Contents

CSC2429 / MAT1304 October 4, 2020

Lecture : Circuit Complexity

Instructor: Benjamin Rossman *Scribe: Lily Li*

1 Administrivia

- **Instructor:** Ben Rossman.
- **Course Info:** Available at the course website. Just in case the website is down: lectures are Thursdays from 16:00 to 18:00 in Bahen B026. Office hours are by appointment.
- **Textbook:** *Boolean Function Complexity* by Stasys Junkha. This is available as a free eBook through the University of Toronto library.
- **Prerequisites:** None, but a previous complexity course is useful. Please read Appendix A.1 of the textbook and understand the material.
- **Workload:** Homework assignment(s), scribe notes, paper report (5 to 10 pages), and presentation if you so choose. No exams.

2 Basic Definitions

2.1 Boolean Functions

Definition 1. *A n-**ary Boolean function** f is a function of the form $f : \{0,1\}^n \to \{0,1\}$. Usually we interpret $(0,1)$ as $(\mathsf{FALSE}, \mathsf{TRUE})$ or as $(1,-1)$ — this makes sense if you think of it as $(-1)^0$ and $(-1)^1$.*

Let $\{0,1\}^* = \cup_{n\in\mathbb{N}}\{0,1\}^n$. We typically refer to a family of Boolean function(s) $f : \{0,1\}^* \to \{0,1\}$. This corresponds to a sequence of functions $f_n : \{0,1\}^n \to \{0,1\}$ and to a language $L \subseteq \{0,1\}^*$ described by its characteristic function $f_L : \{0,1\}^* \to \{0,1\}$.

Example 2. *The following are some examples of n-ary Boolean functions:*

1. $\mathrm{PARITY}(x_1, ..., x_n) = \sum_{i=1}^n x_i \mod 2$.
2. $\mathrm{MOD}_p(x_1, ..., x_n) = 1 \iff \sum_{i=1}^n x_i \equiv 0 \mod p$.
3. $\mathrm{MAJORITY}_n(x_1, ..., x_n) = 1 \iff \sum_{i=1}^n x_i \geq \lceil n/2 \rceil$.
4. $k - \mathrm{CLIQUE} : \{0,1\}^{\binom{n}{2}} \to \{0,1\}$. *Think of each graph G as an indicator vector $\mathbb{1}_G$ over its $\binom{n}{2}$ edges. Then $k - \mathrm{CLIQUE}(\mathbb{1}_G) = 1$ if and only if G has a k-clique.*

Let us consider DeMorgan circuits. These contain logical connectives $\{\vee, \wedge, \neg\}$, input variables $\{x_1, ..., x_n\}$, and constants $\{0, 1\}$.

Definition 3. *A n-**ary DeMorgan circuit** is a finite directed acyclic graph (DAG) with nodes labelled as follows:*

- *Nodes of in-degree zero ("inputs") are labelled by a variable or a constant.*
- *Non-input nodes ("gates") of in-degree one are labelled with $\neg$. Gates of in-degree two are labelled with $\vee$ or $\wedge$.*
- *A subset of the nodes are designated as "outputs" (default:* the *node with out-degree zero).*

*Two circuits are **equivalent** if they compute the same function.*

Formulas are tree-like circuits. Since different branches in a formula depend on different copies of the variables, formulas are memory-less. See Figure 1. Proving that formulas are polynomially weaker than circuits is still an open problem.

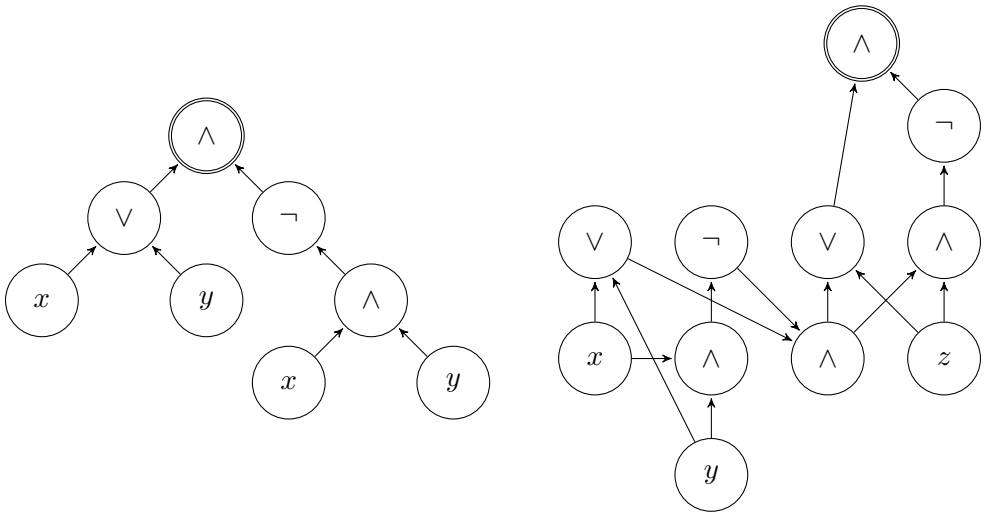

Figure 1: (Left) Formula computing $x \oplus y$. (Right) Circuit computing $x \oplus y \oplus z$.

Definition 4. *The **size** of a circuit is the number of $\vee$ and $\wedge$ gates it contains.*

*The **leaf-size** of a formula is the number of leaves in its associated DAG. This is one more than the circuit size as defined above.*

*The **circuit size** of an n-ary Boolean function $f : \{0,1\}^n \to \{0,1\}$, written $\mathcal{C}(f)$, is the minimum size of a circuit computing f. Similarly, the **formula (leaf) size** of f, written $\mathcal{L}(f)$, is the minimum size of a formula computing f.*

*The **depth** of a circuit is the maximum number of $\wedge$ and $\vee$ gates on any input to output path.*

Example 5. *It is a major open problem to compute the circuit and leaf size lower bounds for various Boolean functions. A couple of known results are as follows.*

f	$\mathcal{L}(f)$	$\mathcal{C}(f)$
AND_n	n	$n-1$
$PARITY_n$	$\Theta\left(n^2\right)$	$3(n-1)$

The results for AND_n are tight since the output depends on all the inputs. Improving the gap size between $\mathcal{L}(PARITY_n)$ and $\mathcal{C}(PARITY_n)$ would separate NC_1 from P.

2.2 Other Ways of Measuring Size

Other ways of counting the size of a circuit include: (1) counting the number of wires and (2) counting all gate types (including $\neg$ gates). It turns out that the result of these calculations differ from our definition above by at most a factor of two. It should be easy to see why this is in the former case. Every $\wedge$ and $\vee$ gate has two incoming wires. Claim 7 shows this in the latter case.

Definition 6. *The input to every $\neg$ gate in a circuit in **negation normal form** is a variable. See Figure 2.*

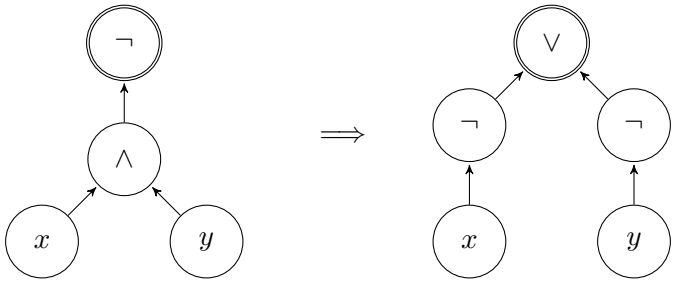

Figure 2: Apply DeMorgan's Law to all $\neg$ gates whose inputs are not literals on the left circuit to get the equivalent right circuit in negation normal form.

Claim 7. *Every circuit C of size m is equivalent to a circuit in negation normal form of size $\leq 2m$.*

Proof. Apply DeMorgan's law to every $\neg$ gate whose input is not a variable. This switches the order of $\neg$ and $\wedge/\vee$ in the DAG and adds an additional $\neg$ gate. By the end of the process we have added at most m $\neg$ gates. □

Thus we can push all $\neg$ gates to the bottom and interpret the inputs as literals (variables and their negation). We can also modify the definition of leaf-size to only count leaves leading to literals (never-mind the constants).

2.3 General Basis

A **basis** B is a set of Boolean functions (or "gate types"). Examples of basis include:

- DeMorgan basis: $\{\wedge, \vee, \neg\}$.

- Full binary basis: all Boolean functions $\{0,1\}^2 \to \{0,1\}$ (for example, you would get $\oplus$).
- Monotone basis: $\{\wedge, \vee\}$ (NOT universal).
- AC^0 basis: $\{\wedge^k, \vee^k, \neg : k \in \mathbb{N}\}$ which are unbounded fan-in $\wedge$ and $\vee$ gates.

For a function f, let $\mathcal{L}_B(f)$ and $\mathcal{C}_B(f)$ be the leaf and circuit size of f with formulas and circuits built from gates of basis B. A basis is **universal** if it computes all functions. For two universal basis B_1 and B_2 it is possible to build a circuit using gates from B_1 which simulates any gate from B_2. If all functions in B_1 and B_2 have constant arity, it follows that $\mathcal{C}_{B_2}(f) = O(\mathcal{C}_{B_1}(f))$; for formula size, the relation is $\mathcal{L}_{B_2}(f) = \mathcal{L}_{B_1}(f)^{O(1)}$. This polynomial blow-up is unavoidable in some cases. Recall the function $PARITY_n$: $\mathcal{L}_{\{\wedge,\vee,\neg\}}(PARITY_n) = \Theta(n^2)$ whereas $\mathcal{L}_{\{\oplus\}}(PARITY_n) = n - 1$.

2.4 Models of Computation

Definition 8. *A **uniform model of computation** is a single machine/program with a finite description which operates on all inputs in $\{0,1\}^*$. Examples range from simple finite automata (where we have lower bounds ala the pumping lemma) to complex Turing Machines (lower bounds much harder to come by).*

*Recall that a language $L \subseteq \{0,1\}^*$ can be interpreted as a sequence of functions $(f_0, f_1, ...)$ where $f_n : \{0,1\}^n \to \{0,1\}$ and $f_n(\mathbf{x}) = 1 \iff \mathbf{x} \in L$ for any $\mathbf{x} \in \{0,1\}^n$. A **non-uniform (concrete) model of computation** is a sequence $(C_0, C_1, ...)$ of combinatorial objects (namely circuits) where C_n computes f_n. Examples include: circuits in the DeMorgan basis, restricted class of circuits (formulas, monotone model), decision trees, etc.*

Observe that the non-uniform model of computation is more powerful than the uniform one since the finite program can be used as every combinatorial objects in the sequence. It follows that lower bounds in the non-uniform model also imply lower bounds in the uniform model. While upper bounds in the uniform model imply upper bounds in the non-uniform model. *We want: unconditional lower bounds.*

Circuits efficiently simulate Turing Machines.

Lemma 9. *Any Turing Machine (TM) M with running time $t(n)$ can be simulated by a circuit (family of) of size $O\left(t(n)^2\right)$.*

Exercise for the reader. *Hint: think about configurations of the Turing Machines as a $t(n) \times t(n)$ grid and construct a circuit for every grid cell.* Fischer and Pipenger (1979) proved an $O(t(n) \log t(n))$ upper bound on *oblivious Turing Machines*[1]. It is unknown if we can do better.

Corollary 10. *If there is a super polynomial lower-bound (better than $\Omega(n^c)$ for all constants $c > 0$) on the circuit size of any language in NP, then $\mathsf{P} \neq \mathsf{NP}$.*

Finding the lower-bound would actually show $\mathsf{NP} \not\subseteq \mathsf{P/poly}$ where $\mathsf{P/poly}$ is the class of languages decidable by poly(n)-size circuits.

[1] An oblivious TM is one whose head motion depends only on the size of the input and not its particular bits. Take a look at this blog post for some entertainment.

We will see some polynomial lower bounds for formulas in the DeMorgan basis later on. As a historical curio, the following is a catalogue of lower bound results for an explicit Boolean function:

1. $\Omega(n^{1.5})$ Subboboskay '61
2. $\Omega(n^2)$ Khrapchenko '71
3. $\Omega(n^{2.5-o(1)})$ Andreev '83
4. $\Omega(n^{3-o(1)})$ Håstad '98 (this is the state of the art until very recently).

3 DeMorgan Basis

3.1 Balancing Formulas

Next we consider the relationship between circuit size and depth. First observe that every circuit of depth d is equivalent to a formula of size at most 2^d. To see this, take the circuit and duplicate any branches that gets reused. The resulting binary tree has at most as many nodes as a perfect binary tree of depth d which itself has circuit size 2^d.

The next theorem shows the converse: every formula of size s can be "balanced" to obtain a formula of depth $O(\log s)$.

Theorem 11. ***(Spira 1971).*** *Every formula with leaf-size s is equivalent to a formula of depth $O(\log s)$ ($2\log_{3/2}(s)$ to be exact) and thus size at most $s^{O(1)}$ ($s^{2/\log_2(3/2)}$).*

Proof. By induction on s. The base case is trivial. Let F be the original formula and g be some gate. Let F_g be the sub-formula rooted at g. For $b \in \{0,1\}$, let $F^{(g \leftarrow b)}$ be the formula with g replaced with the constant value b. See Figure 3. Note that $\mathcal{L}(F) = \mathcal{L}(F_g) + \mathcal{L}\left(F^{(g \leftarrow b)}\right)$. Minimize $\mathcal{L}(F)$ by balancing the two terms on the RHS. By Claim 12, we can find a gate g such that $\frac{s}{3} \leq \mathcal{L}(F_g) \leq \frac{2s}{3}$.

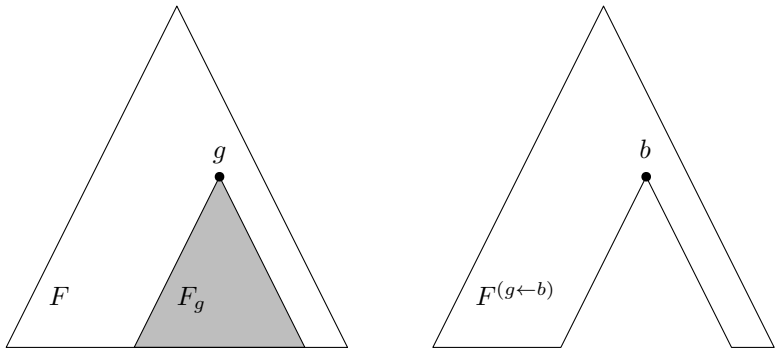

Figure 3: Illustration of gate g and formulas F_g and $F^{g \leftarrow b}$.

Note that $F \equiv \left(F_g \wedge F^{(g \leftarrow 1)}\right) \vee \left((\neg F_g) \wedge F^{(g \leftarrow 0)}\right)$; F_g must evaluate to 0 or 1 and the formula does just that. Apply the induction hypothesis to the four formulas F_g, $F^{(g \leftarrow 1)}$, $\neg F_g$, and $F^{(g \leftarrow 0)}$ to get

formulas of depth $\leq 2\log_{3/2}(2s/3)$. The original formula F can only grow by at most depth two so

$$\begin{aligned}\text{depth}(F) &\leq \max\left\{\text{depth}\left(F_g\right), \text{depth}\left(F^{(g\leftarrow 1)}\right), \text{depth}\left(\neg F_g\right), \text{depth}\left(F^{(g\leftarrow 0)}\right)\right\} + 2 \\ &\leq 2\log_{3/2}\frac{2s}{3} + 2 \\ &= (2\log_{3/2} s - 2) + 2 \\ &= 2\log_{3/2} s\end{aligned}$$

Thus there exists a formula equivalent to F of depth at most $O(\log s)$. □

Claim 12. *There exists a gate g such that F_g has leaf-size between $\frac{s}{3}$ and $\frac{2s}{3}$ leaves.*

Proof. Let $r \rightsquigarrow \ell$ be a root to leaf path in the DAG containing the most $\wedge/\vee$ gates. At the root r, $\mathcal{L}(F_r) = \mathcal{L}(F) = s$ and at the leaf ℓ, $\mathcal{L}(F_\ell) = 1$. Starting at r and moving down to ℓ, the successive leaf-sizes can at most halve after each step. Thus there must exists a gate g for which $\frac{s}{3} \leq \mathcal{L}(F_g) \leq \frac{2s}{3}$. □

3.2 Circuit Size of General Boolean Functions

3.2.1 Upper Bound

Given a function $f : \{0,1\}^n \to \{0,1\}$, let us consider some upper bounds for $\mathcal{C}(f)$.

1. Brute force DNF: $O(n2^n)$. There are 2^n rows in the truth table of f. Each row specifies the output given the n inputs. Thus a clause with $n-1$ $\wedge$ gates represents each row of the table. Formally we consider the expression

$$f(\mathbf{x}) = \bigvee_{\mathbf{a}\in f^{-1}(1)} (\mathbf{x} = \mathbf{a}) = \bigvee_{\mathbf{a}\in\in f^{-1}(1)} (l_1 \wedge l_2 \wedge \cdots \wedge l_{n-1} \wedge l_n)$$

   where $l_i = x_i$ if $a_i = 1$ and $l_i = \overline{x}_i$ otherwise.

2. Function decomposition: $O(2^n)$. Observe that

$$f(\mathbf{x}) \equiv (x_n \wedge f_1(\mathbf{x})) \vee (\overline{x}_n \wedge f_0(\mathbf{x}))\,.$$

   where $f_1 = f(x_1, ..., x_{n-1}, 1)$ and $f_0 = f(x_1, ..., x_{n-1}, 0)$. Thus

$$\mathcal{C}(f) \leq \mathcal{C}(f_1) + \mathcal{C}(f_0) + 3.$$

   Apply the decomposition recursively to f_1 and f_0. Generally at step k,

$$\mathcal{C}(f) \leq \sum_{\mathbf{a}\in\{0,1\}^k} \mathcal{C}(f_\mathbf{a}) + 3(2^k - 1)$$

   where $f_\mathbf{a}(\mathbf{x}) = f(x_1, ..., x_{n-k}, a_1, ..., a_k)$. Since $f(\mathbf{a})$ is a constant at the n^{th} step, $3(2^n - 1)$ is an upper bound on the circuit size of f.

3. Computation reuse: $O(2^n/n)$. See Theorem 14 below.

Let $ALL_n : \{0,1\}^n \to \{0,1\}^{2^{2^n}}$ be the function which calculates all the n-ary Boolean functions at the same time[2]. That is $(ALL_n(\mathbf{x}))_f \coloneqq f(\mathbf{x})$ for any n-ary Boolean function f.

Claim 13. $\mathcal{C}(ALL_n) \leq O(2^{2^n})$.

Proof. Similar to the function decomposition analysis. For every function f in the output of ALL_n, $f(\mathbf{x}) \equiv (x_n \wedge f_1(\mathbf{x})) \vee (\overline{x}_n \wedge f_0(\mathbf{x}))$ where $f_1 = f(x_1, ..., x_{n-1}, 1)$ and $f_0 = f(x_1, ..., x_{n-1}, 0)$. Note that f_1 and f_0 are outputs of ALL_{n-1}. See Figure 4. Since ALL_n has 2^{2^n} outputs,

$$\mathcal{C}(ALL_n) \leq \mathcal{C}(ALL_{n-1}) + 3\left(2^{2^n}\right) = c\left(2^{2^{n-1}}\right) + 3\left(2^{2^n}\right) \in O\left(2^{2^n}\right)$$

for some constant c. □

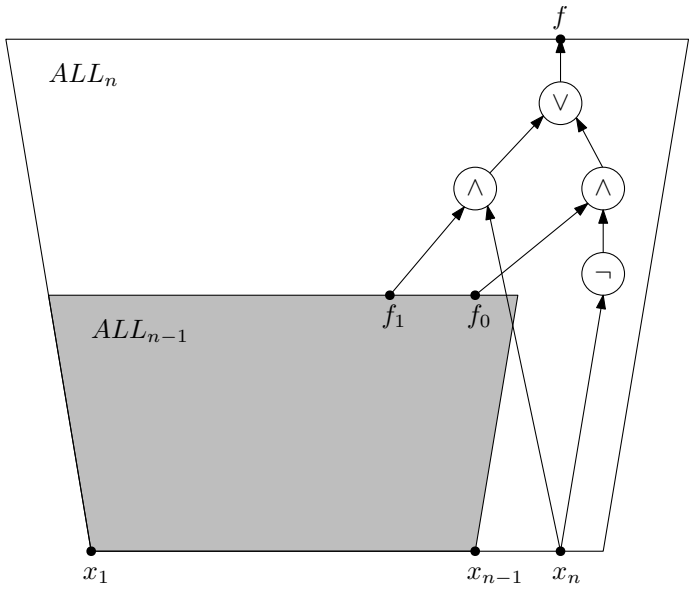

Figure 4: Obtaining a circuit for ALL_n from a circuit for ALL_{n-1}.

Theorem 14. ***(Lupanov 1958).*** *Every n-ary Boolean function has circuit size $O(2^n/n)$*

Proof. The key idea is to use ALL_{n-k} in place of $\{f_{\mathbf{a}} : \mathbf{a} \in \{0,1\}^k\}$ in the analysis of function decomposition. Formally, we have

$$\mathcal{C}(f) \leq \sum_{\mathbf{a} \in \{0,1\}^k} \mathcal{C}(f_{\mathbf{a}}) + 3(2^k - 1) \leq \mathcal{C}(ALL_{n-k}) + 3(2^k - 1) \leq O\left(2^{2^{n-k}}\right) + O(2^k)$$

[2]To see that the range of ALL_n is indeed 2^{2^n}, recall that the domain of every n-ary Boolean function is 2^n. There is a bijection between the set of functions and the power set of $\{0,1\}^n$ (of size 2^{2^n}).

where the last inequality follows from Claim 13. Observe that the two terms on the RHS are balanced when $k = n - \log(n - \log n)$ since

$$\begin{aligned} O\left(2^{2^{n-k}}\right) + O\left(2^k\right) &= O\left(2^{2^{\log(n-\log n)}}\right) + O\left(2^{n-\log(n-\log n)}\right) \\ &= O\left(2^{n-\log n}\right) \\ &= O\left(2^n/n\right) \end{aligned}$$

It follows that the circuit complexity of all n-ary Boolean function is bounded above by $O(2^n/n)$. □

3.2.2 Lower Bound

Prior to Lupanov's result above, Shannon showed a matching lower bound.

Theorem 15. ***(Shannon 1949).*** *Almost all n-ary Boolean functions (as $n \to \infty$) have circuit size $O(2^n/n)$.*

Proof. Use the counting argument. Recall the number of n-ary Boolean functions is 2^{2^n} and let $s = \frac{2^n}{n}$. We will show that the number of Boolean functions which can be computed by circuits of size s is $\ll 2^{2^n}$. Let A be the set of all n-ary circuits with $2n$ literals, $x_1, ..., x_n, \overline{x}_1, ..., \overline{x}_n$, and s gates, denoted $g_1, ..., g_s$. We obtain an upper bound on the number of circuits in A as follows. Each circuit can use any subset of the s gates. Each $\wedge/\vee$ gate can pick two inputs from the $2n$ literals and $s-1$ other gates. If n is sufficiently large (say $n \geq 100$), then $s + 2n < 3s$ so

$$|A| \leq 2^s (s+2n)^{2s} \leq 18^s s^{2s}.$$

Observe that every n-ary function with $\mathcal{C}(f) \leq s$ is computed by at least $s!$ distinct circuits in A since we can permute the labels on the s gates. Thus the total number of Boolean functions computed by circuits in A is at most $\frac{|A|}{s!}$. Recall that $s! \geq \left(\frac{s}{e}\right)^s$. For $s = \frac{2^n}{n}$,

$$\frac{|A|}{s!} \leq \frac{18^s s^{2s}}{(s/e)^s} \leq 50^s s^s = 50^{2^n/n} \left(\frac{2^n}{n}\right)^{2^n/n} = \left(\frac{50}{n}\right)^{2^n/n} (2^n)^{2^n/n} \leq 2^{2^n - 2^n/n}$$

since $n \geq 100$. Thus at least 2^s Boolean formulas have circuit size greater than s. □

3.3 Circuit Size Hierarchy

Theorem 16. *If $n \leq s(n) \leq \frac{2^n - 2}{n}$, then $\mathsf{SIZE}[s] \subsetneq \mathsf{SIZE}[4s]$.*

Proof. Use a combination of Shannon (Theorem 15) and Lupanov (Theorem 14). Pick[3] an $m < n$ such that

$$s(n) \leq \frac{2^m}{m} \leq 2s(n).$$

[3]Such an m must exists. When $m = 1$, $2^m/m \leq s(n)$ and when $m = n - 1$, $2^m/m \geq s(n)$ so there must be some m such that $2^m/m \leq s(n)$ and $2^{m+1}/(m+1) \geq s(n)$. If $2^{m+1}/(m+1) \geq 2 \cdot s(n)$ then

$$s(n) \leq \frac{2^m}{m+1} \leq \frac{2^m}{m}$$

which contradicts our original choice of m.

By Shannon, there exists a function $f : \{0,1\}^m \to \{0,1\}$ such that

$$\mathcal{C}(f) > \frac{2^m}{m} \geq s(n).$$

Thus $f \notin \mathsf{SIZE}[s]$. By the tight bound from Lupanov's theorem, $\mathcal{C}(f) \leq 2^m/m + o(2^m/m)$ so

$$\mathcal{C}(f) \leq \frac{2 \cdot 2^m}{m} \leq 4s(n)$$

and $f \in \mathsf{SIZE}[4s]$. □

4 Lower Bounds for Explicit Functions

4.1 (General) Gate Elimination: $\mathcal{C}(PARITY_n) \in \Omega(n)$

Definition 17. *For $i \in [n]$ and $b \in \{0,1\}$ the **1-bit restriction**, $x_i \leftarrow b$ is the n-ary function $f^{(x_i \leftarrow b)}$. The substitution can be done* syntactically *for circuits C, namely, $C^{(x_i \leftarrow b)}$. The technique is to substitute $x_i \leftarrow b$ and $\overline{x}_i \leftarrow 1 - b$ and performing the relevant simplifications.*

There are a couple of observations to note. (1) If C computes f, then $C^{(x_i \leftarrow b)}$ computes $f^{(x_i \leftarrow b)}$. (2) If x_i appears below a gate in C then for both settings of $b \in \{0,1\}$, size $\left(C^{(x_i \leftarrow b)}\right) \leq \text{size}(C) - 1$ i.e. any setting of b will knock out one gate in C. (2) There exists a setting of b for each gate, such that size $\left(C^{(x_i \leftarrow b)}\right) \leq \text{size}(C) - 2$ i.e. the setting of b knocks out two gates in C.

Theorem 18. ***(Schnorr 1979).*** *$\mathcal{C}(PARITY_n) \geq 3(n-1)$.*

Proof. By induction. The base case where $n = 1$ is trivial. The crucial observation is as follows. If a literal is below k $\wedge/\vee$ gates (of the same type), then there is a setting of the literal such that you can knock out at least k gates. Just think about the different settings of the literal.

Consider any circuit C which calculates the $PARITY_n$ function. Identify three gates in C:

1. A gate whose inputs are two literals. Let these be x_i and x_j.
2. Pick a literal of the previous gate, say x_i. Find another gate with x_i as an input. Suppose such a gate does not exist. Then, by setting x_j appropriately, we could knock out the gate in step 1 and the output would not depend on x_i. This would not calculate the $PARITY_n$ function.
3. The gate above the one in step 2. Such a gate exists if the gate from step 2 is not the output of the circuit. Suppose for a contradiction that it was. Then a setting of x_i would fix the output. This would also not calculate the $PARITY_n$ function.

By setting x_i appropriately, we can kills all three gates above. See Figure 5.

By the induction hypothesis, $C^{(x_i \leftarrow b)}$ has at least $3(n-2)$ gates. Since we were able to eliminate three gates by setting x_i, we know that C has to have $3(n-1)$ gates. □

More sophisticated versions of gate elimination allow for slightly better lower bounds. The current record is $5n - o(n)$ for DeMorgan circuits and $\left(3 + \frac{1}{86}\right) n$ for circuits in the full binary basis.

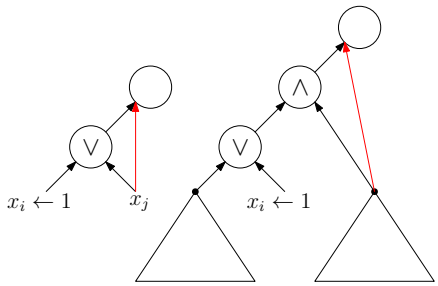

Figure 5: The three gates that get eliminated when we restrict x_i. The actual setting of x_i depends on the gate type.

4.2 (DeMorgan) Linear Algebra Method: $\mathcal{L}(PARITY_n) \in \Omega(n^2)$

Let us define $PARITY_n$ as $\bigoplus_n$ and $1 - PARITY_n$ as $\overline{\bigoplus}_n$. Recall[4] that $\mathcal{C}(\bigoplus_n) \leq 3(n-1)$ and $\mathcal{L}(\bigoplus_n) \leq 2^{2\lceil \log n \rceil}$. We will show that these bounds are tight.

Notation: $\lambda(\mathbf{P})$ is the largest eigenvalue of a symmetric matrix $\mathbf{P}$. Recall[5] that

$$\lambda(\mathbf{P} + \mathbf{Q}) \leq \lambda(\mathbf{P}) + \lambda(\mathbf{Q}).$$

For non-empty $A, B \subseteq \{0,1\}^n$, the matrix $\mathbf{M} \subseteq \{0,1\}^{A \times B}$ is the matrix

$$\mathbf{M}_{a,b} = \begin{cases} 1 & \text{if } a_i \neq b_i \text{ for exactly one } i \\ 0 & \text{otherwise} \end{cases}$$

you can read this as "the hamming distance of $\mathbf{a}$ and $\mathbf{b}$ differs by exactly one". Note that $\mathbf{M}^\intercal\mathbf{M} \in \mathbb{N}^{B \times B}$ with entry (i,j) interpreted as "the number of vectors $\mathbf{a} \in A$ such that both $\mathbf{b}_i$ and $\mathbf{b}_j$ are one away from $\mathbf{a}$". Similarly $\mathbf{M}\mathbf{M}^\intercal \in \mathbb{N}^{A \times A}$ with entry (i,j) interpreted as "the number of vectors $\mathbf{b} \in B$ such that both $\mathbf{a}_i$ and $\mathbf{a}_j$ are one away from $\mathbf{b}$". It is a fact from linear algebra that $\mathbf{M}^\top\mathbf{M}$ and $\mathbf{M}\mathbf{M}^\top$ have the same non-zero eigen-values. In particular, $\lambda(\mathbf{M}^\intercal\mathbf{M}) = \lambda(\mathbf{M}\mathbf{M}^\intercal)$.

Theorem 19. ***(Koutsoupias 1993).*** *For any $f : \{0,1\}^n \to \{0,1\}$, $A \subseteq f^{-1}(0)$, and $B \subseteq f^{-1}(1)$,*

$$\mathcal{L}(f) \geq \lambda(\mathbf{M}^\intercal\mathbf{M}).$$

Proof. By induction on $\mathcal{L}(f)$. The base case occurs when $\mathcal{L}(f) = 1$ and the circuit only reads in one out of the n variables of the input. W.l.o.g assume that the input to the leaf is x_1. Then $f(\mathbf{x}) = x_1$ or $f(\mathbf{x}) = 1 - x_1$; assume the former. Let $A = f^{-1}(0)$ and $B = f^{-1}(1)$. Then $A = \{0s : s \in \{0,1\}^{n-1}\}$ and $B = \{1s : s \in \{0,1\}^{n-1}\}$. Recall that entry (i,j) of $\mathbf{M}^\top\mathbf{M}$ is the number of elements $\mathbf{a} \in A$ such that both $\mathbf{b}_i$ and $\mathbf{b}_j$ differ from $\mathbf{a}$ by one. Notice that $\mathbf{a} = 0s$ and $\mathbf{b} = 1s'$ differ by exactly one if and only if $s = s'$. Thus $\mathbf{M}^\top\mathbf{M}$ is exactly the identity matrix with dimension $|B| \times |B|$ and $\lambda(\mathbf{M}^\top\mathbf{M}) = 1$ satisfying the theorem.

In the inductive step, let F be a formula which computes f of size $\mathcal{L}(f)$. Suppose that $F = F_1 \wedge F_2$ for some circuits F_1 and F_2. Let f_1 and f_2 be the functions computed by F_1 and F_2 respectively.

[4] Construct a circuit with $n-1$ $\oplus$-gates and substituting three DeMorgan gates $(x \wedge \neg y) \vee (\neg x \wedge y)$ for each $x \oplus y$.

[5] I think this can be shown as follows. Take the largest eigen-vector $\mathbf{x}$ of $\mathbf{P}$ and decompose it in the eigen-basis of $\mathbf{Q}$. Then right-multiplying $\mathbf{P} + \mathbf{Q}$ by $\mathbf{x}$.

Notices that $\mathcal{L}(f) = \mathcal{L}(f_1) + \mathcal{L}(f_2)$. Let $A_1 = f_1^{-1}(0)$ and $A_2 = A \backslash A_1$. Since $F = F_1 \wedge F_2$, $A_2 \subset f_2^{-1}(0)$ as at least one of F_1 or F_2 must evaluate to 0. Consider matrices $\mathbf{M}_1 \in \mathbb{N}^{A_1 \times B}$ and $\mathbf{M}_2 \in \mathbb{N}^{A_2 \times B}$. Notice that $\mathbf{M}^\top \mathbf{M} = \mathbf{M}_1^\top \mathbf{M}_1 + \mathbf{M}_2^\top \mathbf{M}_2$ since $A_1 \cup A_2 = A$ and each matrix product counts the number of off-by-one vectors $\mathbf{a}$ matched to by $\mathbf{b} \in B$. Then

$$\begin{aligned}
\lambda(\mathbf{M}^\top \mathbf{M}) &= \lambda(\mathbf{M}_1^\top \mathbf{M}_1 + \mathbf{M}_2^\top \mathbf{M}_2) && \text{(definition)} \\
&\leq \lambda(\mathbf{M}_1^\top \mathbf{M}_1) + \lambda(\mathbf{M}_2^\top \mathbf{M}_2) && \text{(symmtric matrix prop.)} \\
&\leq \mathcal{L}(f_1) + \mathcal{L}(f_2) && \text{(induction hyp.)} \\
&= \mathcal{L}(f)
\end{aligned}$$

The same is true if $F = F_1 \vee F_2$, but this requires decomposing B. Remember however that $\lambda(\mathbf{M}^\top \mathbf{M}) = \lambda(\mathbf{M}\mathbf{M}^\top)$ so it does not make much of a difference. □

Corollary 20. ***(Khrapchenko 1971).***

$$\mathcal{L}(f) \geq \frac{\left(\sum_{\mathbf{a} \in A} \sum_{\mathbf{b} \in B} \mathbf{M}_{a,b}\right)^2}{|A| \cdot |B|}$$

Proof. [6] The idea is to write $\lambda(\mathbf{M}^\top \mathbf{M})$ as a Rayleigh quotient and then substitute in $\mathbb{1}$ to get that lower bound.

$$\begin{aligned}
\lambda\left(\mathbf{M}^\top \mathbf{M}\right) &= \max_{\mathbf{z} \in \mathbb{R}^B \backslash \emptyset} \frac{\mathbf{z}^\top \mathbf{M}^\top \mathbf{M} \mathbf{z}}{\mathbf{z}^\top \mathbf{z}} \\
&\geq \frac{\mathbb{1}^\top \mathbf{M}^\top \mathbf{M} \mathbb{1}}{|B|} \\
&= \frac{\sum_{a \in A} \left(\sum_{b \in B} \mathbf{M}_{a,b}\right)^2}{|B|} \\
&\geq \frac{\left(\sum_{a \in A} \sum_{b \in B} \mathbf{M}_{a,b}\right)^2}{|A| \cdot |B|}
\end{aligned}$$

where the last inequality follows by Cauchy-Schwartz[7]. □

We use the above Corollary 20 to show that $\mathcal{L}(\bigoplus_n) \geq n^2$. Take A and B to be the set of even and odd strings[8] in $\{0,1\}^n$ respectively. Then, by the above,

$$\mathcal{L}(f) \geq \frac{\left(\sum_{a \in A} \sum_{b \in B} \mathbf{M}_{a,b}\right)^2}{|A| \cdot |B|} = \frac{\left(n 2^{n-1}\right)^2}{2^{n-1} \cdot 2^{n-1}} = n^2.$$

This technique can achieve gaps of at most n^2. Exercise: (1)[9] prove lower-bound $\mathcal{L}(MAJ_n) \geq \Omega(n^2)$ and (2) can you devise an upper bound of $\mathcal{L}(MAJ_n) \leq n^{O(1)}$.

[6] :) I like this

[7] The application of Cauchy-Schwartz here is subtle. The key is to multiply top and bottom by $\left(\sum_{a \in A} 1^2\right)$ and combine the two sum of squares.

[8] Here the parity of the string s corresponds to the parity of the sum of ones in s.

[9] Hint: Take $A = \{s \in \{0,1\}^n : s \text{ has exactly } \lceil n/2 \rceil - 1 \text{ ones}\}$ and $B = \{t \in \{0,1\}^n : t \text{ has exactly } \lceil n/2 \rceil \text{ ones}\}$.

4.3 (DeMorgan) Random Restriction: $\mathcal{L}(ANDREEV_{k,m}) \in \Omega(n^3)$

4.3.1 Subbotovskaya's Method

Definition 21. *A formula F is **nice** if for every sub-formula of the form $x_i \wedge F'$, $\overline{x}_i \wedge F'$, $x_i \vee F'$, $\overline{x}_i \vee F'$, the variable x_i does* not *occur in F'.*

Lemma 22. *Every formula is equivalent to a* nice *formula of the same (or less) leaf size.*

Proof. Given sub-formulas of the form $x_i \wedge F$, $\overline{x}_i \wedge F$, $x_i \vee F$, and $\overline{x}_i \vee F$ where F contains literals x_i or $\overline{x}_i$, repeatedly apply

$$x_i \wedge F \to x_i \wedge F^{(x_i \leftarrow 1)}$$
$$\overline{x}_i \wedge F \to \overline{x}_i \wedge F^{(x_i \leftarrow 0)}$$
$$x_i \vee F \to x_i \vee F^{(x_i \leftarrow 0)}$$
$$\overline{x}_i \vee F \to \overline{x}_i \vee F^{(x_i \leftarrow 1)}$$

This shows that every minimal formula for a function f is *nice*. □

Lemma 23. *For every $f : \{0,1\}^n \to \{0,1\}$,*

$$\mathbb{E}_{i \in [n], b \in \{0,1\}} \left[\mathcal{L}\left(f^{(x_i \leftarrow b)} \right) \right] \leq \left(1 - \frac{1}{n} \right)^{1.5} \mathcal{L}(f).$$

Proof. Let F be a minimal nice formula for f. Let ℓ_i be the all leaves of F labelled with x_i or $\bar{x}_i$. Then $\mathcal{L}(f) = \sum_{i=1}^{n} \ell_i$. Notice that every gate g with a leaf λ has an associated sub-formula F' such that λ does not occur in F'.

For a bit $b \in \{0,1\}$, the random restriction $F^{(x_i \leftarrow b)}$ will kill leaf x_i with probability 1 and kill all leaves in F' with probability $\frac{1}{2}$. Thus in expectation, 1.5 leaves are killed under the 1-bit restriction $F^{(x_i \leftarrow b)}$. For each $i \in [n]$ we have

$$\mathbb{E}_{b \in \{0,1\}} \left[\mathcal{L}(F) - \mathcal{L}\left(F^{(x_i \leftarrow b)} \right) \right] \geq 1.5 \ell_i.$$

Averaging over all choices of i, we have that

$$\mathbb{E}_{i \in [n], b \in \{0,1\}} \left[\mathcal{L}(F) - \mathcal{L}\left(F^{(x_i \leftarrow b)} \right) \right] \geq \frac{1.5}{n} \sum_{i=1}^{n} \ell_i = \frac{1.5 \mathcal{L}(F)}{n}.$$

Rearranging the above, we have

$$\mathbb{E}_{i \in [n], b \in \{0,1\}} \left[\mathcal{L}\left(F^{(x_i \leftarrow b)} \right) \right] \leq \left(1 - \frac{1.5}{n} \right) \mathcal{L}(F) \leq \left(1 - \frac{1}{n} \right)^{1.5} \mathcal{L}(F)$$

where the last inequality follows as $1 - ax \leq (1-x)^a$. □

Apparently, this lemma implies that $\mathcal{L}(\oplus_n) \geq n^{1.5}$.

Definition 24. *A **restriction** ρ is a function $\rho : [n] \to \{0, 1, *\}$ which can be thought of as a partial assignment of an n-ary Boolean function f. Denote the restriction of f under ρ as $f \restriction \rho : \{0,1\}^{\rho^{-1}(*)} \to \{0,1\}$. Further ρ is a k**-star restriction** if $|\rho^{-1}(*)| = k$.*

*Let $p \in [0, 1]$. In a p**-random restriction** where you set*

$$R_p(i) = \begin{cases} * & \text{with probability } p \\ 0 & \text{with probability } \frac{1-p}{2} \\ 1 & \text{with probability } \frac{1-p}{2} \end{cases}$$

Theorem 25. *Let $f : \{0,1\}^n \to \{0,1\}$ and let ρ be a uniform random k-start restriction. Then*

$$\mathbb{E}\left[\mathcal{L}(f \restriction \rho)\right] \leq \left(\frac{k}{n}\right)^{1.5} \mathcal{L}(f).$$

Proof. Repeatedly apply Lemma 23 to restrict k bits to get

$$\mathbb{E}\left[f \restriction \rho\right] \leq \left(1 - \frac{1}{n}\right)^{1.5} \cdot \left(1 - \frac{1}{n-1}\right)^{1.5} \cdots \left(1 - \frac{1}{k+1}\right)^{1.5} \mathcal{L}(f) = \left(\frac{k}{n}\right)^{1.5} \mathcal{L}(f).$$

□

Corollary 26. ***(Subbotovskaya 1961).***

$$\mathbb{E}\left[\mathcal{L}\left(f \restriction R_p\right)\right] \leq O\left(p^{1.5}\mathcal{L}(f) + 1\right).$$

According to Håstad (19 something or other) and Tal (2014), this can be improved to $O(p^2\mathcal{L}(f)+1)$.

Open problem: what is the shrinkage exponent of monotone formulas? (this is known to be between 2 and $\left(\log(\sqrt{5}) - 1\right)^{-1} = 3.27$).

4.3.2 DEF: Composition of Boolean Functions

Definition 27. *Let $f : \{0,1\}^k \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$. Let $f \otimes g : (\{0,1\}^m)^k \to \{0,1\}$ is defined as*

$$(f \otimes g)(\mathbf{x}_1, ..., \mathbf{x}_k) = f\left(g(\mathbf{x}_1), ..., g(\mathbf{x})\right).$$

In essence the composition is of the form $f \otimes g = f \circ g^k$.

Think of the input of the composition as a matrix $\mathbf{X} \in \{0,1\}^{k \times m}$ with rows $\mathbf{x}_1, ..., \mathbf{x}_k$. Apply g to each row, then apply f to the resulting column vector. Observe that $\mathcal{L}(f \otimes g) \leq \mathcal{L}(f) \cdot \mathcal{L}(g)$.

Conjecture 28. ***(KRW).*** *For all functions f and g,*

$$\mathcal{L}\left(f \otimes g\right) = \tilde{\Omega}\left(\mathcal{L}(f) \cdot \mathcal{L}(g)\right)$$

where $\tilde{\Omega}(t(n)) = \Omega(t(n))/\left(\log t(n)\right)^{O(1)}$ for any function $t(n)$.

The following is an explicit n-ary Boolean function for which the lower bound is true.

Lemma 29. *For all $k, m \geq 1$ and $f : \{0,1\}^k \to \{0,1\}$,*

$$\mathcal{L}\left(f \otimes XOR_m\right) \geq \mathcal{L}(f) \cdot \Omega\left(\left(\frac{m}{\log k}\right)^2\right).$$

Proof. Let $p = \frac{2 \ln k}{m}$. Apply R_p on $k \times m$ variables of $f \otimes XOR_m$. If R_p has a $*$ in every row then

$$\mathcal{L}\left(\left(f \otimes XOR_m\right) \restriction R_p\right) \geq \mathcal{L}(f)$$

since a formula which calculates the LHS can be used to calculate the RHS. In particular, if there is a $*$ in some row i, then from the perspective of XOR_m, the value of row i is undetermined. If every single row is undetermined, then the input to f is undetermined. Thus $(f \otimes XOR_m) \restriction R_p$ would be able to compute $f(s)$ for any $s \in \{0,1\}^k$.

Let E be the event that there exists a $*$ in every row of the input matrix after applying R_p. We bound $\Pr[E]$ below by bounding $\Pr\left[\overline{E}\right]$ above. Let B_i be the event that some row i is *bad* i.e. does not have a $*$ after applying R_p. Since every element is fixed with probability $1-p$, $\Pr[B_i] = (1-p)^m$ for all $i \in [k]$. Then

$$\Pr\left[\overline{E}\right] \leq \sum_{i=1}^{k} \Pr[B_i] = k(1-p)^m \approx k \exp(-pm) \leq \frac{1}{k}$$

where the first inequality follows by union-bound and the last by the definition of p above. Thus we have the following lower bound

$$1 - \frac{1}{k} \leq \Pr\left[E\right] \tag{1}$$

To get an upper bound for $\Pr[E]$, observe that $\Pr[E] \leq \Pr\left[\mathcal{L}\left(\left(f \otimes XOR_m\right) \restriction R_p\right) \geq \mathcal{L}(f)\right]$. Apply Markov's inequality to get

$$\Pr\left[\mathcal{L}\left(\left(f \otimes XOR_m\right) \restriction R_p\right) \geq \mathcal{L}(f)\right] \leq \frac{\mathbb{E}\left[\mathcal{L}\left(\left(f \otimes XOR_m\right) \restriction R_p\right)\right]}{\mathcal{L}(f)}$$

By the improvement noted after Corollary 26, $\mathbb{E}\left[f \restriction R_p\right] \leq O\left(p^2 \mathcal{L}(f) + 1\right)$, we have

$$\Pr[E] \leq \frac{\mathbb{E}\left[\mathcal{L}\left(\left(f \otimes XOR_m\right) \restriction R_p\right)\right]}{\mathcal{L}(f)} = O\left(\frac{p^2 \mathcal{L}(f \otimes XOR_m) + 1}{\mathcal{L}(f)}\right). \tag{2}$$

Combining the lower bound from Equation (1) and the upper bound from Equation (2), we have

$$1 - \frac{1}{k} \leq \Pr[E] \leq \frac{p^2 \mathcal{L}(f \otimes XOR_m) + 1}{\mathcal{L}(f)}$$

Rearrange with respect to $\mathcal{L}(f \otimes XOR_m)$, taking care to observe that $1 - \frac{1}{k} - \frac{1}{\mathcal{L}(f)} \in O(1)$, to obtain

$$\mathcal{L}\left(f \otimes XOR_m\right) \geq \frac{\mathcal{L}(f)}{p^2} = \mathcal{L}(f) \cdot \Omega\left(\left(\frac{m}{\log k}\right)^2\right)$$

as required. □

4.3.3 FUN: $ANDREEV_{k,m}$

Let us construct an explicit function with cubic lower bound on the leaf size using Lemma 29.

Definition 30. *For parameters $k, m \in \mathbb{N}$,*

$$ANDREEV_{k,m} : \{k\text{-ary Boolean function}\} \times \{0,1\}^{k \times m} \to \{0,1\}$$

such that $ANDREEV(f, \mathbf{X}) = (f \otimes XOR_m)(\mathbf{X})$.

Think of this as follows: consider the $(k+1) \times 2^k$ table T of k-ary Boolean strings and the evaluation of f on these strings. See Table 1. The input matrix $\mathbf{X} \in \{0,1\}^{k \times m}$. Apply the XOR_m function to each row of $\mathbf{X}$ to obtain a k-bit string s. Find the column of T corresponding to s and return $f(s)$.

$f(0^k)$	$\cdots$	$f(1^k)$
0	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$
0	$\cdots$	1

Table 1: Table T of function f.

When $m = 1$, $ANDREEV_{k,1}$ is just the multiplexor function. Let $n = 2^k + mk$. Then $ANDREEV_{k,m}$ can be thought of as an n-ary Boolean function with $\mathcal{C}(ANDREEV_{k,m}) = O(n)$.

Theorem 31. *For every $f : \{0,1\}^k \to \{0,1\}$ we have*

$$\mathcal{L}(ANDREEV_{k,m}) \geq \mathcal{L}(f \otimes XOR_m) \geq \mathcal{L}(f) \cdot \Omega\left(\left(\frac{m}{\log k}\right)^2\right).$$

Proof. By fixing 2^k values $f(s)$ for $s \in \{0,1\}^k$, in the formula for $ANDREEV_{k,m}$, we can calculate $f \otimes XOR_m$. Thus $\mathcal{L}(ANDREEV_{k,m}) \geq \mathcal{L}(f \otimes XOR_m)$. By Shannon's Theorem 15, there exists a k-ary Boolean function f with circuit size, and thus leaf size, $\Omega(2^k/k)$. Let $m = 2^k/k$ and note that $n = 2^k + mk \in \Theta(2^k)$. Then, by Lemma 29,

$$\mathcal{L}(ANDREEV_{k,m}) \geq \Omega\left(\frac{2^k}{k}\right) \cdot \Omega\left(\left(\frac{m}{\log k}\right)^2\right) = \Omega\left(\frac{n^3}{(\log n)^3 (\log\log n)^2}\right).$$

Thus $\mathcal{L}(ANDREEV_{k,m}) \in \tilde{\Omega}(n^3)$. □

This lower bound for the $ANDREEV_{m,k}$ is nearly tight since $\mathcal{L}(ANDREEV_{k,m}) \in \tilde{O}(n^3)$.

4.4 (Full Binary) Subset Subfunction: $\mathcal{L}(ED_n) \in \Omega(n^2)$

4.4.1 DEF: V-Subfunctions

Let B_2 be the full binary basis (all 2-ary gate types). Unfortunately, the random restriction idea does not work in this setting since it is *not true* that

$$\mathbb{E}[\mathcal{L}_{B_2}(f \restriction R_p)] \leq O\left(p^{1+\epsilon} \mathcal{L}_{B_2}(f) + 1\right)$$

for any $\epsilon > 0$. Do you see why?[10]

Definition 32. *For $f : \{0,1\}^n \to \{0,1\}$ and $V \subset [n]$,*

$$\mathrm{sub}_V(f) = \{f \restriction \rho : \rho : [n] \to \{0,1,*\} \text{ such that } \rho^{-1}(*) = V\}$$

*be the set of V-**subfunctions** of f.*

Further, define

$$\mathrm{sub}^*_V(f) = \{f', 1 - f', \underline{0}, \underline{1} : f' \in \mathrm{sub}_V(f)\}$$

where $\underline{b}$ is the constant b function for $b \in \{0,1\}$. Note that $|\mathrm{sub}^*_V(f)| \leq 4 \cdot |\mathrm{sub}_V(f)|$ (actually $|\mathrm{sub}^*_V(f)| \leq 2 \cdot |\mathrm{sub}_V(f)| + 2$).

*Let F be an n-ary formula and $V \subset [n]$ as before. Then **the number of leaves of F labelled by variables in V** be denoted $\ell_V(F)$. Note that $\mathcal{L}(F) = \ell_V(F) + \ell_{[n]\setminus V}(F)$.*

Example 33. *Consider $MAJ_3(x_1, x_2, x_3)$ and $V = \{1, 2\}$. Then*

$$\mathrm{sub}_V(MAJ_3) = \{x_1 \wedge x_2, x_1 \vee x_2\}$$

when x_3 is restricted to 0 and 1 respectively.

4.4.2 Nechiporuk's Bound

Here are two important properties to note:

1. Suppose $F = \mathsf{gate}(G, H)$ for some $\mathsf{gate} : \{0,1\}^2 \to \{0,1\}$. Then

   $$\mathrm{sub}_V(F) \subseteq \{\mathsf{gate}(g, h) : g \in \mathrm{sub}_V(G) \text{ and } h \in \mathrm{sub}_V(H)\}.$$

   and $|\mathrm{sub}^*_V(F)| \leq |\mathrm{sub}^*_V(G)| \cdot |\mathrm{sub}^*_V(H)|$. This should be pretty obvious; let f_V be any function in $\mathrm{sub}_V(f)$. Then this function must be equivalent to the composition of the function computed by gate and some two functions $g \in \mathrm{sub}_V(G)$ and $h \in \mathrm{sub}_V(H)$.

2. Suppose that $F = \mathsf{gate}(G, H)$ and $\ell_V(H) = 0$. Then $\mathrm{sub}^*_V(F) \subseteq \mathrm{sub}^*_V(G)$. Note that $\ell_V(H)$ means that none of the leaves in H are labelled with any indices from V. When considering the V-functions of H, these can only be the constant functions $\underline{0}$ and $\underline{1}$. When composing the function calculated by gate with some $g \in \mathrm{sub}_V(G)$ and a function in $\{\underline{0}, \underline{1}\}$ we can only get $\{g, 1 - g, \underline{0}, \underline{1}\}$. Thus every function in $\mathrm{sub}^*_V(F)$ is also in $\mathrm{sub}^*_V(G)$.

Lemma 34. *If F is an n-ary formula, $V \subseteq [n]$, and $\ell_V(F) \geq 1$, then*

$$|\mathrm{sub}^*_V(F)| \leq 4 \cdot 16^{\ell_V(F)-1}.$$

Proof. By induction on $\mathcal{L}(F)$. The base case where $\mathcal{L}(F) = 1$ is trivial. The inductive case is also not that bad considering the two observations above. Suppose $F = \mathsf{gate}(G, H)$. If one of

[10] Let f be the parity function.

$\ell_V(G) = 0$ or $\ell_V(H) = 0$, then we can use the second observation. W.l.o.g assume $\ell_V(H) = 0$. By the induction hypothesis we have that

$$|\mathrm{sub}_V^*(F)| \leq |\mathrm{sub}_V^*(G)| \leq 4 \cdot 16^{\ell_V(G)-1} \leq 4 \cdot 16^{\ell_V(F)-1}.$$

Next suppose that $\ell_V(G) \geq 1$ and $\ell_V(H) \geq 1$. Then by the induction hypothesis, we have that

$$|\mathrm{sub}_V^*(F)| \leq |\mathrm{sub}_V^*(G)| \cdot |\mathrm{sub}_V^*(G)| = 16^{\ell_V(G)+\ell_V(H)-1} = 16^{\ell_V(F)-1} \leq 4 \cdot 16^{\ell_V(F)-1}$$

as required. □

Corollary 35. *Let F and V be as above, then $|\mathrm{sub}_V(F)| \leq 16^{\ell_V(F)}$.*

This is immediate when $\mathcal{L}(F) \geq 1$. Only $\underline{b}$, $b \in \{0,1\}$, have leaf-size 0, but $|\mathrm{sub}_V(\underline{b})| = 1 \leq 16^0$.

Theorem 36. ***(Nechiporuk's Bound).*** *For any $f : \{0,1\}^n \to \{0,1\}$ and any partition of $[n]$ into t disjoint components $V_1 \uplus \cdots \uplus V_t$*

$$\mathcal{L}_{B_2}(f) \geq \frac{1}{4} \sum_{i=1}^{t} \log |\mathrm{sub}_{V_i}(f)|.$$

Proof. Direct application of Corollary 35. Let F be the minimal formula computing f. Then

$$\mathcal{L}_{B_2}(f) = \sum_{i=1}^{t} \ell_{V_i}(F) \geq \sum_{i=1}^{t} \log_{16} |\mathrm{sub}_{V_i}(F)| = \frac{1}{4} \sum_{i=1}^{t} \log |\mathrm{sub}_{V_i}(F)|.$$

□

4.4.3 FUN: Element Distinctness ED_n

Let us apply Theorem 36 to an explicit function in the full binary basis to get a lower bound.

Definition 37. *For $k \in \mathbb{N}$, let $n = 2^k \cdot 2k$. The **element distinctness** function ED_n is*

$$\mathrm{ED}_n : \{0,1\}^{2^k \times 2k} \to \{0,1\}$$

where

$$\mathrm{ED}_n(X_1, ..., X_{2^k}) = \begin{cases} 1 & \text{if } X_1, ..., X_{2k} \text{ are distinct elements of } \{0,1\}^{2k} \\ 0 & \text{otherwise} \end{cases}$$

Think of this as being given 2^k binary strings of length $2k$ and asked if they are all distinct.

Theorem 38.

$$\mathcal{L}_{B_2}(\mathrm{ED}_n) = \Omega\left(\frac{n^2}{\log n}\right).$$

Proof. Apply Nechiporuk's bound with 2^k disjoints sets V_i were each is a block of length $2k$ corresponding to the coordinates of X_i. Since these blocks are symmetric, we have

$$\mathcal{L}_{B_2}(\mathrm{ED}_n) \geq \frac{1}{4}\sum_{i=1}^{2^k} \log|\mathrm{sub}_{V_i}(\mathrm{ED}_n)| = 2^{k-2}\log|\mathrm{sub}_{V_1}(\mathrm{ED}_n)|.$$

Thus it suffices to bound $|\mathrm{sub}_{V_1}(\mathrm{ED}_n)|$. To ensure that the restriction does not result in a constant function, the $2^k - 1$ fixed $2k$-bit strings in the input to ED_n must all be distinct. There are $\binom{2^{2k}}{2^k-1}$ subsets of $2^k - 1$ distinct length $2k$ Boolean strings. Thus $|\mathrm{sub}_{V_1}(\mathrm{ED}_n)| \geq \binom{4^k}{2^k-1}$. Using the bound $\binom{2^{2k}}{2^k-1} \geq 2^{k(2^k-1)}$ and recalling that $n = 2^k(2k)$,

$$\mathcal{L}_{B_2}(\mathrm{ED}_n) \geq 2^{k-2}\log|\mathrm{sub}_{V_1}(\mathrm{ED}_n)| \geq 2^{k-2}k2^k - O(1) \in \Omega(n^2/\log n).$$

□

Exercise: show that $\Omega(n^2/\log n)$ is the limit on the lower bound achievable by Nechiporuk's method.

5 Non-uniformity is More Powerful than Randomness

Definition 39. *A **randomized circuit** for a function $f : \{0,1\}^n \to \{0,1\}$ is a circuit C with $n+m$ variables $x_1, ..., x_n$ and $y_1, ..., y_m$ (think of $\mathbf{x}$ as the input and $\mathbf{y}$ as a random seed) such that for every $\mathbf{x} \in \{0,1\}^n$*

$$\Pr_{\mathbf{y}\in\{0,1\}^m}[C(\mathbf{x},\mathbf{y}) = 1] \begin{cases} \geq \frac{2}{3} & \text{if } f(\mathbf{x}) = 1 \\ \leq \frac{1}{3} & \text{if } f(\mathbf{x}) = 0 \end{cases}$$

Let $\mathsf{BPP/poly}$ be the class of Boolean functions computable by poly-sized randomized circuits — think of $\mathsf{BPP/poly}$ as the non-uniform version of BPP. Generally $\frac{1}{3}$ and $\frac{2}{3}$ can be replaced with any a, b satisfying $0 < a < b < 1$.

Theorem 40. ***(Adelman 1978).*** *If f is computable by poly-size randomized circuit, then it is computable by poly-sized (deterministic) circuits i.e. $\mathsf{BPP/poly} \subseteq \mathsf{P/poly}$.*

Proof. The intuition is to improve the probability of success by doing repeated trials, taking the majority, then use the probabilistic method to show that there was a good choice of $\mathbf{y}$ which we can hard-wired into the circuit.

Let f be the given n-ary function with $\mathsf{BPP/poly}$ circuit $C(\mathbf{x},\mathbf{y})$. Recall that MAJ_k has $O(k)$ sized circuits. Construct the composite function $g_k : \{0,1\}^n \times \{0,1\}^{k\times m} \to \{0,1\}$ such that

$$g_k(\mathbf{x},\mathbf{Y}) = MAJ_k\left(C(\mathbf{x},\mathbf{y}_1), ..., C(\mathbf{x},\mathbf{y}_k)\right)$$

where each $\mathbf{y}_i$ is the i^{th} row of $\mathbf{Y}$. Observe that $\mathcal{C}(g_k) \leq k \cdot \mathcal{C}(f) + O(k)$ since we can replace each of the k inputs of MAJ_k by a circuit of size $\mathcal{C}(f)$. If k is poly(n) then $g_k \in \mathsf{P/poly}$.

On a randomly sampled seed $\mathbf{y}_i$, let X_i be the indicator r.v. for $C(\mathbf{x}, \mathbf{y}_i) \neq f(\mathbf{x})$. Let $X = X_1 + \cdots + X_k$. Observe that

$$\begin{aligned}\Pr_{\mathbf{Y}\in\{0,1\}^{k\times m}}[g_k(\mathbf{x}, \mathbf{Y}) \neq f(\mathbf{x})] &= \Pr_{\mathbf{y}_1,\ldots,\mathbf{y}_k\in\{0,1\}^m}[MAJ_k(C(\mathbf{x}, \mathbf{y}_1), ..., C(\mathbf{x}, \mathbf{y}_k)) \neq f(\mathbf{x})] \\ &= \Pr\left[X \geq \frac{k}{2}\right] \\ &= \Pr[X \geq (1+\epsilon)pk]\end{aligned}$$

where $p = \Pr[C(\mathbf{x}, \mathbf{y}) \neq f(\mathbf{x})]$ and $\epsilon = \frac{1-2p}{2p}$. From Definition 39, we have $p = \frac{1}{3}$ and $\epsilon = \frac{1}{2}$. Thus by Chernoff bound we have

$$\Pr_{\mathbf{Y}\in\{0,1\}^{k\times m}}[g_k(\mathbf{x}, \mathbf{Y}) \neq f(\mathbf{x})] = \Pr[X \geq (1+\epsilon)pk] \leq \exp\left(\frac{-\epsilon^2 pk}{2+\epsilon}\right) = \exp\left(\frac{-k}{30}\right).$$

When $k > 30$, $\Pr[g_k(\mathbf{x}, \mathbf{Y}) \neq f(\mathbf{x})] \leq 2^{-n}$ and there exists a $\mathbf{Y}$ for which $g_k(\mathbf{x}, \mathbf{Y}) = f(\mathbf{x})$ with probability greater than $1 - 2^{-n}$. Since there are only 2^n inputs $\mathbf{x}$, $g_k(\mathbf{x}, \mathbf{Y})$ matches $f(\mathbf{x})$ on every input. Hard-wiring $\mathbf{Y}$ into the circuit for g_k produces a deterministic poly(n) circuit for f. □

6 Restricted Setting

6.1 Monotone Circuits

Definition 41. *A **monotone function** is a n-ary Boolean function f where $S' \subseteq S \subseteq [n]$ implies $f(\mathbb{1}_{S'}) = 1$ if $f(\mathbb{1}_S) = 1$. A **monotone circuit** is a circuit corresponding to a monotone function.*

6.1.1 Upper Bound: Majority

Recall that $\text{MAJ}_n(\mathbf{x}) = 1$ if and only if $|\mathbf{x}| > \frac{n}{2}$. If $MAJ_n(\mathbf{x}) = 0$, then a random bit x_i chosen uniformly among $\{x_1, ..., x_n\}$ satisfies $\Pr[x_i = 1] \leq \frac{1}{2} - \frac{1}{2n}$.

Definition 42. *Let $m > n$. Then π is a **random projection** from $(y_1, ..., y_m) \to \{x_1, ..., x_n\}^m$ if $\pi(y_i)$ is mapped independently and uniformly among all $\{x_1, ..., x_n\}$.*

For monotone formula $F(\mathbf{y})$, let $F_\pi(\mathbf{x})$ be the monotone formula obtained from F by replacing each variable y_i with $\pi(y_i)$.

We will use random projections to construct a poly-sized monotone formula for $\text{MAJ}_n(\mathbf{x})$. The formula consists of a perfect tree of MAJ_3 gate. At the bottom of this tree we sample i.i.d from the elements of $\mathbf{x}$. Intuitively this works by amplifying the difference between the number of zeros and ones in $\mathbf{x}$. If there are more ones in $\mathbf{x}$, then you would expect $\Pr[\text{MAJ}_3(\pi(y_1), \pi(y_2), \pi(y_3)) = 1] > \frac{1}{2}$ and similarly when there are more zeros. This increased likelihood is then propagated up the tree so that the output at the root matches the output of MAJ_n in expectation. Using standard probabilistic methods, there exists a projection which exactly matches the output of MAJ_n. To formalize this intuition, we need a couple of definitions.

Definition 43. *For $f : \{0,1\}^m \to \{0,1\}$, the **output probability** $\mu_f : [0,1] \to [0,1]$ is the probability that $f(\mathbf{y}) = 1$ given that each $y_i \sim \text{Bern}(p)$ i.e.*

$$\mu_f(p) = \Pr_{\mathbf{y} \sim \text{Bern}(p)^m}[f(\mathbf{y}) = 1].$$

For example, $\mu_{MAJ_3}(p) = p^3 + 3p(1-p)$.

When f is a non-constant, monotone function, then μ_f is increasing with $\mu_f(0) = 0$ and $\mu_f(1) = 1$.

Lemma 44. $\mu_{f \otimes g}(p) = \mu_f(\mu_g(p))$.

Proof. Let f and g be a k and m-ary Boolean functions respectively. The inputs of $f \otimes g$ is a matrix $\mathbf{X} \in \{0,1\}^{k \times m}$ where you apply g to the rows $\mathbf{x}_i$ of $\mathbf{X}$ and then apply f to the resulting column vector. All km bits of $\mathbf{X}$ are distributed like Bern(p). From the perspective of g, each bit is independently set to one with probability p so $\mu_g(p) = \Pr[g(\mathbf{x}_i) = 1]$ for $i \in [k]$. From the perspective of f, each bit is independently set to one with probability $\mu_g(p)$ so $\mu_f(\mu_g(p)) = \Pr[f(g(\mathbf{x}_1), ..., g(\mathbf{x}_k)) = 1] = \mu_{f \otimes g}(p)$. □

Corollary 45. $\mu_{f^{\otimes k}}(p) = \mu_f^{(k)}(p)$.

Lemma 46. *There is a constant $c < 3$ such that $\mu_{\text{MAJ}_3}^{(c \log n)}(p) < \frac{1}{2^n}$ for $0 \le p < \left(\frac{1}{2} - \frac{1}{2n}\right)$ and $\mu_{\text{MAJ}_3}^{(c \log n)}(p) > 1 - \frac{1}{2^n}$ for $\frac{1}{2} + \frac{1}{2n} \le p \le 1$.*

Proof. Keep in mind is the graph of μ_{MAJ_3} shown in Figure 6. The second derivative is positive when $p < \frac{1}{2}$ and negative when $p > \frac{1}{2}$. For positive integer k, $\lim_{k \to \infty} \mu_{\text{MAJ}_3}^{(k)}(p) = 0$ in the former and $\lim_{k \to \infty} \mu_{\text{MAJ}_3}^{(k)}(p) = 1$ in the latter case.

We evaluate the rate at which the composed function approaches its limit in two phases: (1) $\frac{1}{2} > p \ge \frac{1}{4}$ and (2) $\frac{1}{4} > p > \frac{1}{2^n}$.

1. Write $p = \frac{1}{2} - \delta$ where $\delta < \frac{1}{4}$. We want to find a $c_1 > 1$ such that $\mu_{\text{MAJ}_3}(p) = \frac{1}{2} - c_1\delta$. Then $\mu_{\text{MAJ}_3}^{(k_1)}(p) \le \frac{1}{2} - c_1^{k_1}\delta$. Take $k_1 \in O(\log n)$ in order for the RHS to be less than $\frac{1}{4}$. To find c_1

$$\begin{aligned}\left(\frac{1}{2} - \delta\right)^3 + 3\left(\frac{1}{2} - \delta\right)^2\left(\frac{1}{2} + \delta\right) &= \frac{1}{2} - \frac{3\delta}{2} + 2\delta^3 \\ &\le \frac{1}{2} - \left(\frac{3}{2} - \frac{2}{16}\right)\delta \\ &\le \frac{1}{2} - \frac{5}{4}\delta.\end{aligned}$$

   Thus we can choose $c_1 = 5/4$.

2. Here $p < \frac{1}{4}$ so again we want to find a constant $c_2 < 1$ such that $\mu_{\text{MAJ}_3}(p) \le c_2 p$. Then $\mu_{\text{MAJ}_3}^{(k_2)}(p) \le c_2^{k_2} p$. Take $k_2 \in O(\log n)$ in order for the RHS to be less than $\frac{1}{2^n}$. To find c_2

$$\begin{aligned}p^3 + 3p^2(1-p) &= 3p^2 - 2p^3 \\ &\le 3p^2 \le \frac{3}{4}p.\end{aligned}$$

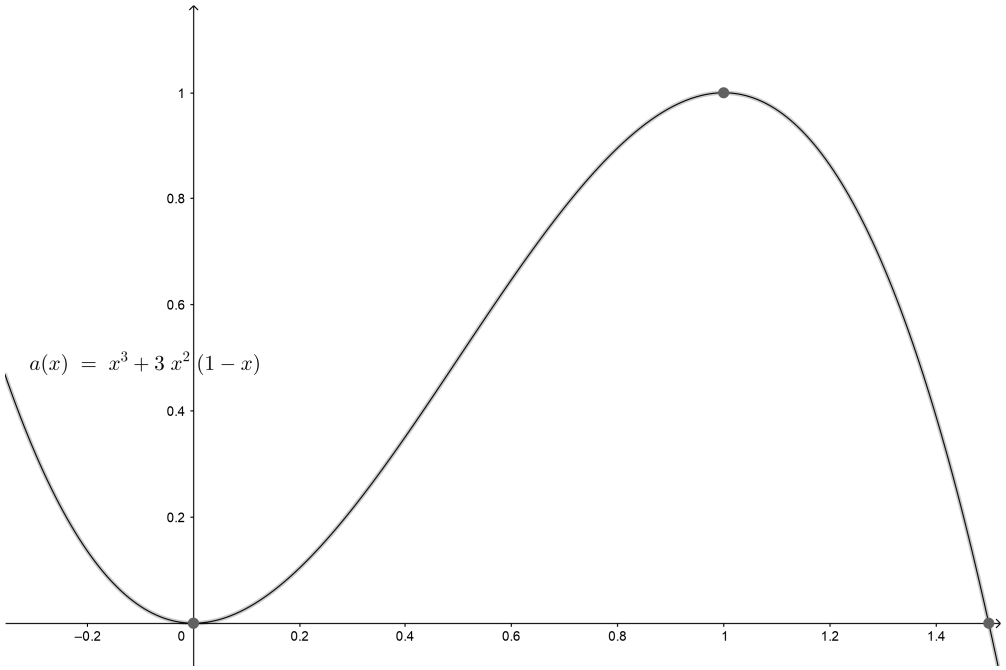

Figure 6: A graph of μ_{MAJ_3}. Pay attention to the interval between $[0, 1]$.

Thus we can choose $c_2 = 3/4$.

Since $\mu_{\mathrm{MAJ}_3}(p)$ is rotationally symmetric about $p = \frac{1}{2}$, the claim also holds for $p \geq \frac{1}{2} + \frac{1}{2n}$. □

Theorem 47. ***(Valiant 1984).*** MAJ_n *has poly-sized monotone circuits*[11]*.*

Proof. We are going to combine the above ideas of amplification and projection. Let $k = c \log n$, where c is the constant from Lemma 46. Build a depth k tree T out of MAJ_3 formulas of minimum size. Since there exists a formula of size five for MAJ_3, see Figure 7, T will have at most 5^k leaves.

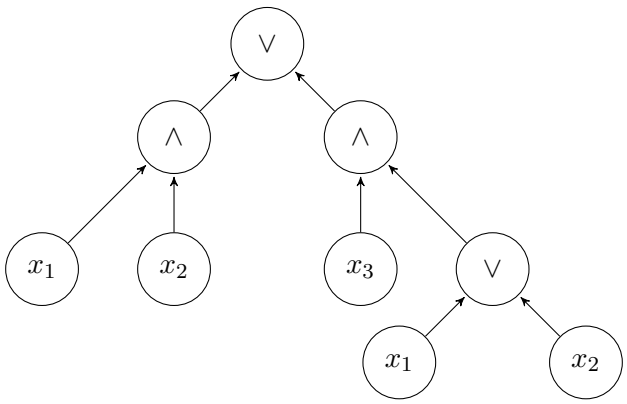

Figure 7: MAJ_3 formula with leafsize five.

[11] And monotone formulas apparently!

Let $F(\mathbf{y})$ be the function computed by T. For a random projection π, $\Pr[F_\pi(\mathbf{x}) \neq \text{MAJ}_n(\mathbf{x})] < \frac{1}{2^n}$ so there exists a projection π^* such that $F_{\pi^*}(\mathbf{x}) = \text{MAJ}_n(\mathbf{x})$ on all inputs $\mathbf{x}$. Hard-wire π^* into T to get a formula which exactly computes MAJ_n. Thus $\mathcal{L}(\text{MAJ}_n) \leq 5^k = n^{c \log 5} \in O(n^7)$. □

6.1.2 Slice Functions

The following is a striking consequence of Theorem 47.

Definition 48. *$f : \{0,1\}^n \to \{0,1\}$ is a **Slice functions** if there exists $k \in \{0, ..., n\}$ such that $f(x) = 0$ if $|x| < k$ and $f(x) = 1$ if $|x| > k$.*

A particular type of slice functions is the threshold function.

Definition 49. *A **threshold function** is a function $\text{THR}_{k,n} : \{0,1\}^n \to \{0,1\}$ such that $\text{THR}_{k,n} = 1$ if and only if $|\mathbf{x}| \geq k$. This generalizes MAJ_n.*

Theorem 50. ***(Berkowitz 1982).*** *If f is a slice function then $\mathcal{L}_{mon}(f) \leq \mathcal{L}(f) \cdot \text{poly}(n)$ and $\mathcal{C}_{mon}(f) \leq \mathcal{C}(f) + \text{poly}(n)$.*

Proof. We will describe the procedure for turning the monotone poly-size formula for MAJ_n into a monotone formula for any slice function. First observe that you can construct a monotone formula for $\text{THR}_{k,n}$ by padding MAJ with an appropriate number of 0s or 1s[12].

Let F be a DeMorgan formula computing k-slice function f. Write F in negation normal form. Let F' be F with each $\bar{x}_i$ replace with $\text{THR}_{k,n}(x_1, ..., x_{i-1}, 0, x_{i+1}, ..., x_n)$. Then observe that $\text{THR}_{k,n} \wedge F'$ is a monotone formula with leafsize $\mathcal{L}f \cdot \text{poly}(n)$ which computes f.

If fewer than k variables are true, then $\text{THR}_{k,n}(\mathbf{x}) = 0$. If more than k variables are true, then each of $\text{THR}_{k,n}(x_1, ..., x_{i-1}, 0, x_{i+1}, ..., x_n) = 1$ and $F'(\mathbf{x}) = 1$ since all its inputs are true. Finally, if exactly k variables are true, then $\text{THR}_{k,n}(\mathbf{x}) = 1$ and $\text{THR}_{k,n}(x_1, ..., x_{i-1}, 0, x_{i+1}, ..., x_n) = \bar{x}_i$ and $F'(\mathbf{x}) = F(\mathbf{x})$. □

Observe that for all monotone functions f, $\mathcal{C}(f) \leq \mathcal{C}_{mon}(f)$ and $\mathcal{L}(f) \leq \mathcal{C}_{mon}(f)$ so for monotone circuits a tight bound is known for its leaf and circuit-sizes.

6.2 Bounded Depth Circuits: AC^0

Definition 51. *An AC^0 **circuit** is one consisting of unbounded fan-in $\wedge$ and $\vee$ gates.*

*The **depth** of an AC^0 circuit is the length of its longest root to leaf path.*

Every AC^0 circuit can be composed of alternating layers of $\wedge$ and $\vee$-gates since two adjacent levels with the same gate type can be merged.

Let $\mathcal{C}_d(f)$ ($\mathcal{L}_d(f)$) be the minimum circuit (resp. leaf)-size of a depth d AC^0 circuit (resp. formula) for f.

[12] If $2k > n$ then you want to add $2k - n$ zeros to MAJ_n. Similarly for when $2k < n$.

AC^0 circuits correspond to algorithms which are very efficient to parallelize. In particular, constant-time on polynomially many processors.

Example 52. *Consider the function $+_n$, addition of two n bit numbers. Let the inputs to $+_n$ be $\mathbf{x} = (x_0, ..., x_{n-1})$ and $\mathbf{y} = (y_0, ..., y_{n-1})$ with values $x = \sum_{i=0}^{n-1} 2^i x_i$ and $y = \sum_{j=0}^{n-1} 2^j y_j$ respectively. We want to output $n+1$ bits $z_0, ..., z_n$ such that $\sum_{k=0}^{n} 2^k z_k = x + y$.*

Though this might not seem possible in constant depth since the carry bit is defined recursively, it can be seen in another way. Suppose $x_n = y_n = 0$, then

$$z_k = x_k \oplus y_k \oplus c_k$$
$$c_k = \bigvee_{i=0}^{k-1} (x_i \wedge y_i) \bigwedge_{j=i+1}^{k} (x_j \vee y_j)$$

The way to think about the carry bit c_k at index k is as the result of some $x_i \wedge y_i$ and for all indices $j \in \{i+1, ..., k\}$, the carry is kept alive by $x_j \vee y_j$.

Some common functions which are not in AC^0 include: multiplication of two n-bit integers and the n-bit parity function XOR_n.

Since AC^0 circuits can be arranged with alternating levels of $\wedge$ and $\vee$-gates, define Π_d and Σ_d to be the subclasses of AC^0 who root consists of a $\wedge$-gate and a $\vee$-gate respectively. For functions f, denote the size of the minimal circuits of f in Π_d and Σ_d by $\mathcal{C}_{\pi_d}(f)$ and $\mathcal{C}_{\sigma_d}(f)$.

For the parity function XOR_n, $\mathcal{C}_{\pi_d}(\mathrm{XOR}_n) = \mathcal{C}_{\sigma_d}(\mathrm{XOR}_n)$. To see this, note that $\mathcal{C}_{\pi_d}(f) = \mathcal{C}_{\sigma_d}(1-f)$ for any function and that $\mathcal{C}_{\pi_d}(\mathrm{XOR}_n) = \mathcal{C}_{\pi_d}(\overline{\mathrm{XOR}_n})$ by negating the first input of XOR_n. The same relationship holds with regards to leafsize.

Challenge. Let us consider the leaf size of an n-ary Boolean function f in AC^0 with depth at most d. Naively, we have that $\mathcal{L}_2(\mathrm{XOR}_n) \leq n2^n$ as you can take an "or" of 2^n "and" gates with fan-in n each specifying a setting of the n input variables[13].

Lemma 53. *For k and $n_1, ..., n_k$ where $\sum_{i=1}^{k} n_i = n$ and $d \geq 2$,*

$$\mathcal{L}_{d+1}(\mathrm{XOR}_n) \leq 2^{k-1} \sum_{i=1}^{k} \mathcal{L}_d(\mathrm{XOR}_{n_i}) \text{ and } \mathcal{C}_{d+1}(\mathrm{XOR}_n) \leq 2^{k-1} + 2 \sum_{i=1}^{k} \mathcal{C}_d(\mathrm{XOR}_{n_i}).$$

Proof. Decompose XOR_n into $\mathrm{XOR}_{n_1}, ..., \mathrm{XOR}_{n_k}$ and then take an XOR of these results. Each XOR_{n_i} has leafsize $\mathcal{L}(\mathrm{XOR}_{n_i})$. Further there exists a formula for the top level XOR with leafsize $k2^{k-1}$ and depth 2, using the trivial construction. By collapsing adjacent levels with the same gate type, $\mathcal{L}_{d+1}(\mathrm{XOR}_n)$ is bounded as required. The circuit-size bound can be similarly constructed. □

Using $\mathcal{L}_2(\mathrm{XOR}_n) \leq n2^n$ for the base case and the recurrence shown in Lemma (53), we can show

$$\mathcal{L}_{d+1}(\mathrm{XOR}_n) \leq n2^{dn^{1/d}}$$

[13]It is also easy to see that $\mathcal{C}_2(f) \leq 2^n$ and $\mathcal{L}_2(f) \leq n2^n$ for an arbitrary function f by writing f as a DNF F.

for any n and $d \geq 2$. Simply choose $n_i = \lceil n^{(d-1)/d} \rceil$ for all $i \in [k]$ and $k = \lceil n^{1/d} \rceil$ to obtain

$$\mathcal{L}_{d+1}(\text{XOR}_n) \leq 2^{\lceil n^{1/d} \rceil - 1} \lceil n^{1/d} \rceil \cdot \mathcal{L}_d(\text{XOR}_{n_i}) \leq 2^{\lceil n^{1/d} \rceil - 1} \lceil n^{1/d} \rceil \cdot \lceil n^{(d-1)/d} \rceil 2^{(d-1)n^{1/d}} \leq n 2^{dn^{1/d}}$$

However, when n is a power of 2, we can get a slightly tighter bound of

$$\mathcal{L}_{d+1}(\text{XOR}_n) \leq n 2^{d\left(n^{1/d}-1\right)}.$$

Ben suspects that the above inequality holds for all n, not just powers of two, since for $d = \lceil \log n \rceil$ it is know that $n^{1/d} - 1 = 2^{\log n / \log n} - 1 = 1$ and the exponent of 2 is sufficient i.e. it is known that

$$\mathcal{L}_{\lceil \log n \rceil}(\text{XOR}_n) \in O(n^2).$$

Further, Ben believes that it is sufficient to achieve this tighter bound by analyzing the recurrence relation more carefully.

7 Håstad's Switching Lemma

Definition 54. *A **decision tree** (DT) is a rooted binary tree whose leaves are labelled by $\{0,1\}$ and whose internal nodes are labelled by variables. The **depth** of a decision tree is the length of the longest root-to-leaf path. For $f : \{0,1\}^n \to \{0,1\}$, let $\mathcal{DT}_{\text{depth}}(f)$ denote depth of the minimum depth DT that computes f.*

*It is useful to consider a function as a DNF or CNF. The **width** of a DNF/CNF formula is the maximum number of literals in any clause.*

Definition 55. *Let $F = C_1 \vee \cdots \vee C_m$ be a k-DNF with an arbitrary fixed order on the clauses and variables. Let $\rho : \{x_1, ..., x_n\} \to \{*, 0, 1\}$ be a restriction. Then the **canonical decision tree** of $F \restriction \rho$, denoted $\mathcal{CDT}(F, \rho)$, is constructed as follows.*

1. *Let $F_1 = F \restriction \rho$ and $C_{1,1} \vee \cdots \vee C_{1,k_1}$ be the set of clauses which have not been set to 1 or 0 by ρ.*
2. *Construct a perfect binary tree where each level of the tree is labelled by a in $C_{1,1}$ in the predetermined order e.g. if $C_{1,1} = x_i x_j$ in F_1 and $i < j$, then set the root to x_i and label its two children x_j.*
3. *Let p be a path in our partially constructed tree. Denote by ρ_p a which sets each variable in $C_{1,1}$ according to the path p. Let ℓ be one of the $2^{|C_{1,1}|}$ leafs in our partially constructed tree. If p_ℓ is the root-to-ℓ path, append $\mathcal{CDT}(F', \rho_p)$ to ℓ.*

See the following example.

Example 56. ***(Constructing a canonical decision tree).*** *Let our k-DNF be*

$$F = \bar{x}_1 x_3 x_5 \vee x_1 x_2 \bar{x}_3 \vee x_2 \bar{x}_4 x_5 \vee x_3 x_4 \bar{x}_6 \vee x_1 \bar{x}_4 \bar{x}_7$$

and our restriction $\rho = \{x_1 \mapsto 1, x_4 \mapsto 0\}$. *Apply the restriction* ρ *to* F, *to obtain the following*

$$F \restriction \rho = 0 \vee x_2\bar{x}_3 \vee x_2x_5 \vee 0 \vee \bar{x}_4\bar{x}_7.$$

Look at the first un-falsified clause $x_2\bar{x}_3$ *and build a tree with these variables on the first and second levels. For every root-to-leaf path, construct a restriction by setting the variables* x_2 *and* $\bar{x}_3$ *according to the edge labels. Continue down the tree until all the variables have been added. The first two restrictions are shown in Figure 8.*

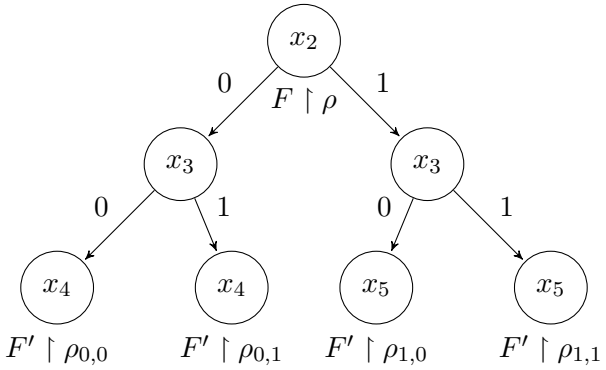

Figure 8: Given the 3-DNF F and the restriction ρ construct the canonical DT for $F \restriction \rho$. Let $F' = F \restriction \rho$ and $\rho_{a,b} = \{x_2 \mapsto a, x_3 \mapsto b\}$.

Observe that every depth d decision tree (DT) is equivalent to a d-DNF (d-CNF) by tracing every root-to-leaf path in the tree which end in a one (resp. end in a zero then apply DeMorgan's rule). It follows that an $\vee$ of depth d DTs is a d-DNF. Further, if f is equivalent to both a k-DNF and l-CNF, then $\mathcal{DT}_{\text{depth}}(f) \leq k \cdot l$. Do you see why this is[14]?

Before looking at the proof of the Switching Lemma, let us consider a simple warm-up pertaining to the depth of a decision tree hit by a random restriction.

Theorem 57. *If* $\mathcal{DT}_{\text{depth}}(f) = k$, *then for any* t

$$\Pr\left[\mathcal{DT}_{\text{depth}}(f \restriction R_p) \geq t\right] \leq (2p)^t \binom{k}{t}.$$

[14]Let f be equivalent to the k-DNF $F = A_1 \vee \cdots \vee A_s$ and $\bar{f}$ be equivalent to the l-DNF $\bar{F} = B_1 \vee \cdots \vee B_t$. Notice that every pair (A_i, B_j) must share a common variable of opposite sign or else there will be a setting of the variable which simultaneously satisfies F and $\bar{F}$. The proof is by induction on the number of variables. When there is only one variable the claim is true. Suppose f has n variables. Build a decision tree of width A_1 which considers all variables in the clause A_1. Consider a restriction ρ corresponding to a root-to-leaf path in the DT we have just created. The DNF-width of $F \restriction \rho$ is at most k. The DNF-width of $\bar{F}$ is *less than* l since at least one variable of each clause was knocked out. Further, by the induction hypothesis, we have

$$\mathcal{DT}_{\text{depth}}(f \restriction \rho) \leq \text{DNF}_{\text{width}}(F \restriction \rho) \cdot \text{DNF}_{\text{width}}(\bar{F} \restriction \rho).$$

By appending the restricted decision tree to every leaf of the depth k DT that we created for the variables in A_1, we have that

$$\mathcal{DT}_{\text{depth}}(f) \leq k + \text{DNF}_{\text{width}}(F \restriction \rho) \cdot \text{DNF}_{\text{width}}(\bar{F} \restriction \rho) \leq k + k(l-1) = kl.$$

Proof. By induction on k. In the base case where $k = 0$, f is a constant function. Suppose that $\mathcal{DT}_{\text{depth}}(f) = d$. Let T be a DT of f of depth d with root labelled x_1. Further let T_0 and T_1 be the left and right sub-trees of T respectively. Either x_1 gets restricted or not. If x_1 gets restricted, then we can apply the induction hypothesis to one of T_0 or T_1, w.l.o.g suppose T_0. This happens with probability $1-p$. If x_1 is unrestricted then it suffices to find the probability that a restriction of T_0 or T_1 has depth $\geq t-1$; w.l.o.g. suppose T_0 is larger. This happens with probability p. Thus

$$\begin{aligned}
\Pr\left[\mathcal{DT}_{\text{depth}}(T \restriction R_p) \geq t\right] &= (1-p)\Pr\left[\mathcal{DT}_{\text{depth}}(T_0 \restriction R_p) \geq t\right] + \\
&\qquad p\Pr\left[(\mathcal{DT}_{\text{depth}}(T_0 \restriction R_p) \geq t-1) \vee (\mathcal{DT}_{\text{depth}}(T_1 \restriction R_p) \geq t-1)\right] \\
&\leq (1-p)\Pr\left[\mathcal{DT}_{\text{depth}}(T_0 \restriction R_p)\right] + 2p\Pr\left[\mathcal{DT}_{\text{depth}}(T_0 \restriction R_p) \geq t-1\right] \\
&\leq (1-p)\cdot(2p)^t\binom{d-1}{t} + 2p\cdot(2p)^{t-1}\binom{d-1}{t-1} \\
&\leq (2p)^t\left(\binom{d-1}{t} + \binom{d-1}{t-1}\right) \\
&= (2p)^t\binom{d}{t}
\end{aligned}$$

where the second inequality follows from a union bound and the equality on the last line is an identity[15]. □

Theorem 58. ***(Håstad's Switching Lemma).*** *If F is a k-DNF (or k-CNF) then*

$$\Pr[\mathcal{DT}_{\text{depth}}(F \restriction R_p) \geq t] \leq (5pk)^t.$$

Proof. This proof due to Razborov as explained in the exposition by Beame. Let[16]

$$B_t \coloneqq \{\rho : \mathcal{DT}_{\text{depth}}(\mathcal{CDT}(F,\rho)) = t\}.$$

We will show that the probability a random restriction $R_p \in B_t$ is bounded by $O(pk)^t$. Let n be the number of variables and m the number of clauses in F.

For each ρ in B_t we want to construct a pair (ρ^*, CODE). ρ^* is a fixed restriction setting a further t variables. CODE is some information needed to uniquely identify ρ. Note that $\text{CODE} = (\mathbf{s}, \pi, \mathbf{n})$ where $\mathbf{s} \in [k]^t$ indicates the location of variables within a clause, $\pi \in \{0,1\}^t$ is the appropriate setting of the variable, and $\mathbf{n} \in \{0,1\}^t$ indicates if we move onto the **n**ext clause. Then, with the pair (ρ^*, CODE) and F in-hand, we can reconstruct ρ.

In the encoding step we are given the k-DNF F and the restriction $\rho \in B_t$. We will build $\rho^* = \rho\sigma_1\cdots\sigma_j$ and $\text{CODE} = \gamma_1\cdots\gamma_j$ incrementally.

1. Let $T = T_1 = \mathcal{CDT}(F,\rho)$ and let $C_1, ..., C_{j_1}$ be the set of clauses with free variables. Let p be the lexicographically first path of length t in T. Such a longest path exists since $\rho \in B_t$ and f is not the constant function.

[15] Remember the binomial coefficients are terms in Pascal's triangle. There we are adding two adjacent terms row $d-1$ to get a term in row d

[16] In many proofs the set B_t is actually called BAD_t. This is illustrative of how you want to think about B_t. It is the set of all *bad* restrictions which cannot be converted into a depth t decision tree.

2. Let σ_1 be the unique setting of the free variables in C_1 such that $C_1 \restriction \sigma_1 \equiv 1$. Let π_1 record the actual setting of the free variables in C_1 along path p as well as their location and number. In particular, $\gamma_1 = (\mathbf{s}_1, \pi_1, \mathbf{n}_1)$ which are the indices, settings, and *is-last-in-clause* properties of the free variables in C_1.

   Consider this first encoding step on Example 56. Suppose the long path p aims to set $x_2 = 1$ and $x_3 = 1$. Then $\sigma_1 = \{x_2 \mapsto 1, x_3 \mapsto 0\}$, $\mathbf{s}_1 = [2, 3]$, $\pi_1 = \{x_2 \mapsto 1, x_3 \mapsto 1\}$, and $\mathbf{n} = [0, 1]$.

3. Proceed down p past all the free variables in C_1. Let T_1 be the remaining subtree. Repeat the process until all t variable on p have been considered.

In the decoding step we are given F and the pair $(\rho^*, \textsc{code})$ where $\rho^* = \rho\sigma_1 \cdots \sigma_j$ and $\textsc{code} = \gamma_1 \cdots \gamma_j$ and want to reconstruct ρ.

1. Evaluate $F \restriction \rho^*$. Let C_1' be the first clause set equal to 1. Observe that $C_1' = C_1$ by construction. Use $\mathbf{n}_1$ to determine how many variables were set during the encoding step and use π_1 and $\mathbf{v}_1$ to reset these values in ρ^* so they match the first $|C_1|$ variables on the longest path p. Let $\rho_1^* = \rho\pi_1\sigma_2 \cdots \sigma_j$.

2. Repeat with ρ_1^* in the place of ρ^* until we have used up all t variables.

All variables consider in the above procedure were originally stars in ρ.

Suppose that ρ has s stars. Then the set $\mathcal{R}_s$ of all restrictions with s stars is of size $\binom{n}{s}2^{n-s}$. Similarly, the set of all restrictions with $s - t$ stars is of size $\binom{n}{s-t}2^{n-s+t}$. Then, since the codes come from a domain of size $(4 \log k)^t$,

$$\Pr[\mathcal{DT}_{\text{depth}}(F \restriction R_p) \geq t] = \frac{|B_t|}{|\mathcal{R}_s|} \leq \frac{|\mathcal{R}_{s-t}|(4 \log k)^t}{|\mathcal{R}_s|} = \frac{\binom{n}{s-t}2^{n-s+t}(4 \log k)^t}{\binom{n}{s}2^{n-s}} \leq (8pk)^t.$$

Considering all bad restrictions with paths of length at least t increases the above probability by a constant. The eight in the upper bound can be improved to five with a more careful analysis. □

It is often natural to choose $p = \frac{1}{10k}$ so that the bound is inverse exponential in t.

Corollary 59. *If F is k-DNF, then*

$$\Pr\left[F \restriction R_p \text{ is not a } t\text{-CNF}\right] \leq (5pk)^t.$$

This corollary follows from Theorem 58 since every depth-t DT is a t-CNF (t-DNF).

7.1 LB Parity Circuit-size

Theorem 60. *Let C be an AC^0 circuit of depth $d + 1$ and size S. Let $p = 10^{-d-1}(2 \log S)^{-d}$.*

$$\Pr[\mathcal{DT}_{\text{depth}}(C \restriction R_p) \geq \ell] \leq \frac{1}{2^\ell} + \frac{1}{S}.$$

Proof. The key idea is to repeatedly apply Theorem 58 being careful to choose appropriate values of k, t and p. Consider the bottom level of C (closest to the literals) and w.l.o.g assume that this is a conjunction ($\wedge$-gate). Since the fan-in consists of literals, you can think of them as width one clauses. Apply the theorem with $k = 1$, $t = 2\log S$, $p_1 = \frac{1}{10}$. According to the switching lemma, we fail[17] at each $\wedge$-node with probability at most $2^{-2\log S} = S^{-2}$. Let E_1 be the event that we successfully completed the switch from 1-CNF to $2\log S$-DNF. Conditioning on E_1, we can collapse this layer of $\vee$-gates with the $\vee$-gates of the level above.

Apply the switching lemma a further $d-1$ iterations. At step i set $k = 2\log S$, $t = 2\log S$, and $p_i = \frac{p_{i-1}}{20\log S}$ each time conditioning on the success of the previous iterations. At the end of these d iterations, take a union bound over the at most S gates we have considered. Since we fail on each level with probability at most S^{-2}, we fail to reach the top-most level with probability at most S^{-1}.

Consider our final iteration. W.l.o.g. the root consists of a $\vee$-gate sitting on top of depth $2\log S$ decision trees (i.e. width $2\log S$-CNF clauses). For the final application of the switching lemma, set $k = 2\log S$, $t = \ell$, and $p_{d+1} = \frac{p_d}{20\log S}$. Thus this iteration fail with probability at most $2^{-\ell}$. In total, this procedure fails with probability $2^{-\ell} + S^{-1}$. □

Corollary 61. $\mathcal{C}_{d+1}(XOR_n) = 2^{\Omega(n^{1/d})}$.

By Theorem 60, the probability that the restriction has depth ≥ 1 is < 1. Thus there is some restriction which sets C to a constant. Since $p = 10^{-d-1}(2\log S)^{-d}$ and $S = 2^{\Omega(n^{1/d})}$, there is a good likely-hood that some variables are un-restricted.

7.2 Switching Lemma for Formulas

The lower bound[18] of $\Omega(dn^{1/d})$ matches the existing upper bound.

8 LB: $\mathsf{AC}^0[p]$ by the Polynomial Method

Goal: want an upper bound on the approximate degree of AND, OR, and MOD_p functions.

Definition 62. *Let $\mathbf{x} = (x_1, ..., x_n)$ then $MOD_p(\mathbf{x}) = 1$ if and only if $\sum_{i=1}^n x_i \equiv 0 \mod p$.*

*Let $A \in \mathbb{F}_p[x_1, ..., x_n]$ be a **random polynomial** over $\mathbb{F}_p$. The **degree** of A is the maximum degree of a polynomial in the support[19] for A.*

Let $f : \{0,1\}^n \to \{0,1\}$. A is an ϵ-approximating for f if

$$\Pr_A[A(x) \neq f(x)] \leq \epsilon$$

for every $x \in \{0,1\}^n$.

[17] That is to say: this node cannot be converted to a short DT under R_p.

[18] A result of Ben's!

[19] Imagine that you have a subset $\mathcal{F} \subset \mathbb{F}_p[\mathbf{x}]$ and that A as a random variable in a distribution over $\mathcal{F}$. Then $\mathcal{F} = \mathrm{supp}(A)$.

The ϵ***-approximate degree of*** f*, denoted* $\deg_\epsilon(f)$*, is the minimum degree of an* ϵ*-approximating random polynomial of* f*. Note that this value is invariant under negation of the inputs and outputs*[20]*.*

Lemma 63. *If* A *is an* ϵ*-approximating random polynomial for* f*, then* $\exists \alpha \in \text{supp}(A)$ *such that*

$$\Pr_{\mathbf{x}\in\{0,1\}^n}[\alpha(x) \neq f(x)] \leq \epsilon$$

Proof. Let A be an ϵ-approximating polynomial for f. Consider the probability for a randomly selected n bit string $\mathbf{x}$ that $A(\mathbf{x}) \neq \mathbf{x}$.

$$\Pr_{\alpha\in\text{supp}(A)}\left[\Pr_{\mathbf{x}\in\{0,1\}^n}[A(\mathbf{x}) \neq f(\mathbf{x})] > \epsilon\right] < \frac{\mathbb{E}_{\alpha\in\text{supp}(A)}\left[\Pr_{\mathbf{x}\in\{0,1\}^n}[\alpha(\mathbf{x}) \neq f(\mathbf{x})]\right]}{\epsilon}$$

by Markov's inequality. Since the RHS is equal to one as A is ϵ-approximating, by the probabilistic method some polynomial $\alpha \in \text{supp}(A)$ must satisfy the above $\leq$ with $\geq$. □

Lemma 64. *Suppose* $f(x) = g(h_1(x), ..., h_m(x))$*. Then for any* $\delta, \epsilon_1, ..., \epsilon_m$*,*

$$\deg_{\delta+\epsilon_1+\cdots+\epsilon_m}(f) \leq \deg_\delta(g) \max_i \deg_{\epsilon_i}(h_i)$$

Proof. Let G and H_i be δ and ϵ_i-approximating polynomials for g and h_i respectively. We will show that $F' \coloneqq G(H_1, ..., H_m)$ is a $\delta + \epsilon_1 + \cdots \epsilon_m$ approximating polynomial for f. Note that

$$\Pr_{\mathbf{x}\in\{0,1\}^n}[f(\mathbf{x}) \neq F'(\mathbf{x})] = \Pr_{\mathbf{x}\in\{0,1\}^n}\left[F'(\mathbf{x}) \neq g(H_1(\mathbf{x}), ..., H_m(\mathbf{x})) \vee \bigvee_{i=1}^m H_i(\mathbf{x}) \neq h_i(\mathbf{x})\right] \leq \delta + \sum_{i=1}^m \epsilon_i$$

where the inequality follows from a union bound. □

Assume that the following are all n-ary Boolean functions.

Lemma 65. ***(***MOD_p ***Function).*** *For any* $\epsilon > 0$ *and* n*,*

$$\deg_\epsilon(\text{MOD}_p) \leq \deg_0(\text{MOD}_p) \leq p - 1$$

Proof. Observe that $\text{MOD}_{p,n} = 1 - (x_1 + \cdots + x_n)^{p-1}$ by Fermat's Little Theorem. Since the RHS is a degree $p-1$ polynomial the inequality holds as required. □

Lemma 66. ***(***OR ***Function).*** $\deg_\epsilon(\text{OR}) \leq p\left(\log_p\left(\frac{1}{\epsilon}\right) + 1\right)$*. One should note that this upper bound is independent of* n*.*

Proof. Consider the polynomial $(\lambda_1 x_1 + \cdots + \lambda_n x_n)^{p-1}$ where $\Lambda = (\lambda_1, ..., \lambda_n)$ is chosen uniformly at random from among $\mathbb{F}_p^n$. Note that we take the inner product to the power $p-1$ to obtain a value in $\{0, 1\}$, again by Fermat's Little Theorem. Observe that

$$\Pr_{\mathbf{x}\in\{0,1\}^n}\left[\text{OR}(\mathbf{x}) \neq \left(\Lambda^\top \mathbf{x}\right)^{p-1}\right] = \begin{cases} 0 & \text{if } \mathbf{x} = \mathbf{0} \\ \frac{1}{p} & \text{if } \mathbf{x} \neq \mathbf{0} \end{cases}$$

[20] Please understand why.

In order to ensure that we have an ϵ-approximate polynomial, we can sample several Λs and take their product. Let $\Lambda_1, ..., \Lambda_t \in \mathbb{F}_p^n$ and let $F(\mathbf{x}) = 1 - \prod_{i=1}^{t} \left(1 - \left(\Lambda_i^\top \mathbf{x}\right)^{p-1}\right)$. Then

$$\Pr_{\mathbf{x} \in \{0,1\}^n} [\mathrm{OR}(\mathbf{x}) \neq F(\mathbf{x})] \leq \frac{1}{p^t}$$

since $F(\mathbf{x}) = 0 \iff (\Lambda_i^\top \mathbf{x})^{p-1} = 0$ for every $i \in [t]$. For $t \geq \left(\log_p \left(\frac{1}{\epsilon}\right) + 1\right)$, $p^{-t} \leq \epsilon^{-1}$. □

Corollary 67. *(*AND ***Function).*** $\deg_\epsilon(\mathrm{AND}) \leq p\left(\log_p \left(\frac{1}{\epsilon}\right) + 1\right)$.

This follows from the results of OR by applying DeMorgan's rule.

Lemma 68. *Any* $\mathsf{AC}^0[p]$ *formula* F*, of size* S *and depth* d*, can be* ϵ*-approximated by a degree* $O(\log_p S/\epsilon)^d$ *i.e.* $\deg_\epsilon(F) \in O(\log_p S/\epsilon)^d$.

Proof. Let the output gate of F be $g \in \{\mathrm{MOD}_p, \mathrm{AND}, \mathrm{OR}\}$. By Lemma 65, Lemma 66, and Corollary 67, we know that $\deg_{1/8}(g) \in O(p)$. □

Corollary 69. *Every* AC^0 *function* F *of depth* d *and size* S *satisfies* $\deg_{1/4}(F) \in O(\log_p S)^d$.

8.1 LB: Parity in $\mathsf{AC}^0[p]$

In order to show the upper bound we need a lemma, an inequality, and an improved version of Corollary 69. These are as follows.

Lemma 70. ***(Kopparty-Srinivasan).*** *For formula* f*,* $\deg_\epsilon(f) \leq \deg_{1/4}(f) \log 1/\epsilon$.

Proof. We will construct an ϵ-approximating polynomial of f by composing several ($\log 1/\epsilon$ to be precise) 1/4-approximating polynomials for f using the majority function.

Let A be a 1/4-approximating polynomial for f, t be a positive integer, and $M(\mathbf{y}) \coloneqq \mathrm{MAJ}_t(\mathbf{y})$ for $\mathbf{y} \in \{0,1\}^t$. Observe[21] that there exists a polynomial for M of degree t. For t independent copies of A, denoted $A_1, ..., A_t$, the polynomial $M'(\mathbf{x}) \coloneqq M(A_1(\mathbf{x}), ..., A_t(\mathbf{x}))$ has degree $t \cdot \deg_{1/4}(f)$.

Define indicator vectors X_i to denote $A_i(\mathbf{x}) = f(\mathbf{x})$. For $X = \sum X_i$, X is a binomial r.v. with probability of success 1/4 and $n = t$. Thus

$$\Pr_{\mathbf{x} \in \{0,1\}^n} [X \geq t/2] \leq \exp(-t/12).$$

When $t \in O(\log(1/\epsilon))$ we have that $\Pr[M'(\mathbf{x}) \neq f(\mathbf{x})] \leq \epsilon$. □

Claim 71. *For* $d \geq 2$*, and arbitrary* a *and* b*,*

$$\left(\frac{a}{d-1} + 1\right)^{d-1} (b+1) \leq \left(\frac{a+b}{d} + 1\right)^d.$$

[21] It turns out that there is a bijection between t-ary Boolean functions f and polynomials in $\mathbb{F}_p[x_1, ..., x_t]$ of degree at most t. It is simplistic to see this in $\mathbb{F}_2[x_1, ..., x_t]$. Take every $\mathbf{x} \in \{0,1\}^t$ such that $f(\mathbf{x}) = 1$ and turn it into a monomial, $\prod_{i=1}^t b_i$ where $b_i = x_i$ if $x_i = 1$ and $b_i = 1 - x_i$ otherwise. For general prime p, simply map $\{0,1\}^t \to \{-1,1\}^t$ and then map the output $\{-1,1\} \to \{0,1\}$.

Proof. Let $\phi(a,b)$ be the polynomial

$$\phi(a,b) = \left(\frac{a+b}{d}+1\right)^d - \left(\frac{a}{d-1}+1\right)^{d-1}(b+1).$$

We will show that $\phi(a,b) \geq 0$ for all a and b. First find the extremum of ϕ by taking the derivative with respect to b,

$$\frac{\partial}{\partial b}\left(\left(\frac{a+b}{d}+1\right)^d - \left(\frac{a}{d-1}+1\right)^{d-1}(b+1)\right) = \left(\frac{a+b}{d}+1\right)^{d-1} - \left(\frac{a}{d-1}+1\right)^{d-1}.$$

Thus the extremum occurs when $b = a/d-1$ with $\phi(a, a/(d-1)) = 0$. Taking the second derivative with respect to b, we see that $\phi(a, a/(d-1))$ is a minimum of the function. □

Now for the improved version of Corollary 69. At first blush, it might not look like that much of an improvement since d and p are constants, but this is exactly the form that we require for the parity bound.

Lemma 72. *For formula F in $\mathsf{AC}^0[p]$ of depth d and size S, $\deg_{1/4}(F) \leq O(p\log_p(S)/d)^d$.*

Proof. By induction on d. Let the output gate of F be $g \in \{\mathrm{MOD}_p, \mathrm{OR}, \mathrm{AND}\}$ with inputs $F_1, ..., F_m$ of sizes $S_1, ..., S_m$ respectively. Then by Lemma 64,

$$\deg_{1/4}(F) \leq \deg_{1/8}(g) \cdot \max_{i\in[m]} \deg_{S_i/(8S)}(F_i).$$

By Lemma 68, we have that $\deg_{1/8}(g) \in O(p)$. By Lemma 70, we have that

$$\begin{aligned}\deg_{S_i/(8S)}(F_i) &\leq \deg_{1/4}(F_i)\log\left(\frac{8S}{S_i}\right)\\ &\leq \log\left(\frac{8S}{S_i}\right)\end{aligned}$$

where the second inequality follows from the induction hypothesis. Finally, it is time to use Claim 71. Don't panic. Just set $a = \log_p(S_i/p^{d-1})$ and $b = \log_p(S/pS_i)$ to get the inequality

$$p^{d-1}\left(\frac{\log_p(S_i)}{d-1}\right)^{d-1} \leq p^{d-1}\left(\frac{\log_p S}{d}\right)^d.$$

Altogether we have

$$\deg_{1/4}(F) \leq \deg_{1/8}(g)\cdot\max_{i\in[m]}\deg_{S_i/(8S)}(F_i) \leq p\left(p^{d-1}\left(\frac{\log_p S}{d}\right)^d\right) = \left(\frac{p\log_p S}{d}\right)^d$$

for sufficiently large n. □

Theorem 73. *For prime $p \geq 3$. If $A \in \mathbb{F}_p[x_1, ..., x_n]$ with $\deg(A) = \Delta$, then*

$$\Pr_{\mathbf{x} \in \{0,1\}^n}[A(\mathbf{x}) = \mathrm{XOR}_n(\mathbf{x})] \leq \frac{1}{2} + O\left(\frac{\Delta}{\sqrt{n}}\right).$$

Proof. Probably the most involved proof in this section. Uses a couple of clever tricks and something known as a *dimension argument*.

First let $\lambda : \{0,1\} \to \{1,-1\}$ be the $\lambda(x) = 1 - 2x$. We want to be able to reduce the degree of a polynomial. To this end, it is helpful to have variable from $y_i \in \{-1, 1\}$ since $y_i^2 = 1$. Observe that

$$\lambda(x_1 \oplus \cdots \oplus x_n) = \prod_{i=1}^{n} (-1)^{x_i} = \prod_{i=1}^{n} \lambda(x_i).$$

Thus, by substituting $\lambda^{-1}(y_i) = x_i$,

$$\lambda\left(\mathrm{XOR}(\mathbf{x})\right) = \lambda\left(\mathrm{XOR}(\lambda^{-1}(\mathbf{y}))\right) = \prod_{i=1}^{n} y_i.$$

Let $\tilde{A}(\mathbf{y}) = \lambda(\mathrm{XOR}(\lambda^{-1}(\mathbf{y})))$ and notice that $\deg(\tilde{A}) = \Delta$. It suffices to show that

$$\Pr_{\mathbf{x} \in \{0,1\}^n}\left[\tilde{A}(\mathbf{y}) = \prod_{i=1}^{n} y_i\right] \leq \frac{1}{2} + \frac{O(\Delta)}{\sqrt{n}}.$$

Consider the set of vectors $S = \{\mathbf{y} \in \{0,1\}^n : \tilde{A}(\mathbf{y}) = \prod_{i \in [n]} y_i\}$. Observe that the probability above is equal to $|S|/2^n$. Thus it suffices to find $|S|$. Consider $\mathcal{F}$, the set of all functions $f : S \to \mathbb{F}_p$. Since each S can be any of $|\mathbb{F}_p| = p$ different values, $|\mathcal{F}| = p^{|S|}$ and $|S| = \log_p |\mathcal{F}|$.

Let us think about the aforementioned function f. We claim that f is equivalent to a degree $(n + \Delta)/2$ polynomial over $\mathbb{F}_p[y_1, ..., y_n]$. To see this, take any polynomial B which matches f over S. Since $y_i^2 = 1$ as $y_i \in \{1, -1\}$, multi-linearize B. Next take each monomial $\prod_{i \in I} y_i$ with $|I| \geq (n + \Delta)/2$ and substitute

$$\tilde{A}(y) \cdot \prod_{j \in [n] \setminus I} y_j.$$

Notice that this monomial is identical to the original since

$$\tilde{A}(y) \cdot \prod_{j \in [n] \setminus I} y_j = \left(\prod_{i \in [n]} y_i\right) \cdot \left(\prod_{j \in [n] \setminus I} y_j\right) = \left(\prod_{i \in I} y_i\right) \cdot \left(\prod_{j \in [n] \setminus I} y_j^2\right) = \prod_{i \in I} y_i$$

and has degree

$$\deg(A) + (n - |I|) \leq \Delta + n - \frac{(n + \Delta)}{2} = \frac{n + \Delta}{2}.$$

Thus we can bound $|S|$ as follows:

$$\begin{aligned}
|S| &= \log_p |\mathcal{F}| \\
&\leq \log_p \left| \text{monomials of degree at most } \frac{n+\Delta}{2} \right| \\
&= \underbrace{\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n/2}}_{2^{n-1}} + \underbrace{\binom{n}{n/2+1} + \cdots + \binom{n}{\frac{n+\Delta}{2}}}_{< \frac{\Delta 2^n}{\sqrt{n}}}
\end{aligned}$$

and the probability as follows:

$$\Pr_{\mathbf{x} \in \{0,1\}^n} \left[\tilde{A}(\mathbf{y}) = \prod_{i=1}^{n} y_i \right] = \frac{|S|}{2^n} \leq \frac{1}{2} + \frac{O(\Delta)}{\sqrt{n}}.$$

□

Theorem 74. *An* XOR_n *formula of depth* d *in* $\mathsf{AC}^0[p]$ *require size* $2^{\Omega(d(n^{1/2d}-1))}$.

Proof. By Lemma 72, there exists a random polynomial A of degree $\left(\frac{p \log_p S}{d}\right)^d$ which 1/4-approximates XOR_n. This means, for all $\mathbf{x} \in \{0,1\}^n$,

$$\Pr_A[A(\mathbf{x}) = \mathrm{XOR}_n(\mathbf{x})] \geq \frac{3}{4}.$$

Thus there exists an polynomial A which matches XOR_n on more than 3/4 of all vectors $\mathbf{x}$. However, by Theorem 73, we know that

$$\Pr_{\mathbf{x} \in \{0,1\}^n}[A(\mathbf{x}) = \mathrm{XOR}_n(\mathbf{x})] \leq \frac{1}{2} + O\left(\left(\frac{p \log_p S}{d} \right)^d \frac{1}{\sqrt{n}} \right).$$

It follows that the RHS must be greater or equal to 3/4. Rearranging this inequality in-terms of S, we have that $S = 2^{\Omega(d(n^{1/2d}-1))}$. □

8.2 Real Approximating Polynomials in Other Norms

Instead of approximating Boolean functions by polynomials over finite fields, we can try to approximate them with polynomials $A \in \mathbb{R}[x_1, ..., x_n]$. The three primary norms we will consider are ℓ_0, ℓ_∞, and ℓ_2. For finite fields, only ℓ_0 makes sense.

8.2.1 ℓ_0-Approximation

We can find a low-degree ϵ-approximating for OR_n in ℓ_0-norm using what is known as the Valiant Vazirani Isolation Lemma. This lemma seems quite useful in other contexts as well.

Lemma 75. *Let $S_0 = [n]$ and $j \in [\log n + 1]$, S_j be a uniformly random subset of S_{j-1}. Then for every nonempty subset $X \subseteq [n]$,*

$$\Pr[\exists j : |X \cap S_j| = 1] \geq \frac{1}{6}.$$

Proof. The key thing to realize is that $[n] = S_0 \supseteq X$ so $|S_0 \cap X| \geq 1$. The only time when the event $E \coloneqq \exists j : |X \cap S_j| = 1$ does not occur is when every set intersects with X too much, or there is some S_j whose intersection with X differs greatly from $S_{j-1} \cap X$. Formally, this means

$$\Pr[\forall j : |X \cap S_j| \geq 1] \leq \Pr[|X \cap S_{\log(n)+1}| \geq 2] + \Pr[\exists j : |X \cap S_{j-1}| \geq 2 \text{ and } |X \cap S_j| = 0].$$

As always, if we can upper bound the two terms on the RHS, then we can upper bound the complement of E and thus lower bound the probability of E.

To bound $\Pr[|X \cap S_{\log(n)+1}| \geq 2]$ consider the probability that any $x_i \in S_{\log(n)+1}$:

$$\Pr[x_i \in S_{\log(n)+1}] = \Pr[x_i \in S_{\log(n)+1} | x_i \in S_{\log n}] \cdots \Pr[x_i \in S_1 | x_i \in S_0] \Pr[x_i \in S_0] = 2^{-\log(n)-1}.$$

Thus $\Pr[|X \cap S_{\log(n)+1}| \geq 2] \leq |X| 2^{-\log(n)-1} \leq n 2^{-\log(n)-1} = 2^{-1}$.

To bound $\Pr[\exists j : |X \cap S_{j-1}| \geq 2 \text{ and } |X \cap S_j| = 0]$ consider all possible values of $|X \cap S_{j-1}|$

$$\Pr[\exists j : |X \cap S_j| \geq \text{ and } |X \cap S_{j-1}| = 0] \leq \frac{1}{3}$$

Together we have

$$\Pr[\exists j : |X \cap S_j| = 1] \geq 1 - \left(\frac{1}{2} + \frac{1}{3}\right) = \frac{1}{6}$$

as required. □

Lemma 76. *(Aspenes et. al. 1993) There exists a random polynomial $A \in \mathbb{R}[x_1, ..., x_n]$ of degree $O(\log(1/\epsilon) \cdot \log(n))$ such that, for every $\mathbf{x} \in \{0,1\}^n$*

$$\Pr_A[A(\mathbf{x}) \neq \mathrm{OR}_n(\mathbf{x})] \leq \epsilon.$$

Proof. Choose random sets $S_0, ..., S_{\log(n)+1}$ as defined in Lemma 75. For $0 \leq j \leq \log(n) + 1$ define the *degree one polynomial*[22] $B_j(\mathbf{x}) = \sum_{i \in S_j} x_i$. Let

$$B(\mathbf{x}) = 1 - \prod_{i=1}^{\log(n)+1} \left(1 - B_j(\mathbf{x})\right).$$

Observe that $B(\mathbf{x})$ is a degree $O(\log n)$ polynomial and $\Pr[B(\mathbf{x}) \neq \mathrm{OR}_n(\mathbf{x})] \leq 1/6$. To see that the latter statement is true consider $\mathbf{x}$ such that $\mathrm{OR}_n(\mathbf{x}) = 0$ and 1 respectively.

[22]Note! This is the *sum* of the terms in S_j and *NOT* the product even though a monomial might seem more natural.

If $\mathrm{OR}_n(\mathbf{x}) = 0$ then $\mathbf{x} = \mathbf{0}$. It follows that $B_j(\mathbf{x}) = 0$ and $B(\mathbf{x}) = 0$. If $\mathrm{OR}_n(\mathbf{x}) = 1$ then

$$\begin{aligned}\Pr[B(\mathbf{x}) = \mathrm{OR}_n] &= \Pr[B(\mathbf{x}) = 1] \\ &= \Pr[\exists j : B_j(\mathbf{x}) = 1] \\ &= \Pr[\exists j : |S_j \cap \{x_j : x_j = 1\}| = 1] \\ &\geq \frac{1}{6}\end{aligned}$$

We will take t independent copies of B, denoted $B^{(1)}, ..., B^{(t)}$ and combine them together to form

$$A(\mathbf{x}) = 1 - \prod_{i=1}^{t} \left(1 - B^{(i)}(\mathbf{x})\right)$$

a degree $O(t \log n)$ real polynomial. Take $t = \log_{5/6}(\epsilon^{-1})$ to be ϵ-approximating[23]. □

8.2.2 ℓ_∞-Approximation

Let f be a n-ary Boolean function. The ℓ_∞-approximating degree of f, denoted $\tilde{\deg}(f)$, is the minimum degree of a real polynomial $\tilde{f} \in \mathbb{R}[x_1, ..., x_n]$ such that $|f(\mathbf{x}) - \tilde{f}(\mathbf{x})| \leq 1/3$ for all $\mathbf{x} \in \{0,1\}^n$. It is an open problem if there exists an AC^0 function $f : \{0,1\}^n \to \{0,1\}$ with $\tilde{\deg}(f) = \Omega(n)$. A bound for depth-$d$ functions, is due to Bun and Thaler with degree $n^{1-1/2^{O(d)}}$. Apparently Sherstov has some work on this question as well.

8.2.3 ℓ_2-Approximation

This is apparently the most natural and well-studied of all polynomial approximations. From what I understand, most of it can be categorized under Fourier analysis.

9 LB: Monotone for k-Clique

9.1 Problem Definition

Let $G = (V, E)$ be a simple graph with no isolated vertex. For a growing parameter n, let $\mathcal{G}$ be the set of all graphs G with $|V_G| \subseteq [n]$. Since graphs with no isolated vertices are determined by their edge sets, identify each graph $G \in \mathcal{G}$ by a vector in $\{0,1\}^{\binom{n}{2}}$. In the following, all functions f, g, and h will be monotone $\mathcal{G} \to \{0,1\}$ Boolean functions.

A graph $X \in \mathcal{G}$ is a **minterm** of f if $f(X) = 1$ and $f(X') = 0$ for all proper subsets $X' \subsetneq X$. Let $\mathrm{Minterms}(f)$ be the set of minterms of f.

For $X \in \mathcal{G}$, let $\mathrm{Ind}_X : \mathcal{G} \to \{0,1\}$ be the X-subgraph indicator r.v. where for graph G,

$$\mathrm{Ind}_X(G) = 1 \iff X \subseteq G.$$

[23]Notice that this is slightly different from Lemma 70. Instead of a binomial r.v., you have t independent Bernoulli r.v.s with probability of success 1/6 and you just need one of them to be successful

Note that every f monotone, is equivalent to the disjunction of all its subgraph indicators over its minterms i.e.

$$f(G) = \bigvee_{X \in \text{Minterms}(g)} \text{Ind}_X(G).$$

For a fixed graph H, with $|V_H| \leq k$, let $\mathcal{G}_H$ be the subset of graphs in $\mathcal{G}$ which are isomorphic to H. Define the function $\text{Sub}_H(G) : \mathcal{G} \to \mathbb{N}$ such that

$$\text{Sub}_H(G) \coloneqq \sum_{X \in \mathcal{G}_H} \text{Ind}_X(G)$$

so $\text{Sub}_H(G)$ counts the number of subgraphs of G isomorphic to H, in the loosest sense. Let $\text{Minterms}_H(f) = \text{Minterms}(f) \cap \mathcal{G}_H$.

For $k \geq 3$, $k-\text{CLIQUE} : \mathcal{G} \to \{0, 1\}$ is the monotone function defined by $\text{Minterms}(k-\text{CLIQUE}) = \mathcal{G}_{K_k}$.

Let $\mathcal{Y}$ be a uniform random graph in $\mathcal{G}_{K_k}$. This is our "Yes"-instance. Let $\mathcal{N}$ be the Erdös-Renyi random graph with probability $p = n^{-2/(k-1)}$. Given this particular choice of p, is it is unlikely that $\mathcal{N}$ will contain a k-clique. This will be our "No" instance.

Lemma 77.

$$\Pr[\mathcal{N} \textit{ contains a } k\textit{-clique}] \leq \frac{3}{4}.$$

Proof. This is just a one-liner if you remember the upper bound for the binomial coefficient.

$$\Pr[\mathcal{N} \text{ contains a } k\text{-clique}] \leq \binom{n}{k} p^{\binom{k}{2}} \leq \left(\frac{en}{k}\right)^k \left(n^{2/(k-1)}\right)^{\binom{n}{2}} = \left(\frac{en}{k}\right)^k n^{-k} = \left(\frac{e}{k}\right)^k \leq \left(\frac{e}{3}\right)^3 \leq \frac{3}{4}$$

where the inequalities are by union bound, upper bound for binomial coefficients, and $k \geq 3$ respectively. □

Corollary 78.

$$\Pr[k - \text{CLIQUE}(\mathcal{Y}) = 1] + \Pr[k - \text{CLIQUE}(\mathcal{N}) = 0] \geq \frac{5}{4}$$

We only require the following theorem:

Theorem 79. *Every monotone circuit C satisfies*

$$\Pr[C(\mathcal{Y}) = 1] + \Pr[C(\mathcal{N}) = 0] \leq 1 + o(1) + \frac{\mathcal{C}(C)}{\Omega(n^{k/4})}.$$

By combining Lemma 78 and Theorem 79 we can obtain a lower bound on $\mathcal{C}(C)$.

Be warned. The proof of Theorem 79 is a dozy. It requires a slew of new definitions and pretty subtle reasoning throughout, so get ready.

9.2 More Definitions

A graph H is **small** if $|V(H)| < k/2$ and **medium** if $|V(H)| \geq k/2$ and there exists small graphs H_1 and H_2 such that $H = H_1 \cup H_2 = H$. *Notice: the union of two small graphs is a small or a medium graph.*

Let $\epsilon \coloneqq n^{-3k}$. A monotone function $f : \mathcal{G} \to \{0,1\}$ is **closed** if, for every small-or-medium graph $X \in \mathcal{G}$,

$$\Pr[f(\mathcal{N} \cup X) = 1] \geq 1 - \epsilon \implies f(X) = 1.$$

If f and g are closed then so is $f \wedge g$. Every monotone function f has a unique **closure**, denoted $\mathrm{cl}(f)$ which is the minimum closed function such that $f \leq \mathrm{cl}(f)$.

A monotone function $f : \mathcal{G} \to \{0,1\}$ is **trimmed** if every minterm of f is small. For any f, let $\mathrm{trim}(f) : \mathcal{G} \to \{0,1\}$ be the monotone function

$$\mathrm{trim}(f)(G) = 1 \iff \exists X \subseteq G : X \text{ is small and } f(X) = 1.$$

Note that $\mathrm{trim}(f) \leq f$, with equality if and only if f is trimmed [24].

Let $\mathcal{A} \coloneqq \{\mathrm{trim}(\mathrm{cl}(h)) : h \text{ monotone}\}$. Every function $h \in \mathcal{A}$ is an **approximator**. Define operations: $\sqcup, \sqcap : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ as follows. For approximators $f, g \in \mathcal{A}$,

$$f \sqcup g \coloneqq \mathrm{trim}(\mathrm{cl}(f \vee g)) \text{ and } f \sqcap g \coloneqq \mathrm{trim}(\mathrm{cl}(f \wedge g)).$$

Note that every edge indicator function Ind_e is an approximator since $\mathrm{Ind}_e = \mathrm{trim}(\mathrm{cl}(\mathrm{Ind}_e))$.

If C is a monotone $\{\wedge, \vee\}$-circuit with inputs labeled by edge-indicators Ind_e, then the **approximator circuit** $C^{\mathcal{A}}$ is the corresponding

9.3 Proof Outline

Again, our main objective is to prove Theorem 79. That is, we want to come up with an upper bound on probability that the circuit C behaves as expected on $\mathcal{Y}$ and $\mathcal{N}$. Instead of doing this directly, we will apply total probability to an approximate version of C, denoted $C^{\mathcal{A}}$. In particular we will upper bound the sum as follows

$$\begin{aligned}
\Pr[C(\mathcal{Y}) = 1] + \Pr[C(\mathcal{N}) = 0] &\leq \Pr[C(\mathcal{Y}) = 1 \wedge C^{\mathcal{A}}(\mathcal{Y}) = 1] + \Pr[C(\mathcal{Y}) = 1 \wedge C^{\mathcal{A}}(\mathcal{Y}) = 0] \\
&\quad + \Pr[C(\mathcal{N}) = 0 \wedge C^{\mathcal{A}}(\mathcal{N}) = 1] + \Pr[C(\mathcal{N}) = 0 \wedge C^{\mathcal{A}}(\mathcal{N}) = 0] \\
&\leq \Pr[C^{\mathcal{A}}(\mathcal{Y}) = 1] + \Pr[C(\mathcal{Y}) = 1 \wedge C^{\mathcal{A}}(\mathcal{Y}) = 0] \\
&\quad + \Pr[C(\mathcal{N}) = 0 \wedge C^{\mathcal{A}}(\mathcal{N}) = 1] + \Pr[C^{\mathcal{A}}(\mathcal{N}) = 0] \\
&\leq 1 + o(1) + \frac{\mathcal{C}(C)}{\Omega(n^{k/4})}.
\end{aligned}$$

Further we will divide the last inequality into two

$$\Pr[C^{\mathcal{A}}(\mathcal{Y}) = 1] + \Pr[C^{\mathcal{A}}(\mathcal{N}) = 0] \leq 1 + o(1)$$

[24] The name actually makes a lot of sense since, $\mathrm{trim}(f)$ is basically just f with the medium graphs "trimmed" off.

and

$$\Pr[C(\mathcal{Y}) = 1 \wedge C^{\mathcal{A}}(\mathcal{Y}) = 0] + \Pr[C(\mathcal{N}) = 0 \wedge C^{\mathcal{A}}(\mathcal{N}) = 1] \leq \frac{\mathcal{C}(C)}{\Omega(n^{k/4})}.$$

It remains to prove these two inequalities. The bulk of the setup is used for the second one. There you will really need to understand the definition of trims and closures and be able to reason about them confidently. The first inequality is actually just a consequence of a simple lemma.

9.4 Proof Proper

Recall that $\mathcal{N}$ is an Erdös-Renyi graph with $p = n^{-2/(k-1)}$.

Lemma 80. *For any $H \subseteq K_k$,*

$$\mathbb{E}[\mathrm{Sub}_H(\mathcal{N})] = \Theta\left(n^{|V_H|} - 2|E_H|/(k-1)\right) \in \Omega(n^{k/4}).$$

Proof.

$$\mathbb{E}[\mathrm{Sub}_H(\mathcal{N})] = |\mathcal{G}_H| \cdot p^{|E_H|} = \frac{|V_H|!\binom{n}{|V_H|}}{\mathrm{auto}(H)} p^{|E_H|} = O\left(n^{|V_H| - 2|E_H|/(k-1)}\right).$$

The equalities are as follows. Apply linearity of expectation to sum over the expectation of each graph isomorphic to H. Each element of $\mathcal{G}_H$ consists of choosing a permutation of $|V_H|$ vertices and removing their duplicates. The number of automorphisms of H divides $|V_H|$, so $|\mathrm{auto}(H)| \leq |V_H|!$ and $|\mathcal{G}_H| \in \Theta(n^{|V_H|})$. □

Suppose $|V_H| = \lambda k$ for some $\lambda \in [0, 1]$. Then we have

$$|V_H| - 2|E_H|/(k-1) \geq \lambda k - \frac{\lambda k(\lambda k - 1)}{k-1} = \frac{k^2\lambda(1-\lambda)}{k-1} \geq k\lambda(1-\lambda).$$

When λ is close to $1/2$, $\mathbb{E}[\mathrm{Sub}_H(\mathcal{N})] \geq n^{k/4}$. For medium graphs, the lower bound is $\Omega(n^{k/4} + \Omega(1/k))$.

Lemma 81. *If f is closed and $H \subseteq K_k$ is small-or-medium, then*

$$|\mathrm{Minterms}_H(f)| \leq \frac{n^{o(1)}}{p^{|E(H)|}}.$$

Proof. Let $\mathcal{M} = \mathrm{Minterms}_H(f)$ and $t = |E(H)|$. We will show a weaker version of this lemma using the Erdös-Rado Sunflower[25]. By the lemma, if $|\mathcal{M}| \geq t!s^t$, then there exists a sunflower of size s i.e. $X_1, ..., X_s \in \mathcal{M}$ such that $X_1 \backslash Z, X_s \backslash Z$ disjoint where $Z \coloneqq X_1 \cap \cdots \cap X_s$. Since we want a lower bound on $|\mathcal{M}|$ we will assume for a contradiction that the above inequality holds.

[25] To get the inequality as shown above, you need to used the "Approximate Sunflower Lemma" shown in Ben's notes.

Let $s = \ln(1/\epsilon)/p^t$. Then there exists a sunflower $X_1, ..., X_s \in \mathcal{M}$ with core Z. Since Z is a proper subset of the minterms of f, $f(Z) = 0$. But we can show that f is not closed as

$$\begin{aligned}
\Pr[f(\mathcal{N} \cup Z) = 0] &= \Pr\left[\bigwedge_{i=1}^{s} (X_i \backslash Z) \not\subseteq \mathcal{N}\right] \\
&= \prod_{i=1}^{s} \Pr[(X_i \backslash Z) \not\subseteq \mathcal{N}] \\
&= \left(1 - p^{|X_i \backslash Z|}\right)^s \\
&\leq \left(1 - p^t\right)^s \leq \exp(-ts) < \epsilon.
\end{aligned}$$

The equality in the first line might be the most difficult to see, but if $X_i \backslash Z \subseteq \mathcal{N}$ for some i then $f(\mathcal{N} \cup Z) = 1$. □

Take $\epsilon = n^{-3k}$.

Lemma 82. *For every monotone function $f : \mathcal{G} \to \{0, 1\}$,*

$$\Pr[f(\mathcal{N}) \wedge \mathrm{cl}(f)(\mathcal{N}) = 1] \leq O(n^{-2k}).$$

Proof. Since $f(\mathcal{N}) \neq \mathrm{cl}(f)(\mathcal{N})$, we know that f is not closed. Thus we can construct an increasing sequence of monotone functions $f = f_0 < f_1 < \cdots < f_t = \mathrm{cl}(f)$. For each $i \in [t-1]$, there exists a graph $X_i \subset K_k$ such that

$$\Pr[f_{i-1}(\mathcal{N} \cup X_i) = 1] \in [1 - \epsilon, 1).$$

Let $f_i = f_{i-1} \vee \mathrm{Ind}_{X_i}$ where Ind_{X_i} is the indicator of X_i. Since every X_i can be added at most once, $t < \infty$. Upper bound the probability as follows

$$\begin{aligned}
\Pr[f(\mathcal{N}) = 0 \text{ and } \mathrm{cl}(f)(\mathcal{N}) = 1] &= \Pr[f_0(\mathcal{N}) = 0 \text{ and } f_t(\mathcal{N}) = 1] \\
&= \sum_{i=0}^{t-1} \Pr[f_i(\mathcal{N} = 0) \text{ and } f_{i+1}(\mathcal{N}) = 1] \\
&= \sum_{i=0}^{t-1} \Pr[f_i(\mathcal{N} = 0) \text{ and } X_{i+1} \in \mathcal{N}] \\
&\leq \sum_{i=0}^{t-1} \Pr[f_i(\mathcal{N} \cup X_i)0] \\
&\leq t\epsilon = O(\epsilon n^k) = O(n^{-2k})
\end{aligned}$$

where the inequality on the last line is due to the probability assumption above. □

Lemma 83. *If f and g are trimmed monotone functions, then all minterms of $\mathrm{cl}(f \wedge g)$ and $\mathrm{cl}(f \vee g)$ are small or medium.*

Proof. This proof is more involved than I originally thought. First you should observe that a minterm of $f \wedge g$ is the union of a minterm of f and one of g. Since both f and g are trimmed, the

union of their minterms is a small or medium graph. Finally we can write $\text{cl}(f \wedge g) = (f \wedge g) \vee X_1 \vee \cdots \vee X_t$ for graphs $X_1, ..., X_t \in \cup_{H \subset K_k} \mathcal{G}_H$ which incrementally extend $f \wedge g$ to its closure (similar to the above lemma). Thus the minterms of $\text{cl}(f \wedge g)$ are small or medium. A similar argument can be made for $\text{cl}(f \vee g)$. □

Lemma 84. *Suppose f is closed and every minterm of f is small ormedium. Then*

$$\Pr[f(\mathcal{Y}) = 1 \wedge \text{trim}(f)(\mathcal{Y}) = 0] \leq \frac{1}{\Omega(n^{k/4})}.$$

Proof. Reason about the probability as follows[26]. If $f(\mathcal{Y}) = 1$ and $\text{trim}(f)(\mathcal{Y}) = 0$, then there exists a medium minterm of f which is a subgraph in $\mathcal{Y}$ i.e. all graphs with a k-clique and *nothing else*.

$$\begin{aligned}
\Pr[f(\mathcal{Y}) = 0 \wedge \text{trim}(f)(\mathcal{Y}) = 0] &\leq \Pr\left[\bigvee_{\text{medium graphs } H} \bigvee_{X \in \text{Minterms}_H(f \wedge g)} (X \in \mathcal{Y})\right] \\
&\leq \sum_{\text{medium graphs } H} \sum_{X \in \text{Minterms}_H(f \wedge g)} \Pr[X \in \mathcal{Y}] && \text{(Union Bound)} \\
&= \sum_{\text{medium graphs } H \subseteq K_k} \sum_{X \in \text{Minterms}_H(f \wedge g)} \frac{\binom{n-|V_H|}{k-|V_H|}}{\binom{n}{k}} && (|\mathcal{Y}| = \tbinom{n}{k}) \\
&\leq \sum_{\text{medium graphs } H \subseteq K_k} |\text{Minterms}_H(f \wedge g)| \cdot \frac{1}{\Omega(n^{|V_H|})} \\
&\leq \sum_{\text{medium graphs } H \subseteq K_k} \frac{n^{o(1)}}{p^{|E_H|}} \cdot \frac{1}{\Omega(n^{|V_H|})} && \text{(Lemma 81)} \\
&= \sum_{\text{medium graphs } H \subseteq K_k} \frac{n^{o(1)}}{\Omega(\mathbb{E}[\text{Sub}_H(\mathcal{N})])} \\
&\leq \sum_{\text{medium graphs } H \subseteq K_k} \frac{n^{o(1)}}{\Omega(n^{k/4} + \Omega(1/k))} && \text{(Lemma 80)}
\end{aligned}$$

Since there are at most $2^{\binom{k}{2}} = O(1)$ medium graphs up to isomorphism with at most k vertices, we can bound this above by $\Omega(n^{-k/4})$. □

Lemma 85.

$$\Pr[C(\mathcal{Y}) = 1 \wedge C^{\mathcal{A}}(\mathcal{Y}) = 0] + \Pr[C(\mathcal{N}) = 0 \wedge C^{\mathcal{A}}(\mathcal{N}) = 1] \leq \frac{\mathcal{C}(C)}{\Omega(n^{k/4})}.$$

Proof. This is the crux of the proof right here. We want to show that $C^{\mathcal{A}}$ closely approximates C on both distributions $\mathcal{Y}$ and $\mathcal{N}$.

[26] I have a pretty big issue with this proof. What is g in the following equation?

For error on $\mathcal{Y}$,

$$\begin{aligned}\Pr[(f \wedge g)(\mathcal{Y}) = 1 \wedge (f \sqcap g)(\mathcal{Y}) = 0] &= \Pr[(f \wedge g)(\mathcal{Y}) = 1 \wedge \mathrm{trim}(\mathrm{cl}(f \wedge g))(\mathcal{Y}) = 0] \\ &\leq \Pr[\mathrm{cl}(f \wedge g)(\mathcal{Y}) = 1 \wedge \mathrm{trim}(\mathrm{cl}(f \wedge g))(\mathcal{Y}) = 0] \qquad (f \wedge g \leq \mathrm{cl}(f \wedge g)) \\ &\leq \frac{1}{\Omega(n^{k/4})} \qquad \text{(Lemma 83 and 84)}\end{aligned}$$

It follows that

$$\Pr[\mathcal{C}(\mathcal{Y}) \wedge \mathcal{C}^{\mathcal{A}}(\mathcal{Y}) = 0] \leq \frac{\mathcal{C}(C)}{\Omega(n^{k/4})}.$$

For error on $\mathcal{N}$,

$$\begin{aligned}\Pr[(f \wedge g)(\mathcal{N}) = 0 \wedge (f \sqcap g)(\mathcal{N}) = 1] &= \Pr[(f \wedge g)(\mathcal{Y}) = 1 \wedge \mathrm{trim}(\mathrm{cl}(f \wedge g))(\mathcal{Y}) = 0] \\ &\leq \Pr[(f \wedge g)(\mathcal{Y}) = 1 \wedge \mathrm{cl}(f \wedge g)(\mathcal{Y}) = 0] \qquad (\mathrm{trim}(\mathrm{cl}(f \wedge g)) \leq \mathrm{cl}(f \wedge g)) \\ &\leq \frac{1}{\Omega(n^{2k})} \qquad \text{(Lemma 82)}\end{aligned}$$

It follows that

$$\Pr[\mathcal{C}(\mathcal{N}) = 0 \wedge \mathcal{C}^{\mathcal{A}}(\mathcal{N}) = 1] \leq \frac{\mathcal{C}(C)}{\Omega(n^{2k})}.$$

□

Lemma 86. *For every approximator $f \in \mathcal{A}$, we have*

$$\Pr[f(\mathcal{Y}) = 1] + \Pr[f(\mathcal{N}) = 0] \leq 1 + o(1).$$

Proof. It suffices to show that $\Pr[f(\mathcal{Y}) = 1] \in o(1)$. The proof is remarkably similar to that of Lemma 84. Since f is an approximator, either $f \equiv 1$ or there exists a monotone function h such that $f = \mathrm{trim}(h)$. In the former case we are done so assume the latter holds. Then

$$f = \mathrm{trim}(h) = \bigvee_{\text{non-empty small graphs}H} \bigvee_{X \in \mathrm{Minterms}_H(h)} \mathrm{Ind}_X(\mathcal{Y}).$$

Therefore

$$\Pr[f(\mathcal{Y} = 1)] \leq \sum_{\text{nonemty small graphs}H} \sum_{X \in \mathrm{Minterms}_H(h)} \Pr[X \in \mathcal{Y}] \in o(1).$$

□

10 Depth-3 Sub-exponential Size Circuits

10.1 Motivation

It is difficult to find an explicit function with circuit $\mathrm{SizeDepth}(n, \log n)$. Valiant shows that such functions have sub-exponential depth-three circuits i.e. $\mathrm{SizeDepth}(n, \log n) \subseteq \mathrm{SizeDepth}(2^{n/\log\log n}, 3) =$

SizeDepth$(2^{o(n)}, 3)$. Thus an explicit function $f \in \text{SizeDepth}(2^{\Omega(n)}, 3)$ breaks the above "linear-size log-depth" barrier.

The proof structure is as follows. Start with a circuit $C \in \text{SizeDepth}(n, \log n)$. Remove a large fraction of the edges so that the disconnected pieces have low-depth. These low-depth pieces can only depend on a few inputs. Construct a brute-force depth-two $\wedge$-$\vee$ circuit for each piece. The evaluation of C is equivalent to a $\vee$ of $\wedge$s over the $\wedge$-$\vee$ sub-functions. Collapsing the two $\wedge$ layers yields a depth-three circuit of sub-exponential size.

Before we can proceed with the proof in detail, we need two crucial observations. Let the $G = (V, E)$ be a forest[27]. Define depth(G) to be the maximum depth of any tree in G which in-turn is the length of the longest path in the tree. Let a *depth function* $\delta_k : V \to \{0, 2^k - 1\}$ be such that if $uv \in E$ then $\delta_k(u) < \delta_k(v)$. Observe that there exists a depth function δ_k if and only if the depth$(G) \leq 2^k$.

Suppose $|E| = m$, and depth$(G) = 2^k$. By removing $\frac{m}{k}$ edges we can reduce the depth of G to at most 2^{k-1}. For every edge uv, let $\sigma(uv)$ be the index of the most significant bit in which $\delta_k(u)$ and $\delta_k(v)$ differ. Let $E_i = \{uv \in E : \sigma(uv) = i\}$. Notice that $E_1, ..., E_k$ partition the set of all edges. Since there are k sets and m edges, there exists some set i with at most $\frac{m}{k}$ edges (pigeon hole principle). Remove all edges in E_i. We claim that the remaining forest has depth at most 2^{k-1}. To see this consider the depth function δ_{k-1} which is just δ_k with index i omitted. Notice that δ_{k-1} satisfies the depth function property on all remaining edges; first $\sigma(uv) \neq i$ since all edges in E_i were removed. For edges uv, with $\sigma(uv) < i$, a low order bit was removed so $\delta_{k-1}(u) < \delta_{k-1}(v)$. Finally for $\sigma(uv) > i$, we must have the i^{th} bit of $\delta_k(u) = \delta_k(v)$ so removing it still satisfies $\delta_{k-1}(u) < \delta_{k-1}(v)$.

Suppose our original graph had size cn and depth $c \log n = 2^\ell$ for some constant c. We will apply the above depth reduction technique $\log 2c$ times. The resulting circuit will have depth $2^\ell / 2^{\log 2c} = \log n/2$. And we would have removed at most

$$cn\left(\frac{1}{\ell} + \frac{1}{\ell - 1} + \cdots + \frac{1}{\ell - \log 2c + 1}\right) \leq cn\left(\frac{\log 2c}{\ell - \log 2c + 1}\right) \in O\left(\frac{n}{\log \log n}\right) \text{ edges.}$$

It remains to put together all these disconnected functions to calculate the value of $C(\mathbf{x})$ on input $\mathbf{x} \in \{0, 1\}^n$. Let the value on edge uv be the output of gate u. Let f_h^e be the function which checks if the output of u on assignment h, matches that of the guess. Since the sub-tree rooted at u has depth at most $\log n/2$, f_h^e can depend on at most $\sqrt{n}$ inputs. Let $\mathbf{x}|_e$ be these inputs. Then $f_h^e(\mathbf{x}|_e)$ can be computed by a depth-two $\wedge$-$\vee$ circuit of size $O(\sqrt{n}2^{\sqrt{n}})$.

Note that $C(\mathbf{x}) = 1$ if and only if there exists a setting h of $\mathbf{x}$ such that, the output at the root is one, and all removed edges have values matching their setting by h. This translates into the following circuit: take a $\vee$ over all $O(2^{n/\log \log n})$ settings of the removed edges. For each setting, take an $\wedge$ over the $\frac{n}{\log \log n}$ functions $f_h^e(\mathbf{x}|_e)$ for checking that the guesses for the removed edges e matches their actual value. Each function f_h^e is then expressed by a $\wedge$-$\vee$ circuit of size at most $\sqrt{n}2^{\sqrt{n}}$. By collapsing the two consecutive $\wedge$-gates, we have a depth-three circuit of size $O\left(n\sqrt{n}2^{\sqrt{n}+n/\log \log n}\right) \in O\left(2^{n/\log \log n}\right)$.

[27]Note that the size of formulas is an upper bound on the size of circuits here since the circuit is of bounded degree.

10.2 Method of Finite Limits

Definition 87. *A vector $\mathbf{y} \in \{0,1\}^n$ is a k-**limit** of a set $B \subset \{0,1\}^n$ if for every subset $S \subseteq [n]$, there exists a $\mathbf{b} \in B$ such that $\mathbf{y} \neq \mathbf{b}$ but $\mathbf{y}$ and $\mathbf{b}$ look indistinguishable on the S-indexed bits i.e. $\mathbf{y}^S = \mathbf{b}^S$.*

*Further, we say that $\mathbf{y}$ is a **lower** k-**limit** if, in addition, $\mathbf{y} \leq \mathbf{b}$, where the inequality is taken entry-wise.*

A circuit has **bottom neg-fan-in** k if the gates at the bottom, next to the inputs, have at most k negated literals[28]. Further, a circuit C **separates** a pair $A, B \subseteq \{0,1\}^n$, with A, B disjoint if $C(\mathbf{x}) = 1$ for all $\mathbf{x} \in A$ and $C(\mathbf{x}) = 0$ for all $\mathbf{x} \in B$.

In order to use this method, we require a useful lemma.

Lemma 88. *If every $\frac{1}{\ell}$ fraction of vectors in B has a lower k-limit in A, then every Π_3 circuit of bottom neg-fan-in k separating (A, B) must have top fan-in larger than ℓ.*

Proof. The top $\wedge$-gate is connected to various $\vee$ gates. Each $\mathbf{b} \in B$ must evaluate to 0 on some $\vee$-gate. By PHP, there exists a $\vee$-gate g which separates A and B', for some $B' \subseteq B$ of size at least $|B|\ell^{-1}$ i.e. $g(\mathbf{a}) = 1$ for every $\mathbf{a} \in A$ and $g(\mathbf{b}) = 0$ for every $\mathbf{b} \in B'$. Fix a lower k-limit $\mathbf{a} \in A$ of B'. Consider an $\wedge$-gate h feeding into g. Let

$$h(\mathbf{x}) = \bigwedge_{i \in S} \overline{x}_i \wedge \bigwedge_{j \in T} x_j.$$

Since $\mathbf{a}$ is a lower k-limit of B', there exists an element $\mathbf{b} \in B'$ such that $\mathbf{a}^S = \mathbf{b}^S$ and $\mathbf{a} \leq \mathbf{b}$. Recall that g rejects $\mathbf{b}$, thus $h(\mathbf{b}) = 0$. Either $\wedge_{i \in S} \overline{b}_i = 0$ or $\wedge_{j \in T} b_j = 0$. In the former case $\wedge_{i \in S} \overline{a}_i = 0$. In the later case, since $\mathbf{a} \leq \mathbf{b}$, for any $b_j = 0$ where $j \in T$, $a_j = 0$ as well. Thus $\wedge_{j \in T} a_j = 0$. Since $\mathbf{a}$ evaluates to zero on every $\wedge$-gate input to g, $g(\mathbf{a}) = 0$ contradicting the assumption.

□

Apply the method as follows. To show that a circuit has size at least ℓ:

1. Suppose f can be computed with a circuit of size $< \ell$.
2. Set a bunch of inputs to constants so that the fan-in of the bottom gates is at most k. This will ensure that we meet the *bottom neg-fan-in k* condition.
3. Find sets $A \subset f^{-1}(1)$ and $B \subset f^{-1}(0)$ s.t. every $\frac{1}{\ell}$ fraction of vectors in B has a k-limit in A.
4. Use Lemma 88 to show that the top fan-in is larger than ℓ.

In order to find the appropriate inputs to set constant, use the following lemma.

[28] This is not with the other definitions since it is pretty peculiar to this technique.

Lemma 89. *Let $\mathcal{F}$ be a family of subsets of $[n]$, each with cardinality at least k. If*

$$|\mathcal{F}| \leq \left(\frac{n}{m}\right)^{k/2},$$

then some subset $T \subseteq [n]$ of size $n-m$ intersects every member of $\mathcal{F}$.

Proof. Build the set T constructively using PHP. Initially there exists an element t_1 which is in at least k sets of $\mathcal{F} = \mathcal{F}_0$. Add t_1 to t and let $\mathcal{F}_1 = \mathcal{F}_0 - \{S \in \mathcal{F}_0 : t_1 \in S\}$. Now there exists an element t_2 contained in at least $k-1$ sets of $\mathcal{F}_1$. Again, add t_2 to T and let $\mathcal{F}_2 = \mathcal{F}_1 - \{S \in \mathcal{F}_1 : t_2 \in S\}$. Generally, at the i^{th} iteration, there exists an element t_i which is contained in at least $k-i+1$ sets of $\mathcal{F}_{i-1}$. Add t_i to T and let $\mathcal{F}_i = \mathcal{F}_{i-1} - \{S \in \mathcal{F}_{i-1} : t_i \in S\}$. After $n-m$ iterations, the

$$|\mathcal{F}_{n-m}| \leq ()$$

□

Lemma 90. *(Limit Lemma for Threshold Functions.) If $B \subset \{0,1\}^n$ is a set of vectors where each entry has exactly r ones, then B has a lower k-limit with fewer than r ones if $|B| > k^r$.*

Proof. □

10.2.1 Application: $Maj_n \in \text{SizeDepth}\left(\Omega\left(2^{\sqrt{n}}\right), 3\right)$

Just a couple of notes after the meeting with Ben: so he is mostly interested in the nitty-gritty of the switching lemma. That has given him some intuition about various functions and their hardness. The sensitivity of n-ary boolean function f on input $\mathbf{x}$, denoted $S_{\mathbf{x}}(f)$, is the number of indices i such that toggling x_i changes the function value i.e. $S_{\mathbf{x}}(f) = |\{i : f(\mathbf{x}) \neq f(\mathbf{x}^{\bar{i}})\}|$ where $\mathbf{x}^{\bar{i}} = \mathbf{x}$ except $x_i^{\bar{i}} = 1 - x_i$. The sensitivity of f is then

$$S(f) = \max_{\mathbf{x} \in \{0,1\}} S_{\mathbf{x}}(f).$$

It also helps to consider the sensitivity of a uniformly randomly chosen string. For parity, for example, a random string $\mathbf{x}$ the sensitivity is $S_{\mathbf{x}}(PARITY_n) = n$. On average the sensitivity is also n. For the majority function however, $S_{\mathbf{x}}(MAJORITY_n)$ depends on the Hamming weight of $\mathbf{x}$. Let $MAJORITY_n(\mathbf{x}) = \|\mathbf{x}\| > \frac{n}{2}$. Then on strings with hamming distance exactly $\frac{n}{2}$, the sensitivity is $\frac{n}{2}$ while on all other strings the sensitivity is 0. It turns out that this constitutes $1/\sqrt{n}$ of all strings[29].

Switching argument type proofs work the best for functions like parity which have high sensitivity. Essentially a random restriction is like a function on a subset of the inputs. If you can preserve the complexity of this sub-function then the lower bound arguments work better. However Ben feels that functions such as Clique might have exponential size lower bound.

[29] This needs to be checked