On the Derandomization of Constant Depth Circuits

Adam R. Klivans*
Laboratory for Computer Science
MIT
Cambridge, MA 02139
klivans@theory.lcs.mit.edu

Abstract. Nisan [18] and Nisan and Wigderson [19] have constructed a pseudo-random generator which fools any family of polynomial-size constant depth circuits. At the core of their construction is the result due to Håstad [10] that no circuit of depth d and size $2^{n^{1/d}}$ can even weakly approximate (to within an inverse exponential factor) the parity function. We give a simpler proof of the inapproximability of parity by constant depth circuits which does not use the Håstad Switching Lemma. Our proof uses a well-known hardness amplification technique from derandomization: the XOR lemma. This appears to be the first use of the XOR lemma to prove an unconditional inapproximability result for an explicit function (in this case parity). In addition, we prove that BPAC_0 can be simulated by uniform quasipolynomial size constant depth circuits, improving on results due to Nisan [18] and Nisan and Wigderson [19].

1 Introduction

1.1 Background

The explicit derandomization of randomized complexity classes remains a central challenge in complexity theory. Considerable interest has of late been given to the seminal work of Nisan [18] and Nisan and Wigderson [19] in which the authors construct a pseudo-random generator powerful enough to derandomize well known complexity classes like BPP under the assumption that a boolean function exists with high average-case circuit complexity. Further work has shown how to amplify the worst-case circuit complexity of a boolean function to achieve the necessary hardness required by the Nisan-Wigderson generator [5, 11, 13, 25]. Depending on the assumption taken, the Nisan-Wigderson generator (or its strengthened relative the Impagliazzo-Wigderson generator) can be used to derandomize a wide range of expressive complexity classes [14]. Still, it seems that the most expressive randomized complexity class we can unconditionally or explicitly derandomize is uniform RAC_0, randomized constant depth polynomial size uniform circuit families [19]. In fact, the case of RAC_0 remains one of the

* Supported in part by NSF grant CCR-97-01304.

very few examples where the Nisan-Wigderson generator can be used to obtain an unconditional derandomization (for another see [17]).

Roughly speaking, the reason we can derandomize RAC_0 is that we can prove strong lower bounds against constant depth circuits. The celebrated results of [1, 9, 10, 21, 23] show that parity cannot be computed by polynomial size constant depth circuits. This fact alone, however, is not sufficient to build a pseudo-random generator useful for derandomizing RAC_0. The Nisan-Wigderson generator requires the existence of a function highly *inapproximable* by constant depth circuits, and only the lower bound techniques due to Håstad [10] yield strong enough results for this type of derandomization. Nisan and Wigderson use the results of Håstad to prove that RAC_0 can be simulated in quasipolynomial time.

1.2 Our Results

In this paper, we give a new proof of the inapproximability of parity by constant depth circuits via a well known hardness amplification technique: the XOR lemma. The advantage of our proof is two-fold. First, it is a simpler proof than the one found in [10] and does not use the Håstad Switching Lemma, a sophisticated combinatorial tool for describing the effect of random restrictions on constant depth circuits. Second, it appears to be the first time the XOR lemma has been used to prove an unconditional inapproximability result for an explicit function (in this case parity). Although our results give a slightly weaker inapproximability result than the optimal ones found in [10], our lower bounds are strong enough to obtain the same qualitative derandomization, namely a quasipolynomial time simulation of RAC_0:

Theorem 1. *Let C be a circuit of depth $d-1$ and size $2^{n^{1/6d}}$. Then C cannot compute parity correctly on more than a $1/2 + 2^{-n^{1/6d}}$ fraction of inputs.*

Further, we will show that BPAC_0 (and RAC_0) has uniform quasipolynomial size constant depth circuit families, improving on results due to Nisan [18] and Nisan and Wigderson [19] who showed that BPAC_0 is contained in quasipolynomial *time*.

1.3 Our Approach

Our starting point is the voting polynomials paper of Aspnes et al. [4] which proves that any circuit of depth d computing parity on any constant fraction of inputs has size $2^{\Omega(n^{\frac{1}{4d}})}$. Their result uses only elementary linear algebra and the fact that constant depth circuits can be approximated by low degree polynomials. In addition, their result holds even if the circuit has a majority gate at the root. We wish to amplify this hardness from a constant to $1/2 - 2^{-n^{1/O(d)}}$. What we prove is that the XOR of several independent copies of parity increases in hardness exponentially with respect to the number of copies taken. To prove such a result, we appeal to the hard-core set construction of Impagliazzo [11] which gives a simple proof of an XOR lemma and works with respect to any model of computation closed under majority.

At first glance, this may seem useless for our case, as it is known that AC_0 circuits cannot compute the majority function [10]. Our main observation is that a lower bound against AC_0 with one majority gate (for example the one found in [4]) gives us enough leeway to carry out the XOR lemma due to Impagliazzo [11]. For clarity of exposition we view Impagliazzo's result from a *boosting* perspective [15] (although details about particular boosting algorithms are not important here).

Boosting is a technique from computational learning theory (see [15]) which combines small circuits that weakly approximate a boolean function f (in our case f is the parity function) to form a circuit that computes f on almost all inputs. The simple boosting algorithm implicit in Impagliazzo's construction [11, 15] combines weakly approximating circuits by taking their majority to form a circuit computing f almost everywhere. From the voting polynomials result, however, we know that circuits with one majority gate at the root cannot compute the parity function. Hence no small circuit can even weakly approximate parity. This observation leads to the existence of a hard-core set, a set of inputs S such that no small circuit can even weakly approximate parity on a randomly chosen input from S. An XOR lemma follows almost immediately, and, since the XOR of several independent copies of parity is simply equal to the parity function on a larger input length, we obtain our desired inapproximability result.

1.4 Comparison with Previous Results

In [10], Håstad gives an optimal result for the inapproximability of parity:

Theorem 2. *[10] Let ϕ be the parity function on inputs of length n. Let C be a circuit of size 2^s and depth $d > 2$ such that $s \geqslant n^{1/d}$. Then $Pr_{x \in \{0,1\}^n}[C(x) = \phi(x)] \leqslant 1/2 + 2^{-\Omega(\frac{n}{s^{d-1}})}$. If $s < n^{1/d}$ then $Pr_{x \in \{0,1\}^n}[C(x) = \phi(x)] \leqslant 1/2 + 2^{-\Omega((\frac{n}{s})^{\frac{1}{d-1}})}$.*

Corollary 1. *[10] Let C be a circuit of size $2^{n^{1/d}}$ and depth d. Then C cannot compute parity correctly on more than a $1/2 + 2^{-n^{1/d+1}}$ fraction of inputs.*

1.5 Organization

We begin by reviewing the "voting polynomials" result of [4]. We then review the Impagliazzo construction from [11] and describe how the "voting polynomials" result implies it can be used in the constant depth circuit setting. In Section 5 we show how to improve known containments of BPAC_0. Section 5 is independent from the other sections in this paper.

2 Preliminaries

For basic definitions regarding circuits and standard complexity classes we refer the reader to [20]. For a definition of a pseudorandom generator see [25].

Definition 1. *A circuit family* $\{C_n\}$ *is* uniform *if there exists a Turing Machine* M *such that* M *on input* 1^n *outputs* C_n *in polynomial time in* n*. A language* L *is in* RAC_0 *if there exists a uniform family of polynomial-size, constant depth circuits* $\{C_n\}$ *where each circuit* C_n *is a circuit taking two inputs* x *and* r *(*$|r| = s = n^k$ *for some* k*) such that if* $x \in L$, $Pr_{r\in\{0,1\}^s}[C(x,r) = 1] \geqslant 1/2$ *and if* $x \notin L$, $Pr_{r\in\{0,1\}^s}[C(x,r) = 1] = 0$*. A language* L *is in* BPAC_0 *if the probabilities* $1/2$ *and* 0 *are replaced with* $2/3$ *and* $1/3$*.*

Definition 2. *Let* f *be a Boolean function on* $\{0,1\}^n$ *and* $\mathcal{D}$ *a distribution on* $\{0,1\}^n$*. Let* $0 < \epsilon < 1/2$ *and let* $n \leqslant g \leqslant 2^n/n$*. We say that* f *is* δ-hard for size g and depth d under $\mathcal{D}$ *if for any boolean circuit* C *of depth* d *with at most* g *gates, we have* $\Pr_{\mathcal{D}}[f(x) = C(x)] \leqslant 1 - \delta$*. We say that* f *is* ϵ-hard-core for size g and depth d under $\mathcal{D}$ *if for any boolean circuit* C *of depth* d *with at most* g *gates, we have* $\Pr_{\mathcal{D}}[f(x) = C(x)] \leqslant 1/2 + \epsilon$*. For a set of inputs* S *we say that* f *is* δ-hard on S for size g and depth d *if* f *is* δ-hard *for size* g *and depth* d *with respect to the uniform distribution over* S *(a similar definition applies to* f *being* ϵ-hard-core *on* S*). If* S *is equal to the set of all inputs then we omit* S*, and if there are no depth restrictions we omit* d*.*

Throughout the paper we use $\mathcal{U}$ to denote the uniform distribution on $\{0,1\}^n$.

Definition 3. *For a distribution* $\mathcal{D}$*, the* minimum entropy *of* $\mathcal{D}$ *is equal to* $\log(L_\infty(\mathcal{D})^{-1})$*.*

3 Mild Hardness via Voting Polynomials

In this section we sketch the elegant "voting polynomials" result due to [4] which shows that parity cannot be approximated to within any constant by a depth d circuit of size $2^{o(n^{\frac{1}{4d}})}$ with one majority gate at the root. At the heart of their result is a simple proof that parity cannot be sign represented well by any low degree polynomial. This idea stems from the now well known *approximation method* originally pioneered by Razborov [21] to give a novel proof that parity is not in AC_0. Smolensky [23] extended Razborov's original idea and showed that since polynomials cannot compute parity over finite fields, the lower bound result holds for constant depth circuits with a mod p gate for any prime p. Our interpretation of the voting polynomials result is that if we work with polynomials over the rationals, instead of finite fields, then we obtain a lower bound for parity against circuits with a *majority* gate, rather than mod p gates, since one can easily embed majority within the rationals.[1] The addition

[1] Beigel [7] has shown how to improve this from one majority gate to $n^{o(1/d)}$ majority gates. His result is tight.

of the majority gate will be essential, as it enables us to carry out the necessary hardness amplification reductions. We begin with some definitions:

Definition 4. *Let f a boolean function on n variables be represented by a function mapping $\{-1,1\}^n$ to $\{-1,1\}$. Notice that in this representation, the parity function on n variables is simply equal to the monomial $x_1,\ldots,x_n$. A rational polynomial $p(x_1,\ldots,x_n)$* weakly represents *f if p is nonzero and for every x such that $p(x)$ is nonzero, $sgn(f(x)) = sgn(p(x))$. The* weak degree *of a function f is the degree of the lowest degree polynomial which weakly represents f. We sometimes refer to a polynomial which weakly represents f as a* voting polynomial *for f.*

Lemma 1. *The weak degree of the parity function on n variables is n.*

Proof. Let ϕ denote the parity function on n variables. Assume that there exists a polynomial p of degree less than n which weakly represents parity. Consider the vector space of rational polynomials in n variables of total degree at most n equipped with the inner product $(f,g) = \sum_{x\in\{-1,1\}^n} f(x)g(x)$. The set of all monomials of degree less than or equal to n forms a basis. Notice that for any monomial m of degree strictly less than n, m does not contain some variable x_i and so (m,ϕ) must equal 0. In other words, ϕ is orthogonal to all monomials of degree less than n. But, since p weakly represents parity, (p,ϕ) must be strictly greater than 0. Hence parity has weak degree n.

Having established this, we can further prove that no low degree polynomial can *approximate* parity well:

Theorem 3. *[4] Let p be a nonzero polynomial of degree d. Let S be the set of inputs for which $sgn(x) \neq \phi(x)$. Then*

$$|S| \geqslant \sum_{i=1}^{\frac{n-d-1}{2}} \binom{n}{i}$$

Proof. Assume $|S|$ is smaller than the above bound. Let q be the degree $\frac{n-d-1}{2}$ polynomial such that for any $x \in S$, $q(x) = 0$. Notice that any polynomial of degree $\frac{n-d-1}{2}$ has, potentially, $\sum_{i=1}^{\frac{n-d-1}{2}} \binom{n}{i}$ nonzero monomials (or degrees of freedom). Thus q is a well defined, non-zero polynomial whose coefficients can be computed by solving an appropriate system of linear equations. Now the polynomial q^2p weakly represents parity and is of degree at most $n-1$, contradicting Lemma 1.

Interestingly, the authors of [4] prove that this theorem is tight by giving a simple construction of a polynomial approximating parity which makes exactly the above number of errors.

Corollary 2. *Any polynomial which weakly represents parity and makes less than $\gamma 2^n$ errors for any $\gamma < 1/2$ must have degree $\Omega(\sqrt{n})$.*

The following theorem builds on work of Valiant and Vazirani [27] and states that constant depth circuits can be accurately approximated by low degree polynomials. The result we state here is succinctly proved in [4] and is a refinement of a theorem from [8].

Theorem 4. *[27] [8] [4] For any $\gamma > 0$ and any function f computed by a boolean circuit of size s and depth d there exists a polynomial of degree $O((\log(s/\gamma)\log(s))^d)$ which* computes *f for all but $\gamma 2^n$ inputs.*

Now we can obtain a similar theorem for a circuit of size s and depth d with one majority gate at the root. For each input to the majority gate (let us say without loss of generality there are s of them) there exists some low degree polynomial approximating that input. Since we are working over the rationals, taking the sum of all of these polynomials and subtracting $s/2$ results in a polynomial weakly representing the output of the majority gate (and thus the entire circuit) on almost every input. From this discussion and Theorem 4 we obtain the following:

Theorem 5. *[4] For any $\gamma > 0$ and any function f computed by a boolean circuit of size s and depth d with one majority gate at the root there exists a polynomial of degree $O((\log(s^2/\gamma)\log(s))^d)$ which* weakly represents *f for all but $\gamma 2^n$ of the inputs.*

Now we can combine the above theorem with Corollary 2:

Theorem 6. *Parity is $1/8$-hard for depth d circuits of size $2^{o(n^{1/4d})}$ with one majority gate at the root.*

4 Extreme Hardness via Hard-Core Sets

In this section we amplify the hardness from a constant to $1/2 + 2^{-n^{(1/O(d))}}$ via Impagliazzo's hard-core set construction [11]. We will show that if there exists a circuit approximating parity to within $1/2 + \epsilon$ (think of ϵ here as $2^{-n^{(1/O(d))}}$) then there exists a subexponential sized circuit with a majority gate at the root computing parity on a constant fraction of the inputs, contradicting Theorem 6.

To achieve this we would like to combine circuits with correctness $1/2 + \epsilon$ with respect to parity to form a slightly larger circuit computing parity on say a $7/8$ fraction of the inputs, i.e. *boost* our ϵ advantage.[2] The existence of these weakly approximating circuits will not be enough, however, to directly carry out such a reduction (this is because it is possible for every weakly approximating circuit to be wrong on the same set of inputs, in which case we cannot hope to boost our advantage). To obtain such a reduction, we will need to assume something stronger, namely that for any distribution with high *min-entropy*, there exists a small circuit approximating parity to within ϵ *with respect to that*

[2] Here again we frame this informal discussion of Impagliazzo's work in terms of boosting for clarity of exposition. The methodology of boosting is not relevant here.

distribution. As a consequence we will have that there exists a distribution D with large *min-entropy* such that no small circuit can weakly approximate parity with respect to D. From this it follows that there is a set of inputs S (of size proportional to the min-entropy of D) such that no small circuit can weakly approximate parity with respect to the uniform distribution on S. This set S is known as a *hard-core* set for parity. With S in hand, it is not hard to prove that computing the XOR of several independent copies of parity is a significantly harder function to approximate– with high probability, one of the inputs to the independent copies will fall within the hard-core set.

An important definition is that of a *measure* which we think of as an un-normalized distribution:

Definition 5. *A* measure *on $\{0,1\}^n$ is a function $M : \{0,1\}^n \to [0,1]$. The* absolute size *of a measure M is denoted by $|M|$ and equals $\sum_x M(x)$; the* relative size *of M is denoted $\mu(M)$ and equals $|M|/2^n$. The distribution $\mathcal{D}_M$ induced by a measure M is defined by $\mathcal{D}_M(x) = M(x)/|M|$.*

The relationship between measure size and min-entropy is established in [15]:

Fact 7 *For any measure M, the distribution induced by M has min-entropy at least $n - \log(1/\mu(M))$.*

4.1 Hard-Core Sets for Constant Depth Circuits

Here we prove that the hard-core set construction of Impagliazzo works with respect to constant depth circuits. We will need to assume initial hardness with respect to circuits with one majority gate at the root to obtain hard-core distributions:

Theorem 8. *[11] Let f be δ-hard for circuits of size g and depth d with one majority gate at the root with respect to $\mathcal{U}$ and let $0 < \epsilon < 1$. Then there is a measure M on $\{0,1\}^n$ with $\mu(M) \geqslant \delta$ such that f is ϵ-hard-core with respect to M (i.e. with respect to the distribution induced by M) for any circuit of size $g' = (1/4)\delta^2\epsilon^2 g$ and depth $d-1$ with one majority gate at the root.*

Proof. Consider the algorithm given in Figure 4.1. We will need the following lemma:

Lemma 2. *Algorithm* **IHA** *terminates after at most $i_0 = 2/\delta^2\epsilon^2$ iterations.*

We defer the proof of Lemma 2 to the appendix.

Once the algorithm terminates it is easy to see that $h \equiv MAJ(C_0, \ldots, C_{i-1})$ agrees with f on all inputs except those which have $N_i(x) \leqslant 0$ and hence $M_i(x) = 1$. Since $\mu(M_i) < \delta$, this implies that $\Pr_{\mathcal{U}}[f(x) = h(x)] \geqslant 1 - \mu(M_i) > 1 - \delta$. But since h is a majority circuit over at most i_0 circuits each of size at most g', and depth at most $d-1$, it follows that h is a depth d circuits with at most $g'i_0 + i_0 \leqslant g$ gates, which contradicts the original assumption that f is δ-hard for circuits of size g and depth d with one majority gate under $\mathcal{U}$.

Input: $\delta > 0$, $\epsilon > 0$, boolean function f
Output: a circuit h such that $\Pr_{\mathcal{U}}[h(x) = f(x)] \geqslant 1 - \delta$

set $i \leftarrow 0$
$M_0(x) \equiv 1$
until $\mu(M_i) < \delta$ **do**
  let C_i be a circuit of size at most g' and depth d with $\Pr_{\mathcal{D}_{M_i}}[C(x) = f(x)] \geqslant 1/2 + \epsilon$
  $R_{C_i}(x) \equiv 1$ if $f(x) = C_i(x)$, $R_{C_i}(x) \equiv -1$ otherwise
  $N_i(x) \equiv \sum_{0 \leqslant j \leqslant i} R_{C_j}(x)$
  $M_{i+1}(x) \equiv 0$ if $N_i(x) \geqslant 1/\delta\epsilon$, $M_{i+1}(x) \equiv 1$ if $N_i(x) \leqslant 0$,
    $M_{i+1}(x) \equiv 1 - \delta\epsilon N_i(x)$ otherwise
  set $i \leftarrow i + 1$
$h \equiv MAJ(C_0, C_1, \ldots, C_{i-1})$
return h

Fig. 1. The `IHA` algorithm.

Hard-core measures of relative size δ correspond to distributions with min-entropy at least $n - \log(1/\delta)$. Since distributions with min-entropy k are convex combinations of uniform distributions over sets of size 2^k ([26]), one would expect the existence of a hard-core measure of relative size δ to imply the existence of a set of inputs S of size $\delta 2^n$ such that f is hard-core with respect to the uniform distribution over S. The following fact confirms this intuition:

Fact 9 *[11] Let f be $\epsilon/2$-hard-core with respect to a measure M of relative size δ for circuits of size g (where $2n < g < (1/8)(2^n/n)(\epsilon\delta)^2$) and depth d. Then there exists a set of inputs S of size $\delta 2^n$ such that f is ϵ-hard-core on S for size g and depth d.*

We note that in [15] it is shown how to improve the circuit size parameter from Theorem 8 from $O(\delta^2\epsilon^2 g)$ to $O(\epsilon^2(\log(1/\delta))^{-1}g)$. In [22] it is shown how to generalize the above algorithm to work with real valued functions.

4.2 Amplifying Hardness with the XOR Lemma

Theorem 10. *(XOR-Lemma) [11] Let f be ϵ-hard-core for some set of $\delta 2^n$ inputs for circuits of size g and depth d, then the function $f'(x_1, \ldots, x_k) = f(x_1) \oplus \cdots \oplus f(x_k)$ is $\epsilon + (1-\delta)^k$-hard-core for circuits of size g and depth d.*

Proof. Let C be a circuit of size h obtaining advantage more than $\epsilon + (1-\delta)^k$. Let A_l be C's advantage given that l inputs lie in the hard-core set. Since the probability that all k inputs lie in the hard-core set is $(1-\delta)^k$, A_l must be greater than ϵ for some $l > 0$. Consider the circuit C' which randomly chooses $l-1$ instances from the hard-core set, $k-(l-1)$ from the complement of the hard-core and, on input x, is equal to $C(x)$ with all the inputs inserted in a random order (and is complemented if the XOR of f of the randomly chosen inputs is 1). Then this circuit C' has average advantage ϵ over the choice of random inputs, so we can fix the random choices non-uniformly to obtain a circuit computing f with advantage ϵ when the input is chosen uniformly from the hard-core, a contradiction.

Using the hard-core set construction and XOR lemma yields our main inapproximability result:

Theorem 11. *The parity function is* $2^{-n^{1/6d}}$-hard-core *for circuits of depth* $d-1$ *and size* $2^{n^{1/6d}}$.

Proof. Corollary 2 tells us that parity is 1/8-hard for circuits of size $2^{o(n^{1/4d})}$ and depth $d-1$ with one majority gate at the root. Now apply Theorem 8 with $\delta = 1/8$ and $\epsilon/2 = 2^{-n^{1/5d}}$ to prove that parity is ϵ-hard-core on a set of size $(1/8)2^n$ for circuits of depth $d-1$ and size $2^{n^{1/5d}}$. The above XOR lemma amplifies our hardness to what is claimed and increases the input length to $O(n^{1+1/5d})$.

Nisan [18] and Nisan and Wigderson [19] show how to use this inapproximability result to build a pseudorandom generator which fools AC_0 and has seed size polylogarithmic in n (more precisely $\log^{O(d)} n$ where d is the depth of the circuit). Our inapproximability result can be used to obtain a similar pseudorandom generator and the following corollary:

Corollary 3. *Uniform* BPAC_0 *is contained in quasipolynomial time.*

5 Approximate Majority and Uniform Circuit Constructions

From the results in the previous section, we would like to conclude that BPAC_0 has uniform, quasipolynomial-size, constant depth circuits (rather than containment in quasipolynomial time). This is immediately true for RAC_0, since it is a one-sided error class, and we accept an input x if and only if all there exists an output from a suitable pseudorandom generator G, call it r, such that $C(x,r) = 1$ where C is the circuit deciding the language. This can be implemented with an OR gate. For BPAC_0, we need to take a majority vote over all possible seeds given to G, and, as we have stated earlier, it is known that majority is not in AC_0. Fortunately, since we can construct strong pseudorandom generators, for every input the fraction of seeds making C accept (or reject) will always be very close to 2/3 (or 1/3). Thus, we only need to compute an *approximate* majority. Ajtai and Ben-Or [3] building on work of Stockmeyer [24] proved the existence of non-uniform constant depth circuit families capable of computing an approximate majority as long as the fraction of ones is always bounded away from 1/2 by at least an inverse polylogarithmic factor. In 1993, Ajtai [2] gave a *explicit* construction of such a circuit family:

Theorem 12. *[2] For any constant* $0 < \delta < 1/2$ *there exists a uniformly constructible polynomial-size, constant depth circuit family such that for any* $s \in \{0,1\}^n$*:*

– $|s| > (1/2 + \delta)n$ *implies* $C(s) = 1$
– $|s| < (1/2 - \delta)n$ *implies* $C(s) = 0$

(Ajtai proves that δ can be replaced with $1/(\log n)^i$ where the depth and size of the circuit will depend on i. This construction is optimal due to the lower bounds found in [10]).

With this construction we can obtain our desired inclusion for $\mathsf{BPAC_0}$:

Theorem 13. $\mathsf{BPAC_0}$ *can be computed by uniform, constant depth, quasipolynomial-size circuits.*

Proof. Let $L \in \mathsf{BPAC_0}$. Then L is computed by a uniform circuit family $\{C_n\}$ of depth d and size n^k for some constants d and k. Let G be a pseudorandom generator which fools $\{C_n\}$, has seed size $(\log n)^{O(d)}$, and approximates the acceptance probability of $\{C_n\}$ to an additive $O(1/n)$ factor. The uniform circuit family for L is obtained by enumerating over all seeds to G and obtaining $2^{(\log n)^{O(d)}}$ circuits where each circuit equals C_n with one of the outputs from G as its random bit sequence. From the approximate majority construction, there exists a circuit M of depth d' and size $p(2^{(\log n)^{O(d)}})$ for some polynomial p computing the approximate majority of any sequence of length $2^{(\log n)^{O(d)}}$ where the number of ones is bounded away from $1/2$. For any input x, the output sequence of all of the above circuits on input x has at least $2/3 - 1/n$ ones or at most $1/3+1/n$ ones. Combining the enumerated circuits as inputs to the approximate majority circuit yields the desired constant depth, quasipolynomial size circuit.

6 Conclusion

In this paper we have given a new proof of the inapproximability of parity by $\mathsf{AC_0}$ without using the Håstad switching lemma. Since the Håstad switching lemma has many important applications in such diverse areas as computational learning theory [16] and proof complexity (see [6]), we hope that the techniques presented in this paper can be used to provide simpler and/or improved results in these areas. In [11], Impagliazzo proves an XOR-lemma where the inputs are chosen pairwise independently. Is it possible to prove such a derandomized XOR lemma in the constant depth circuit setting? What is the most restricted circuit class where this can be carried out? We leave these as interesting open problems.

7 Acknowledgments

Thanks to Rocco Servedio for important conversations about the Impagliazzo construction and its relationship to boosting. Thanks to Ryan O'Donnell and Salil Vadhan for an interesting conversation about approximate majority and to V. Guruswami for his encouragement.

References

1. M. Ajtai. Σ_1^1-Formulae on Finite Structures. *Annals of Pure and Applied Logic*, Vol. 24, 1983.
2. M. Ajtai. Approximate Counting with Uniform Constant-Depth Circuits. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 13, 1993.
3. M. Ajtai and M. Ben-Or. A Theorem on Probabilistic Constant Depth Computations. In *ACM Symposium on Theory of Computing (STOC)*, 1984.
4. J. Aspnes, R. Beigel, M. Furst, and S. Rudich. The Expressive Power of Voting Polynomials. *Combinatorica*, 14(2), 1994.
5. L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.
6. P. Beame. A Switching Lemma Primer. Technical Report UW-CSE-95-07-01, Department of Computer Science and Engineering, University of Washington, November 1994.
7. R. Beigel When do Extra Majority Gates Help? Polylog(n) Majority Gates are Equivalent to One. *Computational Complexity*, 4, 314-324, 1994.
8. R. Beigel, N. Reingold, and D. Spielman. The Perceptron Strikes Back. In *Proceedings of the 6th Annual Conference on Structure in Complexity Theory (SCTC '91)*, June 1991.
9. M. Furst, J. Saxe, and M. Sipser. Parity, Circuits, and the Polynomial-Time Hierarchy. *Mathematical Systems Theory*, 17(1), 1984.
10. J. Håstad. *Computational Limitations of Small Depth Circuits*. MIT-PRESS, 1986.
11. R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science*, pages 538–545. IEEE, 1995.
12. R. Impagliazzo, R. Shaltiel, and A. Wigderson. Near Optimal Conversion of Hardness into Pseudorandomness. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, pages 538–545. IEEE, 1999.
13. R. Impagliazzo and A. Wigderson. P=BPP unless E has sub-exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th ACM Symposium on the Theory of Computing*, pages 220–229. ACM, 1997.
14. A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial hierarchy collapses. In *Proceedings of the 31st ACM Symposium on the Theory of Computing*, pages 659–667. ACM, 1999.
15. A. Klivans and R. Servedio. Boosting and Hard-Core Sets. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science (FOCS)*, 1999.
16. N. Linial, Y. Mansour, and N. Nisan. Constant Depth Circuits, Fourier Transforms, and Learnability. *Journal of the ACM*, 40(3), July 1993.
17. M. Luby, B. Velickovic, and A. Wigderson. Deterministic Approximate Counting of Depth-2 Circuits. In *Proceedings of the Second Israeli Symposium on Theory of Computing and Systems*, 1993.
18. N. Nisan. Pseudorandom Bits for Constant Depth Circuits. *Combinatorica*, 11, 1991.
19. N. Nisan and A. Wigderson. Hardness vs. Randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994.
20. C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
21. A. Razborov. Lower Bounds on the Size of Bounded Depth Circuits over a Complete Basis with Logical Addition. *MATHNAUSSR: Mathematical Notes of the Academ of Sciences of the USSR*, 41, 1987.

22. R. Servedio. Smooth Boosting and Learning with Malicious Noise. To Appear, 2001.
23. R. Smolensky. Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, May 1987.
24. L. Stockmeyer. The Complexity of Approximate Counting. In *ACM Symposium on Theory of Computing (STOC)*, 1983.
25. M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the XOR lemma. Technical Report TR-98-074, Electronic Colloquium on Computational Complexity, 2000. Revision 2.
26. L. Trevisan. Construction of extractors using pseudo-random generators. In *Proceedings of the 31st ACM Symposium on the Theory of Computing*, pages 141–148. ACM, 1999.
27. L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.

A Proof of Lemma 2

Algorithm **IHA** terminates after at most $i_0 = 2/\delta^2\epsilon^2$ iterations.

Proof. Let $\gamma = \epsilon\delta$. Consider the quantity $A_i(x) = \sum_{0 \leqslant j \leqslant i-1} R_{C_j}(x) M_{j-1}(x)$. This is the cumulative advantage that the first i circuits have on input x. It is not hard to see that $\sum_x A_i(x) \geqslant 2^n \gamma i$. For a fixed x, imagine plotting $N_j(x)$ with respect to i (think of j as increasing steps through the iterative algorithm). Match every time $N_j(x)$ increases from k to $k+1$ with every time $N_j(x)$ decreases from $k+1$ to k (if k is less than 0 match changes from k to $k-1$ with changes of $k-1$ to k). An easy calculation reveals that each of these pairs contributes at most γ to $A_i(x)$. For each decreasing unmatched time step we actually subtract from $A_i(x)$ and for each increasing unmatched time step (of which there can be at most $1/\gamma$) we add at most 1 to $A_i(x)$. There are at most $(1/2)i$ pairs. Thus $A_i(x) \leqslant 1/\gamma + (1/2)\gamma i$. Solving for i yields the claimed bound on the number of iterations.