LECTURE 21

Last three lectures:
Discussion of switching lemmas of various kinds

Today

Discuss very recent paper
Lower bounds for bounded-depth Frege (BDF) refutations of Tseitin formulas

Complicated paper. Still no public full-length version. In the version we have still multiple errors (hopefully all fixable) So won't go into details but only give overview

Plan for the lecture

- Background
- Recap of definitions
- Top-down description of proof (but we won't get very far – only get a sense of the overall argument)

Frege proof system

- General formulas as proof lines
- Sound and complete set of derivation rules
- Details not too important — different versions polynomially equivalent [CR '79]

Lower bounds? Almost completely beyond reach (except easy lower bounds based on that proof system has to look at whole formula).

What to do when we cannot prove lower bounds in computational complexity? Weaken the model!

Consider formulas over $\land, \lor, \neg$
But bound the $\boxed{\text{depth}}$ (# alternations $\land / \lor$)
Prove super-polynomial lower bounds on proof size.

Short history:

[Ajtai '94] Super-poly BDF size lower bounds for PHP — major result, but hard-to-read paper and not so explicit bounds

[Bellantoni, Pitassi, Urquhart '92]
Super-poly BDF lower bounds for PHP for depth $O(\log^* n)$ ← grows extremely slowly

[PBI '93, KPW '95]
Superpoly BDF lower bound for PHP for depth $O(\log \log n)$

[Ben-Sasson '02]
Tseitin lower bound via reduction also for depth $O(\log \log n)$

[Buss '87]
Polysize refutations of PHP in depth $O(\log n)$

Also known that this can be achieved for Tseitin formulas over any graph

Big gap between $O(\log \log n)$ and $O(\log n)$...
How to get lower bounds for larger depth?

[Pitassi, Rossman, Servedio, & Tan '16]
Superpolynomial lower bounds for BDF refutations of Tseitin up to $\boxed{\text{depth } O(\sqrt{\log n})}$

Our version of BDF: $\boxed{\text{Schoenfield's system } \mathcal{F}}$
Only $\vee, \neg$. ($A \wedge B$ syntactic sugar for $\neg(\neg A \vee \neg B)$)

Excluded middle $\dfrac{}{p \vee \neg p}$

Expansion rule $\dfrac{p}{p \vee q}$

Contraction rule $\dfrac{p \vee p}{p}$

Associativity $\dfrac{p \vee (q \vee r)}{(p \vee q) \vee r}$

Cut rule $\dfrac{p \vee q \quad \neg p \vee r}{q \vee r}$

Derivation steps: Plug previously derived formulas (or axiom clauses) into p, q, r and derive conclusion

Refutation: Derive empty disjunction $\bot$

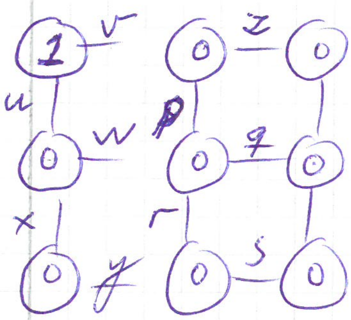
TSEITIN FORMULA $Ts(G, \chi)$

Graph $G = (V, E)$ (assumed connected) charge function $\chi : V \to \{0,1\}$

Variables: $e \in E$

Every vertex v = constraint:

"Sum of edges incident to $v \equiv \chi(v) \pmod 2$"



TOP LEFT

$$(u \lor v) \land (\bar{u} \lor \bar{v})$$

$$\land (x \lor \bar{y}) \land (\bar{x} \lor y)$$

Bottom left

$$\land (\bar{u} \lor w \lor x) \land (u \lor \bar{w} \lor x) \land (u \lor w \lor \bar{x}) \land (\bar{u} \lor \bar{w} \lor \bar{x})$$

middle left

Say that χ has odd charge if

$$\sum_{v \in V} \chi(v) \equiv 1 \pmod 2$$

FACT If G is connected, then $Ts(G, \chi)$ is unsatisfiable iff χ has odd charge.

We saw in one of the first lectures that $Ts(G, \chi)$ is hard for resolution if G is a very well-connected graph (an expander)

[PRST '16] prove a lower bound for Tseitin on expanders for BDF (but in terms of size the lower bound is not exponential, only superpolynomial)

The high-level structure of the proof is the same as for PHP. In one sentence: Assume that there is a ~~short~~ small refutation, and prove that this is just too good to be true — it leads to contradiction.

Or alternatively: If a BDF derivation from $Ts(G, \chi)$ is too small, then it doesn't have enough time to derive contradiction.

Components in proof

(1) Define kind of "GOOD" decision trees that "represent" formulas in derivations in an approximate way {and claim that all clauses in $Ts(G, \chi)$ are true} but say that contradiction $\bot$ is false

(2) Show for any derivation rule that if assumptions have "GOOD" decision trees that always yield true, then conclusion also has GOOD decision tree that always yields true

(3) If the refutation $\pi: Ts(G,\chi) \vdash \bot$ is small enough, then we can hit it with a random restriction ρ s.t.

a) $Ts(G,\chi)\restriction_\rho = Ts(G',\chi')$

b) $\pi\restriction_\rho$ refutes this formula $Ts(G',\chi')$

c) All formulas in $\pi\restriction_\rho$ have ~~good~~ good decision trees as in (1)

But if that is so, $\pi\restriction_\rho$ cannot derive contradiction because of (2).
So π cannot have derived contradiction for $Ts(G,\chi)$ either)

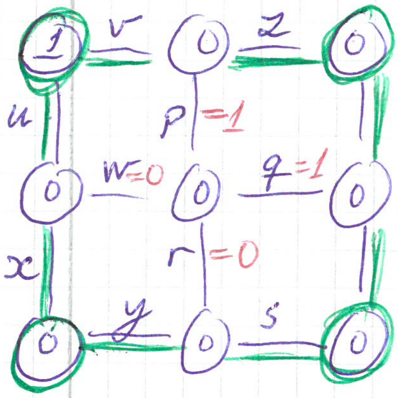
PROJECTIONS

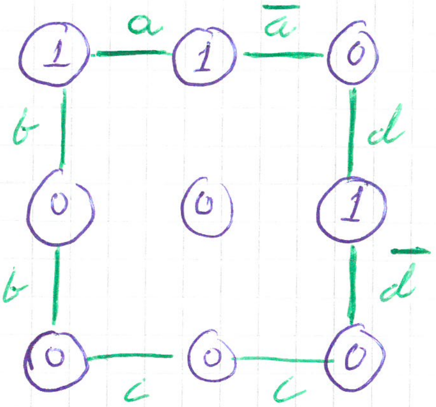
Actually, we are going to use projections — let's make sure to explain properly once how this works

A projection maps a variable to 0, 1, or some literal. Different variables can be projected to same literal, or to literals with opposite signs over the same variable

$$\rho(x) = \begin{cases} 0 & \text{or} \\ 1 & \text{or} \\ y & \text{or} \\ \overline{y} & \end{cases} \Big\} \text{ for some variable } y$$

Consider our Tseitin formula example again:





$(a \vee b)$
$\wedge (\bar{a} \vee \bar{b})$
$\wedge (b \vee \bar{c})$
$\wedge (\bar{b} \vee c)$
$\wedge (c \vee d)$
$\wedge (\bar{c} \vee \bar{d})$
$\wedge (a \vee d)$
$\wedge \bar{a} \vee \bar{d}$

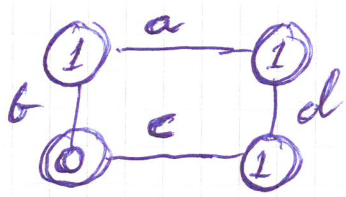
Pick randomly some vertices and paths connecting them (in green)
Set all other variables randomly except that parity constraints need to be respected
i.e., we must have $p+q+r+w \equiv 0 \pmod 2$
Say we set $p=q=1$, $r=w=0$ [All choices except one completely random]
Then define projections to satisfy constraints along paths
So the projection ρ will map

$u \longmapsto b$ $x \longmapsto b$ $p \longmapsto 1$ $r \longmapsto 0$
$v \longmapsto a$ $y \longmapsto c$ $q \longmapsto 1$ $s \longmapsto c$
$w \longmapsto 0$ $z \longmapsto \bar{a}$ et cetera

$\mathrm{Ts}(G, \chi)\restriction_\rho$ is in the top ~~left~~ right column

Note that constraint for center vertex is fixed to true
For, e.g., leftmost middle vertex we get constraints $(b \vee \bar{b}) \wedge (\bar{b} \vee b)$
which are trivial and can be ignored

$\mathrm{Ts}(G, \chi)\restriction_\rho$ will be the Tseitin formula for the graph



In our random projections, important not to violate constraints.

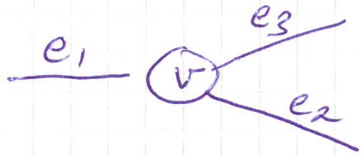
In our example, if we violated the center vertex constraint in a projection, then we would get empty clause $\perp$ already in restricted formula $\Rightarrow$ impossible to appeal to lower bound for restricted formula.

{ Since we won't get very technical today anyway, let us switch back to thinking about restrictions

For our "good" decision trees representing formulas it is also important not to violate constraints.

For instance, axioms should have trees with all leaves labelled 1 (1-trees)

So if we have



decision tree can query e_1 & e_2 but then will simply assume e_3 is set as needed.
(Cannot allow query — might give wrong value)

Cf PHP lower bound — we only had matching trees. When $x_{i,j} = 1$, the tree assumes $x_{i,j'} = 0$ and $x_{i',j} = 0$ for $i' \neq i$ and $j' \neq j$ and never allows asking about such variables.

In fact, we want more from our Tseitin decision trees. Any assignment in any branch should make sure not to push a contradictory odd charge into a small component.

As in the resolution lower bound:
- Any assignment to edges should leave a giant component of size $> |V|/2$
- All other components should have even charge

A BRIDGE (or ISTHMUS) of a graph $G = (V, E)$ is an edge $e \in E$ such that $G' = (V, E \setminus \{e\})$ has one more connected components than G.

Bridges are forced choices
- if disconnects giant component, push charge there
- if disconnects two small components, only one choice makes sure both components have even charge

So good decision trees are not allowed to query such edges. But given assignment σ, we silently assume that bridges in $G' = (V, E \setminus Dom(\sigma))$ are also queried and take the right (forced) choices.

A bit more formally...

Edge set $I \subseteq E$ in $G = (V, E)$ is G-INDEPENDENT if $G \setminus I$ $(= G' = (V, E \setminus I))$ is connected

Decision tree T is G-INDEPENDENT if every branch queries G-independent set

A decision tree T is $\boxed{(k, G)\text{-GOOD}}$ if

a) ~~It~~ T is total, i.e., all leaves labelled 0 or 1 (later we will allow trees with "failed branches" ending in $\bot$)

b) T has depth $< k$

c) T is G-independent

For $G = (V, E)$ and $S \subseteq E$, the $\boxed{G\text{-CLOSURE}}$ $cl_G(S)$ is $S \cup B$ where B is the set of bridges in $G \setminus S$.

Let us say that (G, χ) is $\boxed{\text{NICE}}$ or that G is χ-locally consistent if

a) G has giant component of odd charge

b) All small components have even charge

If σ is charge-preserving, i.e., pushes the odd charge into the giant component, and if $cl_G(Dom(\sigma))$ is small enough so that $G \setminus cl_G(Dom(\sigma))$ also has a giant component, then the closure exists, is charge-preserving, and is unique.

PROPOSITION Let (G, χ) be nice and σ be charge-preserving. Then ~~if~~ ~~before~~ $G \setminus cl(Dom(\sigma))$ has a giant component there is a unique restriction $\supseteq \sigma$, denoted $cl_{G,\chi}(\sigma)$, such that $Dom(cl_{G,\chi}(\sigma)) = cl_G(Dom(\sigma))$ and $cl_{G,\chi}(\sigma)$ is charge-preserving. [Not hard to show — unpack definitions]

(Recall that if we hit $Ts(G,\chi)$ with the restriction σ, then we get the formula $Ts(G \setminus Dom(\sigma), \chi\restriction_\sigma)$ where $\chi\restriction_\sigma$ is defined by

$$\chi\restriction_\sigma(v) = \Big(\chi(v) + \sum_{\substack{e \ni v \\ e \in \sigma^{-1}(1)}} \sigma(e)\Big) \pmod 2 \Big).$$

Now we want to build (k,G)-good trees for all formulas appearing in a BDF derivation such that

All axioms $C \in Ts(G,\chi)$ represented by 1-trees (all leaves labelled 1)

In fact, whole formula $Ts(G,\chi)$ also gets 1-tree

$\perp$ gets 0-tree

Any derivation step preserves 1-trees (so although the representation trees are clearly buggy, BDF cannot detect this)

Representation trees for formulas are constructed inductively in terms of representation trees for subformulas

Recall

$b \in \{0,1\}$

$$\boxed{br_b^{\Delta}(T)} = \{\text{branches } \sigma \text{ in } T \text{ leading to } b\text{-leaves}\}$$

$$\boxed{Disj(T)} = \bigvee_{\sigma \in br_1(T)} \ \bigwedge_{a \in \sigma} a$$

DNF formulas with terms for all 1-labelled branches

As for the PHP lower bound, the tricky part is to represent disjunctions

Fix $(G, \mathcal{X})$ and let $T_1, \ldots, T_m$ be G-independent decision trees. Then a (k', G)-good decision tree T is said to $\boxed{(k', G, \mathcal{X})\text{-represent}}$ $\bigvee_{j=1}^{m} T_j$ if for all $b \in \{0,1\}$ it holds that

$$\sigma \in br_b(T) \Rightarrow \left(\bigvee_j Disj(T_j)\right) \restriction cl_{G,\mathcal{X}}(\sigma) = b$$

That is:

A 1-branch $\sigma \in br_1(T)$ sets some term in $\bigvee_j Disj(T_j)$ to true (after taking closure)

A 0-branch of T falsifies $\bigvee_j Disj(T_j)$ (after closure).

[k' is a parameter that will vary depending on where in the proof we are, but we will ignore this]

Now we can describe what we want from our representing decision trees to prove a lower bound (except that we will be fuzzy on exact parameters)

k-evaluation

Given $G = (V,E)$, χ odd-charge

Let Γ set of formulas over $Vars(Ts(G,\chi)) = E$

s.t. (a) Γ closed under subformulas

(b) Γ includes all clauses $C \in Ts(G,\chi)$

Let k' take the right value.

Then a k-evaluation for Γ is a mapping $\mathcal{T}$ which assigns to each formula $A \in \Gamma$ a (k, G)-good decision tree satisfying

1. $\mathcal{T}(b) = b$ for $b \in \{0,1\}$
2. $\mathcal{T}(\neg A) = \mathcal{T}(A)^c$ [flip labels of all leaves]
3. If $A = \bigvee_j A_j$, then $\mathcal{T}(A)$ (k', G, χ)-represents $\bigvee_j \mathcal{T}(A_j)$
4. For $C \in Ts(G,\chi)$ $\mathcal{T}(C)$ is 1-tree
5. For every tree $T_0 = \mathcal{T}(A_0)$ every collection of at most 6 other trees $T_i = \mathcal{T}(A_i)$, $i \in [6]$, $A_i \in \Gamma$, every branch $\sigma_0 \in br(T_0)$ there exists $\sigma^* \supseteq \sigma_0$ such that

(i) $cl_{G,\chi}(\sigma^*) = \sigma^*$

(ii) σ^* is charge-preserving

(iii) for every i, σ^* forces a value $T_i\restriction_{\sigma^*} = b_i$ for some $b_i \in \{0,1\}$

LEMMA If π is a derivation from $Ts(G, \chi)$, if $\hat{\pi}$ is obtained by taking closure of π under subformulas, and $\hat{\pi}$ has k-evaluation, then π does not refute $Ts(G, \chi)$

Proof sketch

Inductive argument.

Axioms are 1-trees by (4).

(5) kind of says that any branch of a tree can be extended to force a collection of other trees to have a decided opinion in $\{0, 1\}$ about this assignment

Then (2) & (3) say that 1-trees derive 1-trees.

But contradiction is represented by 0-tree by (1). So it is never derived.

The proof isn't actually much fun. The definitions have been set up in such a way that this proof should work.

As before, the hard part is to build the k-evaluation...

How do we build trees for the k-evaluation? By induction over the depth and random restrictions.

To prove depth-d BDF lower bound, fix family of 3-regular expander graphs of decreasing sizes
$G = G^{(0)}, G^{(1)}, G^{(2)}, \ldots, G^{(d)}$

Each graph has $\sim (\log n)$-factor fewer vertices

Build random restrictions (projections) $\rho^{(i)}$ that embed $G^{(i-1)}$ in $G^{(i)}$ "in a random way". The idea is as in our simple example, only immensely more complicated...

Formulas of depth 0 — easy

take obvious (exact) decision trees

Formulas of depth 1 — take trees for depth 0

Hit graph $G = G^{(0)}$ with restriction to get $G^{(1)}$ and $Ts(G^{(1)}, \chi^{(1)})$

Use switching lemma to decrease depth

For depth i, use restriction $\rho^{(i)}$ embedding $G^{(i-1)}$ in $G^{(i)}$

XVI

Technical problems (not exhaustive list):

(A) For PHP we hit matching trees with matching restrictions ⟹ new matching trees

Now we can have (k, G)-good tree with branch σ s.t. $Dom(\sigma)$ independent set of edges

Sample random restriction ρ

If ρ and σ not consistent — not problem

But what if ρ & σ consistent but $Dom(\rho \cup \sigma)$ not independent, but disconnects graph? ~~What~~ Then $T\restriction\rho$ not good tree.

And what if $\rho \cup \sigma$ even push odd charge into small component?!

Need to prune decision trees when this happens

(B) In contrast to the switching lemmas Johan did, variables are not assigned independently

Hard to deal with conditioning in switching lemma. Cannot condition on clauses/terms not being true/false. Have to query all variables ⟹ Pick up extra exponential factor in switching lemma bound

(c) And the whole embedding argument for planting smaller expanders into larger expanders...

It would be nice to simplify the PRST proof considerably

Could lead to better depth lower bounds
better size lower bounds

Pave the way for progress on other problems

- Lower bounds for random 3-CNFs
- Maybe even go beyond BDF to BDF with $\oplus$-connective (parity gates)

But this is all in a possible future. For now we are done with bounded-depth Frege!