Last lecture: lower bounds using interpolation technique.

Clique-colouring formula

$\exists$ m-clique	$\bigvee_{i\in[n]} q_{ik}$	$\forall k \in [m]$	// some vertex is k-th in clique
	$\bar{q}_{ik} \vee \bar{q}_{ik'}$	$\forall i \in [n], k \neq k' \in [m]$	// a vertex only appears once
	$(q_{ik} \wedge q_{jk'}) \to p_{ij}$	$\forall i,j \in [n], k,k' \in [m]$	// i,j in clique $\to$ edge
$\exists$ m-1-colouring	$\bigvee_{\ell\in[m-1]} r_{i\ell}$	$\forall i \in [n]$	// each vertex has a colour
	$(r_{i\ell} \wedge r_{j\ell}) \to \bar{p}_{ij}$	$\forall i,j \in [n], \ell \in [m-1]$	// i,j same colour $\to$ no edge

Plan: → assume resolution proof of clique-colouring.
→ build circuit that outputs 1 if input graph has m-clique and 0 if m-1-colourable, of same size as proof.
→ all such circuits are exponentially large.

We saw how to build a circuit from a proof.

Assume the input (p variables) has a clique. Then it is not m-1-colourable. So, after setting p variables, it is possible to obtain a contradiction using only r clauses. If input m-1-colourable, then contradiction from q clauses.

obs. π resolution proof of F. Then $\pi|_\rho$ resolution proof of $F|_\rho$.

Let us look at proof and label clauses as coming from the r-side or q-side. Then answer is the label of $\bot$.

How do we get new clauses? By resolution steps. 3 cases:

$$\frac{C \vee p \qquad D \vee \bar{p}}{C \vee D}$$

p is set to 0/1. if $p=0$, $D\vee\bar{p}$ is satisfied. therefore, can drop $D\vee\bar{p}$ from proof and replace $C\vee D$ by $C\vee p$ (actually C, since $p=0$); label $C\vee D$ with same label as $C\vee p$.

if $p=1$, pick same label as $D\vee\bar{p}$.

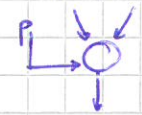
$\frac{C \vee q \quad D \vee \bar{q}}{C \vee D}$ if either is an r-clause, it does not contain q, so can replace $C \vee D$ by that clause, and label by "r".
otherwise, both are q clauses and resolution step is valid; label by "q".

$\frac{C \vee r \quad D \vee \bar{r}}{C \vee D}$ dual case.

At the end, if $\bot$ is labelled "q", it comes from q clauses only (a clause is labelled q if it is the resolvent of two q-clauses or it was copied from another q-clause). Therefore q-part$\restriction_\rho$ is UNSAT. Therefore graph does not have r-clique. Can answer 0.

* maybe graph not $m-1$-colourable either, but in that case any answer is fine.

Can we simulate the labelling process with a circuit?

$\frac{C \vee p \quad D \vee \bar{p}}{C \vee D}$ ⇝ 

$\frac{C \vee q \quad D \vee \bar{q}}{C \vee D}$ 

Yes! Have "q"$=0$, "r"$=1$. Axioms are constants 0/1.

Each resolution step is a sel/OR/AND gate.

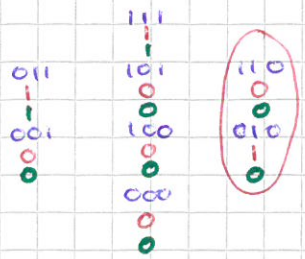
This is what we saw on last lecture.
So are we done yet? Not quite. Need to prove circuit lower bound.
But we are very bad at proving things for circuit, only know special cases. Fortunately, have lower bounds if circuit is monotone, i.e. flipping $0 \mapsto 1$ in input does not decrease output.

Problem! sel gates are not monotone.

triples pxy

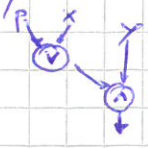
sel

m-sel



sel(010) > sel(110).

Hmm. If sel$(010)=0$ we would be fine, so let us do that.

Replace sel gates by m-sel 

Are you serious? this circuit makes errors! Or does it?

Circuit makes an error if $p=0$, $x=1$, $y=0$. i.e.

$$\frac{\text{"r"}\ C \vee p \qquad D \vee \bar{p}}{C \vee D}$$

"r" should be "r", we say "q".

But "r" clauses only contain negated p. This case never happens.

We proved: let π resolution proof of clique-colouring. There is a monotone circuit for clique-colouring of size $\leq |\pi|$.

Now shift to our main goal: exponential lower bounds for cutting planes.

Overall plan is the same: proof → circuit → large. Except that we need more powerful "real" circuits.

Recall CP rules: work with linear inequalities $\sum c_j x_j \geq D$. $c_j, D \in \mathbb{Z}$.

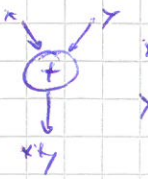
→ Addition $$\frac{\sum a_j x_j \geq A \qquad \sum b_j x_j \geq B}{\sum (a_j + b_j) x_j \geq A + B}$$

→ Multiplication $$\frac{\sum a_j x_j \geq A}{\sum \lambda a_j x_j \geq \lambda A} \quad \lambda \in \mathbb{N}^+$$

→ Division $$\frac{\sum a_j x_j \geq A}{\sum a_j/\lambda \, x_j \geq \lceil A/\lambda \rceil} \quad \lambda \in \mathbb{N}^+;\ \lambda \mid a_j.$$

Division rule says e.g. $x \geq 1/2 \Rightarrow x \geq 1$ (if x integer).

Proof may have very large integers; want our circuit to deal with them without extra overhead. We will be very generous: allow any real numbers on wires, any functions on gates as long as they are monotone.

✓ 

$x \leq x'$, $y \leq y'$ | $x+y \leq x'+y'$

✗ 

$x \leq x' \not\Rightarrow -x \leq -x'$

Monotone real circuits more powerful than monotone boolean circuits (∨ ≡ max, ∧ ≡ min). How more powerful? We don't know! We know better for some functions, but not which.

So, these are the results we want to prove:

th (1): Let Π CP proof of clique-colouring. There is a monotone real circuit for clique-colouring of size $\leq |\Pi| \cdot n$.

th (2): Every monotone real circuit for clique-colouring has size $2^{n^{\Omega(1)}}$.

corollary (3): Every CP proof of clique-colouring has size $2^{n^{\Omega(1)}}$.

* this real circuit l.b. also implies resolution l.b.

Today: prove th (1), give overview on th (2).

What did we do for resolution? Label each clause "q" or "r", then figure out which set of clauses gives a proper proof. Now we will go one step further and actually build two candidates for a proof, then pick one.

Let us look at a generic inequality:

$$\underbrace{\sum e_i x_{p_i}}_{\text{number}} + \underbrace{\sum r_{eq_e}}_{\text{q part}} + \underbrace{\sum g_m r_m}_{\text{r part}} \geq \underbrace{D}_{\text{number}} \qquad (*)$$

We can split (*) into

$$\sum r_{eq_e} \geq D_q \ (q) \ ; \ \sum g_m r_m \geq D_r \ (r)$$

such that $(q) \wedge (r) \Rightarrow (*)$.

We want $\sum f + \sum g \geq D - \sum e$ and have $\sum f + \sum g \geq D_q + D_r$, so as long as $D_q + D_r \geq D - \sum e$ we are fine.

Assume we could do this for every linear inequality in the proof. Then we would have two "proofs" of the same shape, and at the end the original proof is $0 \geq 1$, so we would get $0 \geq D_q$ and $0 \geq D_r$. If we keep the invariant $D_q + D_r \geq D - \sum e$, this means $D_q + D_r \geq 1$, and since both are integers, at least one of D_q and D_r is ≥ 1. This means at least one of our "proofs" is actually a proof.

~~Great! Now let us see~~

As in the resolution case, if the proof comes from the "r" side then we can answer 1, and 0 if it comes from the "q" side.

Great! Now let us see how to build the proofs in detail.

Axiom downloads:

if q-axiom: $\sum \ell_e q_e + \sum e_{x p_x} \geq D \rightsquigarrow \underbrace{\sum \ell_e q_e \geq D - \sum e_{x p_x}}_{(q)}$; $\underbrace{0 \geq 0}_{(r)}$

all of $D - \sum e_{x p_x}$ goes to the (q) part; o/w we have an immediate contradiction.

if r-axiom: dual.

Addition: split in 2 additions: add q parts together, and r parts together.

Multiplication: multiply q/r parts.

Division: divide q/r parts.

obs $\left\lceil \frac{D_q}{\lambda} \right\rceil + \left\lceil \frac{D_r}{\lambda} \right\rceil \geq \left\lceil \frac{D_q + D_r}{\lambda} \right\rceil \geq \left\lceil \frac{D - \sum e_{x p_x}}{\lambda} \right\rceil = \left\lceil \frac{D}{\lambda} \right\rceil + \frac{\sum e_{x p_x}}{\lambda}$,

so invariant holds.

The q side only uses q axioms, and the r side only uses r axioms, so we completed our idea of splitting the proof in two.

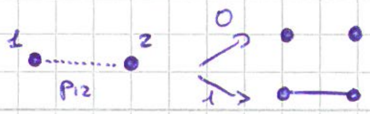
How do we simulate this with a circuit? Observe enough to compute D_r, and answer $[\![D_r \geq 1]\!]$. If $D_r \geq 1$, r side is false, so graph does not have a coloring, so can answer 1. If $D_r \leq 0$, $D_q \geq 1$, q side is false, so can answer 0.

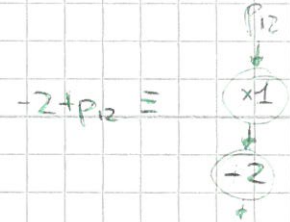
q-axioms	: constant 0 gate.
r-axioms	: compute $D - \sum e_{x p_x}$.
$p_x \geq 0$	: assume q-axiom
$-p_x \geq -1$	: "
addition	: $\oplus$ gate
multiplication	: $\cdot\lambda$ gate
division	: $\lceil / \lambda \rceil$ gate.
answer	: threshold gate

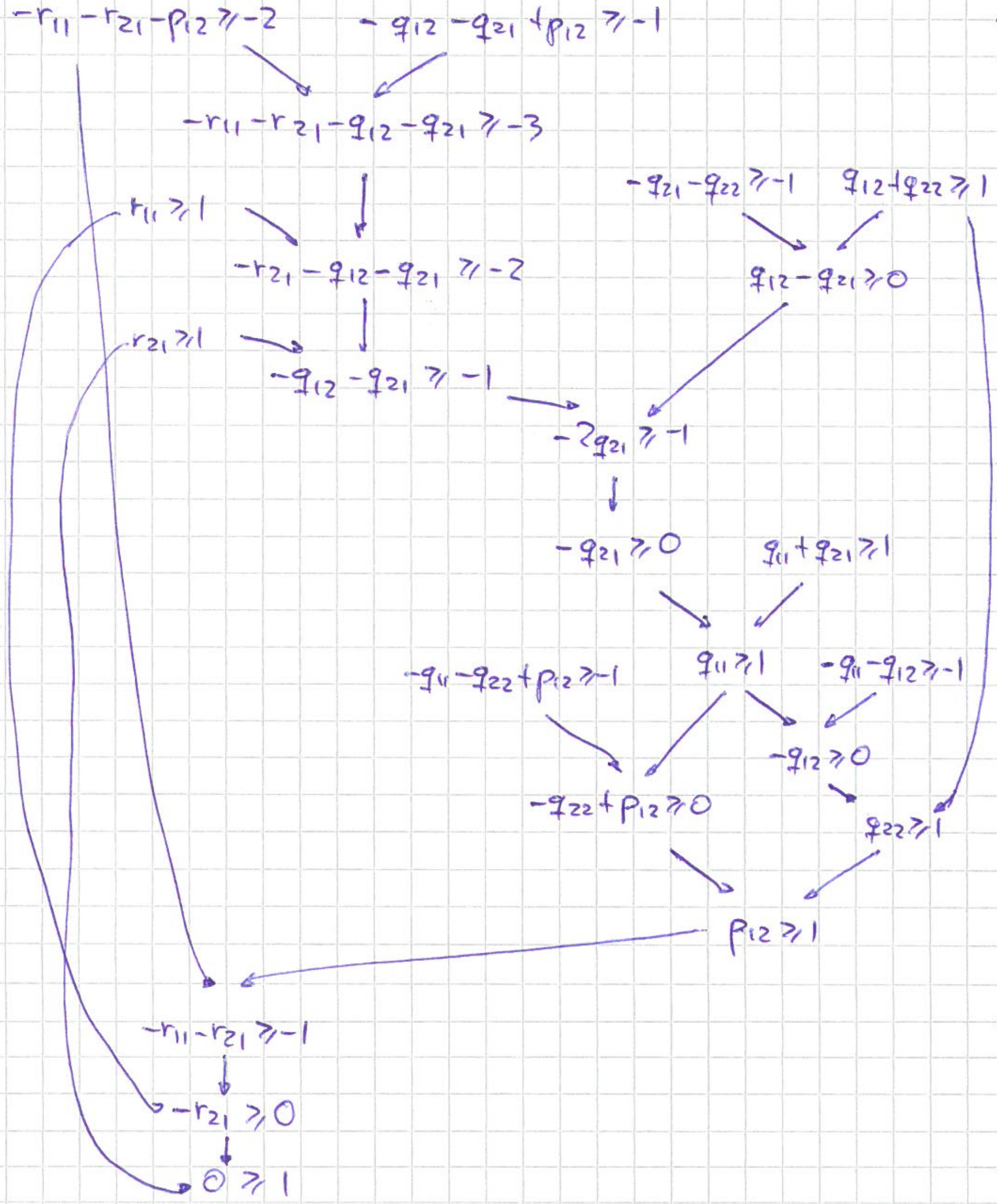
Are all these monotone? Only dangerous to multiply by $-e_x$ (we saw $\cdot -1$ is not monotone). But r axioms have negated p variables, so $e_x < 0$, so $-e_x > 0$ and we are fine.

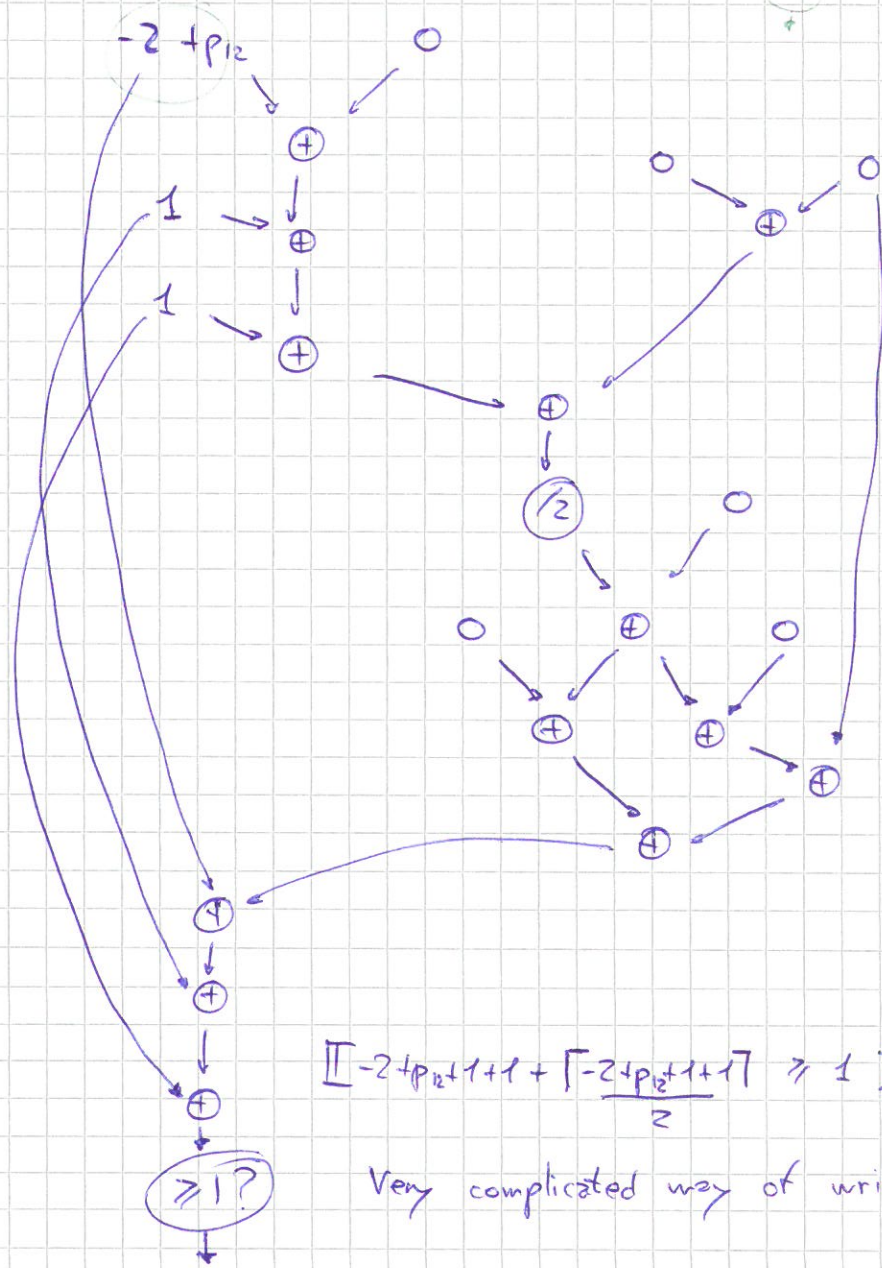
This concludes the proof of th 1. Let us see an example.

eg: $n=2$, $m=2$.









$$\llbracket -2+p_{12}+1+1+\left\lceil \frac{-2+p_{12}+1+1}{2} \right\rceil \geq 1 \rrbracket$$

Very complicated way of writing p_{12}.

For the rest of the lecture we are going to give an overview of how to prove thm (2). There are very few techniques to prove circuit lower bounds. One of them is approximation: Find some class of functions (e.g. polynomials of low degree) such that:

→ Inputs (x_i, 1-variable linear functions) are in the class

→ Applying a gate to two functions in the class can be approximated by another function in the class

→ Output cannot be well approximated within the class.

If we have such a class, then we can build an alternative circuit such that inputs have no error, every gate introduces a small error, but at the end has a huge error; ~~thus~~ thus, to get many errors, we need many gates.

What do you mean, errors? We have errors for an input x when:

→ ~~Answer~~ Answer is 0, but approx > original

→ Answer is 1, but approx < original

(otherwise we are only doing better).

We will prove (next lecture)

lem (4) approximating a gate produces an error in at most a $n^{\ell}\left(\frac{e^2}{2(m-1)}\right)^r$ fraction of 0-inputs.

lem (5) approximating a gate produces an error in at most a $4(\ell+1)2^{-\ell}$ fraction of 1-inputs.

lem (6) for a function in our class, either $f \geq 1$ on all inputs or $f < 1$ in a $2/4$ fraction of 1-inputs.

* 0-inputs and 1-inputs are actually not all inputs s.t. the circuit answers 0 or 1, but only complete m-1-partite graphs and m-cliques.

this is enough to prove thm 2.

proof of thm 2.

fix $m = \frac{1}{8}(n/\log n)^{2/3}$, $\ell = m^{1/2}$, $r = 4m^{1/2}\log n$.

let $\tilde{f}$ be the approximated circuit.

case 1: $\tilde{f} > 1$ on all inputs.

by lem (4) need $\left[n^{\ell}\left(\frac{\ell^2}{2(m-1)}\right)^{r}\right]^{-1} \sim 2^{-\log n \cdot m^{1/2}} \cdot 2^{4m^{1/2}\log n} \sim 2^{n^{1/3}\log n^{2/3}}$ gates.

case 2: $\tilde{f} < 1$ in $2/4$ fraction of cliques

by lem (5) need $2^{4}/[4(e+1)2^{-\ell}] \sim 2^{\ell - \log \ell} \sim 2^{(n/\log n)^{1/3}}$ gates.

This concludes the overview. It remains to define which class of functions we use to approximate the circuit, and then to prove lemmas 4, 5, 6.