^k \log n)$.

We now define our machine $\mathcal{M}$. Informally we want $\mathcal{M}$ to accept an input x if and only if either $x \in 1\{0,1\}^{n-1}$ and $x \in$ PreCIRC, or $x \in 0\{0,1\}^{n-1}$ and $x \in L$, where $L^{=n}$ is a set with no n^k size circuits, for all sufficiently large n, *if* $\mathrm{PreCIRC}^{=n}$ has n^k size circuits for all sufficiently large n. Specifically, $\mathcal{M}$ is designed so that $L^{=n}$ would be $L(\chi_{\mathrm{non}})$ where χ_{non} is lexicographically the first χ of length ℓ with no n^k size circuit, *provided* $\mathrm{PreCIRC}^{=\widetilde{n}}$ has a $\widetilde{n}^k$ size circuit for length $\widetilde{n}$.

More formally, for any given input x of length n, if x starts with 1, then $\mathcal{M}$ accepts it iff $x \in$ PreCIRC. Suppose otherwise; that is, x starts with 0. Then first $\mathcal{M}$ existentially guesses a partial characteristic sequence χ_{non} of length ℓ *and* a circuit C_{pre} of size $\widetilde{n}^k$, more precisely, a string v_{pre} of length $\text{len}(\widetilde{n}^k)$ encoding a circuit for $\text{PreCIRC}^{=\tilde{n}}$ of size $\leq \widetilde{n}^k$. (Below we use C_{pre} to denote the circuit that is encoded by the guessed v_{pre}.) After that, $\mathcal{M}$ enters the universal stage, where it checks the following items.

(1) $\forall \chi$, $|\chi| = \ell$, and $\forall u$, $|u| \leq \text{len}(n^k)$, check that C_{pre} is "locally consistent" on (χ, u) as follows:

$$\begin{array}{lcl} C_{\text{pre}}(\chi, u) = 1 \ \&\ |u| = \text{len}(n^k) & \Longrightarrow & \text{the circuit that } u \text{ encodes is consistent with } \chi, \text{ and} \\ C_{\text{pre}}(\chi, u) = 1 \ \&\ |u| < \text{len}(n^k) & \Longrightarrow & \text{either } C_{\text{pre}}(\chi, u0) = 1 \text{ or } C_{\text{pre}}(\chi, u1) = 1. \end{array}$$

(2) $\forall u$, $|u| = \text{len}(n^k)$, compute the χ_u of length ℓ defined by (the circuit encoded by) u, and verify that C_{pre} works for χ_u and all prefix u' of u, i.e., $C_{\text{pre}}(\chi_u, u') = 1$.

(3) The guessed χ_{non} is lexicographically the first string of length ℓ such that no circuit of size s is consistent with it, *according to* C_{pre}. That is, check $C_{\text{pre}}(\chi_{\text{non}}, \epsilon) = 0$, where ϵ is the empty string, and $\forall \chi$ if $|\chi| = \ell$ and χ is lexicographically smaller than χ_{non} then $C_{\text{pre}}(\chi, \epsilon) = 1$ holds.

Finally on each universal branch, if $\mathcal{M}$ passes the particular test of this branch, then $\mathcal{M}$ accepts the input $x \in 0\{0,1\}^{n-1}$ iff χ_{non} has bit 1 for the string x.

For all (χ, u), such that $|\chi| = \ell$ and $|u| \leq \text{len}(n^k)$, if C_{pre} passes (1), then

$$C_{\text{pre}}(\chi, u) = 1 \implies (\chi, u) \in \text{PreCIRC},$$

and if C_{pre} passes (2), then

$$(\chi, u) \in \text{PreCIRC} \implies C_{\text{pre}}(\chi, u) = 1.$$

Of course there is no guarantee that there exists a circuit C_{pre} (more precisely, v_{pre}) that will pass the tests in items (1) and (2). But if there is such a C_{pre}, then some existential path leads to such a C_{pre} together with the right χ_{non}. this χ_{non} is *the* lexicographically the first string of length ℓ such that no circuit of size n^k is consistent with it, which exists by Lemma 3. In particular it is independent of the particular C_{pre} guessed. Hence, if there is a circuit for $\text{PreCIRC}^{=\widetilde{n}}$ of size $\widetilde{n}^k$, then $\mathcal{M}$ accepts $x \in 0\{0,1\}^{n-1}$ iff x is in $L(\chi_{\text{non}})$, where χ_{non} is the unique lexicographically the first string of length ℓ with no consistent circuit of size $\leq n^k$.

On the other hand, if no circuit of size $\widetilde{n}^k$ can accept $\text{PreCIRC}^{=\widetilde{n}}$ correctly, then no circuit passes the tests in items (1) and (2), and hence, $\mathcal{M}$ simply rejects all $x \in 0\{0,1\}^{n-1}$. But since $\text{PreCIRC}^{=\widetilde{n}}$ has no $\widetilde{n}^k$ size circuit, the hardness is guaranteed by the PreCIRC part of $L(\mathcal{M})$. More formally, if $\text{PreCIRC}^{=n}$ has no circuit of size n^k *infinitely often*, then we are done. Otherwise, for all sufficiently large n, and hence for all sufficiently large $\widetilde{n}$, a circuit C_{pre} exists for $\text{PreCIRC}^{=\widetilde{n}}$ of size $\widetilde{n}^k$, we are also done. Therefore, we can conclude that $L(\mathcal{M})$ has no n^k size circuit (which by definition means that for infinitely many n this is so.) This Σ_2^{p} language proves Theorem 1. It can be easily checked that the machine $\mathcal{M}$ runs in $O(n^{k^2} \log^{k+1} n)$ steps.

3 Proof of Theorem 2

We first give an outline of the proof.

3.1. Proof Outline

Consider any Σ_d^{p} polynomial time bounded oracle alternating Turing machine $\mathcal{M}$, with time bound n^k. We want to design an oracle X so that some family of small size circuits can simulate $\mathcal{M}^X$ with stringent oracle access. More specifically, fix any sufficiently large n, we want a circuit $C_{\mathcal{M}}$ that simulates $\mathcal{M}^X$ on inputs of length n, where, since $\mathcal{M}$ can only query strings of length at most n^k, we require that $C_{\mathcal{M}}$ can also only ask queries of length at most n^k.

It is well known from [FSS81] that a Σ_d^{p} machine $\mathcal{M}$ bounded in time n^k with oracle X, when given an input x of length n, gives rise to a bounded depth Boolean circuit C_x of the following type: The inputs are Boolean variables, and their negations, representing membership of a string $z \in \{0,1\}^{\leq n^k}$ in the oracle X. The Boolean circuit C_x starts with an OR gate at the top, and alternate with AND's and OR's with depth $d+1$, where the bottom level gates have bounded fan-in at most n^k, and all other AND and OR gates are unbounded fan-in, except by the overall circuit size, which is bounded by $n^k 2^{n^k}$. Without loss of generality we may assume the Boolean circuit is tree like, except for the input level, where each Boolean variable corresponding to $\chi_{[z \in X]}$ is represented by a pair of complemented variables, which we will denote by z and $\overline{z}$.

Our first idea is to use random restrictions to "kill" the circuit. Here is what we mean. For any Boolean function f over variables $x_1, \ldots, x_n$, a *random restriction* ρ (for some specified parameter p) is a random function that assigns each x_i either 0, 1, or $*$, with probability $\Pr[\rho(x_i) = *] = p$ 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. It is known that after a random restriction ρ (for a suitably chosen parameter p), the circuit is sufficiently weakened so as to have either small min-terms or small max-terms. Results of this type are generally known as Switching Lemmas, and the strongest form known is due to Håstad [Hås86a]. (See also [Ajt83, FSS81, Yao85, Cai86, Hås86b]). However it turns out that we need a different form, namely a decision tree type Switching Lemma. We want to assign a suitably chosen random restriction ρ, after which the circuit admits a small depth decision tree. We in fact will have to consider an aggregate of 2^n such Boolean circuits C_x simultaneously, one each for an input x of size n. We want to assign ρ, after which all these circuits have small depth decision trees. We then will proceed to set those variables to ensure that all these circuits are "killed", i.e., they all have a definite value now, either 0 or 1. We need to assign those variables consistently over all 2^n small depth decision trees. For decision trees, it is easy to achieve this by always setting "the next variable" asked by the decision tree to 0, say; it is not clear how to maintain this consistency in terms of min-terms and max-terms. If each decision tree has depth bounded by t, then we will have assigned at most $2^n t$ many variables corresponding to those strings of length n^k where ρ initially assigned a $*$ (i.e., they are left unassigned by ρ). We will argue that there are still plenty of unassigned variables left, where we may try to encode the now-determined computational values of these 2^n circuits. We will argue that t is sufficiently small, and yet with high probability all 2^n circuits admit decision trees of depth at most t.

The problem with this idea is that after we have coded the values of all the 2^n circuits in X, there does not seem to be any easy way to recover this information. Since X had already been "ravaged" by the random restriction ρ, it is not clear how to distinguish those "code bits" from those "random bits". Further complicating the matter are those bits assigned during the decision tree settlement. All of this must be sorted out, supposedly, by a polynomial size oracle circuit which is to accept $L(\mathcal{M}^X)^{=n}$. Note that, after a random restriction ρ, it is probabilistically almost impossible to have an easily identifiable segment of the set X all assigned $*$ by ρ, (e.g., all strings in $\{0,1\}^{=n^k}$ with a certain leading bit pattern), not to mention the subsequent all 0 assignment to fix the decision trees. On the other hand, we have 2^n computations to code. It is infeasible for the final polynomial size oracle circuit to "remember" more than a polynomial number of bits as the address of the coding region. So it appears that we must have an easily identifiable region to code, identified with at most a polynomial number of bits for its address, and, to accommodate 2^n computations, this region must be large.

To overcome this difficulty, our idea is to use not true random restrictions, but pseudorandom restrictions via the Nisan-Wigderson generator [Nis91a, Nis91b, NW88]. Nisan and Wigderson designed a pseudorandom generator (which we will call a *NW generator*) provably indistinguishable from true random bits by polynomial size constant depth circuits. While our circuits are not of polynomial size, this can be scaled up easily. Our idea is then to use the output of some NW generator to perform the "random" restriction, and to argue that all 2^n circuits are "killed" with high probability, just as before with true random restrictions. The basic argument is that no constant depth circuits of an appropriate size can tell the difference under either a true random assignment or a pseudorandom assignment coming from the NW generator. However, for our purpose in this paper, we wish to say that a certain behavior of these 2^n constant depth circuits — namely they are likely to possess small depth decision trees after a "random" restriction with 0, 1 and $*$'s — is preserved when "pseudorandom restrictions" are substituted for "random restrictions". It is vitally important that whatever property we wish to claim to have been maintained by the substitution of random bits by pseudorandom bits, the property must be expressible as a constant depth circuit with an appropriate size upper bound. It is not clear the property of "having a small depth decision tree" can be expressed in this way.

We overcome this difficulty by using a weaker property which is a consequence of "having a small depth decision tree", which nonetheless is sufficient for our purpose. Namely, we take directly the property that, after a restriction with 0, 1 and $*$'s, every one of the 2^n circuits can be determined by assigning additionally 0's to a small number of variables, which had been assigned $*$'s. This property is expressible in a constant depth way. Then we will mimic the probability distribution of the 0, 1 and $*$'s under the random restrictions by uniform random bits 0's and 1's, so that we can come up with a constant depth circuit D with the following property: It takes only boolean inputs Ω of 0's and 1's, and D evaluates to 1 iff when a restriction ρ_Ω with 0, 1 and $*$'s defined by Ω is applied to all 2^n circuits C_x, every C_x can be set to either 0 or 1 after a small number of additional variables are set to 0. We will design D in such a way that under a uniform bit sequence Ω, D will almost certainly evaluate to 1.

In fact we need more than that. We also need to have the property that a certain segment of the oracle is untouched by the additional setting of 0's in all 2^n decision tree settlements. We will argue by the pigeonhole principle, that our bounds guarantee a suitable region unspoiled by all these decision tree settlement variables. It is not reasonable to expect that any such region is entirely assigned with $*$'s, but at least there should be many $*$'s.

Assume now we have designed such a D satisfying all these requirements. For this D we apply the NW generator, substituting pseudorandom bits for true random bits Ω given to D as inputs. We conclude that D still evaluates to 1 with high probability. In particular, there must be some setting of the source bits ω for the generator, such that D is evaluated to 1. This implies that we can assign the oracle set X first according to the pseudorandom restriction described by the pseudorandom bits, then according to the 2^n small depth decision trees, which are guaranteed by the evaluation of D, and set these additional variables all to 0. This settles all the decision trees and thus the values of all 2^n circuits C_x are determined. Furthermore, there is a significant segment T_{y_0} of X free from any variables used in any decision tree settlement, where we will code these 2^n results of C_x.

Even though this segment T_{y_0} is free from any variables used in any decision tree settlement, in order to code the computation results of C_x, there must be plenty of $*$ left, and they must be recoverable by polynomial size circuits. We will show that with high probability over a uniformly chosen random seed ω, the pseudorandom restriction defined by ω will leave plenty of $*$ in each segment such as T_{y_0}. We then in fact choose a sequence of bits ω that satisfies both the requirement $D = 1$ and this additional requirement.

Finally, we will show that with a suitable choice of parameters in the *combinatorial design* used in the NW generator, we will be able to recover in polynomial time the location where we assigned $*$'s in X, in particular from within the coding segment of X given the address of this segment. Now our polynomial size circuit $C_{\mathcal{M}}$ is designed as follows: It remembers (is hardwired with) y_0, i.e., the address of T_{y_0}, and remember the seed ω for the NW generator, which is of polynomial length. Then on any input x, it performs the polynomial time computation over a finite field to extract the coded result of C_x from the appropriate location in X.

3.2. Proof Detail

Now we specify the parameters and state our proof precisely.

Fix any Σ_d^{p} polynomial time bounded oracle alternating Turing machine $\mathcal{M}$, with time bound n^k. For notational convenience we will assume $k > 2$ and $d \geq 7$. We assume that n is sufficiently large. On input of length n, $\mathcal{M}$ can only query strings of length at most n^k. We will denote by $m = n^k$ and use M to denote 2^m throughout this proof.

Assume all memberships in the oracle set "$z \in X$?" up to length $< m$ have been decided already. Our task is to fix the membership for "$z \in X$?" of length exactly m in X, so that for all input x of length n, membership "$x \in L(\mathcal{M}^X)$?" can be decided by a polynomial size circuit $C_{\mathcal{M}}$ with oracle gates that can access $X^{=m}$. Since $X^{<m}$ has already been fixed, membership "$x \in L(\mathcal{M}^X)$?" is determined by the set $X^{=m}$. Here we specifically require that the circuit $C_{\mathcal{M}}$ can access only those strings that can be possibly accessed by the simulated machine $\mathcal{M}$ on input of length n.

There are 2^n inputs x of length n, each computation of $\mathcal{M}$ on x gives rise to a depth $d+1$ Boolean circuit C_x with size at most mM, and bottom fan-in at most m. The inputs to each circuit C_x are the $2M$ literals z and $\overline{z}$, where $z \in \{0,1\}^m$ corresponds to the truth value of $\chi_{[z \in X]}$. (To simplify our notation, we will denote by z both a string in $\{0,1\}^m$ as well as the Boolean variable corresponding to $\chi_{[z \in X]}$.) As stated earlier, we may assume in C_x the circuit starts with an OR gate at the top, and alternate with AND's and OR's in a tree like fashion, until inputs z's and $\overline{z}$'s. That is, each circuit C_x is a depth $d+1$ circuit with size at most mM and bottom fan-in at most m.

A Switching Lemma shows that such a constant depth circuit is sufficiently weakened, after a suitably chosen random restriction ρ, so as to have either small min-terms or small max-terms. The strongest form known is due to Håstad [Hås86a]. For our purpose in this paper, however, we will require something more.

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 $\mathrm{DC}(f) \leq t$, 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, clearly, there is a subset of no more than t variables, if one assigns all of them to 0, the function f will be determined. This is an important advantage as we will have to assign many non-disjoint subsets of variables for multiple Boolean functions, and all these assignments need to be consistent.

Adapting Håstad's proof to the decision tree model, one can prove the following lemma. In Section 4 we will discuss these lemmas more thoroughly.

Lemma 4 *For any depth $d+1$ Boolean circuit C on M inputs $z_1, z_2, \ldots, z_M$, of size at most s and bottom fan-in at most t, we have*

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

where the random restriction ρ is defined for $p = \frac{1}{(10t)^d}$.

To reduce all circuits C_x, $x \in \{0,1\}^n$, to small depth decision trees, we apply a random restriction with $p = 1/(20m)^d$ to these circuits. Then by the union bound we have,

Claim 1

$$\mathrm{Pr}_\rho[\bigvee_{x\in\{0,1\}^n} [\mathrm{DC}(C_x\mid_\rho) \geq 2m]\,] \;\leq\; 2^n \cdot \frac{mM}{2^{2m}} \;=\; \frac{m}{2^{m-n}}.$$

That is, with probability close to 1, a random restriction reduces *every* circuit C_x to a decision tree of depth $< 2m$.

Below we will carry out a sequence of transformations on the circuits C_x, $x \in \{0,1\}^n$, with the ultimate goal of constructing the circuit D which, in some sense, is a test for the success of a "random restriction".

Step 1 (C_x^1): C_x^1 takes $2M$ Boolean inputs (a_z, b_z), for $z \in \{0,1\}^m$. The pair (a_z, b_z) will represent the status of the Boolean variable z to C_x as follows: $a_z = 1$ iff the value of z is set (to either 0 or 1, i.e., not set to $*$), and $a_z = 0$ otherwise. If $a_z = 1$, then the 0-1 value of z is represented by b_z. If the pair (a_z, b_z) represents the value of z, then the pair $(a_z, \overline{b_z})$ represents that of $\overline{z_i}$. Clearly, if z is set 0 (resp., 1), then $\overline{z}$ must be set 1 (resp., 0).

C_x^1 is constructed from C_x as follows. Each gate g in C_x will be represented by a pair of gates (g_s, g_v). $g_s = 1$ iff g is set to either 0 or 1, i.e., it is determined; $g_s = 0$ otherwise. If $g_s = 1$ then $g = g_v$. Thus, $(g_s, g_v) = (0,0)$ or $(0,1)$ represent the situation where g has not been determined, and $(g_s, g_v) = (1,0)$, or $(1,1)$ respectively, represent the case where g is set to 0, or 1 respectively.

Suppose g is an OR gate, $g = \bigvee_{i=1}^s g^{(i)}$, where $g^{(i)}$ is an input literal or an internal gate. Suppose $g^{(i)}$ is represented by the pair $(g_s^{(i)}, g_v^{(i)})$. This representation is already defined inductively. Then we let

$$g_s \;=\; \bigvee_{i=1}^{s}\left((g_s^{(i)} \wedge g_v^{(i)})\right) \vee \left(\bigwedge_{i=1}^{s}(g_s^{(i)} \wedge \overline{g_v^{(i)}})\right).$$

That is, g is set iff either some g_i is set to 1, or else all g_i are set to 0. Note that the formula given for g_s is a depth 2 circuit of size $O(s)$. Also let

$$g_v \;=\; \bigvee_{i=1}^{s} g_v^{(i)}.$$

Note that g_v is only a "valid" value for g when $g_s = 1$. Also g_v is depth 1 and has size s.

The case $g = \bigwedge_{i=1}^{s} g^{(i)}$ is dual. In this case, g is set iff either some g_i is set to 0, or else all g_i are set to 1. Thus

$$g_s \;=\; \bigvee_{i=1}^{s}\left((g_s^{(i)} \wedge \overline{g_v^{(i)}})\right) \vee \left(\bigwedge_{i=1}^{s}(g_s^{(i)} \wedge g_v^{(i)})\right), \quad \text{and} \quad g_v \;=\; \bigwedge_{i=1}^{s} g_v^{(i)}.$$

Again they are depth 2, size $O(s)$, and depth 1, size s, respectively.

In order to maintain alternating form of AND's and OR's in the circuit C_x^1, with all negations pushed to the input level, we can represent each gate g by both g and its negated value $\overline{g}$. This can introduce at most a factor of 2 in the size. (In fact we will define only three gates g_s, g_v, and $\overline{g_v}$ in our construction; we do not need $(\overline{g})_s$. But we will omit the detailed analysis of constant factors.) C_x^1 has two output gates g_s and g_v for the output gate g of C_x. It follows that

$$\text{size}(C_x^1) = O(\text{size}(C_x)), \quad \text{and} \quad \text{depth}(C_x^1) = 2\ \text{depth}(C_x).$$

We can take the constant in $O(\text{size}(C_x))$ to be 10, say.

Step 2 (C_x^2): Let $p = \frac{1}{(2m)^d}$. Let $L = \lceil \log_2 \frac{1}{p} \rceil \approx dk \log_2(20n)$. C_x^2 takes Boolean inputs $(a_{z,1}, \ldots, a_{z,L}, b_z)$, for $z \in \{0,1\}^m$. The circuit C_x^2 is identical to C_x^1, except instead of taking inputs a_z, it has $a_z = \bigvee_{j=1}^{L} a_{z,j}$.

Note that, a random restriction ρ with parameter $\Pr[\rho(z) = *] = 1/2^L$ on C_x is simulated by uniformly and independently assigning all the bits $(a_{z,1}, \ldots, a_{z,L}, b_z)$ to 0 or 1, in C_x^2, for $z \in \{0,1\}^m$. The behavior of C_x is represented in C_x^2 exactly. Here we have

$$\text{size}(C_x^2) = \text{size}(C_x^1) + O(ML), \quad \text{and} \quad \text{depth}(C_x^2) = \text{depth}(C_x^1) + 1.$$

Note also that $2^{-L} \le p$. The same upper bound in Lemma 4 and Claim 1 still applies when ρ has parameter 2^{-L}.

Step 3 (C_x^3): In C_x^3 we will check for the existence of a subset $S \subset [M]$ of cardinality $|S| = 2m$ such that, first they are assigned $*$ by the ρ, and second if we further set them all to 0, it would determine the circuit C_x. We know from Claim 1 that this is almost certainly true for our random restriction.

Thus, we let

$$C_x^3 \;=\; \bigvee_{S}\left[\ \bigwedge_{z \in S} \overline{a_z} \wedge [(C_x^2)_s]_S\ \right],$$

where $\bigvee_S$ ranges over all subsets $S \subset [M]$ of cardinality $|S| = 2m$, and $(C_x^2)_s$ is the "set bit output" for C_x^2, and $[(C_x^2)_s]_S$ is obtained from $(C_x^2)_s$ by setting all $b_z = 0$ for $z \in S$. Recall that $a_z = \bigvee_{j=1}^{L} a_{z,j}$. Then we have

$$\mathrm{size}(C_x^3) \le \binom{M}{2m}(\mathrm{size}(C_x^2) + O(m)), \quad \text{and} \quad \mathrm{depth}(C_x^3) = \mathrm{depth}(C_x^2) + 2.$$

Step 4 (D)**:** Finally, define D by

$$D \;=\; \bigwedge_{x \in \{0,1\}^n} C_x^3.$$

Then we have

$$\mathrm{size}(D) = 2^n(\mathrm{size}(C_x^3)), \quad \text{and} \quad \mathrm{depth}(D) = \mathrm{depth}(C_x^3) + 1.$$

This completes the construction of D, with

$$\mathrm{size}(D) < 2^{3m^2}, \quad \text{and} \quad \mathrm{depth}(D) \le 2d + 6 \le 3d - 1.$$

Below we will denote $3d - 1$ by $\widehat{d}$.

From our construction, it follows that (i) the uniform independent distribution on the input bits of D simulates the random restriction ρ with $p = 2^{-L} \approx 1/(20m)^d$, and that (ii) D becomes true if every $C_x|\rho$ has decision tree depth at most $2m$. Hence, the following claim follows from Claim 1.

Claim 2

$$\Pr[D = 1] \;\ge\; 1 - \frac{m}{2^{m-n}},$$

where the probability is over uniform input bits of D.

Now we apply a NW generator to this circuit D. First we recall some basic notions on NW generators from [NW94].

Let U, M, m and q be positive integers. Let $[U]$ be some set of cardinality U, e.g., $\{1, 2, \ldots, U\}$. A collection of subsets $\mathcal{S} = \{S_1, \ldots, S_M\}$ of some domain $[U]$ is called a (m, q)-*design* if it satisfies the following conditions.

(1) $\forall i,\ 1 \le i \le M\ [\ |S_i| = q\ ]$, and

(2) $\forall i, \forall j,\ 1 \le i \ne j \le M\ [\ |S_i \cap S_j| \le m\ ]$.

Based on a given (m, q)-design $\mathcal{S} = \{S_1, \ldots, S_M\}$ with domain $[U]$, we define the following function $g_{\mathcal{S}}$: $\{0,1\}^U \to \{0,1\}^M$, which we call a (parity based) *NW generator*.

$$g_{\mathcal{S}}(x_1 \cdots x_U) \;=\; y_1 \cdots y_M,$$

where each y_i, $1 \le i \le M$, is defined
by $y_i \;=\; x_{s_1} \oplus \cdots \oplus x_{s_q}$ (where $S_i = \{s_1, \ldots, s_q\} \subseteq [U]$).

For the pseudorandomness of this generator, we have the following lemma [NW94].

Lemma 5 *For any positive integers U, M, m, q, s and e, and positive real ϵ, let $g_{\mathcal{S}}$ be the NW generator defined using an (m,q)-design $\{S_1, \ldots, S_M\}$ with domain $[U]$, and suppose for any depth $e+1$ circuit C on q input bits and of size at most $s + c_{\rm nw}2^m M$ (where $c_{\rm nw}$ is some constant), the q bit parity function has the following bias:*

$$\left|\Pr_{(u_1,\ldots,u_q)\in\{0,1\}^q}[C(u_1,\ldots,u_q) = u_1 \oplus \cdots \oplus u_q] - \frac{1}{2}\right| \;\le\; \frac{\epsilon}{M}.$$

Then $g_{\mathcal{S}}$ has the following pseudorandomness against any depth e circuit E on M input bits and of size at most s.

$$\left|\Pr_{\boldsymbol{y}\in\{0,1\}^M}[E(\boldsymbol{y}) = 1] - \Pr_{\boldsymbol{x}\in\{0,1\}^U}[E(g_{\mathcal{S}}(\boldsymbol{x})) = 1]\right| \;\le\; \epsilon.$$

To apply the NW generator to our depth $\widehat{d}$ circuit D constructed above, we set our parameters and define our (m,q)-design, as follows. For the parameters m and M, we will use the same ones that have been used so far, namely $m = n^k$ and $M = 2^m$. We will take a finite field $\mathbf{F}$, and set $q = |\mathbf{F}|$ and $U = q^2$. We will take a specific finite field $\mathbf{F} = \mathbf{Z}_2[X]/(X^{2\cdot 3^u} + X^{3^u} + 1)$ [vL91], where each element $\alpha \in \mathbf{F}$ takes $K = 2\cdot 3^u$ bits, and $q = |\mathbf{F}| = 2^K$. We choose u so that $q \ge (3m^2+1)^{\widehat{d}+2}$. Then $q^{1/(\widehat{d}+2)} \ge \log_2(2^{3m^2} + c_{\rm nw}2^m M)$, where $c_{\rm nw}$ is the constant in the above lemma. Clearly $q \le n^{ckd}$ will do, for some universal constant c, for example $c = 7$. Then $K = O(dk\log n)$. Thus, this field has polynomial size and each element is represented by $O(\log n)$ bits. All arithmetic operations in this field $\mathbf{F}$ are easy.

We will consider precisely $M = 2^m$ polynomials $f_z(\xi) \in \mathbf{F}[\xi]$, each of degree at most m, where each f_z is indexed by its coefficients, concatenated as a bit sequence of length exactly m. The precise manner in which this is done is not very important, but for definiteness, we can take the following. We take polynomials of degree $\delta = \lfloor m/K \rfloor = \Omega(n^k/(dk\log n)) \gg n^2$, with exactly $\delta + 1$ coefficients,

$$f_z(\xi) \;=\; c_\delta \xi^\delta + \ldots + c_1\xi + c_0,$$

where all c_j varies over $\mathbf{F}$, except c_δ is restricted to exactly $2^{m-K\cdot\delta}$ many values. Note that $0 \le m - K\cdot\delta < K$. The concatenation $z = \langle c_\delta \cdots c_0\rangle$ has exactly m bits. Each f_z defines a subset of $\mathbf{F}\times\mathbf{F}$ of cardinality q, $\{(\alpha, f_z(\alpha)) \mid \alpha \in \mathbf{F}\}$, which we denote by S_z. A (m,q)-design that we will use is defined as $\mathcal{S} = \{S_1, \ldots, S_M\}$, indexed by $z \in \{0,1\}^m$, which identify with the index set $\{1,\ldots,M\}$. Note that $\mathbf{F}\times\mathbf{F}$ is a domain $[U]$ with $U = q^2$. The first condition of a (m,q)-design is immediate, and the second condition, i.e., $|S_z \cap S_{z'}| \le m$, for all $z \ne z'$, is also easy to see by noting that $\deg(f_z) < m$ and $\deg(f_{z'}) < m$. Note that our NW generator $g_{\mathcal{S}}$ generates a pseudo random sequence of length $M = 2^m$ from a seed of length $U = q^2$.

For showing the pseudorandomness of $g_{\mathcal{S}}$, we use the following lemma which follows from the decision tree version of the Switching Lemma.

Lemma 6 *For any depth e, and for all sufficiently large q, any circuit C on q inputs and of size at most $2^{q^{1/(e+1)}}$, satisfies*

$$\left|\Pr_{(u_1,\ldots,u_q)\in\{0,1\}^q}[C(u_1,\ldots,u_q) = u_1 \oplus \cdots \oplus u_q] - \frac{1}{2}\right| \;\le\; 2^{-q^{1/(e+1)}}.$$

Then the following claim is immediate from Lemma 5 and Lemma 6.

Claim 3 *Our NW generator $g_{\mathcal{S}}$ has the following pseudorandomness against any circuit E of size at most 2^{3m^2} and depth $\widehat{d}$:*

$$\left|\Pr_{\boldsymbol{y}\in\{0,1\}^M}[E(\boldsymbol{y})=1] - \Pr_{\boldsymbol{x}\in\{0,1\}^U}[E(g_{\mathcal{S}}(\boldsymbol{x}))=1]\right| \ \le\ 2^{m-3m^2}.$$

Recall that the circuit D takes $(L+1)M$ Boolean inputs, i.e., $(a_{z,1},\ldots,a_{z,L},b_z)$, for $z\in\{0,1\}^m$, where $M=2^m$ and $L=\lceil\log_2\frac{1}{p}\rceil$. We provide these input values by our NW generator that produces a M bit pseudorandom string from a q^2 bit random seed. Hence, for the seed to the generator, a random string of length $(L+1)q^2$ is needed, and we use a sequence of independently and uniformly distributed bits $\{u^{(0)}_{\alpha,\beta},u^{(1)}_{\alpha,\beta},\ldots,u^{(L)}_{\alpha,\beta}\}$, for each $\alpha,\beta\in\mathbf{F}$. That is, for each $j=1,\ldots,L$, we use q^2 bits $\{u^{(j)}_{\alpha,\beta}\mid\alpha,\beta\in\mathbf{F}\}$ to generate the M Boolean values of $a_{z,j}$, for $z\in\{0,1\}^m$. Similarly, the set $\{u^{(0)}_{\alpha,\beta}\mid\alpha,\beta\in\mathbf{F}\}$ of q^2 bits is used to generate the M Boolean values of b_z, for $z\in\{0,1\}^m$. More specifically, for each $z\in\{0,1\}^m$ and $j=1,\ldots,L$, we define $a_{z,j}$ and b_z as follows.

$$a_{z,j}\ =\ \oplus_{\alpha\in\mathbf{F}}\,u^{(j)}_{\alpha,f_z(\alpha)},\quad\text{and}\quad b_z\ =\ \oplus_{\alpha\in\mathbf{F}}\,u^{(0)}_{\alpha,f_z(\alpha)}.$$

Then we have the following claim.

Claim 4 *Let $\boldsymbol{g}^{(i)}_{\mathcal{S}}$ denote the pseudorandom output sequence of $g_{\mathcal{S}}$ on random seed bits $\{u^{(i)}_{\alpha,\beta}\mid\alpha,\beta\in\mathbf{F}\}$, for $0\le i\le L$. Then*

$$\Pr[D(\boldsymbol{g}^{(1)}_{\mathcal{S}},\ldots,\boldsymbol{g}^{(L)}_{\mathcal{S}},\boldsymbol{g}^{(0)}_{\mathcal{S}})=1]\ \ge\ 1-o(1),$$

where the probability is over independently and uniformly distributed bits $\{u^{(0)}_{\alpha,\beta},u^{(1)}_{\alpha,\beta},\ldots,u^{(L)}_{\alpha,\beta}\}$, for $\alpha,\beta\in\mathbf{F}$.

Proof. Let us denote by $\boldsymbol{a}_i$ and $\boldsymbol{b}$ respectively a sequence of M true random bits assigned to D's input variables $a_{z,i}$ and b_z, for $z\in\{0,1\}^m$. Then our goal is to show that $\Pr[D(\boldsymbol{g}^{(1)}_{\mathcal{S}},\ldots,\boldsymbol{g}^{(L)}_{\mathcal{S}},\boldsymbol{g}^{(0)}_{\mathcal{S}})=1]$ is close to 1. We claim $\Pr[D(\boldsymbol{g}^{(1)}_{\mathcal{S}},\ldots,\boldsymbol{g}^{(L)}_{\mathcal{S}},\boldsymbol{g}^{(0)}_{\mathcal{S}})\ne 1]\le 1/2^{n-1}$. For a contradiction suppose it is $>1/2^{n-1}$.

Recall that from Claim 2 $\Pr[D(\boldsymbol{a}_1,\ldots,\boldsymbol{a}_L,\boldsymbol{b})\ne 1]\le m/2^{m-n}$.

Then we have

$$\left|\Pr[D(\boldsymbol{a}_1,\ldots,\boldsymbol{a}_L,\boldsymbol{b})\ne 1]-\Pr[D(\boldsymbol{g}^{(1)}_{\mathcal{S}},\ldots,\boldsymbol{g}^{(L)}_{\mathcal{S}},\boldsymbol{g}^{(0)}_{\mathcal{S}})\ne 1]\right|\ >\ \frac{1}{2^{n-1}}-\frac{m}{2^{m-n}}\ >\ \frac{1}{2^n}.$$

This implies, by the telescoping argument,

$$\begin{aligned}
\frac{1}{2^n}\ &<\ |\Pr[D(\boldsymbol{a}_1,\ldots,\boldsymbol{a}_L,\boldsymbol{b})\ne 1]-\Pr[D(\boldsymbol{g}^{(1)}_{\mathcal{S}},\ldots,\boldsymbol{g}^{(L)}_{\mathcal{S}},\boldsymbol{g}^{(0)}_{\mathcal{S}})\ne 1]|\\
&\le\ |\Pr[D(\boldsymbol{a}_1,\ldots,\boldsymbol{a}_L,\boldsymbol{b})\ne 1]-\Pr[D(\boldsymbol{g}^{(1)}_{\mathcal{S}},\boldsymbol{a}_2,\ldots,\boldsymbol{a}_L,\boldsymbol{b})\ne 1]|\\
&\quad +|\Pr[D(\boldsymbol{g}^{(1)}_{\mathcal{S}},\boldsymbol{a}_2,\ldots,\boldsymbol{a}_L,\boldsymbol{b})\ne 1]-\Pr[D(\boldsymbol{g}^{(1)}_{\mathcal{S}},\boldsymbol{g}^{(2)}_{\mathcal{S}},\boldsymbol{a}_3,\ldots,\boldsymbol{a}_L,\boldsymbol{b})\ne 1]|\\
&\quad \cdots\\
&\quad +|\Pr[D(\boldsymbol{g}^{(1)}_{\mathcal{S}},\ldots,\boldsymbol{g}^{(L)}_{\mathcal{S}},\boldsymbol{b})\ne 1]-\Pr[D(\boldsymbol{g}^{(1)}_{\mathcal{S}},\ldots,\boldsymbol{g}^{(L)}_{\mathcal{S}},\boldsymbol{g}^{(0)}_{\mathcal{S}})\ne 1]|,
\end{aligned}$$

that there exists some i such that

$$\left|\Pr[D(\boldsymbol{g}_{\mathcal{S}}^{(1)},\ldots,\boldsymbol{g}_{\mathcal{S}}^{(i-1)},\boldsymbol{a}_i,\ldots,\boldsymbol{a}_L,\boldsymbol{b})\neq 1]-\Pr[D(\boldsymbol{g}_{\mathcal{S}}^{(1)},\ldots,\boldsymbol{g}_{\mathcal{S}}^{(i-1)},\boldsymbol{g}_{\mathcal{S}}^{(i)},\boldsymbol{a}_{i+1},\ldots,\boldsymbol{a}_L,\boldsymbol{b})\neq 1]\right| \;>\; \frac{1}{L2^n}.$$

By an averaging argument, this bound still holds by appropriately fixing random bits other than $\boldsymbol{a}_i$ and the source bits for $\boldsymbol{g}_{\mathcal{S}}^{(i)}$. In other words, for some circuit D' with M input variables of size at most $\mathrm{size}(D)=2^{3m^2}$ and depth $\mathrm{depth}(D)=\widehat{d}$, we have

$$\left|\Pr[D'(\boldsymbol{a}_i)=1]-\Pr[D'(\boldsymbol{g}_{\mathcal{S}}^{(i)})=1]\right| \;>\; \frac{1}{L2^n}.$$

This is a contradiction to Claim 3, the pseudorandomness of the generator $g_{\mathcal{S}}$, since $L=O(d\log m)$. $\square$ (Claim 4)

This claim states that with high probability, a pseudorandom sequence satisfies D, meaning that the random restriction induced from the pseudorandom sequence reduces *every* C_x to a simple function (e.g., a small decision tree) whose value can be fixed by fixing $t=2m$ additional variables (for each C_x) to 0. Next we will argue that, for such a pseudorandom restriction, one can find some space to encode the determined value of each C_x.

Consider a restriction induced by a pseudorandom sequence satisfying D. Apply this restriction to all variables z of circuits C_x, and fix further the value of some set Y of variables to 0 in order to determine the value of circuits C_x for all $x\in\{0,1\}^n$. We may assume that the size of Y is at most $2m2^n$, which is guaranteed by the fact that $D=1$ with our pseudorandom sequence. Then there exists y_0 of length $n^2/2$ such that a segment $T_{y_0}=\{z\in\{0,1\}^m \mid y_0 \text{ is a prefix of } z\}$ has no intersection with Y; that is, all variables in T_{y_0} are free from any variables used to fix the value of circuits C_x . This is simply because $2m2^n\ll 2^{n^2/2}$. Our plan is to code the results of C_x by a Boolean variable z of the form $z=y_0xw$, for some w. The key requirements are that (i) the variable z is assigned $*$ by the pseudorandom restriction, and (ii) it is easy to find such z (i.e., w) from a given x. (We may assume that the string y_0 and the seed for the chosen pseudorandom sequence are remembered by being encoded in the target polynomial-size circuit $C_{\mathcal{M}}$.)

Let $\boldsymbol{u}_{\alpha,\beta}$ be a column vector of 0-1 uniform bits $(u_{\alpha,\beta}^{(1)},u_{\alpha,\beta}^{(2)},\ldots,u_{\alpha,\beta}^{(L)})^{\mathrm{T}}$. Recall that in D's simulation of circuits C_x, a Boolean variable z (of C_x) is assigned $*$ if and only if $a_{z,j}=0$ for all j $=1,\ldots,L$. Hence, z is assigned $*$ by a pseudorandom restriction if and only if $\sum_{\alpha\in\mathbf{F}}\boldsymbol{u}_{\alpha,f_z(\alpha)}=\mathbf{0}$ in $\mathbf{Z}_2^L$. y_0 is determined by the pigeonhole principle, and depends on the source bits $\boldsymbol{u}_{\alpha,\beta}$. We also need to have plenty of $*$'s in the segment T_{y_0}. Since we cannot predetermine y_0, we demand all segments T_y have plenty of $*$'s. So, we want our source bits $\boldsymbol{u}_{\alpha,\beta}$ to satisfy the following condition.

$$\forall y\in\{0,1\}^{n^2/2},\ \forall x\in\{0,1\}^n,\ \exists z=yxw\in\{0,1\}^m\ \left[\ \sum_{\alpha\in\mathbf{F}}\boldsymbol{u}_{\alpha,f_z(\alpha)}=\mathbf{0}\ \right]. \tag{1}$$

Furthermore, such a w should be easy to compute from the source bits $\boldsymbol{u}_{\alpha,\beta}$, and the given y, and x.

Recall that for any $z\in\{0,1\}^m$, f_z is defined by the sequence of the coefficients $\langle c_\delta\cdots c_0\rangle$ which concatenates to z. Let γ be the largest index such that the binary concatenation $\langle c_\delta\cdots c_\gamma\rangle$

becomes longer than $n^2/2+n$ bits, so $n^2/2+n < |\langle c_\delta \cdots c_\gamma\rangle| \le n^2/2+n+K$. Then for any $y \in \{0,1\}^{n^2/2}$ and $x \in \{0,1\}^n$, we have some subsequence of coefficients $c_\delta, \dots, c_\gamma$ such that $yx0^v = \langle c_\delta \cdots c_\gamma\rangle$, with some v for padding. Note that $\gamma > 0$, since $m = n^k$ and $k > 2$. We will show (see Claim 5 below) that with high probability a sequence of random source bits $\boldsymbol{u}_{\alpha,\beta}$ satisfies the following.

$$\forall c_\delta \in \mathbf{F},\ \dots,\ \forall c_\gamma \in \mathbf{F},\ \exists c_0 \in \mathbf{F}\ \left[\ \sum_{\alpha\in\mathbf{F}} \boldsymbol{u}_{\alpha,f_{z^*}(\alpha)} = \mathbf{0}\ \right], \tag{2}$$

where z^* is a string in $\{0,1\}^m$ that is the concatenation $\langle c_\delta \cdots c_\gamma 0 \cdots 0 c_0\rangle$. Observe that this condition (2) is sufficient for our requirement (1). Consider any $y \in \{0,1\}^{n^2/2}$ and $x \in \{0,1\}^n$, and let $c_\delta, \dots, c_\gamma$ be the coefficients corresponding to $yx0^v$. Then from (2), there exists some c_0 by which we can define $z^* = \langle c_\delta \cdots c_\gamma 0 \cdots 0 c_0\rangle$ satisfying the condition of (1). Furthermore, we will show that we can easily find such c_0 (thus z^*) given $\boldsymbol{u}_{\alpha,\beta}$, and $c_\delta, \dots, c_\gamma$. by checking all q elements of $\mathbf{F}$.

We now summarize our oracle construction. Choose any setting of the random bits $\omega = \boldsymbol{u}_{\alpha,\beta}$, such that it generates $(L+1)M$ pseudorandom bits Ω satisfying both $D = 1$ and (2); let ρ_Ω be the restriction induced by this pseudorandom sequence. We construct the segment $X^{=m}$ of our oracle by ρ_Ω as follows. Below z denotes a string in $\{0,1\}^m$ whose membership to X has not been determined yet in the construction. Let X_{fixed} (resp., $\overline{X}_{\text{fixed}}$) be the set of strings in $\{0,1\}^m$ whose membership to X (resp., $\overline{X}$) have been determined. Initially, both X_{fixed} and $\overline{X}_{\text{fixed}}$ are empty. First fix the membership according to ρ_Ω; that is, z is put into X_{fixed} (resp., $\overline{X}_{\text{fixed}}$) if and only if ρ_Ω sets 1 (resp., 0) to the corresponding variable. Secondly, choose a set $Y \subseteq \{0,1\}^m - (X_{\text{fixed}} \cup \overline{X}_{\text{fixed}})$ of at most $2m2^n$ strings such that adding Y to $\overline{X}_{\text{fixed}}$ determines the value of circuits C_x for all $x \in \{0,1\}^n$. This set Y is guaranteed by $D = 1$. Add Y to $\overline{X}_{\text{fixed}}$. Fix one y_0 such that $T_{y_0} \cap Y = \emptyset$. This y_0 exists by the pigeonhole principle. Then for any $x \in \{0,1\}^n$, put any z of the form y_0xw for some w into X_{fixed} (resp., $\overline{X}_{\text{fixed}}$) if and only if the (already determined) value of C_x is 1 (resp., 0). Then put all remaining z into $\overline{X}_{\text{fixed}}$.

Now we explain how to design a polynomial-size circuit $C_{\mathcal{M}}$ simulating $\mathcal{M}^X$. We may assume that the information on the seed ω (of length $(L+1)q^2 = n^{O(kd)}$) and y_0 are hardwired into the circuit and they can be used in the computation. For a given input x, the circuit exhaustively searches for $c_0 \in \mathbf{F}$ satisfying the condition of (2) for the coefficients $c_\delta, \dots, c_\gamma$ corresponding to y_0x0^v. Since the seed is given, for any $z^* = \langle c_\delta \cdots c_\gamma 0 \cdots 0 c_0\rangle$, one can compute $\sum_{\alpha\in\mathbf{F}} \boldsymbol{u}_{\alpha,f_{z^*}(\alpha)}$ within polynomial-time in n. Also the size of $\mathbf{F}$ is $q = n^{O(kd)}$. Thus, the desired c_0 (and hence, z^*) is computable in polynomial-time. When z^* is obtained, the circuit queries the oracle whether "$z^* \in X$?" and accepts the input if and only if $z^* \in X$. It is easy to check that the whole computation can be implemented by some circuit of size n^{ckd} for some constant $c > 0$.

We complete the proof by proving the following claim.

Claim 5 *Over q^2L independent and uniform random bits $\{u^{(1)}_{\alpha,\beta}, \dots, u^{(L)}_{\alpha,\beta} \mid \alpha,\beta \in \mathbf{F}\}$, the condition (2) holds with probability $1 - o(1)$.*

Proof. For any fixed $c_\delta, \dots, c_\gamma$, let $z^*(c)$ denote $\langle c_\delta \cdots c_\gamma 0 \cdots 0 c\rangle$. Then $f_{z^*(c)}(\xi)$ is expressed as $f_{z^*(c)}(\xi) = g(\xi) + c$, where the polynomial $g(\xi) = c_\delta\xi^\delta + \cdots + c_\gamma\xi^\gamma$ is independent of c.

Define $\boldsymbol{u}^*_{\alpha,c} = \boldsymbol{u}_{\alpha,g(\alpha)+c}$. Then since $\boldsymbol{u}^*_{\alpha,c} = \boldsymbol{u}_{\alpha,f_{z^*(c)}(\alpha)}$, the condition (2) can be stated as

$$\forall c_\delta,\ \ldots,\ \forall c_\gamma,\ \exists c_0 \left[\sum_{\alpha\in\mathbf{F}} \boldsymbol{u}^*_{\alpha,c_0}(\alpha) = \mathbf{0} \right].$$

Notice that for any fixed $c_\delta, \ldots, c_\gamma$, for any α, α', c and c', the vectors $\boldsymbol{u}^*_{\alpha,c}$ and $\boldsymbol{u}^*_{\alpha,c'}$ consist of disjoint sets of bits, unless $\alpha = \alpha'$ and $c = c'$. Hence, if $c \neq c'$, they are (probabilistically) independent, from which the following bound follows: $\forall c_\delta, \ldots, c_\gamma$,

$$\Pr\left[\forall c_0 \, [\sum_{\alpha\in\mathbf{F}} \boldsymbol{u}^*_{\alpha,c_0} \neq \mathbf{0}\,] \right] \ = \ \prod_{c\in\mathbf{F}} \Pr\left[\sum_{\alpha\in\mathbf{F}} \boldsymbol{u}^*_{\alpha,c} \neq \mathbf{0} \right] \ = \ \left(1 - \frac{1}{2^L}\right)^q \ < \ e^{-\Omega(q/(20m)^d)},$$

where the probability is taken uniformly over all the bits $u^{(1)}_{\alpha,\beta}, \ldots, u^{(L)}_{\alpha,\beta}$, for all $\alpha, \beta \in \mathbf{F}$. Then the claim is proved as follows:

$$\Pr\left[\forall c_\delta,\ \ldots,\ \forall c_\gamma,\ \exists c_0 \, [\sum_{\alpha\in\mathbf{F}} \boldsymbol{u}^*_{\alpha,c_0}(\alpha) = \mathbf{0}\,] \right] \ \geq \ 1 - 2^{n^2/2+n+K} e^{-\Omega(q/(20m)^d)} \ = \ 1 - o(1).$$

□ (Claim 5)

Remark 1: For convenience we assumed in the proof that $k > 2$ and $d \geq 7$. This is only to simplify notations. Clearly $d \geq 7$ is unnecessary. We only need to forgo the estimate of $2d + 6 \leq 3d - 1$, and use $2d + 6$. Also any machine $\mathcal{M}$ in Σ^{p}_d for $d < 7$ can always be considered in a higher level. Similarly, $k > 2$ is not necessary. If one traces through the proof, with slight modification, any real number $k > 1$ is sufficient.

Remark 2: The final computation by the polynomial size circuit can be done in NC^1. We only need to evaluate some arithmetic operations in the finite field $\mathbf{F}$. It turns out that since elements in $\mathbf{F}$ are represented by $O(\log n)$ bits, the only step that really requires NC^1 is the parity sum of $n^{O(1)}$ terms, when we evaluate the polynomial f_z.

Remark 3: Though the proof is stated for simulating one machine $\mathcal{M}$, it is also possible to construct a single oracle X such that for every d and k, and every Σ^{p}_d-machine $\mathcal{M}$ running in time $O(n^k)$, the language $L(\mathcal{M}^X)$ can be recognized by some polynomial-size circuit family with stringent access to oracle X.

4 Some results on constant depth circuits

As lower bound results on constant depth circuits play a crucial role in this work, we take this opportunity to present some unpublshed older results of the first author on these circuits. In particular we emphasize the decision tree viewpoint, and give some better constants in the exponents than previously published lower bounds. We give a historical account at the end of the section.

The decision tree perspective was first proposed in [Cai86] where a weaker version of this Lemma was proved. The following proof essentially adapts the techniques from [Hås86a].

We say a boolean function G on variables $\{x_1, \ldots, x_n\}$ is a *t-And-Or* if $G = G_1 \wedge G_2 \wedge \ldots \wedge G_w$, where each G_i is the Or of at most t literals, (a literals is a variable or its complement). Similarly, we say G is a *t-Or-And* if $G = G_1 \vee G_2 \vee \ldots \vee G_w$, where each G_i is the And of at most t literals.

A restriction is a partial assignment of some of the variables to $\{0, 1\}$. More formally, it is a map ρ from the set $\{1, 2, \ldots, n\}$ to the set $\{0, 1, *\}$. The restriction of G by ρ, denoted by $G|_\rho$, is the boolean function obtained by setting x_i to be $\rho(i)$ if $\rho(i) \in \{0, 1\}$ and leaving x_i as a variable otherwise. A random p-restriction is a restriction ρ picked by independently assigning $\rho(i) = *$ with probability p and either 0 or 1 with probability $(1 - p)/2$.

Lemma 7 *Let G be a t-And-Or formula $G_1 \wedge G_2 \wedge \ldots \wedge G_w$. Let ρ be a random p-restriction. Then, for all $\Delta \geq 0$,*

$$\Pr[\ \mathrm{DC}(G|_\rho) \geq \Delta\ ] \ \leq\ (5pt)^\Delta. \tag{3}$$

Proof. The Lemma is proved by an induction on w. Concerning G_1, immediately there are 2 cases, either $G_1|_\rho \equiv 1$ or $G_1|_\rho \not\equiv 1$. By renaming literals, we may assume $G_1 = \bigvee_{i \in T} x_i$. Then $G_1|_\rho \equiv 1$ is equivalent to $\rho(i) = 1$ for some $i \in T$. If $G_1|_\rho \equiv 1$, we want to prove that the conditional probability that the rest of G has $\mathrm{DC}(G|_\rho) \geq \Delta$ is no larger. If however $G_1|_\rho \not\equiv 1$, we want to carefully analyze what happens to the variables in T. All of this will accumulate as some prior condition on ρ. It will be seen that the inductive step will carry a condition that refers to some collection of subsets of variables on each of which ρ has assigned some variable of it in some definite way. In the earlier proof of Yao [Yao85], as well as in the proof of Cai [Cai86], these conditions are explicitly carried along in the proof. The following device used in Håstad's proof [Hås86a] is more elegant.

One makes the stronger claim, that for *any* boolean function F, we have

$$\Pr[\ \mathrm{DC}(G|_\rho) \geq \Delta \mid F|_\rho \equiv 1\ ] \ \leq\ \alpha^\Delta, \tag{4}$$

where α will be set to $5pt$, and we agree that the conditional probability is 0 if the condition is not satisfied. The Lemma follows from (4) by taking F to be the constant function 1.

The statement (4) is trivially true for $\Delta = 0$, since the RHS becomes 1 in this case. Similarly, if $\alpha \geq 1$, then the statement is true. Thus, we may assume $\Delta > 0$ and $\alpha < 1$.

We prove (4) by induction on w. If $w = 0$, then $G \equiv 1$ by definition and the statement holds since the LHS is 0. Let $w > 0$. Put $G = G_1 \wedge G'$, where $G' = G_2 \wedge \ldots \wedge G_w$. Now, either $G_1|_\rho \equiv 1$ or $G_1|_\rho \not\equiv 1$. If $G_1|_\rho \equiv 1$, then we have, by induction

$$\begin{aligned}&\Pr[\ \mathrm{DC}(G|_\rho) \geq \Delta \mid F|_\rho \equiv 1, G_1|_\rho \equiv 1\ ] \\ &\ = \ \Pr[\ \mathrm{DC}(G'|_\rho) \geq \Delta \mid (F \wedge G_1)|_\rho \equiv 1\ ] \ \leq\ \alpha^\Delta.\end{aligned}$$

Now consider the case $G_1|_\rho \not\equiv 1$. We want to prove

$$\Pr[\ \mathrm{DC}(G|_\rho) \geq \Delta \mid F|_\rho \equiv 1, G_1|_\rho \not\equiv 1\ ] \ \leq\ \alpha^\Delta \tag{5}$$

as well. We have renamed the variables so that $G_1 = \bigvee_{i \in T} x_i$. Then $G_1|_\rho \not\equiv 1$ means that for each $i \in T$, $\rho(i) = 0$ or $*$. Moreover, since $\Delta > 0$, it cannot be that $\rho(i) = 0$ for all $i \in T$, or else $G_1|_\rho \equiv 0$, and $\mathrm{DC}(G|_\rho) = 0$. Thus, the set of restrictions ρ such that $F|_\rho \equiv 1, G_1|_\rho \not\equiv 1$ and $\mathrm{DC}(G|_\rho) \geq \Delta$ is contained in

$$\bigcup_{\emptyset \neq Y \subseteq T} \{\rho \,:\, \rho(Y) = *, \rho(T-Y) = 0, F|_\rho \equiv 1, \mathrm{DC}(G|_\rho) \geq \Delta\,\}.$$

Suppose $\rho(Y) = *$ and $\rho(T-Y) = 0$, for some $\emptyset \neq Y \subseteq T$.

First we assume $|Y| < \Delta$. Then there must be some assignment $\sigma_Y : Y \to \{0,1\}$, and $\sigma_Y \neq 0^Y$, where we denote by 0^Y the all 0 assignment on Y, such that $\mathrm{DC}(G|_\rho|_{\sigma_Y}) \geq \Delta - |Y|$. For otherwise, one could obtain *some* decision tree of depth $< \Delta$ for $G|_\rho$ by first asking all the variables in Y. Note that such a $\sigma_Y \neq 0^Y$ because the all 0 assignment leads to $G_1|_\rho|_{0^Y} \equiv 0$.

For $\sigma_Y \neq 0^Y$, $G_1|_\rho|_{\sigma_Y} \equiv 1$, so that $G|_\rho|_{\sigma_Y} \equiv G'|_\rho|_{\sigma_Y}$. Then

$$\begin{aligned}&\Pr[\ \mathrm{DC}(G|_\rho) \geq \Delta \mid F|_\rho \equiv 1 \wedge \rho(Y) = * \wedge \rho(T-Y) = 0\ ] \\ &\quad \leq\ \sum_{\substack{\sigma_Y : Y \to \{0,1\} \\ \sigma_Y \neq 0^Y}} \Pr[\ \mathrm{DC}(G'|_\rho|_{\sigma_Y}) \geq \Delta - |Y| \mid F|_\rho \equiv 1 \wedge \rho(Y) = * \wedge \rho(T-Y) = 0)\ ]. \qquad (6)\end{aligned}$$

Set

$$\begin{aligned} 0^{T-Y} &= \text{the all 0 assignment on } T-Y, \\ \widetilde{F} &= \textstyle\bigwedge_{\tau_Y : Y \to \{0,1\}} F|_{0^{T-Y}}|_{\tau_Y} \ \text{ and} \\ \widetilde{\rho} &= \rho \text{ restricted to the complement of } T, \end{aligned}$$

then under the condition $\rho(Y) = * \wedge \rho(T-Y) = 0$ we have

$$F|_\rho \equiv 1 \iff \widetilde{F}|_{\widetilde{\rho}} \equiv 1.$$

Hence, the sum in (6) has the upper bound

$$\sum_{\substack{\sigma_Y : Y \to \{0,1\} \\ \sigma_Y \neq 0^Y}} \Pr[\ \mathrm{DC}(G'|_{0^{T-Y}}|_{\sigma_Y}|_{\widetilde{\rho}}) \geq \Delta - |Y| \mid \widetilde{F}|_{\widetilde{\rho}} \equiv 1\ ]\ \leq\ (2^{|Y|} - 1)\alpha^{\Delta - |Y|}, \qquad (7)$$

by induction.

The upper bound (7) holds for

$$\Pr[\ \mathrm{DC}(G|\rho) \geq \Delta \mid F|\rho \equiv 1 \wedge \rho(Y) = * \wedge \rho(T-Y) = 0\ ] \qquad (8)$$

for all $Y \neq \emptyset$ with $|Y| < \Delta$. However, for $|Y| \geq \Delta$, the bound in (7) holds trivially for a probability (8), since in this case the bound in (7) is ≥ 1, as $|Y| \geq \Delta > 0$ and $\alpha < 1$. Hence in fact it holds for all $Y \neq \emptyset$.

Let

$$\begin{aligned} a_Y &= \Pr[\ \rho(Y) = * \wedge \rho(T-Y) = 0 \mid F|_\rho \equiv 1 \wedge G_1|_\rho \not\equiv 1\ ], \\ b_Y &= \Pr[\ \rho(Y) = * \mid F|_\rho \equiv 1 \wedge G_1|_\rho \not\equiv 1\ ]. \end{aligned}$$

Then

$$b_Y\ =\ \sum_{Y \subseteq Z \subseteq T} a_Z,$$

and by the Möbius Inversion Formula,

$$a_Y = \sum_{Y\subseteq Z\subseteq T} (-1)^{|Z-Y|} b_Z.$$

It follows that

$$\begin{aligned}
&\Pr[\ \mathrm{DC}(G|_\rho) \geq \Delta \mid F|_\rho \equiv 1, G_1|_\rho \not\equiv 1\ ] \\
&\leq \sum_{\emptyset\neq Y\subseteq T} a_Y \cdot (2^{|Y|}-1)\alpha^{\Delta-|Y|} \\
&= \sum_{Y\subseteq T} a_Y \cdot (2^{|Y|}-1)\alpha^{\Delta-|Y|}.
\end{aligned}$$

Substituting b_Z for a_Y, we have

$$\begin{aligned}
&\Pr[\ \mathrm{DC}(G|_\rho) \geq \Delta \mid F|_\rho \equiv 1, G_1|_\rho \not\equiv 1\ ] \\
&\leq \sum_{Y\subseteq T}\sum_{Y\subseteq Z\subseteq T} (-1)^{|Z-Y|} b_Z \cdot (2^{|Y|}-1)\alpha^{\Delta-|Y|} \\
&= \sum_{Z\subseteq T} b_Z \sum_{Y\subseteq Z} (-1)^{|Z-Y|}(2^{|Y|}-1)\alpha^{\Delta-|Y|} \\
&= \sum_{Z\subseteq T} b_Z(-1)^{|Z|}\alpha^{\Delta} \sum_{Y\subseteq Z}\left[\left(\frac{-2}{\alpha}\right)^{|Y|} - \left(\frac{-1}{\alpha}\right)^{|Y|}\right] \\
&= \alpha^{\Delta}\sum_{Z\subseteq T} b_Z(-1)^{|Z|}\left[\left(1-\frac{2}{\alpha}\right)^{|Z|} - \left(1-\frac{1}{\alpha}\right)^{|Z|}\right] \\
&= \alpha^{\Delta}\sum_{Z\subseteq T} b_Z\left[\left(\frac{2}{\alpha}-1\right)^{|Z|} - \left(\frac{1}{\alpha}-1\right)^{|Z|}\right].
\end{aligned}$$

Concerning b_Z, intuitively, under the condition that $F|_\rho \equiv 1 \wedge G_1|_\rho \not\equiv 1$, the probability of $\rho(Z) = *$ is at most $q^{|Z|}$, where $q = p/(p+\frac{1-p}{2}) \approx 2p$, i.e., $b_Z \leq q^{|Z|}$. We already saw that $G_1|_\rho \not\equiv 1$ means that each variable in Z is assigned either 0 or $*$. The additional condition that $F|_\rho \equiv 1$ can only decrease the probability that some variable is assigned a $*$. We will argue this point more carefully. For the moment, we accept the upper bound $b_Z \leq q^{|Z|}$.

Then, since the coefficients of b_Z are non-negative, we have

$$\begin{aligned}
&\alpha^{\Delta}\sum_{Z\subseteq T} b_Z\left[\left(\frac{2}{\alpha}-1\right)^{|Z|} - \left(\frac{1}{\alpha}-1\right)^{|Z|}\right] \\
&\leq \alpha^{\Delta}\sum_{Z\subseteq T} q^{|Z|}\left[\left(\frac{2}{\alpha}-1\right)^{|Z|} - \left(\frac{1}{\alpha}-1\right)^{|Z|}\right] \\
&= \alpha^{\Delta}\left\{\left[1+q\left(\frac{2}{\alpha}-1\right)\right]^{|T|} - \left[1+q\left(\frac{1}{\alpha}-1\right)\right]^{|T|}\right\} \\
&\leq \alpha^{\Delta}\left\{\left[1-q+\frac{2q}{\alpha}\right]^{t} - \left[1-q+\frac{q}{\alpha}\right]^{t}\right\}. \qquad (9)
\end{aligned}$$

At this point, we can recover the bound $(5pt)^{\Delta}$ as follows [Hås86a]. Observe that

$$\left[1-q+\frac{2q}{\alpha}\right]^t - \left[1-q+\frac{q}{\alpha}\right]^t \;\le\; \left(1+\frac{2q}{\alpha}\right)^t - \left(1+\frac{q}{\alpha}\right)^t. \tag{10}$$

If we set $c = 1/\log\phi \approx 2.078$, where $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$ is the golden ratio, then we have $e^{2/c} - e^{1/c} = 1$. Then set $\alpha = cqt < 5pt$, we get

$$\left(1+\frac{2q}{\alpha}\right)^t - \left(1+\frac{q}{\alpha}\right)^t \;=\; \left(1+\frac{2}{ct}\right)^t - \left(1+\frac{1}{ct}\right)^t \;<\; e^{2/c} - e^{1/c} \;=\; 1.$$

Then

$$\Pr[\ \mathrm{DC}(G|_\rho) \ge \Delta \mid F|_\rho \equiv 1, G_1|_\rho \not\equiv 1\ ] \;<\; \alpha^\Delta.$$

This completes the proof of

$$\Pr[\ \mathrm{DC}(G|_\rho) \ge \Delta \mid F|_\rho \equiv 1\ ] \;<\; (5pt)^\Delta.$$

Finally, we show that $b_Z \le q^{|Z|}$. Note that for $Z \subseteq T$, we have

$$\Pr[\ \rho(Z) = * \mid G_1|_\rho \not\equiv 1\ ] \;=\; q^{|Z|},$$

This is because $G_1|_\rho \not\equiv 1$ is the same as ρ assigns only $*$ or 0 on T.

We show that $F|_\rho \equiv 1$ cannot increase the probability of $\rho(Z) = *$. This is trivial if $Z = \emptyset$. Suppose $Z \ne \emptyset$. Consider any fixed restriction ρ' on the complement of Z, $\rho' : Z^c \to \{0, 1, *\}$. Then, there is a unique extension of ρ' over Z, call it ρ^*, that satisfies $\rho^*(Z) = *$.

We claim that

$$\Pr[\ \rho(Z) = * \mid F|_\rho \equiv 1, G_1|_\rho \not\equiv 1, \rho|_{Z^c} = \rho'\ ] \;\le\; q^{|Z|}.$$

The event $\rho(Z) = *$ refers to the unique ρ^*, under the condition $\rho|_{Z^c} = \rho'$. If $F|_{\rho^*} \not\equiv 1$, then the above conditional probability is 0 and the claim trivially holds. Otherwise, $F|_\rho \equiv 1$ for all extensions ρ of ρ' to Z. Hence $F|_\rho \equiv 1, G_1|_\rho \not\equiv 1, \rho|_{Z^c} = \rho'$ refers to exactly $2^{|Z|}$ assignments ρ, such that $\rho(i) \in \{0, *\}$ for all $i \in Z$. The claim follows. Lemma 7 is proved. □

If we take $p = \frac{1}{10t}$, then we get the following bound: For any G as in Lemma 7, and for all $\Delta \ge 0$,

$$\Pr[\ \mathrm{DC}(G|_\rho) \ge \Delta\ ] \;\le\; 2^{-\Delta}. \tag{11}$$

Using this bound as the base case, we can inductively prove Lemma 4.

On the other hand, it is possible to obtain a slightly stronger bound from (9). In fact the use of the inclusion-exclusion formula has been ignored in (10). In the following, we will show this slightly stronger bound.

We will set $q = \beta/t$ for some constant $\beta > 0$, to be determined later. Set

$$\alpha \;=\; \beta \Big/ \ln\left[\frac{1+\sqrt{1+4e^\beta}}{2}\right].$$

Then

$$e^{2\beta/\alpha} - e^{\beta/\alpha} \;=\; e^{\beta}.$$

It follows that

$$\begin{aligned}
&\left[1 - q + \frac{2q}{\alpha}\right]^t - \left[1 - q + \frac{q}{\alpha}\right] \\
=\;& \left[1 + \left(\frac{2\beta}{\alpha} - \beta\right)\frac{1}{t}\right]^t - \left[1 + \left(\frac{\beta}{\alpha} - \beta\right)\frac{1}{t}\right]^t \\
<\;& e^{\frac{2\beta}{\alpha} - \beta} - e^{\frac{\beta}{\alpha} - \beta} \;=\; 1.
\end{aligned}$$

Replacing the analysis after (9) in the above proof, we obtain the following lemma.

Lemma 8 *Let G be a t-And-Or formula $G_1 \wedge G_2 \wedge \ldots \wedge G_w$. For any β, $0 < \beta < t$, let ρ be a random p-restriction, where $p = \frac{\beta}{t-\beta}$, and let $\alpha = \beta / \ln\left[\frac{1+\sqrt{1+4e^{\beta}}}{2}\right]$. Then for all $\Delta \geq 0$, we have*

$$\Pr[\; \mathrm{DC}(G|_{\rho}) \geq \Delta \;] \;\leq\; \alpha^{\Delta}.$$

Minimizing α we find at $\beta_0 = 0.227537$, $\alpha_0 = \alpha(\beta_0) \approx 2^{-1.2638031} \approx 0.4164447$. Let $\gamma_0 = \beta_0/2 \approx 0.1137685$. Then we have the following bound. This is a strengthening of (11).

Lemma 9 *Let G be a t-And-Or formula $G_1 \wedge G_2 \wedge \ldots \wedge G_w$, and let ρ is a random γ_0/t-restriction. Then for all $\Delta \geq 0$, we have*

$$\Pr[\; \mathrm{DC}(G|_{\rho}) \geq \Delta \;] \;\leq\; \alpha_0^{\Delta}.$$

Proof. Let $q = \beta_0/t$ and $p = \frac{q}{2-q}$. Then $q = \frac{2p}{1+p}$ is the probability of getting a 0 or a $*$ in a random p-restriction.

We have shown that

$$\Pr[\; \mathrm{DC}(G|_{\rho'}) \geq \Delta \;] \;\leq\; \alpha_0^{\Delta},$$

where ρ' is a random p-restriction.

Since $p > q/2 = \gamma_0/t$, a random γ_0/t-restriction ρ can be realized by first applying a random p-restriction ρ', followed by a $\gamma_0/(pt)$-restriction. Note that if $\mathrm{DC}(G|_{\rho'}) < \Delta$ then $\mathrm{DC}(G|_{\rho}) < \Delta$. The Lemma follows. $\Box$

Now consider general constant depth circuits. Denote by $C^d(s,t)$ the class of depth d circuits with bfi[1] $\leq t$, and the number of gates above the first level $\leq s$. Denote by $C^d(s)$ the class of depth d circuits without a bfi condition but with total size $\leq s$. By extending one level with fan-in 1, clearly $C^d(s) = C^{d+1}(s,1)$. (Here in this notation we suppress the number n of variables and the depth d, where s and t are understood to be functions of one or both of them.)

[1] *bfi* is the abbreviation of bottom fan-in, the maximum fan-in of the bottom level gates. By a "bfi condition" we mean a bound of the form bfi $\leq t$ that is given in each context.

Lemma 10 *For all $C \in C^d(s, \gamma_0 n^{1/d})$, we have*

$$\Pr[\ \mathrm{DC}(G|_\rho) \geq \gamma_0 n^{1/d}\ ] \ \leq\ s \cdot \alpha_0^{\gamma_0 n^{1/d}} \ \approx\ s \cdot 2^{-0.143781 \cdot n^{1/d}},$$

where ρ is a random $1/n^{\frac{d-1}{d}}$-restriction.

Proof. Apply Lemma 9 repeatedly $d-1$ times, each time with a random $1/n^{\frac{1}{d}}$-restriction. Note that any function with decision tree depth $\leq \Delta$ can be expressed both as a Δ-And-Or as well as a Δ-Or-And. After switching bottom level And-Or formulas to Or-And's, or vice versa, one can merge two successive levels of gates and reduce the depth by 1. Then the lemma follows. $\square$

Let $C \in C^d(s)$ with no bfi requirement. By considering $C \in C^{d+1}(s, 1)$ we may first apply Lemma 9 to each of the bottom depth 2 subcircuit with bfi 1, with a random γ_0-restriction. But we can actually do slightly better by looking at it directly.

Fix any 1-Or-And formula S. (The case with any 1-And-Or is dual.) S is just a simple Or, by renaming variables, we may assume $S = \bigvee_{i=1}^{m} x_i$. Fix any $\Delta > 0$. If we apply a random p-restriction ρ, and if ρ assigns any $x_i = 1$, or if ρ assigns all x_i to 0 or $*$ but fewer than Δ of them are assigned $*$, then $\mathrm{DC}(S|_\rho) < \Delta$. Thus

$$\begin{aligned}\Pr[\ \mathrm{DC}(S|_\rho) \geq \Delta\ ] \ &\leq \sum_{\substack{J \subseteq \{1,\ldots,m\} \\ |J| \geq \Delta}} \Pr[\ \rho(J) = *, \rho(J^c) = 0\ ] \\ &= \left(\frac{1+p}{2}\right)^m \sum_{j=\Delta}^{m} \binom{m}{j} q^j (1-q)^{m-j},\end{aligned}$$

where $\Pr[\,\rho(i) \neq 1\,] = p + \frac{1-p}{2} = \frac{1+p}{2}$, and $q = \Pr[\,\rho(i) = * \mid \rho(i) \neq 1\,] = \frac{2p}{1+p}$. Hence

$$\Pr[\ \mathrm{DC}(S|_\rho) \geq \Delta\ ] \ \leq\ q^\Delta \left(\frac{1+p}{2}\right)^m \sum_{i=0}^{m} \binom{m}{i} (1-q)^{m-i} \ =\ q^\Delta \ <\ (2p)^\Delta.$$

So if we first apply a random restriction with $p = \frac{\alpha_0}{2} \approx 0.2082223$, with probability $> 1 - s_1 \alpha_0^{\gamma_0 n^{1/d}}$, all bottom level 1-Or-And subcircuits are switched to $\gamma_0 n^{1/d}$-And-Or (or all 1-And-Or switched to $\gamma_0 n^{1/d}$-Or-And), where s_1 is the total number of level 1 gates in the depth d circuit C, which are the depth 2 gates in the depth $d+1$ circuit with bfi 1. After the switching we get a circuit of depth $d+1$ with bfi $\leq \gamma_0 n^{1/d}$, but with the same type of gates on the 2 levels just above the bottom level gates. After merging these two levels, we get a circuit in $C^d(s', \gamma_0 n^{1/d})$, where $s' = s - s_1$. Now we apply Lemma 10. This gives the following bound.

Lemma 11 *For all $C \in C^d(s)$, we have*

$$\Pr[\ \mathrm{DC}(G|_\rho) \geq \gamma_0 n^{1/d}\ ] \ <\ s \cdot \alpha_0^{\gamma_0 n^{1/d}} \ \approx\ s \cdot 2^{-0.143781 \cdot n^{1/d}},$$

where ρ is a random $\alpha_0/(2n^{\frac{d-1}{d}})$-restriction.

These results can be used to prove circuit lower bounds for such circuits. Consider any circuit C in $C^d(s, \gamma_0 n^{1/(d-1)})$. Apply $d-2$ rounds of random $1/n^{1/(d-1)}$-restrictions, with probability

$> 1 - s \cdot 2^{-0.143781 \cdot n^{1/(d-1)}}$, we get a circuit in $C^2(1, \gamma_0 n^{1/(d-1)})$ after switching and merging. The number of variables N left has expectation $\mathrm{Exp}[N] = n^{1/(d-1)}$. By Chernoff bound, we have

$$\Pr[\ N \le \gamma_0 n^{\frac{1}{d-1}}\ ] \ = \ \Pr[\ N - n^{\frac{1}{d-1}} \le -(1-\gamma_0) n^{\frac{1}{d-1}}\ ] \ < \ e^{-\frac{(1-\gamma_0)^2}{2} \cdot n^{\frac{1}{d-1}}} \ < \ e^{-0.3927 n^{\frac{1}{d-1}}}.$$

Hence, if $s < 2^{0.143781 \cdot n^{1/(d-1)}}$, the probability is approaching 1 that both C is reduced to a circuit in $C^2(1, \gamma_0 n^{1/(d-1)})$ and $N > \gamma_0 n^{1/(d-1)}$. Therefore C does not compute the parity.

Lemma 12 *For all $C \in C^d(s, \gamma_0 n^{1/(d-1)})$, if C computes the parity function, then its size s must satisfy*

$$s \ \ge \ 2^{0.143781 \cdot n^{1/(d-1)}}.$$

Let $C \in C^d(s)$ with no bfi requirements. As in the proof of Lemma 11 we will separate the bottom level gates from the rest. Thus we first apply a random $\alpha_0/2$-restriction followed by $d-2$ rounds of random $(n^{1/(d-1)})^{-1}$-restrictions. Thus altogether we applied a random $\alpha_0 \cdot (2n^{\frac{d-2}{d-1}})^{-1}$-restriction, and with probability $> 1 - (s-1) \cdot 2^{-0.143781 \cdot n^{1/(d-1)}}$ we end up with a circuit in $C^2(1, \gamma_0 n^{1/(d-1)})$. By Chernoff bound again, if N is the number of variables left, then $\mathrm{Exp}[N] = \alpha_0 n^{1/(d-1)}/2$, and therefore

$$\Pr[\ N \le \gamma_0 n^{\frac{1}{d-1}}\ ] \ = \ \Pr\left[\ N - \frac{\alpha_0 n^{1/(d-1)}}{2} \le -\left(\frac{\alpha_0}{2} - \gamma_0\right) n^{\frac{1}{d-1}}\ \right] \ < \ e^{-0.021423 n^{\frac{1}{d-1}}}.$$

Hence, if $s < 2^{0.143781 \cdot n^{1/(d-1)}}$, C does not compute the parity.

Theorem 13 *For all $C \in C^d(s)$, if C computes the parity function, then its size s must satisfy*

$$s \ \ge \ 2^{0.143781 \cdot n^{1/(d-1)}}.$$

Now we consider the inapproximability type lower bound. The decision tree depth lower bound is ideally suited for deriving the inapproximability type lower bound, and the decision tree perspective was introduced precisely for this reason.

Denote for the rest of this section $m = n^{1/d}$. Let C be a depth d circuit. Note that after some restriction ρ, if C is reduced to a decision tree of depth smaller than the number of variables left, then for exactly half of the 0-1 extensions of ρ, C agrees with parity. This is because at every leaf of the decision tree, the circuit C is completely determined. (This property was called *monochromaticity* in [Cai86].)

Consider $\Pr[\ C(x_1, \ldots, x_n) = \oplus(x_1, \ldots, x_n)\ ]$, where $\oplus(x_1, \ldots, x_n)$ denotes the parity function, and the probability is over all 2^n assignments. This can be evaluated by first assigning any random restriction, followed by an unbiased 0-1 assignments for all the remaining variables. Let $E = E_1 \wedge E_2$, where E_1 denotes the event that after the random restriction, we end up with a decision tree of depth t, and E_2 denotes the event that the number of variables N assigned to $*$ is more than t, where t will be specified later as $O(m)$. Let $[C = \oplus]$ denote $[\,C(x_1, \ldots, x_n) = \oplus(x_1, \ldots, x_n)\,]$ as a short hand.

Note first

$$
\begin{aligned}
\Pr[\, C = \oplus \,] &= \Pr[E] \cdot \Pr[\, C = \oplus \,|\, E \,] + \Pr[\neg E] \cdot \Pr[\, C = \oplus \,|\, \neg E \,] \\
&= \Pr[\, C = \oplus \,|\, E \,] + \Pr[\neg E] (\, \Pr[\, C = \oplus \,|\, \neg E \,] - \Pr[\, C = \oplus \,|\, E \,]\,).
\end{aligned}
$$

As we noted $\Pr[\, C = \oplus \,|\, E \,] = 1/2$. Then

$$
\left| \Pr[\, C = \oplus \,] - \frac{1}{2} \right| \;\leq\; \frac{1}{2} \Pr[\neg E].
$$

Also since $\Pr[\, C = \oplus \,] - \Pr[\, C \neq \oplus \,] = 2(\Pr[\, C = \oplus \,] - \frac{1}{2})$, we have

$$
|\Pr[\, C = \oplus \,] - \Pr[\, C \neq \oplus \,]| \;\leq\; \Pr[\neg E].
$$

Now we specify the parameter of the random restrictions.

First consider any $C \in C^d(s, \gamma_0 m)$. Then let $t = \gamma_0 m$, and we apply Lemma 10. With a random $(n^{\frac{d-1}{d}})^{-1}$-restriction, we have

$$
\Pr[\neg E_1] \;\leq\; s\alpha_0^t \;\approx\; s \cdot 2^{-0.143781 \cdot m}.
$$

Then again by using the Chernoff bound, we estimate $\Pr[\neg E_2] = \Pr[\, N \leq \gamma_0 m \,]$ as follows.

$$
\Pr[\neg E_2] \;\leq\; e^{-\frac{(1-\gamma_0)^2}{2} m} \;<\; e^{-0.3927 m}.
$$

Thus $\Pr[\neg E_2]$ is dominated by $\Pr[\neg E_1]$. This analysis gives the following bound.

Lemma 14 *For all $C \in C^d(2^{0.07189 n^{1/d}}, \gamma_0 n^{1/d})$, we have*

$$
|\Pr[\, C = \oplus \,] - \Pr[\, C \neq \oplus \,]| \;\leq\; 2^{-0.07189 n^{1/d}}.
$$

Finally we consider $C \in C^d(s)$ with no bfi condition. This time we have to work harder to optimize the exponents. Our strategy is as follows. We will first assign a $\frac{\alpha_0}{2}$-restriction, and this will give us a depth d circuit *with* bfi $\leq \gamma_0 m$. Then we assign $d-2$ rounds of $1/m$-restrictions, each time using Lemma 9 with the same parameters $p = 1/m$, and $t = \Delta = \gamma_0 m$. This will give us a depth 2 circuit *with* bfi $\leq \gamma_0 m$. So far the failure probability is $(s-1)\alpha_0^{\gamma_0 m}$. Finally we assign another $1/m$-restriction, but this time use the parameters $t = \gamma_0 m$ and $\Delta = x\gamma_0 m$, where $0 < x < 1$ is to be determined later. The overall failure probability is $< s\alpha_0^{\gamma_0 m} + \alpha_0^{\Delta} + \Pr[\, N \leq \Delta \,]$, where N is the number of variables left.

It turns out that if we used the same values for t and Δ for the estimate in the last round, the bound for $\Pr[\, N \leq \Delta \,]$ would be too weak. We will use a more exacting form of the Chernoff bound, and then optimize the overall bound by balancing the last two terms with a judicious choice of x.

We use the following version of the Chernoff bound. (See, for example, p. 70 of [MR95].)

$$
\Pr[\, N < (1-\delta)\mathrm{Exp}[N] \,] \;<\; \exp\left[-\mathrm{Exp}[N] \cdot (\delta + (1-\delta) \cdot \ln(1-\delta)) \right].
$$

Here we have

$$\mathrm{Exp}[N] \; = \; \frac{\alpha_0}{2}m, \;\; \text{and} \;\; \delta \; = \; 1 - \frac{2\gamma_0 x}{\alpha_0}.$$

We balance the two bounds by setting x such that

$$x\gamma_0 \ln \frac{1}{\alpha_0} \; = \; \frac{\alpha_0}{2}\left[\, \delta + (1-\delta)\cdot \ln(1-\delta) \,\right].$$

This leads to choosing $x = 0.617945$, and we get both

$$\alpha_0^{\Delta} \; < \; 2^{-0.0888488 \cdot m} \;\; \text{and} \;\; \Pr[\, N \le \Delta \,] \; < \; 2^{-0.0888488 \cdot m}.$$

Then by setting $s = 2^{0.07189 \cdot m}$ we get a balanced discrepancy lower bound.

Theorem 15 *For all $C \in C^d(2^{0.07189 n^{1/d}})$, we have*

$$|\Pr[\, C = \oplus \,] - \Pr[\, C \neq \oplus \,]| \;\; \le \;\; 2^{-0.07189 n^{1/d}}.$$

Note that the bound in Theorem 15 is the same as that of lemma 14 with the bfi condition. Lemma 6 follows from Theorem 15 for large input size.

Remark: The original motivation for Furst-Saxe-Sipser [FSS81] where super polynomial lower bounds were proved for parity against constant depth circuits, was to provide an oracle separation of PH and PSPACE. This was achieved in a breakthrough result by Yao [Yao85] who proved a lower bound of the form $2^{n^{\Omega(1/d)}}$ for parity on n bits for depth d circuits. Yao's bound was further improved by Håstad[Hås86a] from $2^{-n^{\Omega(1/d)}}$ 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 investigated in [Cai86] whether constant depth circuits of size $2^{n^{\Omega(1/d)}}$ must err on an asymptotically 50 % of inputs against parity. This was motivated by another long standing open problem, that of random oracle separation of PH and PSPACE (see also [Bab87]). To attack this problem, the decision tree point of view was first adopted in [Cai86], although a different but completely synonymous terminology (Master-Player Game and t-monochromaticity) was used. It was proved in [Cai86] that after a suitable random restriction ρ, with high probability, the constant depth circuit $C \mid_\rho$ has decision tree depth smaller than the number of unassigned Boolean variables. In such cases, $\Pr[C = \oplus]$ is exactly $\frac{1}{2}$. Thus the discrepancy

$$|\Pr[C = \oplus] - \Pr[C \neq \oplus]| \tag{12}$$

was shown to be $o(1)$ for circuits of depth d and size $2^{n^{\Omega(1/d)}}$. Implicitly a bound of the form $2^{-n^{\Omega(1/d)}}$ for the discrepancy (12) was proved there as well [Cai86]. The $o(1)$ upper bound for the discrepancy was sufficient for the random oracle separation result which was the purpose of [Cai86], but one needs Håstad's technique to improve the bound from $2^{-n^{\Omega(1/d)}}$ to $2^{-cn^{\frac{1}{d}}}$ as in Lemma 15. However, the weaker bound $2^{-n^{\Omega(1/d)}}$ would have sufficed for our Theorem 2.

It was a marvelous application by Nisan and Wigderson [Nis91a, Nis91b, NW88] who turned this inapproximability type of lower bounds based on decision trees on its head, and produced an explicit construction—usually considered an upper bound—of a pseudorandom generator provably indistinguishable from true random bits by polynomial size constant depth circuits. A central ingredient in [Nis91a, Nis91b, NW88] is a suitable combinatorial design. Seen in this way, our proof of Theorem 2 can be viewed as using a *lower bound* (Switching Lemma), to get an *upper bound* (the NW pseudorandom generator), to prove a *lower bound* (to kill all 2^n circuits C_x simultaneously with the pseudorandom assignments), to finally prove an *upper bound* (to be able to code all the computations). And all this, is to show that it is impossible to prove super polynomial circuit *lower bound* for any fixed language in Polynomial-time Hierarchy, with a relativizable proof with stringent access to an oracle.

References

[Ajt83] M. Ajtai, Σ^1_1-formulae on finite structures, *Ann. Pure Applied Logic*, 24, 1–48, 1983.

[Bab87] L. Babai, Random oracles separate PSPACE from the polynomial-time hierarchy, *Information Processing Letters*, 26(1): 51-53 (1987).

[BCGKT] N. Bshouty, R. Cleve, R. Gavaldà, S. Kannan, and C. Tamon, Oracles and queries that are sufficient for exact learning, *J. Comput. and System Sci.* 52(3), 421–433, 1996.

[BFT98] H. Buhrman, L. Fortnow, and T. Thierauf, Nonrelativizing separations, in *Proc. the 13th IEEE Conference on Computational Complexity* (CCC'98), IEEE, 8–12, 1998.

[Cai86] J-Y. Cai, With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy, in *Proc. 18th ACM Symposium on Theory of Computing* (STOC'86), ACM, 21–29, 1986. See also the journal version appeared in *J. Comput. and System Sci.* 38(1): 68-85 (1989).

[Cai01] J-Y. Cai, $\mathrm{S}_2^{\mathrm{P}} \subseteq \mathrm{ZPP}^{\mathrm{NP}}$, in *Proc. 42th IEEE Symposium on Foundations of Computer Science* (FOCS'01), IEEE, 620–628, 2001.

[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.

[Hås86a] 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.

[Hås86b] J. Håstad, *Computational Limitations for Small-Dept Circuits*, MIT Press, 1986.

[HPV77] J.E. Hopcroft, W.J. Paul, and L.G. Valiant, On time versus space, *J. ACM* 24(2), 332-337, 1977.

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

[KL80] R.M. Karp and R.J. Lipton, Some connections between nonuniform and uniform complexity classes, in *Proc. 12th ACM Symposium on Theory of Computing* (STOC'80), ACM, 302–309, 1980.

[KW98] J. Köbler and O. Watanabe, New collapse consequences of NP having small circuits, *SIAM J. Comput.*, 28, 311–324, 1998.

[MVW99] P.B. Miltersen, N.V. Vinodchandran, and O. Watanabe, Super-Polynomial versus half-exponential circuit size in the exponential hierarchy, in *Proc. 5th Annual International Conference on Computing and Combinatorics* (COCOON'99), Lecture Notes in Computer Science 1627, 210–220, 1999.

[MR95] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge Univ. Press, 1995.

[Nis91a] N. Nisan, Pseudorandom bits for constant depth circuits, *Combinatorica* 11(1), 63–70, 1991.

[Nis91b] N. Nisan, *Using Hard Problems to Create Pseudorandom Generators*, MIT Press, 1991.

[NW88] N. Nisan and A. Wigderson, Hardness vs randomness, in *Proc. 29th IEEE Symposium on Foundations of Computer Science* (FOCS'88), IEEE, 2–12, 1988.

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

[vL91] J. van Lint, *Introduction to Coding Theory*, Springer-Verlag, 1991.

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