Lower Bounds **(CS294-152, UC Berkeley, Fall'18)**

Linear-Size Lower Bounds for Functions in P

Notes for 10/1/18

Ryan Williams

1 Linear-Size Circuit Lower Bounds by Gate Elimination

A couple of lectures ago, we considered functions with maximum circuit complexity, i.e. $C(f) \geq \Omega(2^n/n)$. Last lecture, we looked at known results about functions f with $C(f) >$ poly(n), for fixed polynomials and for arbitrary polynomials.

In this lecture, we'll discuss what is known about proving circuit size lower bounds of the form $C(f) \geq c \cdot n$, where $c \geq 1$ and the f are in P (or easier). The known lower bounds are proved by the so-called *gate elimination method*.

Recall that we'd like to know if $\mathsf{NP} \not\subset \mathsf{P}/\text{poly}$. A baby step towards this would be to show that

$$\mathsf{NP} \not\subset \bigcup_{c \geq 1} \mathsf{SIZE}[cn].$$

That is, show that for every $c \geq 1$, there is a function in NP that does not have cn size circuits. (Recall that the gate basis does not matter, as long as the gates have bounded fan-in.)

We don't know how to prove this yet, but in the last lecture we saw that

$$\Sigma_2\mathsf{P} \not\subset \mathsf{SIZE}(n^c) \quad \text{and} \quad \mathsf{MA}/1 \not\subset \mathsf{SIZE}[n^c],$$

for any desired $c \geq 1$. $\Sigma_2\mathsf{P}$ and MA are a little bigger than NP, we don't know by how much. MA is believed to be equal to NP, and it is not hard to show that $\mathsf{NP}/1 \not\subset \mathsf{SIZE}[n^c]$ implies $\mathsf{NP} \not\subset \mathsf{SIZE}[n^c]$. So it is seems reasonable to believe that $\mathsf{NP} \not\subset \mathsf{SIZE}[n^c]$ could be proved.

We have worked hard to make the proofs here as simple as possible. We'll build up our treatment of gate elimination slowly, getting more and more complex as we go on.

1.1 Non-Degenerate Functions

We begin with a very simple lower bound. Recall $B_n := \{f : \{0,1\}^n \to \{0,1\}\}$. For an $f \in B_n$ on input variables $x_1, \ldots, x_n$, and a constant $c \in \{0,1\}$, the *restriction of f to $x_i = c$* is:

$$f|_{x_i=c}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) := f(x_1, \ldots, x_{i-1}, c, x_{i+1}, \ldots, x_n).$$

That is, $f|_{x_i=c}$ is a function on $n-1$ variables which behaves as f when $x_i = c$. This is a very convenient notation for assigning values to variables.

Definition 1. *A function $f \in B_n$ is* non-degenerate *if for all $i \in [n]$, $f|_{x_i=0} \neq f|_{x_i=1}$.*

In other words, f is non-degenerate if the value of f depends on all of its input variables: for every index i, there is *some* assignment $(a_1, \ldots, a_n)$ such that the value of f on $(a_1, \ldots, a_n)$ changes when you flip the value of a_i.

Proposition 1. *For all non-degenerate $f \in B_n$, $C_{B_2}(f) \geq n - 1$.*

Proof. Let f be non-degenerate, and let C be a min-size circuit for f. Since each non-input gate has fan-in 2,

$$\text{(number of edges of } C) = 2C_{B_2}(f).$$

Observe:

1. For all $i = 1, \ldots, n$, $outdeg_C(x_i) \geq 1$. That is, all input gates in C have outdegree at least one. (Otherwise, f is degenerate: that variable x_i plays *no* role in the value of f.)
2. For every non-input, non-output gate g, $outdeg_C(g) \geq 1$ (otherwise, g could be removed, as its output is unused).

Therefore

$$\text{(number of edges of } C) = \sum_{g \in C} outdeg(g) \geq n + C_{B_2}(f) - 1,$$

as this is the total number of gates in the circuit, minus one for the output gate. Hence

$$2C_{B_2}(f) \geq n + C_{B_2}(f) - 1 \Longrightarrow C_{B_2}(f) \geq n - 1.$$

□

This simple proposition is already enough to get tight lower bounds in some interesting cases. Recall the PARITY function on n bits is

$$\mathrm{P}_n(x_1, \ldots, x_n) = \sum_i x_i \pmod 2.$$

A key property of the parity function is is its "extreme" non-degeneracy:

Fact 1.1 (PARITY is Flippy)**.** *For all $i \in [n]$ and $a \in \{0,1\}^{n-1}$, PARITY$|_{x_i=0}(a) \neq$ PARITY$|_{x_i=1}(a)$.*

That is, PARITY is highly non-degenerate: its value flips on *every* variable assignment, given a single bit flip of an input. (Note that the only functions satisfying this "flippy" property are PARITY and its negation.)

Theorem 1.1 (Folklore). $C_{B_2}(P_n) = n - 1$.

Proof. For the upper bound, note that by associativity of addition, we can easily compute PARITY on n bits by making a tree of $n - 1$ copies of PARITY on 2 bits. The lower bound follows from Fact 1.1 and Proposition 1. □

What about computing PARITY in the basis U_2? How efficiently can PARITY be computed when we *do not* have PARITY on 2 bits? Well, PARITY on two inputs can be done with three U_2 gates:

$$(x + y) = (x \wedge \neg y) \vee (\neg x \wedge y).$$

If we replace each 2-input parity gate in our $(n-1)$-gate tree-like circuit for P_n with 3 U_2 gates, the circuit size becomes $3(n - 1)$.

Can we do better? This is basically the most naive construction one could do. It is in fact optimal.

Theorem 1.2 (Schnorr '74). $C_{U_2}(P_n) = 3(n - 1)$.

For notational brevity, in the following let $C(f)$ denote $C_{U_2}(f)$. The approach used to prove this result is called *gate elimination*. It is an inductive method; in the case of P_n, we argue as follows:

1. Prove: $C(\mathsf{P}_2) \geq 3$.
2. Inductively assume: $C(\mathsf{P}_{n-1}) \geq 3(n - 2)$.
3. Show: For every minimum size circuit D for P_n, we can "eliminate" at least three (non-input) gates of D, and get a circuit for P_{n-1}.
4. Hence $size(C) \geq 3 + C(\mathsf{P}_{n-1}) \geq 3(n - 2) = 3(n - 1)$.

How will we remove gates from D? We will assign one input variable to a constant value, and argue that the restricted circuit on $n-1$ inputs can be simplified to have three fewer internal gates. Formally, let D be a U_2-circuit, with the constants 0 and 1 as sources along with the n inputs. We run the following procedure on D:

Procedure Simplify(D)
While applicable, apply the following rules:
- For all $h = NOT(b)$ where $b \in \{0, 1\}$, replace all wires out of h with wires out of $\neg b$.
- For all $h = AND(1, g)$ in D, replace all wires out of h with wires out of g.
- For all $h = AND(0, g)$ in D, replace all wires out of h with wires out of 0.
- For all $h = OR(1, g)$ in D, replace all wires out of h with wires out of 1.
- For all $h = OR(0, g)$ in D, replace all wires out of h with wires out of g.

Output the circuit D' obtained.

Every simplification rule eliminates a gate, replacing it with either a wire from other gates, or with a constant (which can be further simplified).

Incidentally, simplification rules like the above are commonly used in SAT solvers that run on circuits. We have written Simplify in terms of AND, OR, and NOT gates separately; it easily extends to every function in U_2 (which can have NOTs at either of the two inputs, or at the outputs).

Proposition 2. *For all D, the circuit $E = Simplify(D)$ computes the same function as D. Furthermore, E is either a constant or every inner gate in E has no constant inputs.*

The Simplify procedure also makes sense for B_2, but the significant difference is that when we set only one of the inputs of an XOR to 0 or 1, the XOR *never* outputs a constant. In fact, this is **the** primary difference between working with circuits over U_2 and B_2. (We will return to this point later.) But for all U_2 functions, we have this important "restriction" property:

Proposition 3 (Restriction Proposition for U_2)**.** *For all $f(x, y) \in U_2$, there are $b_1, b_2, c_1, c_2 \in \{0, 1\}$ such that $f(b_1, y) \equiv c_1$ and $f(x, b_2) \equiv c_2$.*

That is, every $f \in U_2$ can be forced to be a constant function, by setting only one of the inputs.

The proof is very easy; first note the proposition is trivial for constant functions, and functions depending on only one variable. It is also obvious for f equal to AND. The remaining functions can all be written as AND gates with negations on some of their inputs and possibly their outputs.

Proof of Theorem 1.2. We already proved $C(\text{PARITY}_n) \leq 3(n-1)$.

We first prove $C(\text{PARITY}_2) \geq 3$. Since $\text{PARITY}_2 \notin U_2$, we have $C(\text{PARITY}_2) \geq 2$. Suppose $C(\text{PARITY}_2) = 2$. Consider a 2-gate circuit with output gate g. Since g has two inputs and there's only one other gate h, g must take some variable x as input. (If g had both inputs coming from h, the circuit could be simplified to 1 gate.) By the Restriction Proposition, there is a b such that $x_i := b$ forces g to output a constant, independent of the other input variable. This contradicts the fact that PARITY is flippy. We have proved $C(PARITY_2) = 3$.

Now let $n > 2$. Let D be a min-size circuit for P_n. Observe that for all variables x_i, $D' = D|_{x_i=0}$ computes P_{n-1} and $D'' = D|_{x_i=1}$ computes $\neg\text{P}_{n-1}$. (Note for all f, $C(f) = C(\neg f)$, since NOT gates are free over U_2.) Let us inductively assume $C(\text{PARITY}_{n-1}) \geq 3(n-2)$. Now we want to prove:

> *For all min-size C for PARITY_n, there is an input x_i and $v \in \{0, 1\}$ such that $D' = Simplify(D|_{x_i=v})$ satisfies $size(D') \leq size(D) - 3$.*

If we can prove this, we are done. Why? Every D' obtained from Simplify can be used to compute PARITY and $\neg$PARITY on $n-1$ bits by negating the output). By induction, $size(D') \geq 3(n-2)$. Hence $size(D) \geq 3 + 3(n-2) \geq 3(n-1)$.

We now show how to find such an x_i and v. The *idea* is to take an input x_i with wires to (at least) two distinct gates g and g' (later in topological order). Setting x_i to a constant that forces g' to be a constant eliminates g, g', and at least one other later gate that g' points to.

Consider the "first" inner gate g of C: the gate numbered $n+1$ in topological order. This g takes two input variables x_i and x_j (because C is min-size, g cannot take only one input).

We claim that outdeg$(x_i) \geq 2$. If $outdeg(x_i) = 1$, then by the Restriction Proposition, there is an assignment to the other variable $x_j := v$ that makes g output a constant. But then $D|_{x_j=v}$ does not depend on x_i at all, contradicting the fact that PARITY is flippy. Therefore, x_i must have an edge to some other gate.

Consider the "last" gate g' that x_i connects to: the closest to the output gate in topological order. (It must be different from the "first" gate g, otherwise C is not min-size.) By the Restriction Proposition, we can assign $x_i := v$ that makes g' a constant function. Therefore g' cannot be the output, because PARITY is flippy. Thus g' has an outgoing wire to another later gate g'' (which also cannot be g). Therefore, $D' = \text{Simplify}(D|_{x_i=v})$ has at least three gates are eliminated: g, g', and g''. □

Review. We have a set of functions $\{f_n\}$. We want to prove for all n, $C(f_n) \geq cn - d$ for some constants c, d. We do this as follows:

1. Prove $C(f_2) \geq 2c - d$. (By case analysis, or computer?)
2. Prove for all n, $C(f_{n+1}) \geq c + C(f_n)$.
   Given a min-size circuit D for f_{n+1}, assign an input $x_i = b$ so that Simplify$(D|_{x_i=b})$ produces a circuit for f_n with (at least) c fewer gates than D.

1.2 Improved Lower Bounds for U2

The best known circuit size lower bound over U2 is the following.

Theorem 1.3 (Iwama-Lachish-Morizumi-Raz'05)**.** *There is an $f \in \mathsf{P}$ such that for all n, $C_{U2}(f_n) \geq 5n - o(n)$.*

Let's give a brief overview of what they do. The authors identify a particular Boolean function property which implies a size lower bound.

Definition 2. *A function $f \in B_n$ is ℓ-mixed if for* **all** *subsets S of variables of size at most ℓ, all 2^ℓ assignments to those variables produces 2^ℓ distinct functions on $n - \ell$ variables.*

For example, a non-degenerate function is 1-mixed: for any variable, if you set it true you get a function f_0, and if you set it false you get another function f_1. The intuition is that the more "mixed" a function is, the more "complex" it gets. Their key lemma roughly shows that for every circuit computing a $(n - o(n))$-mixed function, there is a set of S of either 1 or 2 variables such that, when restricting f to particular values on those variables, we can eliminate $5|S|$ gates. After restricting v variables of an ℓ-mixed function to values, the result is still $(\ell - v)$-mixed.

The proof is a nasty case analysis (and you have to control how the $o(n)$ parameter may change over time). What I want to emphasize here is the consequence:

Corollary 1.1. *Every $(n - o(n))$-mixed $f \in B_n$ has $C_{U_2}(f) \geq 5n - o(n)$.*

It would seem that "being $(n - o(n))$-mixed" is a strong property for a function. Quite interestingly, there are efficiently computable highly-mixed functions:

Theorem 1.4 (Savicky and Zak '96)**.** *There is an $f : \{0,1\}^\star \to \{0,1\}$ such that for almost all n, f_n is $(n - 3n^{1/2})$-mixed, and $f \in \mathsf{P}$.*

It is also known that there are highly mixed functions computable in polynomial time, with circuits of size $5n - o(n)$. That is, the lower bound of $5n$ is actually optimal for highly mixed functions!

Theorem 1.5 (Amano-Tarui)**.** *For all n, there is an $f_n \in B_n$ such that f_n is $(n - n^{2/3})$-mixed and $C_{U_2}(f_n) \leq 5n + o(n)$.*

This shows that any further progress on circuit size lower bounds for U2 will have to use a stronger property than "being highly mixed"! (Note that the property of "being flippy" only got us a $3n$ lower bound.)

2 Size Lower Bounds for B2

For the basis of all 2-bit functions, the situation is even more embarrassing! The best lower bounds for functions in P are $3n - o(n)$ and $3.01n - o(n)$.

The main problem with B_2-circuits is that the Restriction Proposition does not hold for XOR gates. When we feed a constant into one input of an XOR gate, the XOR will *never* output a constant, it will always get replaced with wires from other gates.

But no one said we had to simplify circuits by setting variables to constants! Suppose we *generalize* the kinds of variable restrictions that we allow. Rather than merely setting $x := 1$ or

$x := 0$ for a variable x, we could set x to be an affine form, *a degree-one polynomial over $\mathbb{F}_2$ in some other variables*, e.g., $x := y + z + 1$. This is equivalent to simplifying C to a circuit C' that agrees with C on the set $\{(x, y, z, \ldots) \mid x = y + z + 1 \bmod 2\}$. (Note that $x = 1$ and $x = 0$ are also equations, so this is a generalization.)

To give a concrete example, suppose we had the gates $XOR(XOR(x, y), z)$ inside a circuit, and we set $x = y + z + 1 \bmod 2$. We could then simplify these gates to

$$XOR(XOR(XOR(y, z, 1), y), z) \equiv 1.$$

We would force the top XOR to output a constant! This is the basic idea behind a new approach to gate elimination.

2.1 Demenkov and Kulikov's $3n - o(n)$ size lower bound over B2

Perhaps the most important decision in a lower bound proof of this kind is: *what* function are we going to try lower bounding? In the case of PARITY, we heavily exploited that fact that we can assign up to $n - 1$ variables to be constants, and yet PARITY will still not be constant. Once we generalize our kind of variable assignments, and allow variable assignments to be mod-2 sums of constants and other variables, intuitively we will need a function that will not be constant under many such assignments. These are called *affine dispersers*,

Demenkov and Kulikov gave a simple proof of a $3n - o(n)$ size lower bound for affine dispersers. We recall some basic linear algebra needed to define these functions.

A Little Linear Algebra Mod 2. Think of $\{0, 1\}^n$ as $\mathbb{F}_2^n$, and think of $\mathbb{F}_2^n$ as a vector space over $\mathbb{F}_2$, where $u + v$ is component-wise addition of vectors. (In the following, $+$ denotes sum modulo 2.)

Definition 3. *A subset $S \subseteq \{0, 1\}^n$ is an* affine subspace *of dimension d if $S = v_0 + span(v_1, ..., v_d)$, for some $v_0 \in \mathbb{F}_2^n$ and some linearly independent $v_1, \ldots, v_d \in \mathbb{F}_2^n$.*

Note that linear independence over $\mathbb{F}_2$ is equivalent to: for all $j = 1, \ldots, d$, and all $T \subseteq [d] - j$, $v_j \neq \sum_{i \in T} v_i$. So we have

$$S = \left\{ v \in \mathbb{F}_2^n \mid \exists T \subseteq [d] \text{ s.t. } v = v_0 + \sum_{j \in T} v_j \right\},$$

where $|S| = 2^d$.

Affine subspaces are the sets of solutions to feasible systems of linear equations over $\mathbb{F}_2$: Given a system $Ax = b$, the set of solutions (provided there is one) has the form $U + x^\star$ where

$$U = \ker(A) = \{x \mid Ax = 0\}$$

and $x^\star$ is an arbitrary solution of $Ax = b$. The dimension of the affine subspace $U + x^\star$ is $n - \text{rank}(A)$, and hence the number of solutions to $Ax = b$ will be $2^{n-\text{rank}(A)}$.

(Example: when $Ax = b$ has exactly one solution, $\text{rank}(A) = n$, the dimension of the solution space is 0, and $U = \{\vec{0}\}$.)

Definition 4. *$f : \{0,1\}^n \to \{0,1\}$ is an* affine disperser for dimension d *if for every affine subspace S of dimension at least d, there are $v, v' \in S$ such that $f(v) \neq f(v')$.*

That is, f is *non-constant* on all affine subspaces with at least 2^d vectors. For *small* d, it is instructive to think of this property as generalizing PARITY, which is non-constant on all variable assignments on up to $n - 1$ variables (Any partial assignment on $n - k$ variables can be seen as a system of $n - k$ linear equations, which has $O(2^k)$ solutions.) In the following, let f_d (with n inputs) denote an arbitrary affine disperser for dimension d.

Our lower bound will actually be against a more powerful circuit model than just B_2-circuits.

Definition 5. *An XOR-layered B_2-circuit with n inputs is a B_2 circuit on t inputs composed with an arbitrary affine transformation $A : \mathbb{F}_2^n \to \mathbb{F}_2^t$ for some t.*

Such a circuit can be written as a B_2-circuit of size s, on top of t XOR gates, each of which may have *arbitrary* fan-in, and point directly at the n inputs along with the constants 0 and 1.

Theorem 2.1 (Main)**.** *For all XOR-layered B_2-circuits C with t gates in the XOR layer and s gates elsewhere, and for all $k \geq d$, if there is an affine subspace $S \subseteq \mathbb{F}_2^n$ with $dim(S) \geq k$ such that $C(x) = f_d(x)$ on all $x \in S$, then $s + t \geq 4(k - d)$.*

As C agrees with f_d on a "larger" affine subspace, the lower bound gets "larger". Before we prove the Main Theorem, we note some of its corollaries.

Corollary 2.1 (Lower Bound)**.** *Let C be an XOR-layered B_2 circuit with t gates in the XOR layer and s other gates. If C computes f_d then $s + t \geq 4(n - d)$.*

Proof. Set $k = n$ in the Main Theorem. If C computes f_d, then we can set the affine subspace $S = \mathbb{F}_2^n$. Therefore $s + t \geq 4(n - d)$. □

Corollary 2.2. *$C_{B_2}(f_d) \geq 3n - 4d$.*

Proof. Every B_2-circuit C of size s' can be expressed as an XOR-layered circuit C' with n XOR gates. (For all $i = 1, \ldots, n$, we can simply add the trivial XOR gate $x_i + 0$ to the circuit.) By the Lower Bound Theorem,

$$4n - 4d \leq s' + n,$$

implying that $s' \geq 3n - 4d$. □

In the proof of the Main Theorem, we use an "XOR version" of the Restriction Proposition for U_2:

Proposition 4 (XOR Restriction Proposition)**.** *Let g be a gate in the XOR layer of C, i.e.*

$$g = b + \sum_{i \in T} x_i, \;\; b \in \{0,1\}, \;\; T \subseteq [n].$$

For all $j \in T$ and $c \in \{0,1\}$, consider the circuit C' which replaces x_j with the gate $c + \sum_{i \in T - j} x_j$. Then on all inputs to C', the gate g is equivalent to a constant function, and C' agrees with C on the set of inputs $\{x \mid \sum_{i \in T} x_j = c\}$.

Proof. After the replacement of x_j, we can write

$$g = \left(b + \sum_{i \in T - \{j\}} x_i \right) + \left(c + \sum_{i \in T - j} x_j \right) = b + c,$$

because the sum is modulo 2. So $g \equiv b + c$. Clearly C' agrees with C on the above set. □

This proposition means that we can eliminate gates from the XOR layer of g by "assigning affine forms to variables". This replacement does two things.

(a) Restricts the space of inputs that our circuit works over: if our circuit worked over a set S then it now works over a set $\{x \in S \mid \sum_{i \in T} x_j = c\}$.
(b) Simplifies the circuit we are working with.

(Recall we had an analogous set of two things in the lower bounds for U_2.) We are ready to prove the Main Theorem:

Proof of Main Theorem. Induction on k. When $k = d$ the theorem is trivial.

Let $k \geq d + 1$. Let C be min-size and let S have dimension at least k, where $C = f_d$ on S.

> **Goal:** Find an XOR restriction $x_j := \sum_{i \in T} x_i + c$ so that in the remaining subcircuit, $s + t$ reduces by at least 4.

We claim that achieving the **Goal** will prove the theorem. Assuming the **Goal**, after setting the XOR restriction, the circuit C simplifies to a circuit C' with

$$s' + t' \leq s + t - 4$$

gates, such that $C'(x) = C(x)$ for all $x \in S' = \left\{x \in S \mid x_j = \sum_{i \in T} x_i + c\right\}$. Observe that S' has dimension at least $k-1$, since S had dimension at least k.

So there's some S' of dimension at least $k-1$ such that C' agrees with f on S'. By induction,

$$s' + t' \geq 4((k-1) - d).$$

Therefore

$$s + t \geq s' + t' + 4 \geq 4((k-1) - d) + 4 = 4(k - d).$$

Now we set our sights on achieving the **Goal.** We start with some simple claims.

Generalized R.P.: For any gate L_i from the XOR layer pointing to a U_2 gate g, there is an XOR restriction that makes both L_i and g output constants. The resulting circuit agrees with C on $\{x \mid L_i(x) = c\}$.

Proof: By the XOR Restriction Proposition, we can set L_i to output any desired constant c. By the Restriction Proposition, we can choose c that makes g output a constant. **QED**

Claim: There's at least one U_2-gate g in C depending on both of its inputs, which is not the output gate.

Proof: Assume not. Then C has only XOR and ¬XOR gates and one-variable projections. Thus we can write

$$C(x) = \sum_{i \in T} x_i + c.$$

But for some S of dimension k, $C(x) = f(x)$ for all $x \in S$. Thus f is constant on $S' = \{x \in S \mid \sum_{i \in T} x_i = c\}$, where $dim(S') \geq k - 1 \geq d$. This contradicts the assumption that C agrees with an affine disperser for dimension d on the set S.

If g is the output of C, then by **Generalized R.P.** we can XOR-restrict L_1 and g to constants. But then C is constant on $S' = \{x \in S \mid L_1 = c\}$ of dimension $k - 1 \geq d$, contradicting that C agrees with an affine disperser for dimension d on the set S. **QED**

Applying the **Claim**, take the topologically first U_2-gate g depending on both inputs, and let g' be a gate that g points to (g is not the output). The two inputs L_1 and L_2 going into g must be from the XOR layer. We consider a few cases, similar (but not identical) to the proof for PARITY over U2.

(Case 1) outdeg$(L_2) = 1$. By **Generalized R.P.**, we can XOR-restrict a variable to make L_1 and g constant. Then we can eliminate g and g', and L_1 and L_2 (because L_2 had outdegree 1).
So in this case, $s + t$ decreases by 4 by setting L_1 to a constant.

(Case 2) outdeg$(L_2) \geq 2$. Let $h \neq g$ be another gate that L_2 has a wire to. Suppose we XOR-restrict a variable to make L_2 and g both constant.

(2a) Suppose $h = g'$. The two inputs to h are L_2 and g, both of which are now constants, so h is constant. This h cannot be the output (similar to our proof of the claim: the whole circuit would be constant after one XOR restriction). Thus h points to a later gate g''. Thus we can set L_2 to a constant which eliminates L_2, g, h, g'', so $s + t$ decreases by 4.

(2b) Suppose $h \neq g'$. In this case, we eliminate L_2, g, h, and g', and $s + t$ decreases by 4.

Thus in all cases, $s + t$ decreases by 4 after one linear restriction. □

Note the above lower bound is basically tight (as a function of $(n - d)$), up to additive constants:

Claim 1 (Exercise?)**.** $IP_n(x_1, \ldots, x_{n/2}, y_1, \ldots, y_{n/2}) = \sum_i x_i y_i \bmod 2$ *is an affine disperser for dimension* $n/2 + 1$.

We have $C_{B_2}(IP_n) \leq (n/2 - 1) + n/2 = n - 1$: we can compute IP_n with $n/2 - 1$ XOR gates and $n/2$ AND gates.

So there is an XOR-layered circuit for IP_n with $s + t \leq n - 1 + n = 2n - 1$.

But by the above claim for IP_n, we have $4(n - d) \geq 4(n/2 - 1) = 2n - 4$.

Therefore $s + t + 3 \leq 4(n - d)$, while the Main Theorem showed $s + t \geq 4(n - d)$.

However, we could potentially do better when d is $o(n)$.

2.2 Constructions of Affine Dispersers

How to get affine dispersers of low dimension?

Claim 2 (Exercise?)**.** *Whp, a random* $f : \{0,1\}^n \to \{0,1\}$ *is an affine disperser for dimension* $O(\log n)$.

Thus for such f, we have $C_{B_2}(f) \geq 3n - O(\log n)$, but of course we have stronger lower bounds for random functions!

There is an explicit construction of affine dispersers for low dimension:

Theorem 2.2 (Ben-Sasson and Kopparty'09, Shaltiel'11, Li'15)**.** *There is a family* $\{f_n\}$ *of affine dispersers for dimension* poly$(\log n)$ *that can be computed in polynomial time.*

It is rather technical to describe these, so we'll skip them.

Open: Suppose f_n is an affine disperser of dimension $o(n)$.

We know $C_{B_2}(f_n) \leq$ poly(n) and $C_{B_2}(f_n) \geq 3n - o(n)$. Which is closer to the truth?

Since 2011, I have been saying in my course:

The proof above is very simple. I am sure that affine dispersers require larger circuit size.

Recently, the "$3n$-size barrier" for B_2-circuits was **finally** broken:

Theorem 2.3 (Find-Golovnev-Hirsh-Kulikov'16)**.** *For every affine disperser f_n of dimension $o(n)$, we have $C_{B_2}(f_n) \geq 3.01n - o(n)$.*

This theorem uses a *lot* of extra machinery on top of gate elimination. They use XOR restrictions, but they also allow the "XOR layers" in circuits to actually be *cyclic* graphs. They sometimes make restrictions which are read-one quadratic polynomials in other variables. And they choose a "size measure" much crazier than $s + t$ to measure their progress in the gate elimination steps.

It seems likely to me that there's a somewhat short proof of (at least) a $3.5n - o(n)$ lower bound for affine dispersers for poly$(\log n)$ dimension.

2.3 A Barrier

It seems clear that we need a new technique beyond gate elimination to prove super-linear lower bounds. Here we note a strong "barrier" result that was recently proved.

Say that an *arbitrary* variable restriction has the form

$$x_i := r(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n),$$

where r can be *any* Boolean function on $(n-1)$ variables. Note that in this lecture, we have considered r which are constants and r which are "XOR restrictions": degree-1 $\mathbb{F}_2$ polynomials over the variables.

Theorem 2.4 (Golovnev-Hirsch-Knop-Kulikov'18)**.** *For every function f on n inputs, there is a function f' on $3n$ inputs such that*

$$C_{B_2}(f) \leq C_{B_2}(f') \leq C_{B_2}(f) + 13.5n,$$

yet every arbitrary *restriction to every input variable of f' eliminates at most 5 gates from a minimum circuit for f'.*

Corollary 2.3. *There exist circuits such that it is impossible to remove more than 5 gates by an* arbitrary *variable restriction.*

The authors also generalize their results to restrictions on $O(1)$ variables.

This shows that, if we are to push gate elimination much further, we cannot expect to get even "5 gates of progress" in every variable restriction, or even in $O(1)$ variable restrictions: we would have to do something that only eliminates 5 gates *on average* over a long period of variable substitutions.