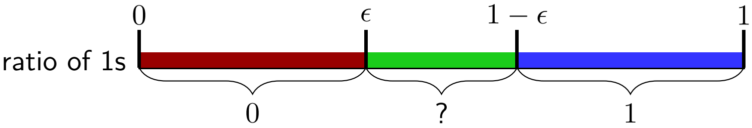
Size Bounds on Low Depth Circuits for Promise Majority

Joshua Cook

The University of Texas at Austin

July 3, 2022

Promise Majority



Approximate majority[1], promise majority[3], approximate selector[2], etc.

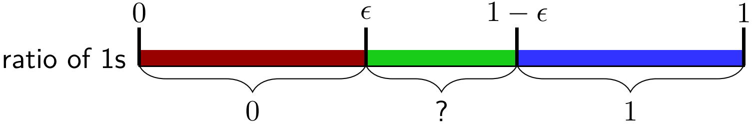
Definition (Promise Majority)

For $n \in \mathbf{N}$, $\epsilon \in (0, 1/2)$, and function $f : \{0,1\}^n \to \{0,1\}$, we say f solves ϵ-promise majority if for all $x \in \{0,1\}^n$ with $\sum_{i\in[n]} x_i < \epsilon n$ and for all $y \in \{0,1\}^n$ with $\sum_{i\in[n]} 1 - y_i < \epsilon n$

$$f(x) = 0, f(y) = 1.$$

- Often usable in place of majority, in circuit derandomization.

Promise Majority



Approximate majority[1], promise majority[3], approximate selector[2], etc.

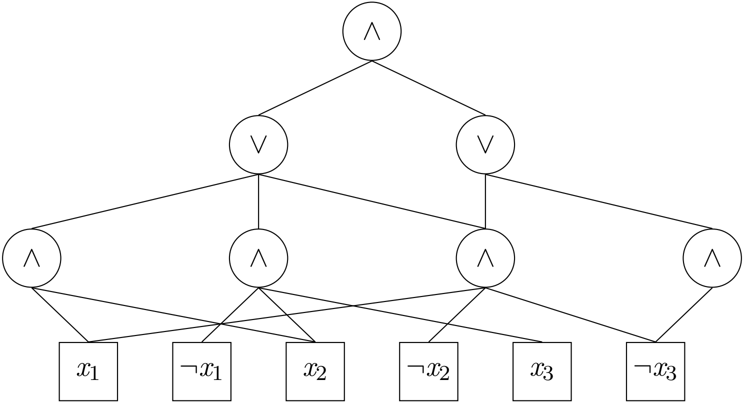
Definition (Promise Majority)

For $n \in \mathbf{N}$, $\epsilon \in (0, 1/2)$, and function $f : \{0,1\}^n \to \{0,1\}$, we say f solves ϵ-promise majority if for all $x \in \{0,1\}^n$ with $\sum_{i\in[n]} x_i < \epsilon n$ and for all $y \in \{0,1\}^n$ with $\sum_{i\in[n]} 1 - y_i < \epsilon n$

$$f(x) = 0, f(y) = 1.$$

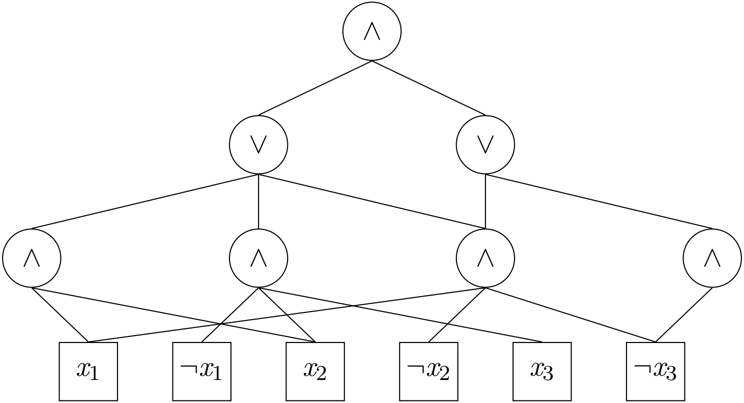
- Often usable in place of majority, in circuit derandomization.
- Widely studied, computable by AC0.

AC0



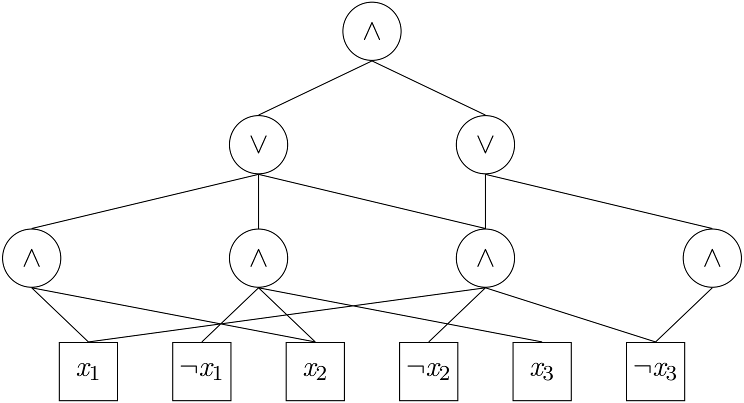
- Alternating circuit: unbounded fan in "AND" and "OR" gates.

AC0



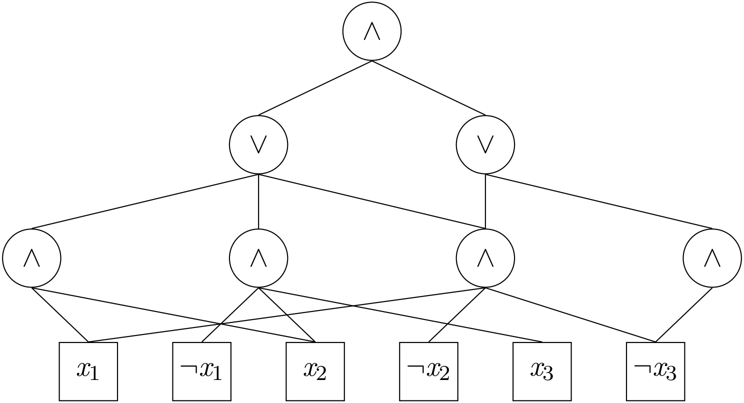
- Alternating circuit: unbounded fan in “AND” and “OR” gates.
- Layers “Alternate” between “AND” and “OR” gates.

AC0



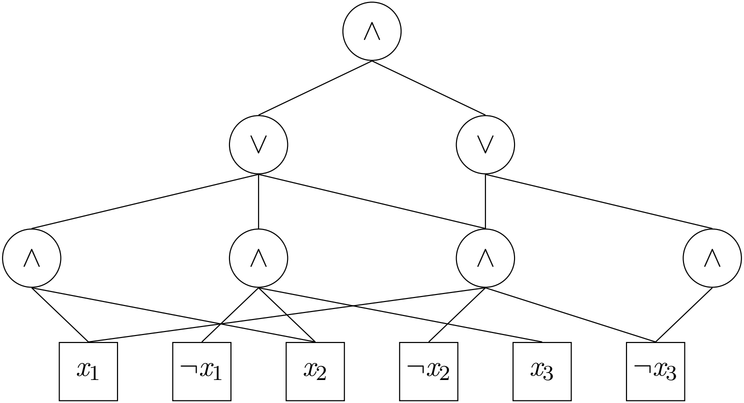
- Alternating circuit: unbounded fan in "AND" and "OR" gates.
- Layers "Alternate" between "AND" and "OR" gates.
- Bottom layer includes negated inputs.

AC0



- Alternating circuit: unbounded fan in "AND" and "OR" gates.
- Layers "Alternate" between "AND" and "OR" gates.
- Bottom layer includes negated inputs.
- Size is number of gates (same results for wires).

AC0



- Alternating circuit: unbounded fan in "AND" and "OR" gates.
- Layers "Alternate" between "AND" and "OR" gates.
- Bottom layer includes negated inputs.
- Size is number of gates (same results for wires).
- AC0 constant depth, polynomial size.

Depth-3 ϵ-Promise Circuit Bounds

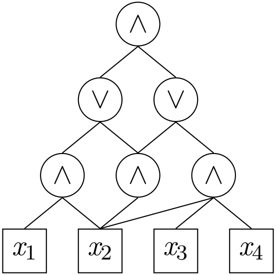
Depth-3 Lower Bounds (Suppressing polylogarithmic factors):

Author	Size	Monotone
Trivial	n	General
Chaudhuri, Radhakrishnan 1996 [2]	$n^{\frac{64}{63}}$	General
Viola 2011 [5]	$n^{\Omega(-\ln(1-2\epsilon))}$	General
Us	$n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)}}$	Monotone
Us	$n^{2+\frac{\ln(1-\epsilon^2)}{2\ln(\epsilon)}}$	General

Circuit Upper Bound by Ajtai 1983 [1]:

$$n^{2+\frac{\ln(1-\epsilon)}{\ln(\epsilon)-\ln(1-\epsilon)}}.$$

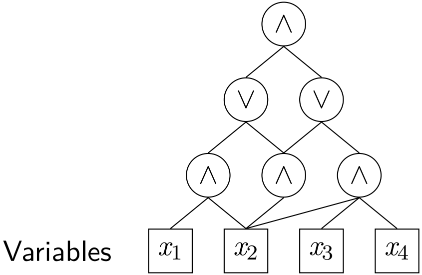
Depth-3 Circuits Terminology



Focus on depth-3 promise Majority

- Negation of promise majority circuit, also promise majority. Assume lowest level gate is “AND”.

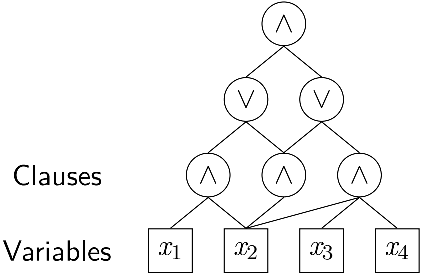
Depth-3 Circuits Terminology



Focus on depth-3 promise Majority

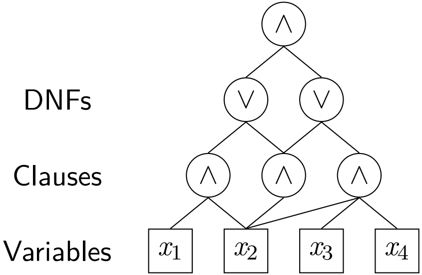
- Negation of promise majority circuit, also promise majority. Assume lowest level gate is "AND".
- Call input bits "variables".

Depth-3 Circuits Terminology



Focus on depth-3 promise Majority

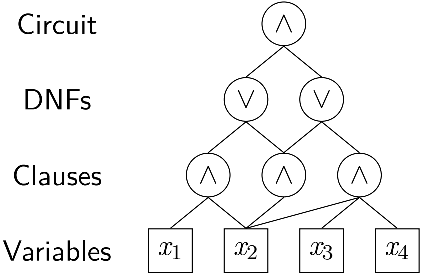
- Negation of promise majority circuit, also promise majority. Assume lowest level gate is “AND”.
- Call input bits “variables”.
- First level, AND gates “clauses”.



Focus on depth-3 promise Majority

- Negation of promise majority circuit, also promise majority. Assume lowest level gate is "AND".
- Call input bits "variables".
- First level, AND gates "clauses".
- Second level, OR gates "DNFs".

Depth-3 Circuits Terminology



Focus on depth-3 promise Majority

- Negation of promise majority circuit, also promise majority. Assume lowest level gate is “AND”.
- Call input bits “variables”.
- First level, AND gates “clauses”.
- Second level, OR gates “DNFs”.
- Third level, AND gate “circuits”.

Biased Coin Distributions

Definition

Let D_ϵ be the distribution on $\{0,1\}^n$ that sets each bit independently to 1 with probability ϵ.

Biased Coin Distributions

Definition

Let D_ϵ be the distribution on $\{0,1\}^n$ that sets each bit independently to 1 with probability ϵ.

Example: $D_{1/3}$ with 3 coins:

outputs	probabilities
111	$\left(\frac{1}{3}\right)^3$
$011, 101, 110$	$\left(\frac{1}{3}\right)^2 \frac{2}{3}$
$100, 010, 001$	$\left(\frac{1}{3}\right)\left(\frac{2}{3}\right)^2$
000	$\left(\frac{2}{3}\right)^3$

By central limit theorem, with probability almost one half, D_ϵ has less than ϵ fraction ones.

Restriction

Definition

We say $\rho \in \{0, 1, *\}^n$ is a restriction on n bits. We say the size of ρ, $|\rho|$, is the number of 1s and 0s in ρ.
If $f : \{0, 1\}^n \rightarrow \{0, 1\}$, then define $f \restriction_\rho$ as the function where the values from ρ are passed into f where it is 1 or 0, and otherwise the corresponding variable from the argument is passed in.

Restriction

Definition

We say $\rho \in \{0, 1, *\}^n$ is a restriction on n bits. We say the size of ρ, $|\rho|$, is the number of 1s and 0s in ρ.
If $f : \{0, 1\}^n \to \{0, 1\}$, then define $f \restriction_\rho$ as the function where the values from ρ are passed into f where it is 1 or 0, and otherwise the corresponding variable from the argument is passed in.

Example:

$$\rho = (1, *, 0, *)$$

$$f \restriction_\rho (x_1, x_2) = f(1, x_1, 0, x_2)$$

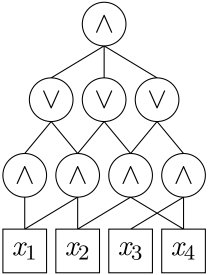
Definition

We say $\rho \in \{0, 1, *\}^n$ is a restriction on n bits. We say the size of ρ, $|\rho|$, is the number of 1s and 0s in ρ.
If $f : \{0, 1\}^n \to \{0, 1\}$, then define $f \restriction_\rho$ as the function where the values from ρ are passed into f where it is 1 or 0, and otherwise the corresponding variable from the argument is passed in.

Example:

$$\rho = (1, *, 0, *)$$

$$f \restriction_\rho (x_1, x_2) = f(1, x_1, 0, x_2)$$



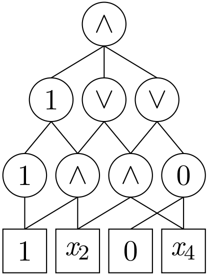
Definition

We say $\rho \in \{0,1,*\}^n$ is a restriction on n bits. We say the size of ρ, $|\rho|$, is the number of 1s and 0s in ρ.
If $f:\{0,1\}^n \to \{0,1\}$, then define $f\restriction_\rho$ as the function where the values from ρ are passed into f where it is 1 or 0, and otherwise the corresponding variable from the argument is passed in.

Example:

$$\rho = (1,*,0,*)$$

$$f\restriction_\rho (x_1, x_2) = f(1, x_1, 0, x_2)$$



Restriction

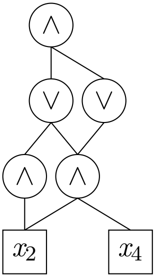
Definition

We say $\rho \in \{0, 1, *\}^n$ is a restriction on n bits. We say the size of ρ, $|\rho|$, is the number of 1s and 0s in ρ.
If $f : \{0, 1\}^n \to \{0, 1\}$, then define $f \restriction_\rho$ as the function where the values from ρ are passed into f where it is 1 or 0, and otherwise the corresponding variable from the argument is passed in.

Example:

$$\rho = (1, *, 0, *)$$

$$f \restriction_\rho (x_1, x_2) = f(1, x_1, 0, x_2)$$



Monotone Lower Bound Idea

Idea: Lower bound the fan in at each layer.
Pretend $\epsilon \in (0, 1/2)$ is constant for simplicity. Let $\alpha = \frac{\epsilon}{\ln(1/\epsilon)}$.

1. From Viola [4], clauses have size $\frac{\ln(n)}{\ln(1/\epsilon)}$.

Monotone Lower Bound Idea

Idea: Lower bound the fan in at each layer.
Pretend $\epsilon \in (0, 1/2)$ is constant for simplicity. Let $\alpha = \frac{\epsilon}{\ln(1/\epsilon)}$.

1. From Viola [4], clauses have size $\frac{\ln(n)}{\ln(1/\epsilon)}$.
2. DNFs have size $\tilde{\Omega}(n^{1+\alpha})$.

Monotone Lower Bound Idea

Idea: Lower bound the fan in at each layer.
Pretend $\epsilon \in (0, 1/2)$ is constant for simplicity. Let $\alpha = \frac{\epsilon}{\ln(1/\epsilon)}$.

1. From Viola [4], clauses have size $\frac{\ln(n)}{\ln(1/\epsilon)}$.
2. DNFs have size $\tilde{\Omega}(n^{1+\alpha})$.
3. Circuit has $\tilde{\Omega}(n^{2+\alpha})$ clauses.

Greedy Set Cover

Theorem

Let $S = \{S_1, ..., S_m\}$ *be subsets of* $[n]$ *where each* $i \in [m]$ *has* $|S_i| \geq w$*. Then for any* $t \in [n]$ *there is some* $T \subseteq [n]$ *with* $|T| = t$ *so that* T *intersects all but at most*

$$|S|e^{-w\frac{t}{n}}$$

of the sets in S*.*

Greedy Set Cover

Theorem

Let $S = \{S_1, ..., S_m\}$ *be subsets of* $[n]$ *where each* $i \in [m]$ *has* $|S_i| \geq w$*. Then for any* $t \in [n]$ *there is some* $T \subseteq [n]$ *with* $|T| = t$ *so that* T *intersects all but at most*

$$|S|e^{-w\frac{t}{n}}$$

of the sets in S*.*

In particular, if

- S is the set of clauses in a monotone DNF, F, and
- ρ is some restriction restricting variables in T to 0,

then $|F\restriction_\rho| \leq |F|e^{-w\frac{t}{n}}$ variables remaining.

Monotone DNF Size

Theorem

Let $\epsilon \in (0, 1/2)$ *and monotone DNF* F *be such that*

- *For all x with less than* ϵn *zeros,* $F(x) = 1$*.*
- $\Pr[F(D_\epsilon) = 0] \geq$ *poly*$(1/n)$.

Then F has $n^{1+\alpha}$ *clauses for some* $\alpha = \Omega(\frac{\epsilon}{\ln(1/\epsilon)})$.

All DNFs in circuit must satisfy condition 1.

But For DNF to “help” by much, it must satisfy condition 2.

Monotone Circuit Size Lower Bounds

Theorem

Depth-3 Circuit C solving ϵ-promise majority has size $n^{2+\Omega\left(\frac{\epsilon}{\ln(1/\epsilon)}\right)}$.

Idea: Eliminate many DNFs with few clauses.

Can eliminate too many DNFs if there are not enough clauses.

Monotone Circuit Lower Bound Proof Idea

- Remove large clauses.
- Use DNF lower bounds to get each DNF bigger than $n^{1+\alpha}$.
- Fix whole clauses to apply set cover on DNFs.

Monotone Circuit Lower Bound Proof Idea

- Remove large clauses.
- Use DNF lower bounds to get each DNF bigger than $n^{1+\alpha}$.
- Fix whole clauses to apply set cover on DNFs.
- If there are few clauses, DNFs must share some clauses many times.
- Must be many clauses or many DNFs.

Monotone Circuit Lower Bound Proof Idea

- Remove large clauses.
- Use DNF lower bounds to get each DNF bigger than $n^{1+\alpha}$.
- Fix whole clauses to apply set cover on DNFs.
- If there are few clauses, DNFs must share some clauses many times.
- Must be many clauses or many DNFs.

Issue: Some DNFs might be small.

Monotone Circuit Lower Bound Proof Idea

- Remove large clauses.
- Use DNF lower bounds to get each DNF bigger than $n^{1+\alpha}$.
- Fix whole clauses to apply set cover on DNFs.
- If there are few clauses, DNFs must share some clauses many times.
- Must be many clauses or many DNFs.

Issue: Some DNFs might be small.

Solution: Focus on large DNFs during elimination.

Insight: Some large DNF must survive if few variables fixed.

Non Monotone Lower Bound Overview

Monotone Idea: Bound size at each level, using restrictions from set cover algorithm.

General Idea: Same!

Non Monotone Lower Bound Overview

Monotone Idea: Bound size at each level, using restrictions from set cover algorithm.

General Idea: Same!

- Clause lower bounds, works!

Non Monotone Lower Bound Overview

Monotone Idea: Bound size at each level, using restrictions from set cover algorithm.

General Idea: Same!

- Clause lower bounds, works!
- DNF lower bounds, *almost* works.

Non Monotone Lower Bound Overview

Monotone Idea: Bound size at each level, using restrictions from set cover algorithm.

General Idea: Same!

- Clause lower bounds, works!
- DNF lower bounds, *almost* works.

  Following first proof, may set DNF to one early due to negations.
  Then, can't argue restriction left any clauses.
  Will discuss next.

Non Monotone Lower Bound Overview

Monotone Idea: Bound size at each level, using restrictions from set cover algorithm.

General Idea: Same!

- Clause lower bounds, works!
- DNF lower bounds, *almost* works.

  Following first proof, may set DNF to one early due to negations.
  Then, can't argue restriction left any clauses.
  Will discuss next.
- Circuit lower bounds, works!

Non Monotone Lower Bound Overview

Monotone Idea: Bound size at each level, using restrictions from set cover algorithm.

General Idea: Same!

- Clause lower bounds, works!
- DNF lower bounds, *almost* works.

  Following first proof, may set DNF to one early due to negations. Then, can't argue restriction left any clauses.
  Will discuss next.
- Circuit lower bounds, works!
  - At worst, might eliminate or shrink DNFs and clauses early.
  - But circuit still solves a promise problem, so it still has large DNFs after restriction.

Probabilistic Restriction Idea

Idea: Define sequence of restrictions, each restricting one more variable such that:

Probabilistic Restriction Idea

Idea: Define sequence of restrictions, each restricting one more variable such that:

- Each restriction in the sequence adds one more restriction, sampled from D_ϵ.

Probabilistic Restriction Idea

Idea: Define sequence of restrictions, each restricting one more variable such that:

- Each restriction in the sequence adds one more restriction, sampled from D_ϵ.
- Each restriction has a good chance of eliminating many clauses.

Probabilistic Restriction Idea

Idea: Define sequence of restrictions, each restricting one more variable such that:

- Each restriction in the sequence adds one more restriction, sampled from D_ϵ.
- Each restriction has a good chance of eliminating many clauses.
- Focuses on deleting clauses bigger then w.

Probabilistic Restriction Idea

Idea: Define sequence of restrictions, each restricting one more variable such that:

- Each restriction in the sequence adds one more restriction, sampled from D_ϵ.
- Each restriction has a good chance of eliminating many clauses.
- Focuses on deleting clauses bigger then w.

Use greedy set cover algorithm to choose variables like monotone case.

Instead of just setting them to 0, we set them to 1 with probability ϵ.

Probabilistic Restriction Idea

Idea: Define sequence of restrictions, each restricting one more variable such that:

- Each restriction in the sequence adds one more restriction, sampled from D_ϵ.
- Each restriction has a good chance of eliminating many clauses.
- Focuses on deleting clauses bigger then w.

Use greedy set cover algorithm to choose variables like monotone case.

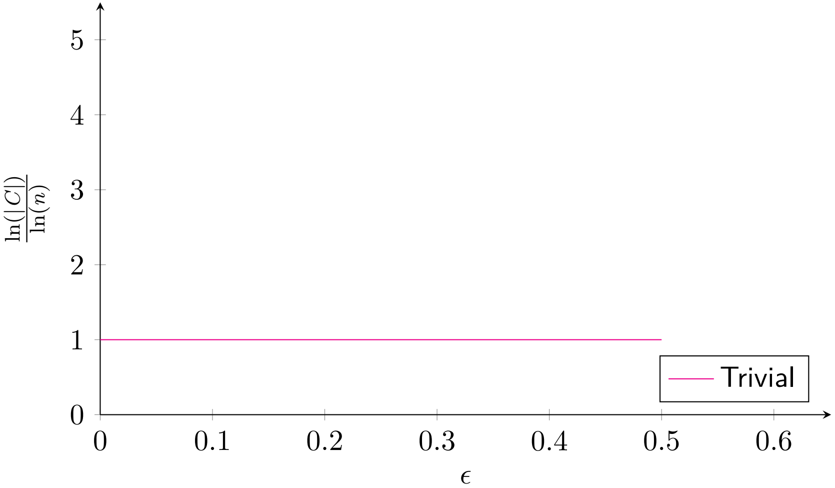
Instead of just setting them to 0, we set them to 1 with probability ϵ.

Then by Chernoff bounds, its likely that we eliminate many clauses.

By definition, restricting the rest of the variables is the same as using D_ϵ.

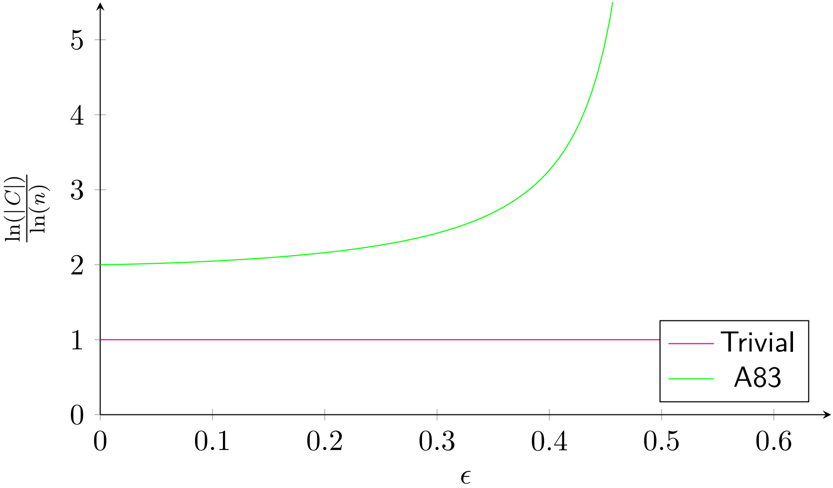
Depth-3 Bounds, Constant ϵ

Note: Graphs *slightly* adjusted for visibility.
Y-axis is c for circuit size n^c.



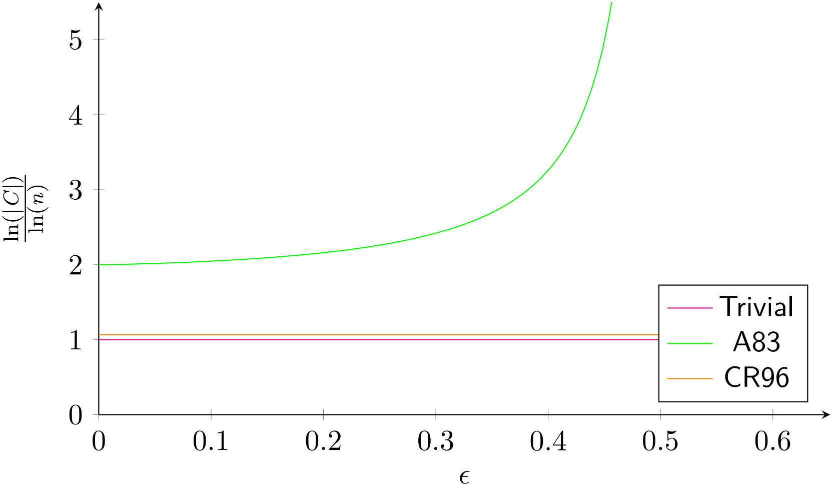
Depth-3 Bounds, Constant ϵ

Note: Graphs *slightly* adjusted for visibility.
Y-axis is c for circuit size n^c.



Depth-3 Bounds, Constant ϵ

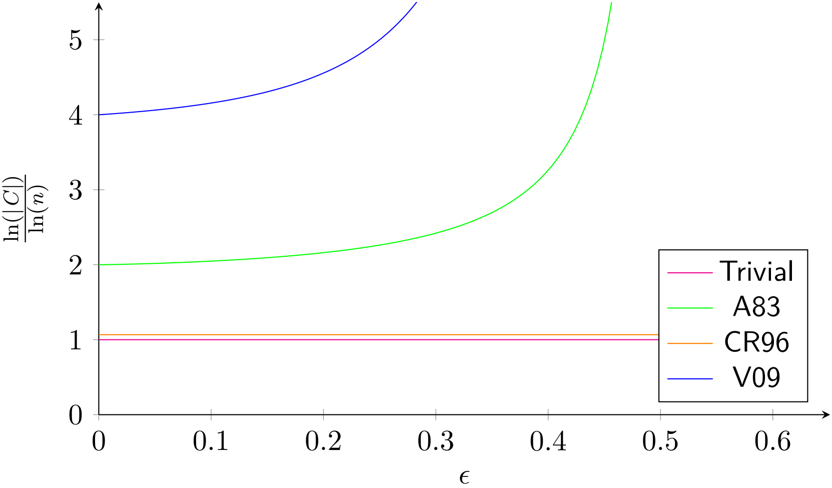
Note: Graphs *slightly* adjusted for visibility.
Y-axis is c for circuit size n^c.



Depth-3 Bounds, Constant ϵ

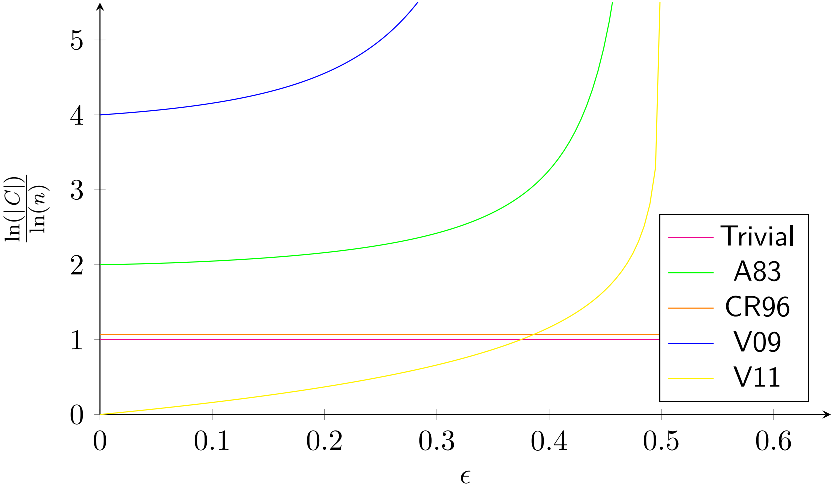
Note: Graphs *slightly* adjusted for visibility.

Y-axis is c for circuit size n^c.

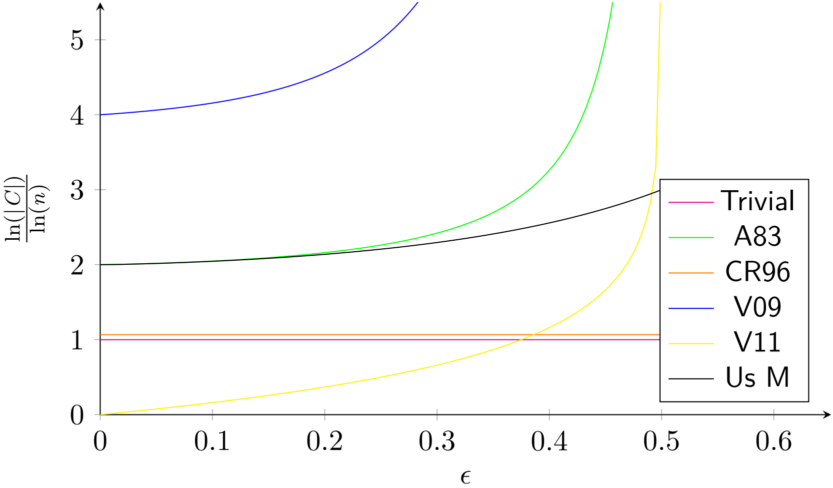


Depth-3 Bounds, Constant ϵ

Note: Graphs *slightly* adjusted for visibility.
Y-axis is c for circuit size n^c.

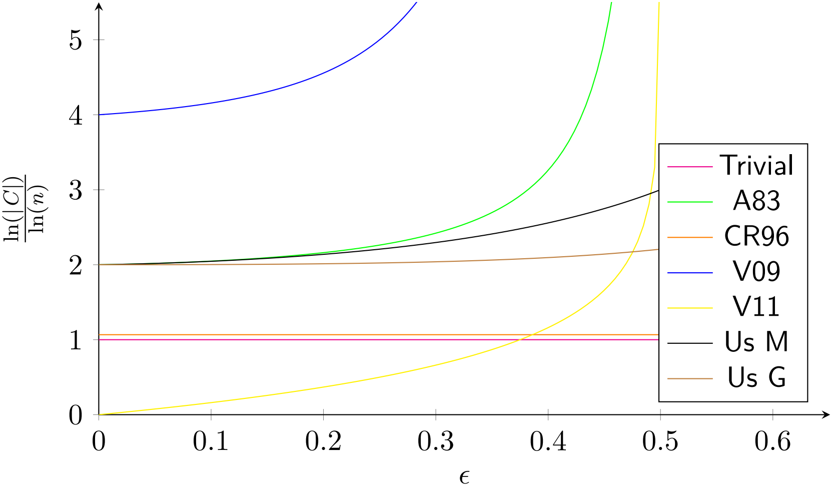


Note: Graphs *slightly* adjusted for visibility.
Y-axis is c for circuit size n^c.



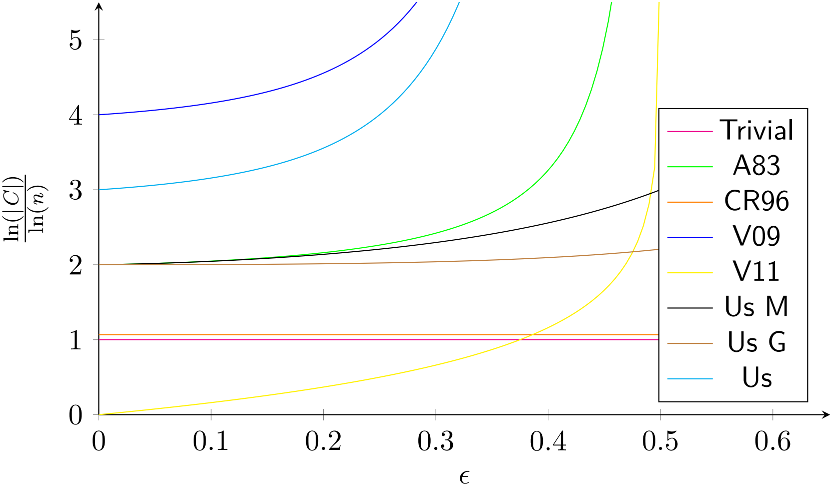
Depth-3 Bounds, Constant ϵ

Note: Graphs *slightly* adjusted for visibility.
Y-axis is c for circuit size n^c.



Depth-3 Bounds, Constant ϵ

Note: Graphs *slightly* adjusted for visibility.
Y-axis is c for circuit size n^c.



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

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.

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.

References II

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

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