CSE 599 Proof Complexity
Lecture 4 12 Oct 2020

For $m \geq n$, PHP^m_n: clauses

pigeon $x_{i1} \vee \cdots \vee x_{in}$ for all $i \in [m]$

hole $\overline{x_{ij}} \vee \overline{x_{i'j}}$ for all $i \neq i' \in [m]$, $j \in [n]$

Variants function $\overline{x_{ij}} \vee \overline{x_{ij'}}$ for all $i \in [m]$, $j \neq j' \in [n]$

surjectivity $x_{1j} \vee \cdots \vee x_{mj}$ for all $j \in [n]$

function-PHP^m_n, onto-PHP^m_n, $\boxed{\text{bijective-}PHP^m_n}$

easiest

<u>Theorem</u> PHP^n_{n-1} requires resolution proof size $\geq 2^{n/20}$

(Same holds for bijective-PHP^n_{n-1})

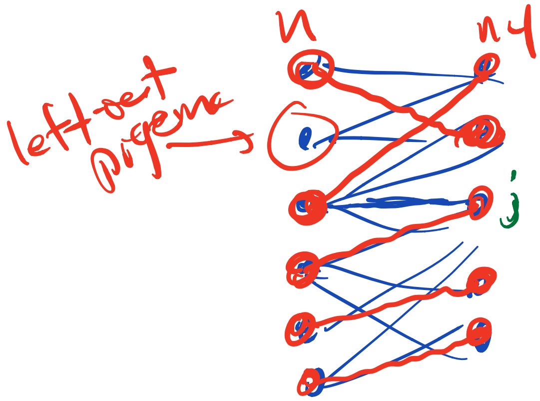
<u>Proof</u> Idea: • Any resolution refutation of PHP^n_{n-1} requires wide clauses (lots of variables)

* Each wide clause rules out few truth assignments so we need lots of them.

* If proof is short then we can set few^t vars to keep PHP^n_{n-1} after there are $\geq t$ & all wide clauses gone

PHP^{n-t}_{n-t-1} ← hard

Last time critical truth assignments (cta)



- vars for a matching of size n-1 are 1
- all other vars are 0

i-CTA if i is the leftout pigeon

Monotone conversion of a clause C

$$C \qquad\qquad M(C)$$

$$x_{ij} \mapsto x_{ij}$$

$$\bar{x}_{ij} \mapsto x_{1j} \vee \dots \vee x_{(i-1)j} \vee x_{(i+1)j} \vee \dots \vee x_{nj}$$

no x_{ij}

C and M(C) behave exactly the same on all CTA's

$C_1 \dots C_s = \perp$ resolution proof

$\Downarrow$

$M(C_1) \dots M(C_s) = \perp$ monotone conversion

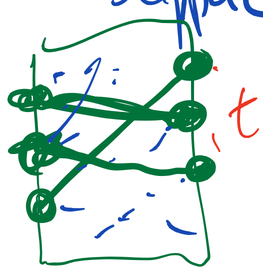
Partial matching restriction

Suppose T matching of size t

PHP^{n}_{n-1}

$\Downarrow T$

PHP^{n-t}_{n-t-1}



restriction based on t:

set $x_{ij}=1$ for $(i,j) \in T$

$x_{ij}=0$ for i or j matched by T but $(i,j) \notin T$

$n(n-1)$ vars

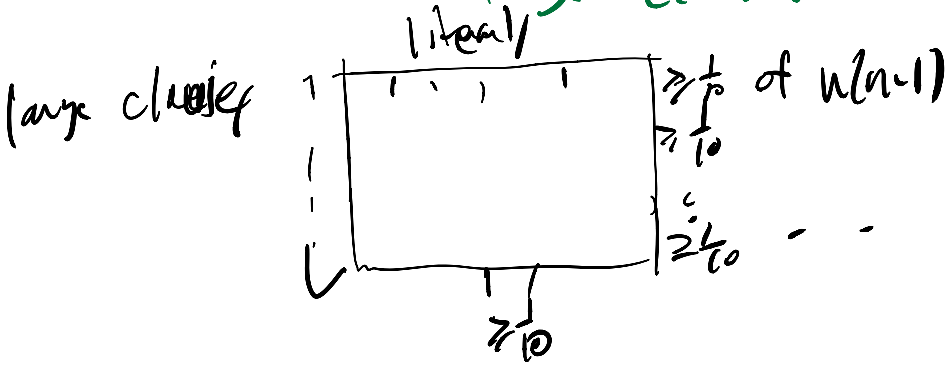
We call C **large** iff $M(C)$ has $\geq n^2/10$ (literals)

Let L = # of large clauses $\leq$ proof size

Claim: There is small matching T that kills off all large clauses

$T = \emptyset$

$\therefore \exists\, x_{ij}$ that appears in $\geq \frac{L}{10}$ large clauses



add (i,j) to T. forces all clauses contain x_{ij} to true (kills them)

# of large clause $L \mapsto \frac{9}{10} L$

Repeat this $\lceil \log_{\frac{10}{9}} L \rceil$ times $\Rightarrow$ no large clauses left. □

So for $L \le 2^{n/20}$ then

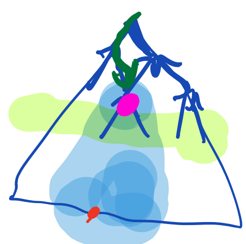
$$n' = n - t \ge n - \log_{\frac{10}{9}} L \ge n - \left(\log_{\frac{10}{9}} 2\right)\frac{n}{20}$$

$$\ge 0.671\, n$$

Claim: Any resolution refutation of PHP^{n}_{n+1} has a clause C s.t. $M(C)$ has $\ge 2n^2/9$ literals

$$\boxed{\frac{2(n')^2}{9} > \frac{n^2}{10}}$$

Proof:



C in proof

$\mathrm{bad}(C) = \{ i \mid$ there is some i-CTA α making C false $\}$

progress at C $= |\mathrm{bad}(C)|$

$\mathrm{bad}(\bot) = [n]$

$\mathrm{bad}(\mathrm{pigeon}_i) = \{i\}$

$C = A \cup B$

$C' = A \vee x \qquad B \vee \bar{x} = C''$

$\mathrm{bad}(C) \subseteq \mathrm{bad}(C') \cup \mathrm{bad}(C'')$

$|\mathrm{bad}(C)| \le |\mathrm{bad}(C')| + |\mathrm{bad}(C'')|$

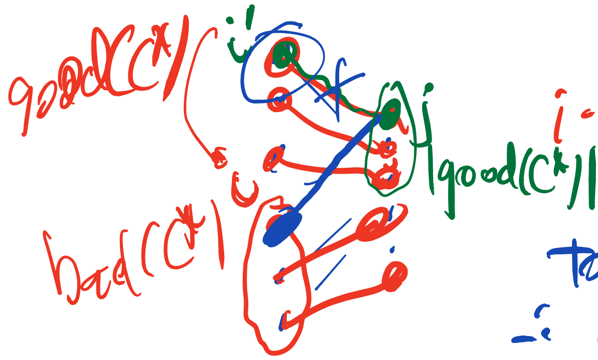
Start at output $\bot$ walk back through the proof choosing predecessor with larger $|\mathrm{bad}(C)|$ until C^*

π

$|\mathrm{bad}(C^*)| \le \frac{2}{3} n$

$\Rightarrow |\mathrm{bad}(C^*)| \ge n/3$

Claim: C^* has $\geq 2n^2/9$ literals



$i \in bad(C^*)$

i-CTA makes C^* false

α

$i' \notin bad(C^*)$

Toggle α to get β

$\therefore C^*(\beta)$ is true, since $i' \notin Bad(C^*)$

$\therefore |bad(C^*)| \cdot |good(C^*)|$ x_{ij} must be in C^*

literals in C^* $\therefore$ we get $|good(C^*)|$ vars x_{ij}

lits $\geq \frac{2n}{3} \cdot \frac{n}{3} = \frac{2n^2}{9}$ ∎

Width of resolution

Def^n Given a CNF formula F (or set of clauses, eg a resolution proof)

$w(F)$ or $w(P)$ = length longest clause in F or P

$\text{width}(F) = \min_P \{w(P) : P \text{ is a resolution refutation of } F\}$

$Res_{tree}(F)$ = min size of treelike ref'n. of F

$Res(F)$ = min.

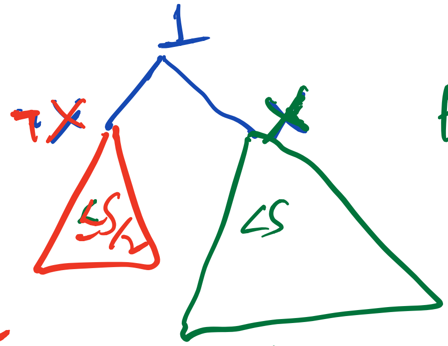
→ **Thm** $Res_{tree}(F) \geq 2^{width(F) - w(F)}$

We will actually prove:

Thm Every tree-like resolution refutation of F of size $\leq S$ $\Rightarrow$ $\exists$ refutation of width $\lceil \log_2 S \rceil + w(F)$

$$\boxed{\text{i.e. } width(F) \leq \log_2(Res_{tree}(F)) + w(F)}$$

Proof by induction.



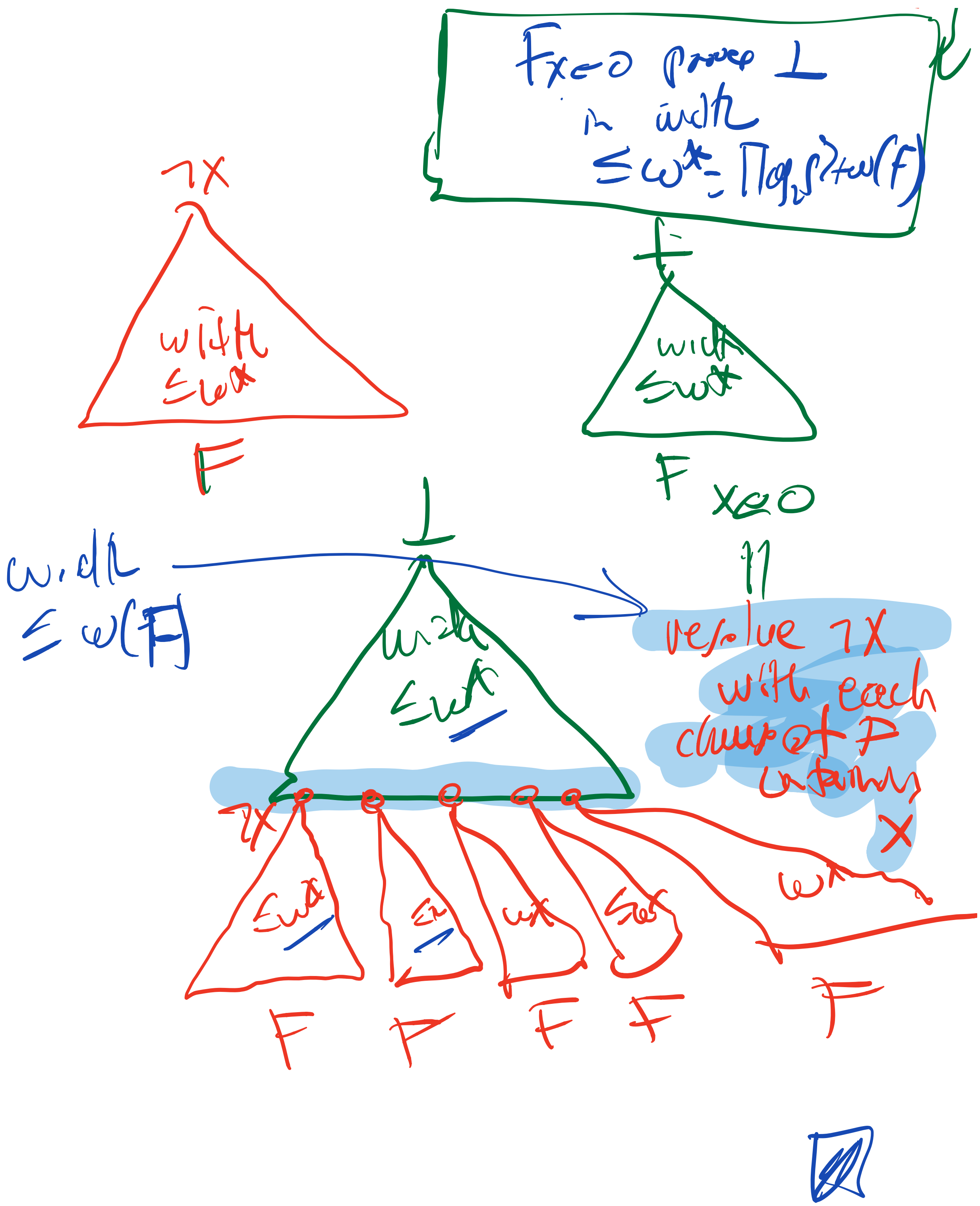
F prove ¬x in size $\leq S/2$

F proves x in size $< S$

$F_{x\leftarrow 1}$ prove ⊥ in size $\leq S/2$

$F_{x\leftarrow 0}$ proving ⊥ in size $< S$

I.H. width $\leq \lceil \log_2 S/2 \rceil + w(F)$
$= \lceil \log_2 S \rceil + w(F) - 1$ $\Rightarrow$ F proves ¬x in width $\lceil \log_2 S \rceil + w(F) = (\omega)$


$F_{x \leftarrow 0}$ proves $\bot$
in width
$\leq w^* = \lceil \log_2 S \rceil + w(F)$
$\bot$
width $\leq w^*$
$F_{x \leftarrow 0}$
¬X
width $\leq w^*$
F
width
$\leq w(F)$
⊥
width $\leq w^*$
resolve ¬X
with each
clause of F
containing
X
¬X
$\leq w^*$
w^*
F
F
F
F
F