Electronic Colloquium on Computational Complexity, Revision 3 of Report No. 122 (2020)

Size Bounds on Low Depth Circuits for Promise Majority

Joshua Cook
The University Of Texas At Austin, Texas, USA
jac22855@utexas.edu

Abstract

We give two results on the size of AC0 circuits computing promise majority. ϵ-promise majority is majority promised that either at most an ϵ fraction of the input bits are 1 or at most ϵ are 0.

- First, we show super-quadratic size lower bounds on both monotone and general depth-3 circuits for promise majority.
  - For any $\epsilon \in (0, 1/2)$, monotone depth-3 AC0 circuits for ϵ-promise majority have size
    $$\tilde{\Omega}\left(\epsilon^3 n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)}}\right).$$
  - For any $\epsilon \in (0, 1/2)$, general depth-3 AC0 circuits for ϵ-promise majority have size
    $$\tilde{\Omega}\left(\epsilon^3 n^{2+\frac{\ln(1-\epsilon^2)}{2\ln(\epsilon)}}\right).$$

  These are the first quadratic size lower bounds for depth-3 ϵ-promise majority circuits for $\epsilon < 0.49$.
- Second, we give both uniform and non-uniform sub-quadratic size constant-depth circuits for promise majority.
  - For integer $k \geq 1$ and constant $\epsilon \in (0, 1/2)$, there exists monotone *non* uniform AC0 circuits of depth-$(2 + 2k)$ computing ϵ-promise majority with size
    $$\tilde{O}\left(n^{\frac{1}{1-2^{-k}}}\right).$$
  - For integer $k \geq 1$ and constant $\epsilon \in (0, 1/2)$, there exists monotone *uniform* AC0 circuit of depth-$(2 + 2k)$ computing ϵ-promise majority with size
    $$n^{\frac{1}{1-\left(\frac{2}{3}\right)^k}+o(1)}.$$

  These circuits are based on incremental improvements to existing depth-3 circuits for promise majority given by Ajtai [2] and Viola [17] combined with a divide and conquer strategy.

2012 ACM Subject Classification Theory of computation → Circuit complexity

Keywords and phrases AC0, Approximate Counting, Approximate Majority, Promise Majority, Depth 3 Circuits, Circuit Lower Bound

Funding This research was supported by NSF grant number 1705028.

Acknowledgements Thanks to Dana Moshkovitz for suggesting I study the size cost of derandomizing AC0 circuits. Thanks to Justin Yirka, Amanda Priestly and an anonymous reviewer for feedback on this paper.

1 Introduction

The majority function is a classic function that cannot be computed in AC0 [10]. But AC0 can compute majority promised the input is either mostly 1s or mostly 0s.

ISSN 1433-8092

▶ **Definition 1** (ϵ-Promise Majority). *Let $W : \{0,1\}^n \to [n]$ be the function giving the number of ones in the input*[1]*. Let $\epsilon \in (0, 1/2)$. Then define the ϵ promise inputs to be:*

$$Maj_\epsilon^0 = \{x \in \{0,1\}^n : W(x) \leq \epsilon n\}$$
$$Maj_\epsilon^1 = \{x \in \{0,1\}^n : W(x) \geq (1-\epsilon)n\}$$
$$Maj_\epsilon = Maj_\epsilon^0 \cup Maj_\epsilon^1$$

We say that function f solves the ϵ-promise majority[2] *problem if:*

$$f(Maj_\epsilon^0) = 0$$
$$f(Maj_\epsilon^1) = 1$$

That is, f computes the majority promised the input is in Maj_ϵ.

We give size[3] lower bounds to depth-3 circuits[4] computing ϵ-promise majority. Then we give small circuits solving promise majority with larger depth.

1.1 Motivation

Promise majority is an important tool in derandomizing circuits. We say a function $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ is a randomized function for $g : \{0,1\}^n \to \{0,1\}$ if for all $x \in \{0,1\}^n$, $\Pr_{r \in \{0,1\}^m}[f(x,r) = g(x)] \geq 2/3$. A circuit implementing f is called a randomized circuit for g. We call $r \in \{0,1\}^m$ a seed for f.

Adleman [1] showed that for any randomized function $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$, implementing some $g : \{0,1\}^n \to \{0,1\}$, there is some choice of seeds $R \subseteq \{0,1\}^m$ with $|R| = O(n)$ such that for all x and the majority of seeds in R, f computes g, i.e., $\Pr_{r \in R}[f(x,r) = g(x)] > 1/2$. If f has size-$O(n)$ random circuits, this gives a size-$O(n^2)$ deterministic circuits by computing majority of $|R|$ copies of f and taking majority.

Unfortunately, AC0 cannot compute majority, but it can compute ϵ-promise majority. With the same argument, we can get R with $|R| = O(n)$ such that $\Pr_{r \in R}[f(x,r) = g(x)] > 3/5$. So, we only need to compute 2/5-promise majority since f only outputs the wrong bit for at most 2/5 of $r \in R$.

Ajtai [2] gave depth-3 circuits of size $O\left(n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}}\right)$ solving the ϵ-promise majority problem. Applying a depth-d promise majority circuit, M, to the output of a depth-k circuit, C, gives a depth-$(k+d-1)$ circuit since the kind of gate at the lowest level of M can be made the same as the top level of C. Combining this result with Adleman takes a size-$O(n)$, depth-d randomized circuit and gives a depth-$(d+2)$, size-$O\left(n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}}\right)$ deterministic circuit. This is bigger than the ideal $O(n^2)$ size from the unbounded depth setting.

This paper gives new, super-quadratic size lower bounds for depth-3 circuits computing ϵ-promise majority. Thus applying Adleman's technique on AC0 circuits to get size-$O(n^2)$ deterministic circuits using promise majority requires a depth-3 increase. We show this is tight by giving size-$O(n^2)$ depth-4 circuits for ϵ-promise majority. Thus Adleman's technique can be used to get size-$O(n^2)$ deterministic circuits with a depth-3 increase.

---

[1] For functions and circuits, we implicitly refer to a family of functions, one for each size n where n is implicit. The same holds for Maj_ϵ.

[2] Prior work often called promise majority "approximate majority" [17, 18]. But, approximate majority also refers to the standard notion of approximating a Boolean function [5]. To avoid confusion, we follow the convention suggested in [13] to refer to the promise problem version of majority as promise majority.

[3] In this paper, we use size of a circuit to mean the number of gates.

[4] In this paper, all circuits are constant depth alternating circuits (AC0 circuits) unless stated otherwise.

1.2 Our Results

For notation, let $\tilde{O}(x)$ indicate order x up to polylogarithmic factors:

▶ **Definition 2.** *$f(n) = \tilde{O}(g(n))$ if for some integer c, $f = O(g(n)\ln(n)^c)$.*
$f(n) = \tilde{\Omega}(g(n))$ if for some integer c, $f = \Omega(g(n)\ln(n)^c)$.

First, we give a size lower bound for monotone, depth-3 circuits for promise majority. Note that the best known depth-3 circuits are monotone.

▶ **Theorem 3.** *For any $\epsilon \in (0, 1/2)$, a* monotone*, depth-3 circuit solving the ϵ-promise majority problem must have size $\tilde{\Omega}\left(\epsilon^3 n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)}}\right)$.*

We follow this up with some weaker, but still super-quadratic, size lower bounds for depth-3 circuits computing promise majority.

▶ **Theorem 4.** *For any $\epsilon \in (0, 1/2)$, a depth-3 circuit solving the ϵ-promise majority problem must have size $\tilde{\Omega}\left(\epsilon^3 n^{2+\frac{\ln(1-\epsilon^2)}{2\ln(\epsilon)}}\right)$.*

Minor improvements to Ajtai's promise majority circuits [2] gives depth-4, quadratic size, promise majority circuits.

▶ **Theorem 5.** *For constant $\epsilon \in (0, 1/2)$, there exists* non *uniform, monotone, depth-4, size-$O\left(n^2\right)$ circuits solving the ϵ-promise majority problem.*

We then show how to solve ϵ-promise majority with even smaller circuits with larger depths using a divide and conquer strategy.

▶ **Theorem 6.** *For constant $\epsilon \in (0, 1/2)$, there exists* non *uniform, monotone, depth-$(2+2k)$ circuits solving the ϵ-promise majority problem with size $\tilde{O}\left(n^{\frac{1}{1-2^{-k}}}\right)$.*

The above circuits are not explicit, or uniform: we do not know how to construct it efficiently. However, we next give P-Uniform circuits for promise majority: circuits with a polynomial-time algorithm to construct them. These circuits use a slight improvement to Viola's depth-3 promise majority circuits [17] with a divide and conquer strategy.

▶ **Theorem 7.** *For constant $\epsilon \in (0, 1/2)$, there exists P-uniform, monotone, depth-$(2+2k)$ circuits solving the ϵ-promise majority problem with size $n^{\frac{1}{1-\left(\frac{2}{3}\right)^k}+o(1)}$.*

For $k = 2$, this gives depth-6, size-$o(n^2)$, P-uniform, monotone circuits for promise majority.

▶ **Corollary 8.** *For constant $\epsilon \in (0, 1/2)$, there exists P-uniform, monotone, depth-6 circuits solving the ϵ-promise majority problem with size $n^{\frac{9}{5}+o(1)}$.*

Thus a P-uniform PRG with $O(n)$ seeds for AC0 could derandomize linear-size, randomized circuits to get quadratic-size, deterministic circuits with a depth increase of 5. Finding such PRGs, or even PRGs with polynomially many seeds, is still open. Though, work by Dean Doron, Dana Moshkovitz, Justin Oh and David Zuckerman constructs nearly quadratic PRGs conditioned on some complexity theoretic assumptions [8].

1.3 Related Work

There are well known polynomial-size AC0 circuits for promise majority. First, Ajtai gave polynomial size, depth-3 circuits for ϵ-promise majority [2]. Ajtai later gave uniform, even deterministic log time uniform, AC0 circuits for promise majority [3]. But, these uniform circuits have large depth and their constructions are complicated. Viola later gave simpler P-Uniform, depth-3 AC0 circuits for promise majority [17].

Chaudhuri and Radhakrishnan [6] proved that any depth-d circuit computing ϵ-promise majority must have size $\Omega\left((\epsilon n)^{\frac{1}{1-1/4^d}} - n\right)$. This gives super-linear lower bounds for depth-3 circuits, but not close to quadratic. Their paper uses deterministic restrictions for lower bounds similar to ours, but our paper uses fan-in lower bounds from Viola [17] and different restrictions to get better depth-3 lower bounds.

In the same work [6], Chaudhuri and Radhakrishnan gave, for any k, depth-$O(k)$ circuits with size $O\left(n^{1+\frac{1}{2^k}}\right)$ for ϵ-promise majority. Like our paper, it uses a recursive strategy, but we use a different recursive strategy that gives shallower circuits.

Exact threshold functions in AC0 have been studied extensively. Ragde and Wigderson [15] show that for integer $r > 0$, the $\ln(n)^r$ threshold function, which computes whether $W(x) > \ln(n)^r$, has AC0 circuits with depth $O(r)$ and size $o(n)$. This improves on a result by Ajtai and Ben-Or [4]. Further, Håstad, Wegener, Wurm, and Yi [11] show that polylogarithmic threshold functions have sub-polynomial size, constant-depth circuits.

Results by Amano [5] building on work by O'Donnell and Wimmer [14] prove the minimum size for a depth-d circuit computing majority on most inputs is $\Theta\left(n^{\frac{1}{2d-1}}\right)$. This is consistent with promise majority results because most inputs are close to balanced, within a $O(1/\sqrt{n})$ factor, but promise majority is only guaranteed to give majority on inputs that are far from balanced.

For $\epsilon = \left(\frac{1}{2} - \frac{1}{\ln(n)^k}\right)$, Viola proved that randomized, depth-$(k+1)$, polynomial-size circuits can solve ϵ-promise majority, but deterministic, depth-$(k+2)$, polynomial-size circuits cannot. Further, there are deterministic, depth-$(k+3)$, polynomial-size circuits for ϵ-promise majority [18].

The same work [18] gave size lower bounds for depth-3 ϵ-promise majority circuits, but the bounds are less than linear for $\epsilon < 0.49$. Closer analysis gives better lower bounds, but we could not get quadratic lower bounds for $\epsilon < 0.49$ with this technique.

A later work by Limaye, Srinivasan and Tripathi [13] showed that deterministic, depth-$(k+1)$, polynomial-size AC0 circuits with parity gates also cannot solve $\left(\frac{1}{2} - \frac{1}{\ln(n)^k}\right)$-promise majority.

2 Proof Ideas

2.1 Monotone Depth-3 Circuit Lower Bounds

For depth-3 promise majority circuits, without loss of generality, assume the first level of gates are AND gates[5]. Call the inputs "variables", the first level gates "clauses", and the second level gates "DNFs".

[5] Switching the ANDs to ORs and ORs to ANDs in a circuit solving ϵ-promise majority still solves ϵ-promise majority. To see this, observe that flipping all the input bits will flip a Maj_ϵ^1 input to a Maj_ϵ^0 input. Then apply de Morgan's law.

To prove lower bounds for a depth-3 circuit, we construct adversarial restrictions that simplify the circuit while setting too few variables to violate the promise. To do this, we use two main tools. The first is a lemma from Viola [17] that we use to remove gates with very small fan in at the first level.

The second is a greedy set cover algorithm which shows that any collections of large subsets of variables can have a large fraction of the subsets hit by a small fraction of variables. To do this, we repeatedly select a variable in at least the average number of sets per variable.

First, we show DNFs have $\tilde{\Omega}(n^{1+\alpha})$ clauses for some $\alpha > 0$. To do this, we eliminate small clauses using the first idea, then eliminate a large fraction of clauses with few 0s using the second idea. This leaves many clauses while eliminating a large fraction of clauses, thus we started with many clauses.

Then, we show the circuit has $\tilde{\Omega}(n^{2+\alpha})$ clauses. First, we use the second idea to remove any very large clauses. This lets us fix clauses to 1 without using too many variables. Then, using the second idea again, we can hit many DNFs using few clauses. Thus there must be many clauses so we can not hit every DNF using few clauses.

We generally will not worry about integrality. This only becomes an issue when $\epsilon = \tilde{O}\left(n^{-1/2}\right)$ as some restrictions would not have size greater than one. In that case, our lower bounds hold trivially as ϵn gates can be fixed to a constant assigning only ϵn variables.

2.2 General Depth-3 Circuit Lower Bounds

The proof for non-monotone circuits is similar but with an additional hurdle. In monotone circuits, setting variables to 0 only makes clauses 0. But with negations, we can actually shrink clauses without eliminating them. This is an issue for showing DNFs must be large, but the rest of the argument only needs minor changes.

The solution is to set adversarial bits probabilistically. We independently set each bit to 1 with probability ϵ. With good probability, this will give an input in Maj_ϵ^0. Some DNFs then must have a good probability of "noticing" and becoming 0.

With high probability, fixing a small fraction of variables according to D_ϵ will eliminate many clauses. For some $\alpha > 0$, if a DNF is smaller than $n^{1+\alpha}$ this will make it constant. With good probability, setting the rest of the variables gives an input this DNF must "notice" and become 0. Thus, if the DNF is small, for some input it will be fixed to the constant 0 with only a few variables fixed. This cannot happen, so the DNF must be larger than $n^{1+\alpha}$.

2.3 Small Sized Circuits

To get small circuits, first we amplify the ϵ promise input to a $\frac{1}{\text{polylog}(n)}$ promise input by taking majority over $O(\ln(\ln(n)))$ length walks on an expander graph. Then we separate our input into polynomially small groups and run a $\frac{1}{\ln(n)}$-promise majority on each. This gives a polynomially smaller layer which satisfies just an $\ln(n)$ factor worse promise. Applying this several times computes majority of the promise input.

Ajtai's promise majority strategy gives a quadratic-sized $\frac{1}{\ln(n)}$-promise majority circuit. Using this with the divide and conquer strategy above gives non uniform small circuits.

For our uniform circuit, we look at Viola's circuit [17]. It uses a hitting property that requires $n^{3+o(1)}$ many random walks for each of our n bits, requiring an overall size of $n^{4+o(1)}$. We reduce this by showing it suffices to let each bit only range over random walks starting at that bit, giving a size-$n^{3+o(1)}$ circuit for $\frac{1}{\ln(n)}$-promise majority.

Applying this improved version of Viola's depth-3 circuit with our divide and conquer strategy gives our uniform small circuits.

2.4 Terminology

We will use biased inputs in our proofs.

▶ **Definition 9** (ϵ Biased Input). *For any $\epsilon \in [0,1]$ the ϵ biased input D_ϵ is a random variable over $\{0,1\}^n$ where each bit independently is 1 with probability ϵ.*

As with Maj^0_ϵ and Maj^1_ϵ, n in D_ϵ is implicit. D_ϵ is related to Maj^0_ϵ by a central limit theorem: $\Pr[D_\epsilon \in \text{Maj}^0_\epsilon] > \frac{1}{3}$ for large enough n.

We will make sub DNFs by only taking some clauses from a larger DNF.

▶ **Definition 10** (Sub DNF). *Let G be a DNF with clauses $C = \{C_i : i \in [k]\}$ so that $G = \bigvee_{i\in[k]} C_i$. Let $\Lambda \subseteq [k]$ and H be a DNF with $H = \bigvee_{i\in\Lambda} C_i$.*

Then we say that H is sub DNF of G or G has sub DNF H.

Restrictions fix some bits in the input to a function. We formalize this as a function that takes unrestricted bits as input and outputs the restricted and unrestricted bits together[6].

▶ **Definition 11** (Restriction). *A restriction ρ on n variables of size m is a function $\rho : \{0,1\}^{n-m} \to \{0,1\}^n$ such that for some $c \in \{0,1\}^m$ and some permutation of $[n]$, π, for all $x \in \{0,1\}^{n-m}$ and $i \in n$:*

$$\rho(x)_i = \begin{cases} c_{\pi_i} & \pi_i \leq m \\ x_{\pi_i - m} & \pi_i > m \end{cases}$$

We write the size of ρ as $|\rho| = m$ and define $f \restriction_\rho = f \circ \rho$.

When we apply a restriction, ρ, to a DNF, F, we let $F \restriction_\rho$ be the DNF which is F with variables restricted in ρ set to their restricted value. We simplify such a DNF to remove any clause that has been set to 0. We count the size of a DNF by its number of clauses.

▶ **Definition 12** (DNF Size and Width). *For a DNF F, the size of F, $|F|$, is the number of clauses in F. Any DNF that is the constant 1 or 0 function has size 0.*

We say a DNF F has width w if no clause in F has width greater than w.

3 Monotone Depth-3 Circuit Size Lower Bounds

3.1 Removing Small Clauses

We use a result from Viola [17], Lemma 11 therein. Intuitively, this lemma says for a DNF with small width, either there is some setting to a small number of variables that makes it 0, or under a randomized input it is unlikely to be 0.

▶ **Lemma 13.** *Let G be a DNF with a sub DNF F. Assume for some positive integers w and m, F has width at most w and $\Pr[G(D_\epsilon) = 0] \geq e^{-\epsilon^w \cdot m/w^2}$. Then there exists a restriction ρ with $|\rho| \leq m$ such that $F \restriction_\rho = 0$ and $\Pr[G \restriction_\rho (D_\epsilon) = 0] \geq \Pr[G(D_\epsilon) = 0]$.*

Our result is slightly generalized over the original and the proof is given in Appendix A. As a corollary, we can can apply small restrictions to eliminate small width clauses.

[6] This is an equivalent but slightly nonstandard way to define restrictions.

▶ **Corollary 14.** *Suppose we have $\epsilon \in (0, 1/2)$, DNF F and constant $\alpha > 0$ such that $\Pr[F(D_\epsilon) = 0] \geq n^{-\alpha}$. Then for sufficiently large n and*

$$w = \log_\epsilon \left(\frac{\ln(n)^5}{n\epsilon \ln(\epsilon)^2} \right)$$

there is a restriction ρ restricting at most $m = \frac{\epsilon n}{\ln(n)}$ variables so that any clause C in F with width less than w has $C \restriction_\rho = 0$ and $\Pr[F \restriction_\rho (D_\epsilon) = 0] \geq \Pr[F(D_\epsilon) = 0]$.

Proof. Let F' be the sub DNF of F with clauses of width less than w. Then

$$\mathbb{E}[F(D_\epsilon) = 0] \geq n^{-\alpha} = e^{-\alpha \ln(n)} = e^{-\alpha \frac{\ln(n)^5}{n\epsilon \ln(\epsilon)^2} \frac{\epsilon n}{\ln(n)} \frac{\ln(\epsilon)^2}{\ln(n)^3}} = e^{-\alpha \epsilon^w m \frac{\ln(\epsilon)^2}{\ln(n)^3}} \geq e^{-\epsilon^w m \frac{1}{w^2}}$$

From Lemma 13, there is a restriction ρ of size m with $\mathbb{E}[F \restriction_\rho (D_\epsilon) = 0] \geq \mathbb{E}[F(D_\epsilon) = 0]$ setting $F' \restriction_\rho = 0$. Any width w clause C would be in F', thus $C \restriction_\rho = 0$ since $F' \restriction_\rho = 0$. ◀

3.2 Covering Many Large Sets with Few Elements

We prove the simplest version of the clause elimination result, but slight variations will be used in multiple places. In particular, in the non-monotone lower bounds, we can't quite reduce the problem to set cover, but the same algorithm still works with a similar bound. Since the proofs look very similar, we only present one in detail. We show how to remove many clauses from a monotone DNF with a small restriction

▶ **Lemma 15.** *Let F be a monotone DNF where each clause has width at least w. Then for any positive integer b, there is some restriction ρ with $|\rho| = b$ only fixing variables to 0 such that $|F \restriction_\rho| < |F| e^{w \ln\left(1 - \frac{b}{n+1}\right)}$*

Proof. The idea is to restrict the variable that intersects the most clauses to 0. This removes at least the average number of clauses per variable, which when we have m clauses and have fixed i variables is at least $\frac{mw}{n-i}$. After b restrictions, we get ρ with $|\rho| = b$ and

$$|F \restriction_\rho| \leq |F| \prod_{i=0}^{b-1} \left(1 - \frac{w}{n-i}\right).$$

We prove this by induction then simplify. For the base case where $b = 0$, F is unchanged and we get the empty product, so the inequality holds.

For $b > 0$, we have some ρ' restricting $b-1$ variables with $|F \restriction_{\rho'}| \leq |F| \prod_{i=0}^{b-2} \left(1 - \frac{w}{n-i}\right)$. Then $F \restriction_{\rho'}$ is a function on $n + 1 - b$ variables. Let s be the variable in the most clauses of $F \restriction_{\rho'}$. Then s is in at least $|F \restriction_{\rho'}| \frac{w}{n+1-b}$ clauses. Let ρ be ρ' also fixing s to 0. Then:

$$|F \restriction_\rho| \leq |F \restriction_{\rho'}| - |F \restriction_{\rho'}| \frac{w}{n+1-b} = |F| \prod_{i=0}^{b-1} \left(1 - \frac{w}{n-i}\right),$$

completing our induction. The above equation simplifies to:

$$|F \restriction_\rho| = |F| \prod_{i=0}^{b-1} \left(1 - \frac{w}{n-i}\right) < |F| e^{\sum_{i=0}^{b-1} -\frac{w}{n-i}} = |F| e^{-w \sum_{i=n+1-b}^{n} 1/i}. \tag{1}$$

From calculus we have

$$\sum_{i=a}^{b} \frac{1}{i} \geq \int_a^{b+1} \frac{1}{x} dx = \ln\left(\frac{b+1}{a}\right),$$

which applied to Equation (1) gives

$$|F\restriction_\rho| < |F|e^{-w\sum_{i=n+1-b}^{n}\frac{1}{i}} \leq |F|e^{-w\ln\left(\frac{n+1}{n+1-b}\right)} = |F|e^{w\ln\left(1-\frac{b}{n+1}\right)}.$$

◀

The same idea gives the simpler bound:

▶ **Corollary 16.** *Let F be a monotone DNF where each clause has width at least w. Then for any integer b there is some restriction ρ with $|\rho| \leq b$ such that $|F\restriction_\rho| < |F|e^{-wb/n}$.*

With this idea, we can remove all large clauses fixing few variables. For the non-monotone case, we only remove half the average number of clauses with each variable, giving:

▶ **Corollary 17.** *Let F be a collection of clauses. Then there is some restriction ρ fixing n/p variables such that $F\restriction_\rho$ has width $w = 2\ln(|F|)p$.*

3.3 Monotone DNF Size

We prove that any DNF with a good chance of "noticing" inputs from D_ϵ has a large size.

▶ **Lemma 18.** *Suppose for $\epsilon \in (0, 1/2)$, there is a monotone DNF F with $F(Maj^1_\epsilon) = 1$ and $\Pr[F(D_\epsilon) = 0] \geq 1/n^\alpha$ for constant α. Then F has $\tilde{\Omega}\left(\epsilon n^{1+\frac{\ln(1-\epsilon)}{\ln(\epsilon)}}\right)$ clauses.*

Proof. The idea is to restrict our function until we are only promised it outputs 1 on an $\mathrm{Maj}^1_{\epsilon/\ln(n)}$ input. Using Lemma 15, we can do this in such a way that eliminates a large fraction of clauses. Then since we still need to output 1 if we have fewer than $\frac{\epsilon n}{\ln(n)}$ more zeros, we can choose these remaining $\frac{\epsilon n}{\ln(n)}$ zeros to each eliminate one clause, showing that there are still $\frac{\epsilon n}{\ln(n)}$ clauses left. This will imply that we must have started with the claimed number of clauses.

For $w = \log_\epsilon\left(\frac{\ln(n)^5}{n\epsilon\ln(\epsilon)^2}\right)$, by Corollary 14, there is restriction ρ with $|\rho| \leq \frac{\epsilon n}{\ln(n)}$ and $F\restriction_\rho$ that has no clauses smaller than w. Denote $F_2 = F\restriction_\rho$. Note F_2 solves $F_2\left(\mathrm{Maj}^1_{\epsilon(1-1/\ln(n))}\right) = 1$ and has no clauses smaller than w.

Now we use Lemma 15 to get restriction ρ_2 that assigns $\epsilon n(1 - 2/\ln(n))$ variables and:

$$|F_2\restriction_{\rho_2}| \leq |F_2|e^{w\ln\left(1-\frac{\epsilon n(1-2/\ln(n))}{n+1}\right)}.$$

Now we simplify the above exponent. For $0 < x < \frac{1}{2}$ and $0 < y$, by a Taylor argument we have $\ln(1 - x + y) \leq \ln(1 - x) + 2y$. Then for sufficiently large n:

$$\ln\left(1 - \frac{\epsilon n(1-2/\ln(n))}{n+1}\right) = \ln\left(1 - \epsilon + \frac{\epsilon}{n+1} + \frac{2\epsilon n}{n+1}\frac{1}{\ln(n)}\right) \leq \ln(1-\epsilon) + \frac{5\epsilon}{\ln(n)}.$$

Now including w,

$$\begin{aligned} w\ln\left(1 - \frac{\epsilon n(1-2/\ln(n))}{n+1}\right) \leq& \frac{\ln(n) - \ln\left(\frac{\ln(n)^5}{\epsilon\ln(\epsilon)^2}\right)}{\ln(1/\epsilon)}\left(\ln(1-\epsilon) + \frac{5\epsilon}{\ln(n)}\right) \\ <& \frac{\ln(n)\ln(1-\epsilon)}{\ln(1/\epsilon)} - \frac{\ln\left(\frac{\ln(n)^5}{\epsilon\ln(\epsilon)^2}\right)\ln(1-\epsilon)}{\ln(1/\epsilon)} + \frac{5}{\ln(1/\epsilon)}. \end{aligned}$$

Then applying this to our size bound

$$\begin{aligned}|F_2 \restriction_{\rho_2}| \leq & |F_2| e^{w \ln\left(1-\frac{\epsilon n(1-2/\ln(n))}{n+1}\right)} \\ < & |F_2| e^{\frac{\ln(n)\ln(1-\epsilon)}{\ln(1/\epsilon)} - \frac{\ln\left(\frac{\ln(n)^5}{\epsilon \ln(\epsilon)^2}\right)\ln(1-\epsilon)}{\ln(1/\epsilon)} + \frac{5}{\ln(1/\epsilon)}} \\ < & |F_2| n^{\frac{\ln(1-\epsilon)}{\ln(1/\epsilon)}} 2\ln(n)^5 e^8.\end{aligned}$$

Since ρ and ρ_2 only restricts $\epsilon n\left(1-\frac{1}{\ln(n)}\right)$ clauses, $F_2 \restriction_{\rho_2} (\mathrm{Maj}^1_{\epsilon/\ln(n)}) = 1$. Further, since F is monotone, ρ and ρ_2 only fixed variables to 0. Therefore, $F_2 \restriction_{\rho_2} \neq 1$. Then $F_2 \restriction_{\rho_2}$ must have at least $\frac{\epsilon n}{\ln(n)}$ clauses. Thus:

$$\begin{aligned}\frac{\epsilon n}{\ln(n)} \leq & |F_2 \restriction_{\rho_2}| \leq e^8 |F_2| n^{\frac{\ln(1-\epsilon)}{\ln(1/\epsilon)}} 2\ln(n)^5 \\ \frac{\epsilon n^{1+\frac{\ln(1-\epsilon)}{\ln(\epsilon)}}}{2e^8 \ln(n)^6} \leq & |F_2|.\end{aligned}$$

F has at least as many clauses as F_2, thus $|F| = \tilde{\Omega}\left(\epsilon n^{1+\frac{\ln(1-\epsilon)}{\ln(\epsilon)}}\right)$. ◀

3.4 Monotone Circuit Size Lower Bounds

Now we prove the monotone depth-3 promise majority circuit lower bounds.

▶ **Theorem 3.** *For any $\epsilon \in (0, 1/2)$, a* monotone*, depth-3 circuit solving the ϵ-promise majority problem must have size $\tilde{\Omega}\left(\epsilon^3 n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)}}\right)$.*

Proof. Let F be a monotone depth-3 circuit computing ϵ-promise majority. We will refer to the first level gates as clauses, and the second level gates as DNFs. Let $|F|$ refer to the number of clauses in F, and $\|F\|$ refer to the number of DNFs. If F has more than $n^{2+\alpha}$ gates, we are done. So suppose it does not.

Let $\alpha = \frac{\ln(1-\epsilon+3\epsilon/\ln(n))}{\ln(\epsilon-3\epsilon/\ln(n))}$. We can show that

$$\alpha > \frac{\ln(1-\epsilon)}{\ln(\epsilon)} - O\left(\frac{1}{\ln(n)}\right).$$

So if we show $|F| = \tilde{\Omega}\left(\epsilon^3 n^{2+\alpha}\right)$, then the second term in α becomes a constant.

First, from Corollary 17, we have a restriction ρ fixing $\frac{\epsilon n}{\ln(n)}$ variables such that any clause wider than $w = 2\ln(|F|)\frac{\ln(n)}{\epsilon}$ is set to 0. Let $F_2 = F \restriction_{\rho}$. See that F_2 solves the $\epsilon\left(1-\frac{1}{\ln(n)}\right)$-promise majority problem and has no clauses wider than $6\frac{\ln(n)^2}{\epsilon}$.

By Lemma 18, every DNF G with $\Pr[G(D_{\epsilon(1-3/\ln(n))}) = 0] \geq 1/n^{3+\alpha}$ has at least $c\epsilon n^{1+\alpha}$ clauses for some polylogarithm c. Let F_3 be the sub circuit of F_2 with only the DNFs of F_2 larger than $c\epsilon n^{1+\alpha}$.

Since no clauses are wider than w, we can set any m clauses in F_3 to 0 by fixing only mw variables. Then, analogous to Corollary 16, there exists a restriction ρ_2 fixing $\epsilon n/\ln(n)$ variables to 1 such that:

$$\|F_3 \restriction_{\rho_2}\| \leq \|F_3\| e^{-c\epsilon n^{1+\alpha}(|\rho_2|/w)/|F_3|},$$

where $\|F_3 \restriction_{\rho_2}\|$ is the number of DNFs in F_3 not fixed to 1 or 0 under the restriction ρ_2.

See that $F_2 \restriction_{\rho_2}$ still solves the $\epsilon(1-2/\ln(n))$-majority problem. By a central limit theorem, $D_{\epsilon(1-2/\ln(n))}$ has a constant nonzero probability of being in $\mathrm{Maj}^0_{\epsilon(1-2/\ln(n))}$. Since F_2 has fewer than $n^{2+\alpha}$ DNFs (by assumption), some DNF in F_2, A, must be 0 on $D_{\epsilon(1-2/\ln(n))}$ with probability greater than $1/n^{3+\alpha}$. By Lemma 18, A has size at least $c\epsilon n^{1+\alpha}$. Thus A must also be in F_3. Thus $\|F_3 \restriction_{\rho_2}\| \geq 1$.

Now we can compute a lower bound for $|F_3|$:

$$\begin{aligned} 1 \leq & \|F_3 \restriction_{\rho_2}\| \leq \|F_3\| e^{-c\epsilon n^{1+\alpha} \frac{|\rho_2|}{w|F_3|}} \\ e^{c\epsilon^2 n^{2+\alpha} \frac{1}{w|F_3|\ln(n)}} \leq & \|F_3\| \\ c\epsilon^3 n^{2+\alpha} \frac{1}{2\ln(|F|)\ln(n)|F_3|\ln(n)} \leq & \ln(\|F_3\|) \\ \tilde{\Omega}\left(\epsilon^3 n^{2+\alpha}\right) \leq & |F_3|. \end{aligned}$$

Using the definition of α and that $|F| > |F_3|$ we get:

$$|F| \geq \tilde{\Omega}\left(\epsilon^3 n^{2+\alpha}\right) \geq \tilde{\Omega}\left(\epsilon^3 n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)} - O\left(\frac{1}{\ln(n)}\right)}\right) \geq \tilde{\Omega}\left(\epsilon^3 n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)}}\right).$$

◀

4 General Depth-3 Circuits

The proof of the size lower bound for general depth-3 circuits computing promise majority is almost the same as the monotone case, except for the proof that DNFs must be large. We only prove our DNF bound here, but leave a brief proof of Theorem 4 in Appendix B.

▶ **Lemma 19.** *Suppose $\epsilon \in (0, 1/2)$, and F is a DNF such that $\Pr[F(D_\epsilon) = 0] \geq 1/n^\alpha$ for some constant α and $F(Maj^1_\epsilon) = 1$. Then F has size at least $\tilde{\Omega}\left(\epsilon n^{1+\frac{\ln(1-\epsilon^2)}{2\ln(\epsilon)}}\right)$.*

Proof. First, see that if $F(\mathrm{Maj}^1_\epsilon) = 1$ and $F \neq 1$, there must be at least ϵn clauses. Otherwise we could fix one variable in each clause to 0 using fewer than $n\epsilon$ zeros. Then for $\epsilon = \tilde{O}\left(\frac{1}{\sqrt{n}}\right)$ the lemma is satisfied. So take $\epsilon = \omega\left(\frac{\ln(n)^3}{\sqrt{n}}\right)$.

Let $m = \epsilon n(1-2/\ln(n))$ and $w = \log_\epsilon(\frac{\ln(n)^5}{n\epsilon\ln(\epsilon)^2})$. We will define a sequence of probabilistic restrictions, $\rho_0, ..., \rho_m$, each restricting one more variable according to D_ϵ. At the same time we will construct a sequence of sub DNFs of F, $F_0, ..., F_m$, each a subset of the last, so that each $F_i \restriction_{\rho_i}$ has width at least w.

Informally, with decent probability each F_i is significantly smaller than the last. Thus by a Chernoff bound, with high probability F_m has a small fraction of the clauses of F. Then we use Corollary 14 to eliminate the small width clauses in $F_m \restriction_{\rho_m}$. With good probability the DNF will still not be 1, in which case it must still have an almost linear number of clauses. Thus there must have been many clauses to destroy so many and have so many left.

Let ρ_0 restrict no variables and F_0 be F restricted to clauses wider than w. Then for any i, let ρ_i be ρ_{i-1} plus restricting whichever variable appears in the most clauses in $F_{i-1} \restriction_{\rho_{i-1}}$ to one with probability ϵ and 0 otherwise. Then let F_i be the clauses such that they have width greater than w in $F \restriction_{\rho_i}$. See that $F_i \subseteq F_{i-1}$, since further restrictions will only decrease the size and number of clauses.

With probability at least ϵ, ρ_i will eliminate at least $\frac{|F_{i-1}|w}{2(n-i+1)}$ clauses. Thus:

$$\Pr\left[|F_{i+1}| \leq |F_i|\left(1 - \frac{w}{2(n-i)}\right)\right] \geq \epsilon.$$

Let k be the number of times the above inequality holds. By an argument similar to Lemma 15:

$$|F_m| \leq |F_0| \prod_{i=0}^{k-1} \left(1 - \frac{w}{2(n-i)}\right)^k \leq |F_0| e^{\frac{w}{2}\ln\left(1-\frac{k}{n+1}\right)}.$$

See the expected value of k is at least $m\epsilon$. By a Chernoff bound, we have:

$$\Pr[k < (1 - \frac{1}{\ln(n)})\epsilon m] \leq e^{-\frac{\epsilon m}{2\ln(n)^2}} < e^{-\frac{\epsilon^2 n}{\ln(n)^3}}.$$

Now, notice that ρ_m only sets variables according to an ϵ biased distribution. So if we just finish sampling the rest of the variables from D_ϵ, it is the same as sampling all the variables from D_ϵ. Thus:

$$\mathbb{E}_{\rho_m}[\Pr[F \restriction_{\rho_m} (D_\epsilon) = 0]] = \Pr[F(D_\epsilon) = 0].$$

We need high probability that $F \restriction_{\rho_m}$ still outputs 0 with polynomial probability on D_ϵ. Applying the above equation and our assumption we get:

$$\begin{aligned}
\frac{1}{n^\alpha} \leq& \mathbb{E}_{\rho_m}[\Pr[F \restriction_{\rho_m} (D_\epsilon) = 0]] \\
\leq& \frac{1}{n^{2\alpha}} + \Pr_{\rho_m}\left[\Pr[F \restriction_{\rho_m} (D_\epsilon) = 0] > \frac{1}{n^{2\alpha}}\right] \\
\frac{1}{n^\alpha} - \frac{1}{n^{2\alpha}} \leq& \Pr_{\rho_m}\left[\Pr[F \restriction_{\rho_m} (D_\epsilon) = 0] > \frac{1}{n^{2\alpha}}\right].
\end{aligned}$$

The probability that ρ_m has both $\Pr[F \restriction_{\rho_m} (D_\epsilon) = 0] > 1/n^{2\alpha}$ and $k > (1 - \frac{1}{\ln(n)})\epsilon m$ is at least $\frac{1}{n^\alpha} - \frac{1}{n^{2\alpha}} - e^{-\frac{\epsilon^2 n}{\ln(n)^3}}$, which for large n is positive. Then take such ρ_m as ρ.

By Corollary 14, we have a restriction of $F|_\rho$, ρ', which restricts $\epsilon n/\ln(n)$ variables and leaves no clauses of width less than w, and has

$$\Pr[F \restriction_\rho \restriction_{\rho'} (D_\epsilon) = 0] \geq \Pr[F \restriction_\rho (D_\epsilon) = 0] \geq \frac{1}{n^{2\alpha}}.$$

Now call $F' = F \restriction_\rho \restriction_{\rho'}$. See that F' has fixed $\epsilon n(1 - \frac{1}{\ln(n)})$ variables. Thus it still satisfies $F'(\mathrm{Maj}^1_{\epsilon/\ln(n)}) = 1$. Since $F' \neq 1$, $|F'| \geq \epsilon n/\ln(n)$. The clauses in F' had width greater than w in F_m, otherwise ρ' would have set them to 0. Thus $|F_m| \geq \epsilon n/\ln(n)$. Together we have:

$$\begin{aligned}
\frac{\epsilon n}{\ln(n)} \leq& |F_0| e^{\frac{w}{2}\ln\left(1-\frac{k}{n+1}\right)} \\
\leq& |F_0| e^{\frac{w}{2}\ln\left(1-\frac{(1-1/\ln(n))\epsilon m}{n+1}\right)} \\
\leq& |F_0| e^{\frac{\ln\left(\frac{n\epsilon\ln(\epsilon)^2}{\ln(n)^5}\right)}{2\ln(1/\epsilon)}\left(\ln\left(1-\epsilon^2\right)+6\epsilon^2/\ln(n)\right)} \\
\tilde{\Omega}\left(\epsilon n^{1+\frac{\ln\left(1-\epsilon^2\right)}{2\ln(\epsilon)}}\right) \leq& |F_0|.
\end{aligned}$$

Thus F has at least $\tilde{\Omega}\left(\epsilon n^{1+\frac{\ln\left(1-\epsilon^2\right)}{2\ln(\epsilon)}}\right)$ clauses. ◀

5 Circuit Upper Bounds

This section mostly uses standard techniques and the details are given in the appendix. A close analysis of Ajtai's [2] promise majority circuits (shown in Appendix C.1) gives:

▶ **Theorem 20.** *For any $\epsilon \in (0, 1/2)$, there exists monotone, depth-3 circuits solving the ϵ-promise majority problem with size $O\left((\epsilon \ln(\epsilon))^2 n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}}\right)$.*

This also gives the corollary we will use for our stronger upper bounds for higher depth.

▶ **Corollary 21.** *For any $\epsilon = O\left(\frac{\ln(\ln(n))}{\ln(n)}\right)$, there are monotone, depth-3 circuits solving the ϵ-promise majority problem with size $O\left(n^2\right)$.*

Using random walks on expander graphs, we can amplify our promise (proof in Appendix C.2). The polylogarithmic factor in the size depends on ϵ and k.

▶ **Lemma 22.** *For any constant k and $\epsilon \in (0, 1/2)$, there exists P-Uniform, monotone, depth-3 circuits with size $\tilde{O}(n)$ amplifying a Maj_{ϵ}^{0} input to a $Maj_{\frac{1}{\ln(n)^k}}^{0}$ output and a Maj_{ϵ}^{1} input to a $Maj_{\frac{1}{\ln(n)^k}}^{1}$ output.*

With amplification and quadratic-size circuits, we can trivially prove the existence of depth-4, size-$\tilde{O}(n^2)$ circuits. But the circuit size only depends on the number of potential inputs (*not* the number of bits used to represent them). Thus the circuit has size $O(n^2)$.

▶ **Theorem 5.** *For constant $\epsilon \in (0, 1/2)$, there exists* non *uniform, monotone, depth-4, size-$O\left(n^2\right)$ circuits solving the ϵ-promise majority problem.*

We can apply promise majority circuits in a divide and conquer fashion to get the following (proved in Appendix C.3):

▶ **Lemma 23.** *If there are depth-3 circuits with size n^{α} solving $\frac{1}{\ln(n)}$-promise majority, then for any positive integer k, there are depth-$(1+2k)$ circuits solving $\frac{1}{\ln(n)^k}$-promise majority with size*

$$kn^{\frac{1}{1-\left(\frac{\alpha-1}{\alpha}\right)^k}},$$

which is uniform and monotone if the depth-3 circuits are uniform and monotone.

Combining Lemma 23 with amplification and our quadratic-sized majority gives:

▶ **Theorem 6.** *For constant $\epsilon \in (0, 1/2)$, there exists* non *uniform, monotone, depth-$(2+2k)$ circuits solving the ϵ-promise majority problem with size $\tilde{O}\left(n^{\frac{1}{1-2^{-k}}}\right)$.*

For uniform circuits, we refine Viola's result [17] by giving a more efficient way to use the random walks in the existing algorithm. Details in Appendix D.1.

▶ **Theorem 24.** *There exists P-uniform, monotone, depth-3, size-$O\left(n^{3+o(1)}\right)$ circuits solving the $\frac{1}{\ln(n)}$-promise majority problem.*

Again applying amplification and divide and conquer we get:

▶ **Theorem 7.** *For constant $\epsilon \in (0, 1/2)$, there exists P-uniform, monotone, depth-$(2+2k)$ circuits solving the ϵ-promise majority problem with size $n^{\frac{1}{1-\left(\frac{2}{3}\right)^k}+o(1)}$.*

6 Closing Statements & Open Problems

These results are essentially tight in the following sense. For a wide range of ϵ, between $\epsilon = o(1)$ and $\epsilon = n^{-o(1)}$, the optimal size of depth-3 circuits for ϵ-promise majority is $n^{2\pm o(1)}$.

These lower bounds do not obviously extend to depth-4 circuits, so the right size for promise majority at higher depths is less clear. Better amplification plus Ajtai's promise majority circuit can actually achieve circuits with size significantly smaller than n^2. So our upper bounds are not optimal for depths greater than 3.

For depth-3 circuits computing promise majority, we gave four different size bounds: a lower bound for monotone circuits, a lower bound for general circuits, an upper bound for monotone circuits, and an upper bound for uniform monotone circuits. Each of these bounds differs by a polynomial factor, but we suspect they are equal.

Finally, here are some open problems:

1. Is there a way to derandomize any depth-d, size-$O(n)$, randomized circuit to get a depth-$(d+2)$, size-$O(n^2)$, deterministic circuit?
   We did not find any function f that has a randomized, depth-d, size-$O(n)$ circuit, R, computing f, but no deterministic, depth-$(d+2)$, size-$O(n^2)$ circuit computing f. We only showed that taking promise majority over $O(n)$ copies of R (as you would with an ideal PRG) would give super-quadratic circuits. There may always be some other deterministic, depth-$(d+2)$, size-$O(n^2)$ circuit computing f.
2. Do negations help solve promise majority?
   Our lower bounds for monotone circuits are better than our general lower bounds. It does not seem like negations should help, but we were unable to rule it out.
3. What is optimal size for depth-3 circuits computing ϵ-promise majority?
   For constant $\epsilon \in (0, 1/2)$, there is a polynomial gap between even our monotone lower bounds $\tilde{\Omega}\left(n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)}}\right)$, and upper bounds $O\left(n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}}\right)$.
   For constant $\alpha \in (0, 1/2)$ and $\epsilon = n^{-\alpha}$, there is a polynomial gap between our lower bounds $\tilde{\Omega}\left(n^{2-3\alpha}\right)$ and our upper bounds $\tilde{O}\left(n^{2-2\alpha}\right)$.
4. What is the optimal size for depth greater than 3?
   Chaudhuri and Radhakrishnan [6] gave size lower bounds of roughly $\Omega\left(n^{1+\frac{1}{2^{2d}}}\right)$ for depth-d ϵ-promise majority circuits, while we only achieve upper bounds of roughly $\Omega\left(n^{1+\frac{1}{2^{d/2-1}}}\right)$.
5. Do these bounds extend to AC0 with parity, or other circuit classes below TC0?
6. Are there uniform depth-3 circuits for promise majority with the same size as Ajtai's construction? Can we get uniform, depth-4, quadratic size circuits for promise majority?

References

1 Leonard Adleman. Two theorems on random polynomial time. In *Proceedings of the 19th Annual Symposium on Foundations of Computer Science*, SFCS '78, page 75–83, USA, 1978. IEEE Computer Society.

2 Miklós Ajtai. Sigma11-formulae on finite structures. *Ann. Pure Appl. Log.*, 24:1–48, 1983.

3 Miklós Ajtai. Approximate counting with uniform constant-depth circuits. In *Advances In Computational Complexity Theory*, volume 13, pages 1–20, 1993.

4 Miklós Ajtai and Michael Ben-Or. A theorem on probabilistic constant depth computations. In *STOC '84*, pages 471–474, 1984.

5 Kazuyuki Amano. Bounds on the size of small depth circuits for approximating majority. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming: Part I*, ICALP '09, page 59–70, Berlin, Heidelberg, 2009. Springer-Verlag.

6 Shiva Chaudhuri and Jaikumar Radhakrishnan. Deterministic restrictions in circuit complexity. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 30–36, New York, NY, USA, 1996. Association for Computing Machinery. doi:10.1145/237814.237824.

7 Michael B. Cohen. Ramanujan graphs in polynomial time. *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 276–281, 2016.

8 Dean Doron, Dana Moshkovitz, Justin Oh, and David Zuckerman. Nearly optimal pseudorandomness from hardness. In *To appear in The proceedings of the 61st IEEE Symposium on Foundations of Computer Science*, 2020.

9 Alexander Healy. Randomness-efficient sampling within nc1. *Computational Complexity*, 17:3–37, 04 2008.

10 Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, page 6–20, New York, NY, USA, 1986. Association for Computing Machinery.

11 Johan Håstad, Ingo Wegener, Norbert Wurm, and Sang-Zin. Yi. Optimal depth, very small size circuits for symmetrical functions in ac0. *Information and Computation*, 108(2):200 – 211, 1994.

12 Clemens Lautemann. Bpp and the polynomial hierarchy. *Information Processing Letters*, 17(4):215 – 217, 1983.

13 Nutan Limaye, Srikanth Srinivasan, and Utkarsh Tripathi. More on $AC^0[\oplus]$ and variants of the majority function. In *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2019)*, volume 150, pages 22:1–22:14, 2019.

14 Ryan O'Donnell and Karl Wimmer. Approximation by dnf: Examples and counterexamples. In *Proceedings of the 34th International Conference on Automata, Languages and Programming*, ICALP'07, page 195–206, Berlin, Heidelberg, 2007. Springer-Verlag.

15 Prabhakar Ragde and Avi Wigderson. Linear-size constant-depth polylog-threshold circuits. *Information Processing Letters*, 39:143–146, 1991.

16 Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

17 Emanuele Viola. On approximate majority and probabilistic time. *Computational Complexity*, 18:337–375, 2009.

18 Emanuele Viola. Randomness buys depth for approximate counting. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 230–239, 2011.

A Proof of Lemma 13

This proof was first presented by Viola [17] in the more straightforward case where $G = F$: where there is only one DNF. We use this lemma on even non-monotone circuits to eliminate small clauses in a DNF without fixing it to one, so we have to present it in this slight generalization. This does not require any new ideas, just more careful analysis.

▶ **Lemma 13.** *Let G be a DNF with a sub DNF F. Assume for some positive integers w and m, F has width at most w and $\Pr[G(D_\epsilon) = 0] \geq e^{-\epsilon^w \cdot m/w^2}$. Then there exists a restriction ρ with $|\rho| \leq m$ such that $F \restriction_\rho = 0$ and $\Pr[G \restriction_\rho (D_\epsilon) = 0] \geq \Pr[G(D_\epsilon) = 0]$.*

Proof. The proof follows closely from a basic idea: if there are many small independent clauses in a DNF, it will accept with high probability. If there aren't, then we can set a small number of variables to decrease the size of each clause.

See that since F is a sub DNF of G, for any ρ, $G \restriction_\rho (x) = 0 \implies F \restriction_\rho (x) = 0$. Thus for any distribution y, $\Pr[F \restriction_\rho (y) = 0] \geq \Pr[G \restriction_\rho (y) = 0]$.

We will use a proof by induction. First the base case, where $w = 1$. If there are only m clauses in F, then let ρ be restriction setting each variable in F to make its corresponding clause 0. Then since this is the only assignment to these variables so that G is not one, we have

$$\Pr[G(D_\epsilon) = 0] \leq \Pr[G \restriction_\rho (D_\epsilon) = 0]$$

Otherwise F must have at least m + 1 clauses. Then $\Pr[F(D_\epsilon) = 0]$ is just the probability that none of its at least m variables is one. Thus:

$$\Pr[G(D_\epsilon) = 0] \leq |\Pr[F(D_\epsilon) = 0] \leq (1-\epsilon)^{m+1} < e^{-\epsilon m}$$

but by assumption $\Pr[G(D_\epsilon) = 0] \geq e^{-\epsilon m}$, thus this case cannot occur.

Now for the general case where $w \geq 2$. We say that a set of variables, S, is a cover for DNF F if every clause in F contains at least one variable from S. We say that a set of clauses T is independent if no two clauses in T have any variables in common.

Consider a minimum cover of F, S. Let T be a maximal independent set of clauses in F. See that $|T|w \geq |S|$ because if we take every variable in a maximal independent set of clauses, we must get a cover for F (otherwise we would have another independent clause) and we already said S is the smallest cover.

Then the probability F is 0 is at most the probability each clause in T is 0. Thus:

$$\begin{aligned}\Pr[F(D_\epsilon) = 0] \leq & (1-\epsilon^w)^{|T|} \\ < & e^{-|T|\epsilon^w} \\ \leq & e^{-|S|\epsilon^w/w}\end{aligned}$$

Now if $|S| \geq m/w$, then we have violated the assumption and are done. Thus we have $|S| < m/w$.

Choose some assignment to the variables in S to get ρ' so that $\Pr[G \restriction_{\rho'} (D_\epsilon) = 0] \geq \Pr[G(D_\epsilon) = 0]$. By an averaging argument, some ρ' must do this. This gives

$$\begin{aligned}\Pr[G \restriction_{\rho'} (D_\epsilon) = 0] \geq & \Pr[G(D_\epsilon) = 0] \\ \geq & e^{-\epsilon^w \cdot m/w^2} \\ > & e^{-\epsilon^{w-1} \cdot m \frac{w-1}{w-1} \frac{1}{(w-1)w}} \\ \geq & e^{-\epsilon^{w-1} \cdot m \frac{1-1/w}{(w-1)^2}}\end{aligned}$$

Thus $F \restriction_{\rho'}$ is a new width $w-1$ instance. Then by the inductive hypotheses, we have a restriction ρ'' of size at most $m(1-1/w)$ setting $F \restriction_{\rho'} \restriction_{\rho''} = 0$ and not decreasing the probability $G \restriction_{\rho'} (D_\epsilon)$ is 0. Thus letting $\rho = \rho' \circ \rho''$, we have

$$|\rho| \leq m/w + m(1-1/w) = m$$

$$F \restriction_\rho = 0$$

$$\Pr[G \restriction_\rho (D_\epsilon) = 0] \geq \Pr[G(D_\epsilon) = 0]$$

◀

B Proof of Theorem 4

Here we present the proof for the general case depth-3 circuit for promise majority lower bounds. As previously mentioned, the proof is very similar to the proof of Theorem 3.

▶ **Theorem 4.** *For any $\epsilon \in (0, 1/2)$, a depth-3 circuit solving the ϵ-promise majority problem must have size $\tilde{\Omega}\left(\epsilon^3 n^{2+\frac{\ln(1-\epsilon^2)}{2\ln(\epsilon)}}\right)$.*

Proof. As before, we first use $\epsilon n/\ln(n)$ variables to remove all very large clauses. Then we will use $\epsilon n/\ln(n)$ more getting rid of many large DNFs. This will leave us with at least 1 large DNF since we still output 0 on $D_{\epsilon(1-2/\ln(n))}$ with high probability.

Let F be a polynomial size, depth-3 circuit computing ϵ-promise majority. Then as in the proof of Theorem 3, assume its first level of gates are AND gates, call them clauses and our second level gates DNFs.

First, using Corollary 17, there is a restriction ρ with $|\rho| = \epsilon n/\ln(n)$ variables and removes any clauses width greater than $O(\ln(n)^2/\epsilon)$. Then let $F' = F|_\rho$.

Now let F_2 be the subcircuit of F' that only has DNFs of size at least $c\epsilon n^{1+\alpha}$ for $\alpha = \frac{\ln(1-\epsilon^2)}{2\ln(\epsilon)}$ and some $c = \text{polylog}(n)$ given by Lemma 19. Now we fix $\epsilon^2 n/\ln(n)^4$ clauses in F_2 that eliminate as many DNFs as we can to get F_3. This requires us to fix fewer than $\epsilon n/\ln(n)$ variables. By a similar proof to Theorem 3, we can get

$$\|F_3\| \leq \|F_2\| e^{-\epsilon^3 \frac{n^{2+\alpha}}{|F_2|\text{polylog}(n)}}$$

Now, $\|F_3\|$ must have at least 1 DNF of size $c\epsilon n^{1+\alpha}$ because F_3 still solves a $\epsilon\left(1-\frac{2}{\ln(n)}\right)$-promise majority problems. Thus:

$$1 \leq \|F_2\| e^{-\epsilon^3 \frac{n^{2+\alpha}}{|F_2|\text{polylog}(n)}}$$

This only holds if $|F_2| = \tilde{\Omega}(\epsilon^3 n^{2+\alpha})$ or $\|F_2\|$ is exponentially large. Thus the size of F is $\tilde{\Omega}(\epsilon^3 n^{2+\alpha})$. ◀

C Circuit Upper Bounds

C.1 Ajtai's Construction

Ajtai gave a randomized construction of a depth-3 circuit for promise majority. It is defined recursively with each layer being a conjunction or disjunction of a specific number of random circuits of smaller depth.

Intuitively, it gives a circuit which solves a coin problem, determining whether an input comes from D_ϵ or $D_{1-\epsilon}$, with exponentially high probability. Then feeding in random indexes from a Maj_ϵ instance is equivalent to feeding in an input from D_ϵ or $D_{1-\epsilon}$. Then by an averaging argument, some choice of indexes must always work.

So first we prove:

▶ **Lemma 25.** *For any $\epsilon \in (0, 1/2)$ and positive integer n, there exists a monotone depth-3 circuit, C, with size $O\left(n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}}\right)$ so that for any $0 \leq \delta \leq \epsilon$:*

$$\Pr[C(D_\delta) = 1] < 2^{-n}$$

$$\Pr[C(D_{1-\delta}) = 0] < 2^{-n}$$

Note, breaking from convention, C will NOT have n input bits, but $\tilde{O}(|C|)$ input bits.

Proof. Let $\alpha = \frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}$. Let C_1 AND together $\ln_\epsilon(n^{-1-\alpha})$ bits. Then define C_2 to OR together $n^{1+\alpha}$ independent copies of C_1. Finally, let C_3 AND together $3n$ independent copies of C_2. Then C_3 circuits has $O(n^{2+\alpha})$ gates.

Now for $\delta < \epsilon$, we prove C_3 fail to output 0 on D_δ or 1 on $D_{1-\delta}$ with exponentially low probability. First observe that since C_3 is monotone, increasing the probability a bit is 1 will only increase the probability C_3 outputs 1. That is $\Pr[C_3(D_\delta) = 1] \leq \Pr[C_3(D_\epsilon) = 1]$. Similarly $\Pr[C_3(D_{1-\delta}) = 0] \leq \Pr[C_3(D_{1-\epsilon}) = 0]$. So it suffices to show it distinguishes D_ϵ from $D_{1-\epsilon}$.

First take an input from D_ϵ. Then, the probability a random gate at each level will output 1 is:

- C_1 (AND):

$$\epsilon^{\ln_\epsilon(n^{-1-\alpha})} = \frac{1}{n^{1+\alpha}}$$

- C_2 (OR): See that for $k \geq 2$, $(1 - 1/k) \geq e^{-4/3k}$. Then:

$$1 - \left(1 - \frac{1}{n^{1+\alpha}}\right)^{n^{1+\alpha}} \leq 1 - e^{-4/3} \leq \frac{3}{4}$$

- C_3 (AND):

$$(3/4)^{3n} < 2^{-n}$$

Now take an input from $D_{1-\epsilon}$. Then the probability of C_3 outputting a 0 (or equivalently 1 on the negated circuit with negated input) is:

- C_1 (OR): Notice that by our choice of α that $\alpha = (1+\alpha)\frac{\ln(1-\epsilon)}{\ln(\epsilon)}$. Then:

$$1 - (1-\epsilon)^{\ln_\epsilon(n^{-1-\alpha})} = 1 - e^{\ln(1-\epsilon)\frac{-(1+\alpha)\ln(n)}{\ln(\epsilon)}} = 1 - n^{-\alpha} \leq e^{-n^{-\alpha}}$$

- C_2 (AND):

$$e^{-n^{-\alpha}n^{1+\alpha}} = e^{-n}$$

- C_3 (OR): For $n \geq 12$

$$1 - (1 - e^{-n})^{3n} \leq 3ne^{-n} < 2^{-n}$$

Thus with probability less than 2^{-n} will C_3 fail to distinguish D_ϵ from $D_{1-\epsilon}$ and the size of C_3 is the fan in of C_3 plus the fan in of C_3 times the fan in of C_2, which is $O\left(n^{2+\alpha}\right) = O\left(n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}}\right)$. ◀

Then feeding in random indexes from a Maj_ϵ input and an averaging argument gives an ϵ-promise majority circuit.

▶ **Theorem 26.** *For any $\epsilon \in (0, 1/2)$ there exists a monotone depth-3 circuit solving the ϵ-promise majority problem with size $O\left(n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}}\right)$.*

Proof. Let $\alpha = \frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}$. Take C_3 from Lemma 25. Construct a distribution of circuits C which is C_3 with inputs from C_3 mapped to random bits in our n bit input.

An input into C with ϵn ones is equivalent to an input into C_3 from D_ϵ. Thus for any $x \in \text{Maj}_\epsilon$, C outputs the wrong answer with probability less than 2^{-n}. Thus by an averaging argument, some circuit in the distribution of circuits C always outputs the right answer. Then that circuit solves ϵ-promise majority. ◀

C.1.1 Ajtai's Circuit for Small Promises

We can improve our result for small ϵ by observing we only need to get failure probability below $|\mathrm{Maj}_\epsilon|^{-1}$, which will be more than 2^{-n} for small ϵ. This gives us:

▶ **Theorem 20.** *For any $\epsilon \in (0, 1/2)$, there exists monotone, depth-3 circuits solving the ϵ-promise majority problem with size $O\left((\epsilon \ln(\epsilon))^2 n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}}\right)$.*

Proof. First, I show that

$$|\mathrm{Maj}_\epsilon^0| < 2^{\log(2e/\epsilon)\epsilon n}$$

Consider sampling an element from Maj_ϵ^0 by choosing ϵn indexes out of $(1+\epsilon)n$ bits. For any location chosen from the first n locations will be the locations of the 1 bits. See that this selects from every input in Maj_ϵ^0 with some probability. Further see that this only samples at most $\binom{(1+\epsilon)n}{\epsilon n}$ values. Then we have:

$$|\mathrm{Maj}_\epsilon^0| \leq \binom{(1+\epsilon)n}{\epsilon n} \leq \left(\frac{e(1+\epsilon)n}{\epsilon n}\right)^{\epsilon n} < \left(\frac{2e}{\epsilon}\right)^{\epsilon n} = 2^{\log(2e/\epsilon)\epsilon n}$$

Then we trivially get that $|\mathrm{Maj}_\epsilon| \leq 2^{2\log(2e/\epsilon)\epsilon n}$.

Now using Lemma 25 with $n = 2\log(2e/\epsilon)\,\epsilon n$, we get a circuit C with size

$$O\left((2\epsilon\log(2e/\epsilon))^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}}\, n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}}\right) = O\left((\epsilon\ln(\epsilon))^2 n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}}\right)$$

For convenience, I am assuming $\frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}$ is bounded. If not, then ϵ is close to one half, $\epsilon\ln(\epsilon)$ is constant, and the bound holds by Theorem 26.

So that for any $\delta < \epsilon$

$$\Pr[C(D_\delta) = 1] < 2^{-2\log(2e/\epsilon)\epsilon n} < |Maj_\epsilon|^{-1}$$

$$\Pr[C(D_{1-\delta}) = 0] < 2^{-2\log(2e/\epsilon)\epsilon n} < |Maj_\epsilon|^{-1}$$

Then applying C to random indexes of our input, any $x \in \mathrm{Maj}_\epsilon$ will give the wrong answer with probability less than $|\mathrm{Maj}_\epsilon|^{-1}$. Thus some choice of indexes works for every $c \in \mathrm{Maj}_\epsilon$. Then C at these indexes solves ϵ-promise majority. ◀

Given amplification, our depth 4 upper bounds in Theorem 5 follows in a similar manner.

These marginal improvements tell us that for polynomially small ϵ, depth-3 ϵ-promise majority circuits can be smaller than quadratic. This still leaves an ϵ factor between our lower bounds in Theorem 4 and our best known circuits in Theorem 20.

C.2 Amplification

Now to get our smaller circuits, we need amplify a constant Maj_ϵ input to a $\mathrm{Maj}_{1/\log(n)}$ input.

One way to perform this amplification is to use a randomized construction similar to Ajtai's that has a $1/\mathrm{polylog}(n)$ chance of outputting correctly whether this is a Maj_ϵ^1 input or a Maj_ϵ^0 input. Then we can use a Chernoff bound to show that if we select $n\mathrm{polylog}(n)$ of these circuits at random, with probability all but 2^{-n} a Maj_ϵ^1 input will map to a $\mathrm{Maj}_{1/\mathrm{polylog}(n)}^1$ input and a Maj_ϵ^0 input will map to a $\mathrm{Maj}_{1/\mathrm{polylog}(n)}^0$. Thus some circuit will do that mapping.

Instead, our amplification procedure will be a majority of short walks on an expander. This, combined with a Chernoff bound for expander graphs, will give us our amplification.

Note this solution actually gives a significantly WORSE polylogarithmic overhead than the randomized construction above. But we are ignoring polylogarithmic factors in our result and we can use this same construction in our uniform bounds.

For notation:

▶ **Definition 27** (Random walks). *If G is a graph, let G_t be the length t walks on G. For random walk $w \in G_t$, and $i \in [t]$, let w_i be the ith bit visited in walk w.*

The expander Chernoff bound is given explicitely by Healy [9] as theorem 1:

▶ **Lemma 28.** *For G a regular graph with n vertices and spectral expansion λ and $f : [n] \to [0,1]$ be any function. Let $\mu = \mathbb{E}_{v\in[n]}[f(v)]$. Then for any $\epsilon > 0$:*

$$\Pr_{w\in G_t}\left[\left|\mathbb{E}_{i\in[t]}[f(w_i)] - \mu\right| \geq \epsilon\right] \leq 2e^{-\frac{\epsilon^2(1-\lambda)t}{4}}$$

Then this almost directly gives our amplification .

▶ **Lemma 29.** *For constant $\epsilon \in (0, 1/2)$ and any $\delta \in (0, 1/2)$, there is a P-Uniform, depth-2 monotone circuit, C, outputting $O(n/\delta^{O(1)})$ bits such that:*

$$C\left(Maj^1_\epsilon\right) \subseteq Maj^1_\delta$$

$$C\left(Maj^0_\epsilon\right) \subseteq Maj^0_\delta$$

And C has size

$$|C| = O(n/\delta^{O(1)})$$

Where we allow both constants to depend on ϵ.

Proof. Take a constant degree d regular expander graph G with constant spectral expansion $\lambda < 1$ over the n input bits. That such a graph exists is standard result [7]. Now we just need our walk to be long enough that we have only probability δ that our sample differs from the mean by more than $1/2 - \epsilon$. This happens when

$$t = \frac{4(\ln(2) - \ln(\delta))}{(1/2-\epsilon)^2(1-\lambda)} = \frac{16(\ln(2) - \ln(\delta))}{(1-2\epsilon)^2(1-\lambda)}$$

For notation, let M take an index to value of a bit at that index. See that for an input with at most ϵ fraction of the bits one, $\mathbb{E}_{v\in[n]}[M(v)] = \mu \leq \epsilon$ and:

$$\begin{aligned}
\Pr_{w\in G_t}[\mathbb{E}_{i\in[t]}[M(w_i)] \geq 1/2] &= \Pr_{w\in G_t}[\mathbb{E}_{i\in[t]}[M(w_i)] - \epsilon \geq 1/2 - \epsilon] \\
&\leq \Pr_{w\in G_t}[\mathbb{E}_{i\in[t]}[M(w_i)] - \mu \geq 1/2 - \epsilon] \\
&\leq \Pr_{w\in G_t}\left[\left|\mathbb{E}_{i\in[t]}[M(w_i)] - \mu\right| \geq 1/2 - \epsilon\right] \\
&\leq 2e^{-\frac{(1/2-\epsilon)^2(1-\lambda)t}{4}} \\
&= 2e^{\ln(\delta)-\ln(2)} \\
&= \delta
\end{aligned}$$

Thus with probability at most δ will the majority of the bits in a random walk differ from the majority. Then majority on all random walks on G of length t will give a Maj^0_δ

input. Similarly for an input with at most ϵ fraction 0s. Now we just need to show that there aren't too many walks. See that the number of random walks is:

$$\begin{aligned}|G_t| &= n \cdot d^{t-1} \\ &= n \cdot d^{\frac{4(\ln(2)-\ln(\delta))}{(1/2-\epsilon)^2(1-\lambda)}-1} \\ &= n \cdot d^{\frac{4\ln(2)}{(1/2-\epsilon)^2(1-\lambda)}-1}\left(\frac{1}{\delta}\right)^{\frac{4\ln(d)}{(1/2-\epsilon)^2(1-\lambda)}} \\ &= O(n)\left(\frac{1}{\delta}\right)^{O(1)}\end{aligned}$$

Now for the size of the circuit computing majority for one of these walks. For each of these walks, we have a DNF computing the majority. Any DNF on t variables is of size at most

$$2^t = 2^{\frac{16(\ln(2)-\ln(\delta))}{(1-2\epsilon)^2(1-\lambda)}} = O(1)\,(1/\delta)^{O(1)}$$

Then the total size of the circuit is just that, times the number of walks, which is just $O(n/\delta^{O(1)})$. ◀

And the specific delta we need is inverse poly logarithmic. This immediately gives us Lemma 22.

▶ **Lemma 22.** *For any constant k and $\epsilon \in (0, 1/2)$, there exists P-Uniform, monotone, depth-3 circuits with size $\tilde{O}(n)$ amplifying a Maj^0_ϵ input to a $Maj^0_{\frac{1}{\ln(n)^k}}$ output and a Maj^1_ϵ input to a $Maj^1_{\frac{1}{\ln(n)^k}}$ output.*

C.3 Recursive Majority Proof

We can get sub-quadratic sized circuits for promise majority, down to size $n^{1+\alpha}$ for arbitrarily small α with a constant-depth depending on α. The basic idea is to first amplify the input. Then separate it into n^α sized sections and use our promise majority circuit on each of these. This will give us a new input which satisfies a slightly worse promise but is only of size $n^{1-\alpha}$. Then we can just repeat this process $\log_2(1/\alpha)$ times before the output is a single value which solves our promise majority problem.

This is the iterative equivalent of a recursive divide an conquer technique where we separate the input into $n^{\frac{1+\alpha}{2}}$ groups of size $n^{\frac{1-\alpha}{2}}$ and do our recursive promise majority on each of these groups. Then do a final promise majority on the output. These produce the same circuits and the analysis is almost the same.

This process will work using any circuit computing promise majority as a subroutine as long as we can easily perform whatever amplification it needs. We use $\epsilon = \frac{1}{\ln(n)}$ for our promise majority sub routine. This is convenient, but it could be smaller, even down to an appropriate polynomial as long as we can still do the amplification efficiently.

We first prove the strategy works on a $\mathrm{Maj}_{1/\mathrm{polylog}(n)}$ input, and this combined with amplification solves our promise majority problem.

▶ **Lemma 23.** *If there are depth-3 circuits with size n^α solving $\frac{1}{\ln(n)}$-promise majority, then for any positive integer k, there are depth-$(1+2k)$ circuits solving $\frac{1}{\ln(n)^k}$-promise majority with size*

$$kn^{\frac{1}{1-\left(\frac{\alpha-1}{\alpha}\right)^k}},$$

which is uniform and monotone if the depth-3 circuits are uniform and monotone.

Proof. The proof is by induction. For the base case, $k = 1$, we have by assumption a size-n^α, depth-3 circuit.

In general, for $k \geq 2$, let $\beta = \frac{1}{1-\left(\frac{\alpha-1}{\alpha}\right)^k}$, and $\gamma = \frac{\beta-1}{\alpha-1}$. Separate the input into sets of size n^γ. Then let C_1 be the depth-3 circuit that just computes a $\frac{1}{\ln(n)}$-promise majority on each of these groups. See that the number of groups is $n^{1-\gamma}$. Then the size of C_1 will be:

$$n^{1-\gamma}\left(n^\gamma\right)^\alpha = n^{1-\gamma+\alpha\gamma} = n^{1+(\alpha-1)\frac{\beta-1}{\alpha-1}} = n^\beta$$

By a counting argument, if an input satisfies the $\frac{1}{\ln(n)^k}$ promise, since our promise majority at depth-3 solves the $\frac{1}{\ln(n)}$ promise, the output of C_1 will satisfy a $\frac{1}{\ln(n)^{k-1}}$ promise. This is because fooling one of the promise majority will require $\frac{n^\gamma}{\ln(n)}$ of the minority bits. Since we only have $\frac{n}{\ln(n)^k}$ bits, we can only fool $\frac{n^{1-\gamma}}{\ln(n)^{k-1}}$ groups, which is only a $\frac{1}{\ln(n)^{k-1}}$ fraction of the groups.

Now we use our induction hypothesis on the output of C_1. By assumption, we have a circuit C_2 with depth $(1 + 2(k-1)$ that solves the $\frac{1}{\ln(n)^{k-1}}$-promise majority on $n^{1-\gamma}$ bits and has size $(k-1)n^{\frac{1-\gamma}{1-\left(\frac{\alpha-1}{\alpha}\right)^{k-1}}}$. Then $C_2 \circ C_1$ solves the $\frac{1}{\ln(n)^k}$ promise majority problem. Now for the size of C_2. First we will simplify the number of groups. See that

$$\begin{aligned}
1-\gamma =& 1 - \frac{\beta-1}{\alpha-1} \\
=& \frac{\alpha-\beta}{\alpha-1} \\
=& \frac{\alpha - \frac{1}{1-\left(\frac{\alpha-1}{\alpha}\right)^k}}{\alpha-1} \\
=& \frac{\alpha\left(1-\left(\frac{\alpha-1}{\alpha}\right)^k\right) - 1}{(\alpha-1)\left(1-\left(\frac{\alpha-1}{\alpha}\right)^k\right)} \\
=& \frac{\alpha - 1 - \alpha\left(\frac{\alpha-1}{\alpha}\right)^k}{(\alpha-1)\left(1-\left(\frac{\alpha-1}{\alpha}\right)^k\right)} \\
=& \frac{1-\left(\frac{\alpha-1}{\alpha}\right)^{k-1}}{1-\left(\frac{\alpha-1}{\alpha}\right)^k}
\end{aligned}$$

Then the size of C_2 is

$$\begin{aligned}
(k-1)n^{\frac{1-\gamma}{1-\left(\frac{\alpha-1}{\alpha}\right)^{k-1}}} =& (k-1)\left(n^{\frac{1-\left(\frac{\alpha-1}{\alpha}\right)^{k-1}}{1-\left(\frac{\alpha-1}{\alpha}\right)^k}}\right)^{\frac{1}{1-\left(\frac{\alpha-1}{\alpha}\right)^{k-1}}} \\
=& (k-1)\left(n^{\frac{1}{1-\left(\frac{\alpha-1}{\alpha}\right)^k}}\right) \\
=& (k-1)\left(n^\beta\right)
\end{aligned}$$

Then $C_2 \circ C_1$ has size

$$n^\beta + (k-1)n^\beta = kn^\beta$$

Further, see that for any promise input, if you flip all the bits, you get a promise input that outputs the opposite value. So switching all the AND gates to OR gates and all the OR gates to AND gates still gives a promise majority circuit. This can be seen by negating the circuit twice, once to use De Morgan's law, and once to flip all the input bits.

Thus, we can choose C_1 such that its top level of gates are the same as the bottom level gates of C_2, so they can merge. Thus $C_2 \circ C_1$ has depth $3 + 1 + 2 \cdot (k-1) - 1 = 1 + 2 \cdot k$. So $C_2 \circ C_1$ is our promise majority circuit as we wanted. ◀

D Viola's Promise Majority Circuit

Viola [17] gave a depth-3 circuit ϵ-promise majority circuit where the lowest level has fan in $O\left(\frac{\ln(n)}{\ln(\ln(n))}\right)$ and solves the $\frac{1}{\ln(n)}$ promise majority problem. This combined with the amplification from Lemma 22 gives a constant-depth circuit for ϵ-promise majority.

Normally this would give a depth-4 circuit. But the second level of this circuit, only depends on $O(\ln(n))$ bits, so we can switch them from CNFs to DNFs or DNFs to CNFs with only polynomial overhead. This switching collapses a layer yielding a polynomial size depth-3 circuit.

Unfortunately, the depth-3 circuit is significantly larger than cubic, thus we will consider amplification separately. This leaves us is in much the same situation as the non uniform construction, except that our $\frac{1}{\ln(n)}$-promise promise majority circuit is larger, requiring greater depths to get the same size.

D.1 Improvement on Viola's Promise Majority

First, we will give some intuition for Viola's promise majority. Viola's construction builds off of Lautemann's proof [12] that

$$\text{BPTime}(n) \subseteq \Sigma_2\text{Time}(n^2\text{polylog}(n))$$

We will look at a slight generalization of Lautemann's approach. Let $M : [n] \to \{0,1\}$ simply be the function that takes an index to the value of the input bit at that index. The idea is to take a family of bijections, F, and some collection of tuples of bijections $G \subseteq F^m$ and check if:

$$\exists f_1, ..., f_m \in G : \forall u \in [n] : \exists i \in [m] : M(f_i(u))$$

The idea is that each function mixes the input bits in some way. Then we look at groups of functions as spreading the input by applying all the functions and taking the OR of their outputs. Then we ask, can one of our groups spread the 1s to cover the entire space? If we had less than a $1/m$ fraction of 1s, we can't do this. This is because at best our functions will multiply our number of ones by m.

If we have more than $1/m$ fraction of ones, SOME choice of functions will give us the spread we need. But we can't enumerate over ALL functions. Lautemann used bitwise xor with random strings as his family of functions and used a probabilistic argument to show this works. But every choice of m n bit strings is still too many options, so Viola used random walks on expander graphs to get values to xor with. We take this one step further and let the functions themselves be the random walks.

In the ones case, we argue that some choice of random walk will "spread" the ones to cover the space. The idea is to view the construction of the family of functions as randomized. Then we argue the probability over functions and indexes that we don't cover that index is

less than the number of indexes. Thus, by an averaging argument, some family of functions actually obtains the average, and thus never outputs a 0.

That is, we want to prove

$$\Pr_{f_1,\ldots,f_m \in G, u \in [n]}[\forall i \in [m] : M(f_i(u)) = 0] < 1/n$$

Which implies:

$$\exists f_1, ..., f_m \in G : \forall u \in [n] : \exists i \in [m] : M(f_i(u))$$

▶ **Theorem 24.** *There exists P-uniform, monotone, depth-3, size-$O\left(n^{3+o(1)}\right)$ circuits solving the $\frac{1}{\ln(n)}$-promise majority problem.*

Proof. First, observe that there exists an expander, G, on n vertices (representing the n bits) that is $d = 5\ln(n)$ regular with spectral expansion $\lambda \leq \frac{2}{\sqrt{d}} = \frac{2}{\sqrt{5\ln(n)}}$. Specifically, we will use a bipartite Ramanujan graph as is shown to be constructable by Cohen [7]. To do this we will assume that n is even. Otherwise we can add a 0 bit and our promise will be changed negligibly. We will use the bipartite structure later for defining our random walks in such a way to make our functions bijections.

We want a way to enumerate the paths of length t over both the starting index, and the choice of steps in an independent way. Let G_t be the length t walks over graph G. Let W_d^t be the set of walk strategies for length t walks regardless of origin. That is the choices of edges to take on a length t path in a d regular graph. This will correspond to at every step, giving an index of the next edge to take. We will show later how to index the edges in such a way that each step forms a bijection.

Let $w \in W_d^t$ be a function $w : [n] \to G_t$ that takes a starting vertex and a walk strategy to that walk. Thus we can see that $G_t = \{w(i) : i \in [n], w \in W_d^t\}$, or that the set of all length t walks is equal to all walks from every starting vertex with every walk strategy.

We will let the length of our walks be $t = 2\ln(n)/\ln(\ln(n)) + 1$. Then we have number of walk strategies:

$$|W_d^t| = d^{t-1} = (5\ln(n))^{2\ln(n)/\ln(\ln(n))} = n^{2+o(1)}$$

As above, let M be the function that takes a variable index to the value of that bit in the input. We claim that the proposition

$$\exists w \in W_d^t : \forall u \in [n] : \exists i \in [t] : M(w(u)_i)$$

is true if our input is in $\mathrm{Maj}^1_{1/\ln(n)}$, and false if it is in $\mathrm{Maj}^0_{1/\ln(n)}$. Finally, this statement is computed by a depth-3 circuit with size $O(n^{3+o(1)})$.

0 Input is in $\mathrm{Maj}^0_{1/\ln(n)}$: This comes from a counting argument. But we need our functions, $w(u)_1, ..., w(u)_t$, to be permutations of the vertices, that is bijections. To do this, we make each choice of edge a permutation.
A walk strategy will just be a series of t numbers saying whether to take the first, second, up to dth edge. We label each edge with an index so that every vertex has exactly one edge with each label adjacent to it. Then to take edge i is just to go across the edge labelled i adjacent to the vertex. Then we need d disjoint perfect matchings on our graph. Our graph is regular and bipartite, so it has a perfect matching. Take a perfect matching and label each edge in that matching 1 and remove them. This leaves us with again a bipartite regular graph, so we can get another matching for label 2, and so on until we

have labeled every edge. By definition of perfect matching, each vertex and each label will have exactly one edge incident to that vertex with that label.
Then with this labeling, each step is a permutation, so every length i walk strategy is a permutation. Therefore the number of pairs, (u, i), such that $M(w(u)_i)$ is one is at most

$$\frac{n}{\ln(n)} \cdot t < \frac{n}{\ln(n)} \frac{3\ln(n)}{\ln(\ln(n))} < n$$

Therefore, for some u, there must not be any i so that $M(w(u)_i) = 1$. Thus there does not exist such a walk strategy so that for all u, that strategy starting at u hits a one.

1 Input is in $\mathrm{Maj}^1_{1/\ln(n)}$: As a reminder, our strategy is to prove that the probability of outputting a 0 for every step of a random walk is so low that some strategy doesn't do this on any bit. That is:

$$\Pr_{w \in W_d^t, u \in [n]}[\forall i \in [m] : M(w(u)_i) = 0] < 1/n$$

This comes from the well known result [16] of expander walks that for any expander on n vertices G with spectral expansion λ and any set of vertices A with density $\mu = |A|/n$:

$$\Pr_{w \in G_t}[\forall i \in [t] : w_i \in A] \leq \mu(\mu + (1-\mu)\lambda)^{t-1}$$

In particular, for our graph

$$\begin{aligned}
\Pr_{w \in W_d^t, u \in [n]}[\forall i \in [m] : M(w(u)_i) = 0] = &\Pr_{w \in G_t}[\forall i \in [m] : M(w_i) = 0] \\
\leq &\mu(\mu + (1-\mu)\lambda)^{t-1} \\
< &\left(1/\ln(n) + 2/\sqrt{5\ln(n)}\right)^{t-1} \\
< &\left(1/\sqrt{\ln(n)}\right)^{t-1} \\
< &\left(1/\sqrt{\ln(n)}\right)^{2\ln(n)/\ln(\ln(n))} \\
= &n^{-1}
\end{aligned}$$

By an averaging argument, there must be some walk strategy w so that

$$Pr_{u \in [n]}[\forall i \in [m] : M(w(u)_i) = 0] < n^{-1}$$

But there are only n choices for u. Thus for all u, there is some i so that $M(w(u)_i) = 1$. That is exactly what we needed to prove.

Finally, see the statement represents a small AC0 circuit. This is purely syntactic. Each quantifier is a layer of gates where the fan in is represented by the domain of the variables. 3 quantifiers means 3 layers, and the size is at most the product of number of choices at each level. Thus the size is at most:

$$|W_d^t| \cdot n \cdot t = n^{2+o(1)} \cdot n \cdot (2\ln(n)/\ln(\ln(n)) + 1) = n^{3+o(1)}$$

◀

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il