Narrow Proofs May Be Maximally Long

Jakob Nordström
KTH Royal Institute of Technology

University of Chicago
May 9, 2014

Joint work with:

Albert Atserias
(UPC, Barcelona)

Massimo Lauria
(KTH, Stockholm)

Proof complexity: Study of succinct, polynomial-time verifiable certificates for ***unsatisfiable*** CNF formulas

Generally believed impossible to provide certificates of length at most polynomial in formula size

If proven, would imply $coNP \neq NP$ and hence $P \neq NP$

Still very distant goal...

More recent motivation for proof complexity:
Applied SAT solving

A SAT solver looks for satisfying assignments of a CNF

When CNF formula is unsatisfiable, solver implicitly searches for certificate/proof of unsatisfiability using some method of reasoning (i.e., a proof system)

Proof complexity: study of potential and limitations of methods of reasoning used by SAT solvers

Suitable proof systems for SAT solving?
Trade-off between:

Expressiveness: stronger proof system ⇒ shorter proofs

Simplicity: weaker proof system ⇒ simpler search space
⇒ better heuristics

Resolution proof system

- Simple enough to allow efficient proof search
- Powerful enough to be useful in practice
- Davis-Putnam-Logemann-Loveland (DPLL) algorithm
- CDCL SAT solvers (Conflict-Driven Clause Learning)
- Algorithms in Tarjan (1972), Tarjan & Trojanowski (1977), Jian (1986), and Shindo & Tomita (1990) for finding **independent sets** can be simulated in resolution (see Chvátal, 1977)
- McDiarmid (1984) proof system for **colourability**

Definition of resolution

Show $F = \bigwedge_{i=1}^{m} C_i$ unsatisfiable

Initial clauses: C_i

Resolution rule: $$\frac{A \vee x \qquad B \vee \neg x}{A \vee B}$$

Goal: Derive empty clause $\perp$

SIZE: # clauses in resolution proof

SPACE: # clauses in memory during verification

WIDTH: # literals in largest clause in proof

width lower bounds

proof size lower bounds

[Ben-Sasson & Wigderson ’99]
[Bonet & Galesi ’99]

proof space lower bounds

[Atserias & Dalmau ‘03]
[Ben-Sasson & Nordström ‘08]

Small width implies small size

Resolution proof in width $\leq w$ must have size $\leq |Vars|^{O(w)}$

Proof: Just count total # distinct clauses

But all known (natural) formulas with proof width $< \sqrt{|Vars|}$
in fact have ***linear proof size*** measured in size of formula

Small width implies small size

Resolution proof in width $\leq w$ must have size $\leq |Vars|^{O(w)}$

Proof: Just count total # distinct clauses

But all known (natural) formulas with proof width $< \sqrt{|Vars|}$
in fact have ***linear proof size*** measured in size of formula

So is this simple counting argument tight?

Small width makes CDCL run fast

Theorem [Atserias, Fichte, Thurley '09]

If there exists a width-w proof, then w.h.p. the formula is solved in time $|Vars|^{O(w)}$ by CDCL (with enough randomness)

Note that CDCL couldn't care less about narrow proofs...

Small width makes CDCL run fast

Theorem [Atserias, Fichte, Thurley '09]

If there exists a width-w proof, then w.h.p. the formula is solved in time $|Vars|^{O(w)}$ by CDCL (with enough randomness)

Note that CDCL couldn't care less about narrow proofs...

Is this running time optimal?

Our results

We exhibit family $F_{n,k}$ of polynomial-size 3-CNF formulas that:

- have narrow **resolution** proofs of width $O(k)$
- require proofs of size $n^{\Omega(k)}$ in

  resolution

  polynomial calculus (Gröbner basis computations)

  Sherali-Adams (linear programming hierarchy)

i.

description of the formula

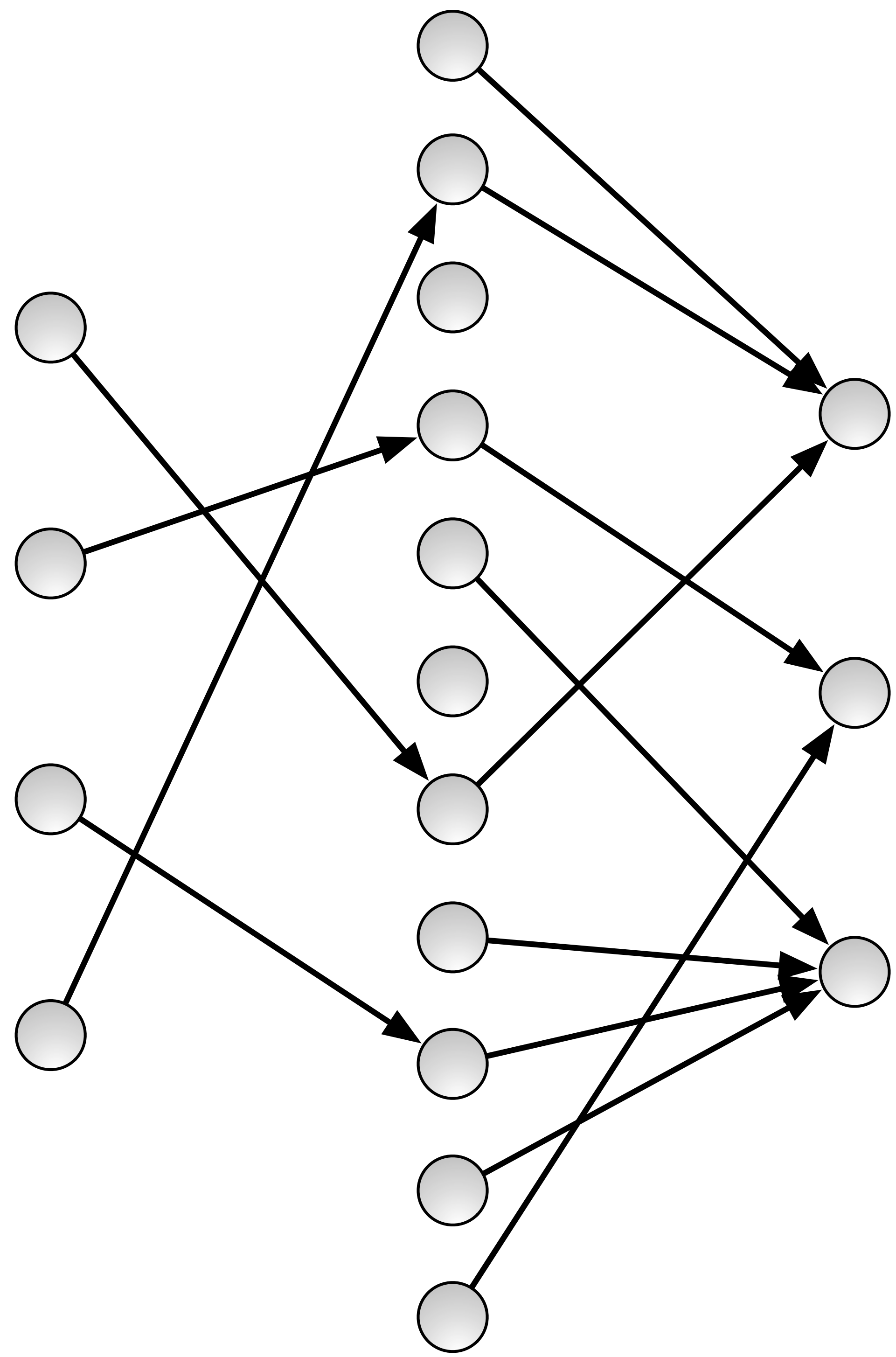
An “obfuscated” pigeonhole principle

Our 3-CNF formula claims that it is possible to

- pick k among a set of n pigeons
- map the chosen pigeons one-to-one to k-1 holes

$F_{n,k}$:"The composition of *p* and *q* is one-to-one"

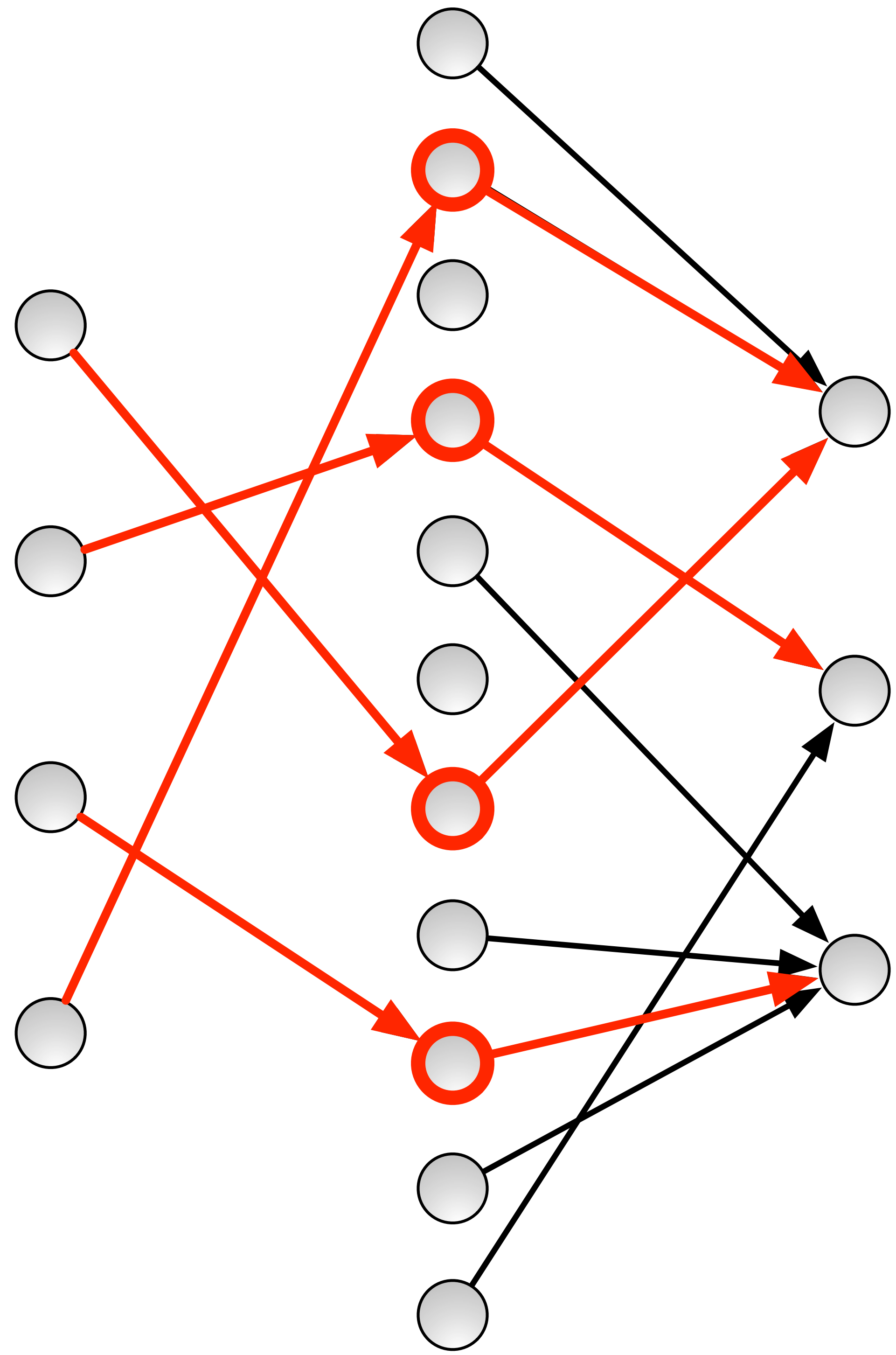
$p : [k] \longrightarrow [n]$



$q : [n] \longrightarrow [k-1]$

$F_{n,k}$:"The composition of *p* and *q* is one-to-one"

$p : [k] \longrightarrow [n]$



$q : [n] \longrightarrow [k-1]$

function p picks k pigeons

$$p_{u,1} \vee \cdots \vee p_{u,n}$$

$$\overline{p}_{u,v} \vee \overline{p}_{u',v}$$

set $r \subseteq [n]$ of picked pigeons

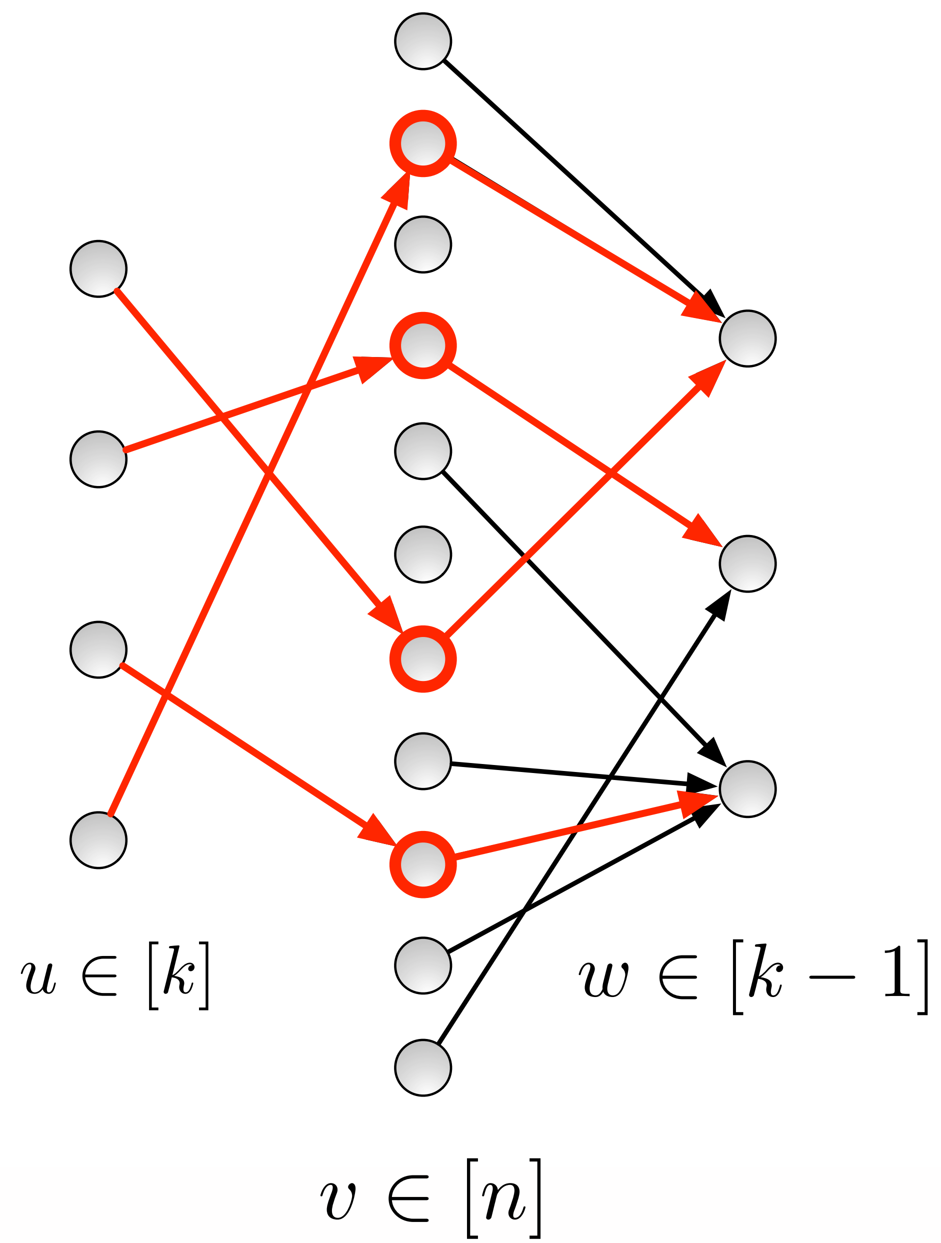
$$\overline{p}_{u,v} \vee r_v$$

q maps the picked pigeons into holes

$$\overline{r}_v \vee q_{v,1} \vee \cdots \vee q_{v,k-1}$$

$$\overline{r}_v \vee \overline{r}_{v'} \vee \overline{q}_{v,w} \vee \overline{q}_{v',w}$$

standard conversion to 3-CNF



The actual formula in the paper is converted to 3-CNF

$$l_1 \vee l_2 \vee \ldots \vee l_m \longrightarrow \begin{array}{r} l_1 \vee l_2 \vee e_1 \\ \neg e_2 \vee l_3 \vee e_3 \\ \vdots \\ \neg e_{i-1} \vee l_i \vee e_i \\ \vdots \\ \neg e_{m-2} \vee l_{m-1} \vee l_m \end{array}$$

Ignore this detail to simplify the talk

The 3-CNF version of the formula has

$O(n^2)$ variables

$O(kn^2)$ clauses

Refutation by brute-force DPLL procedure

For each $u \in [k]$:

choose a $v_u \in [n]$ and fix p_{u,v_u} to true

if there is a conflict, then <u>backtrack</u>

For each $u \in [k]$:

choose a $w \in [k-1]$ and fix $q_{v_u,w}$ to true

if there is a conflict then <u>backtrack</u>

Yields proof in tree-like resolution (= DPLL)

Size $n^k k^k = n^{O(k)}$

Width $2k + 1$

ii.

lower bound for resolution

Key tool: random restriction

A partial assignment ρ

$$F_{n,k} \xrightarrow{\quad\rho\quad} F_{n,k}\restriction_{\rho}$$ simplifies formula

$$\pi \xrightarrow{\quad\rho\quad} \pi\restriction_{\rho}$$ proof of simplified formula

Idea:

Simplified formula requires proof with complex clause

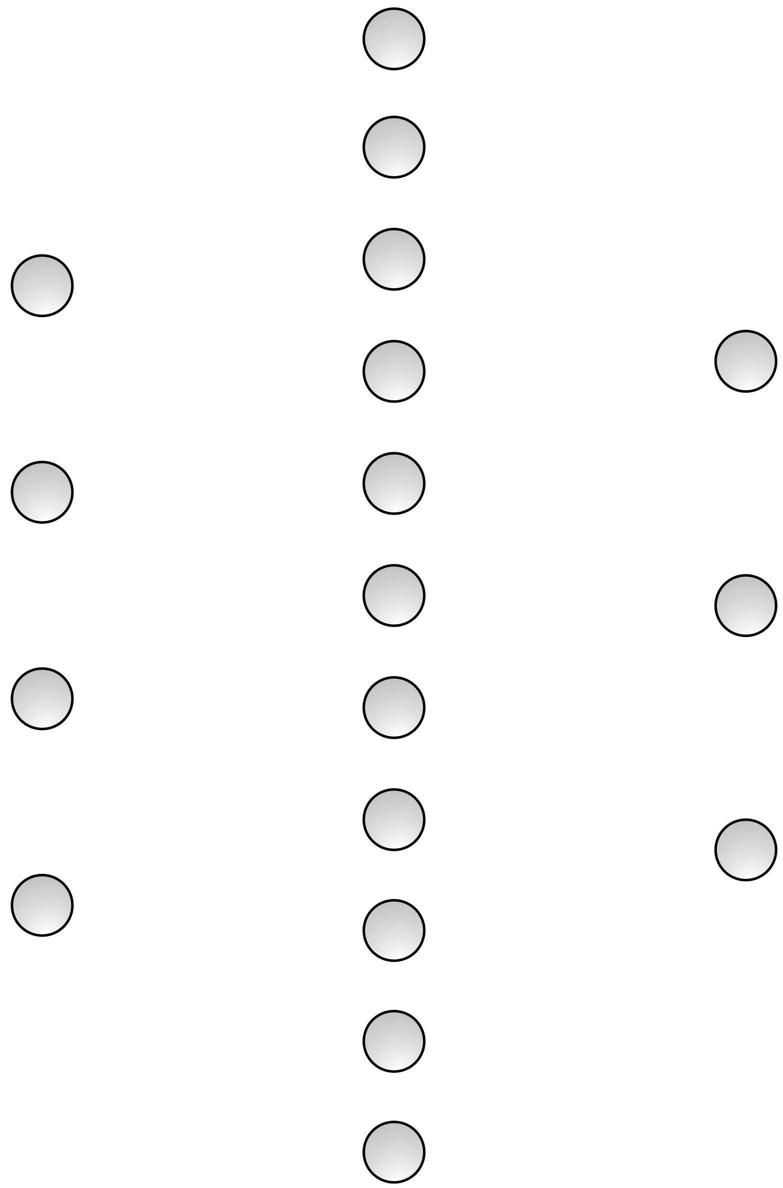
If proof is small, restriction removes all complex clauses

Usually, restriction arguments give exponential lower bounds, which **cannot work here**...

... we need to fine tune the restriction to make it work in the right range of parameters.

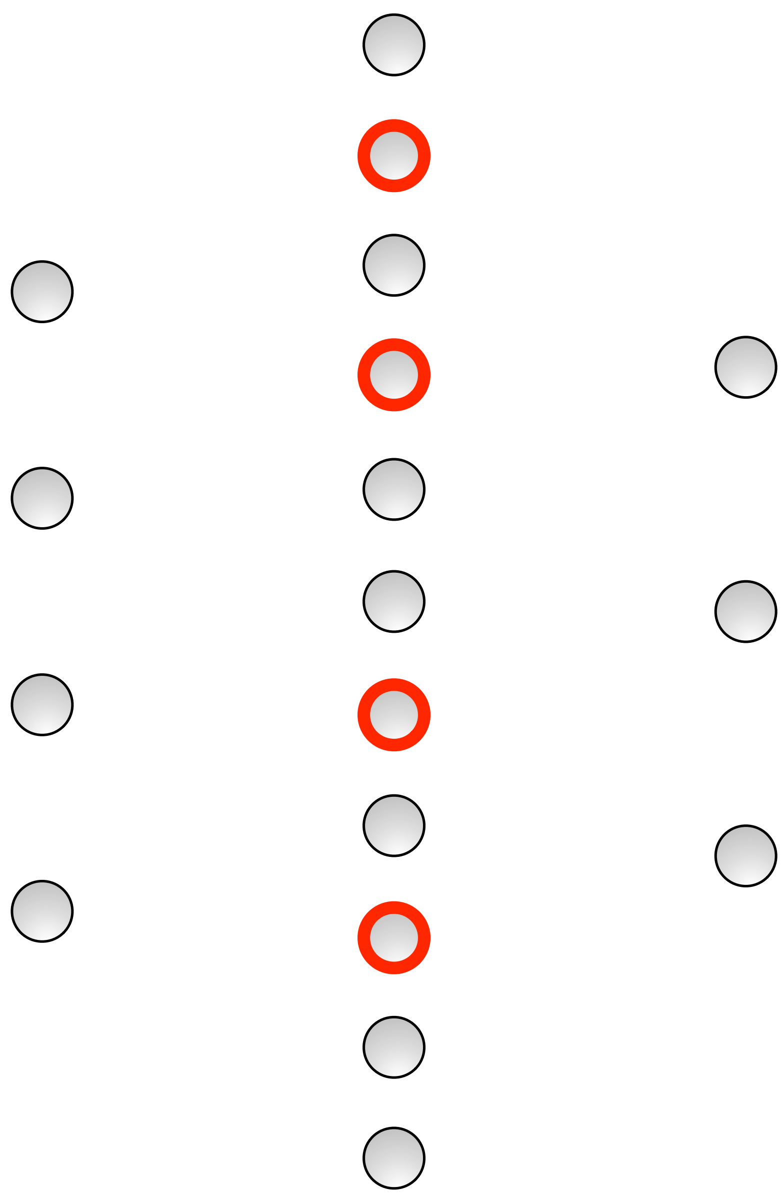
For the experts:
Furst-Saxe-Sipser style
instead of Håstad style

Random restriction argument: take ρ as follows



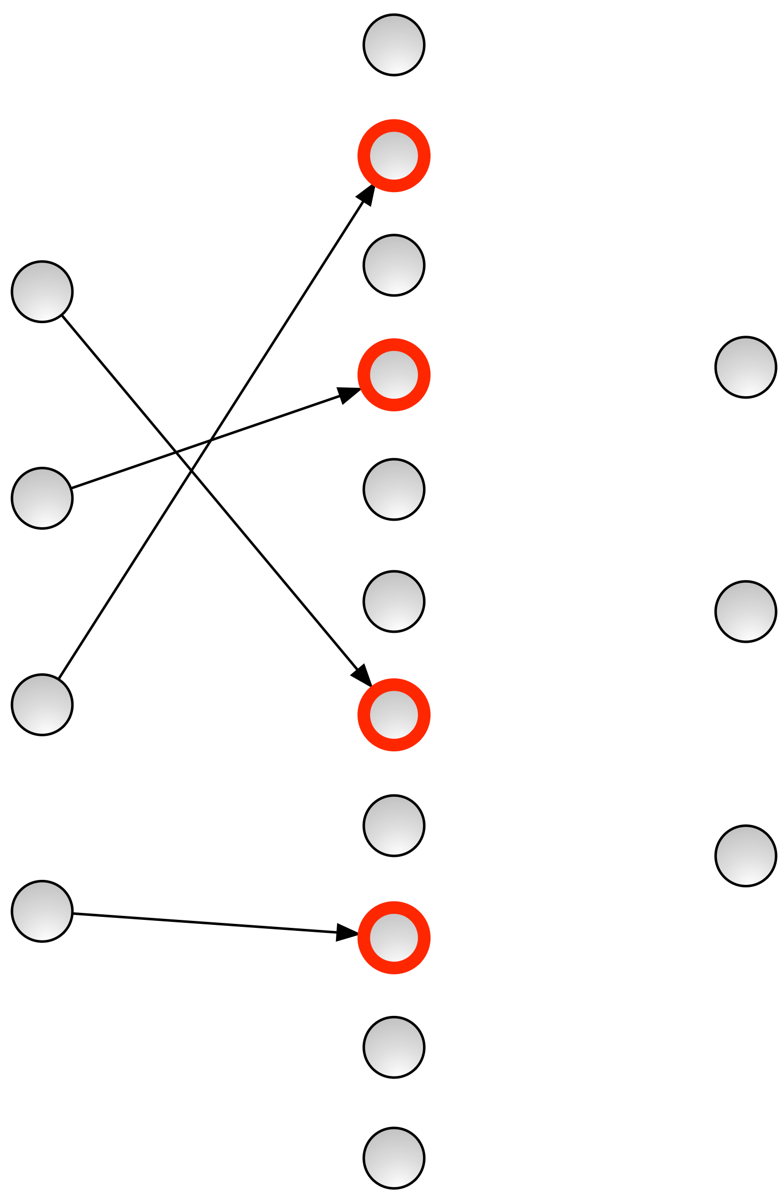
- Pick a set $S \subseteq [n], |S| = k$
- Fix $r_v := v \in S$
- Match $[k]$ with S arbitrarily
- If $r_v = 0$ fix all $q_{v,w}$ at random
- Resulting formula is PHP^k_{k-1} on surviving $q_{v,w}$ variables

Random restriction argument: take ρ as follows



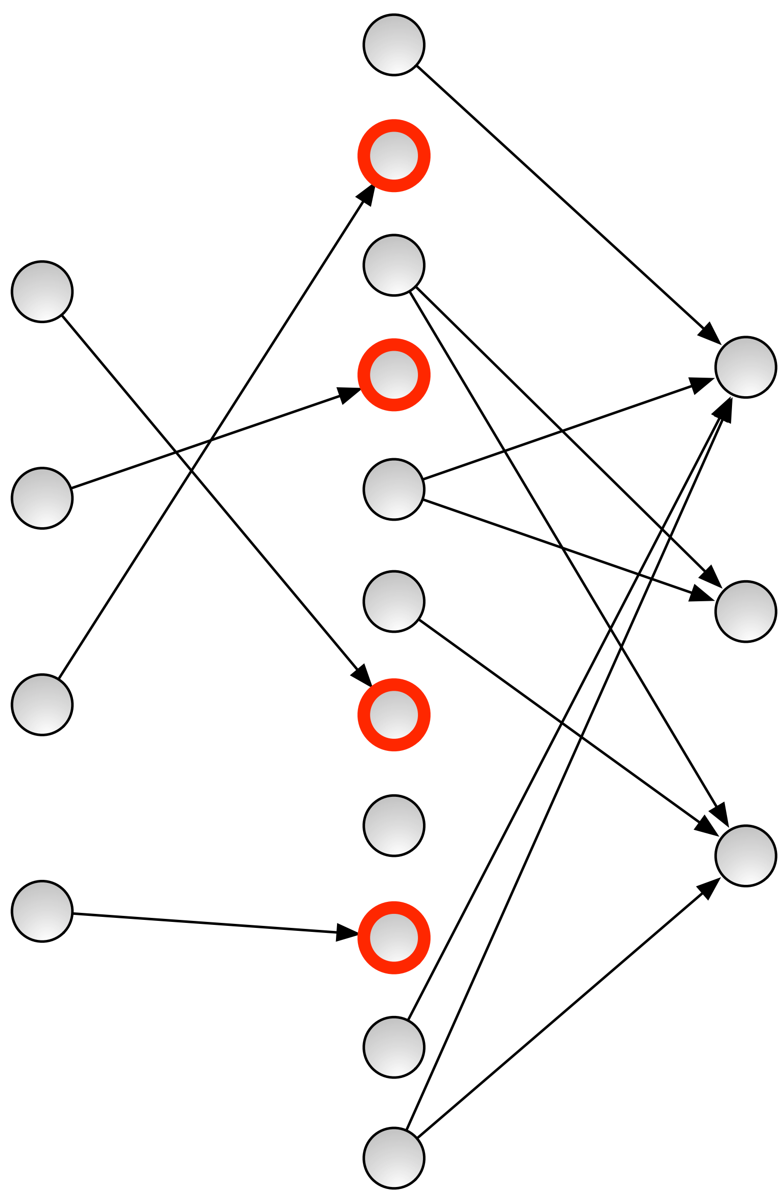
- Pick a set $S \subseteq [n], |S| = k$
- Fix $r_v := v \in S$
- Match $[k]$ with S arbitrarily
- If $r_v = 0$ fix all $q_{v,w}$ at random
- Resulting formula is PHP^k_{k-1} on surviving $q_{v,w}$ variables

Random restriction argument: take ρ as follows



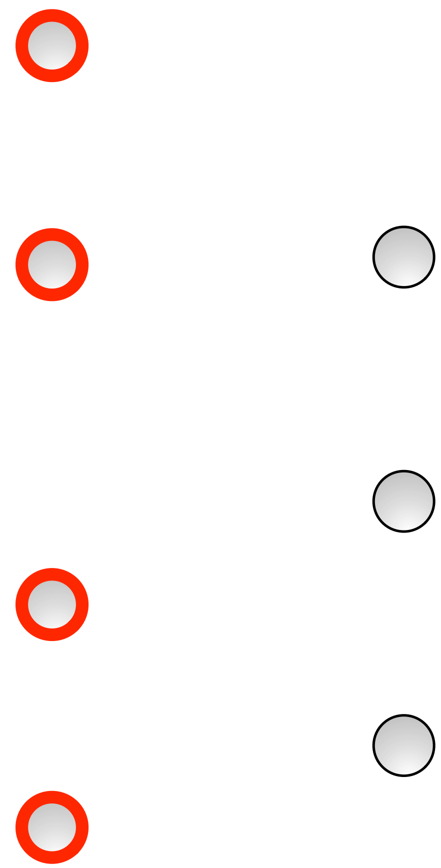
- Pick a set $S \subseteq [n], |S| = k$
- Fix $r_v := v \in S$
- Match $[k]$ with S arbitrarily
- If $r_v = 0$ fix all $q_{v,w}$ at random
- Resulting formula is PHP^k_{k-1} on surviving $q_{v,w}$ variables

Random restriction argument: take ρ as follows



- Pick a set $S \subseteq [n], |S| = k$
- Fix $r_v := v \in S$
- Match $[k]$ with S arbitrarily
- If $r_v = 0$ fix all $q_{v,w}$ at random
- Resulting formula is PHP^k_{k-1} on surviving $q_{v,w}$ variables

Random restriction argument: take ρ as follows



- Pick a set $S \subseteq [n], |S| = k$
- Fix $r_v := v \in S$
- Match $[k]$ with S arbitrarily
- If $r_v = 0$ fix all $q_{v,w}$ at random
- Resulting formula is PHP^k_{k-1} on surviving $q_{v,w}$ variables

Pigeons "mentioned" by the restricted clause:

$$p_{1,2} \vee \bar{p}_{3,2} \vee \bar{p}_{1,3} \vee r_2 \vee r_4 \vee \bar{r}_3 \vee q_{\mathbf{3},4} \vee q_{\mathbf{3},2} \vee q_{\mathbf{2},4} \vee \bar{q}_{\mathbf{1},2} \vee \bar{q}_{\mathbf{2},5}$$

$$\Big\downarrow \rho$$

$$q_{\mathbf{3},4} \vee q_{\mathbf{3},2} \vee \bar{q}_{\mathbf{1},2}$$ "mentions" 2 pigeons

Lemma 1. After restriction, a clause mentions *k*-1 pigeons with probability < $n^{-\Omega(k)}$

Hence, if proof is small there exists restriction yielding proof where no clause mentions *k*-1 pigeons

Let $S \subseteq [n], |S| = k$ be picked by the restriction

An OR of variables mentioning the same pigeon

E.g. $\bar{q}_{2,1} \vee q_{2,4} \vee \bar{q}_{2,5}$

is not set to true with probability at most

$$\left(\frac{1}{2} + \frac{k}{n-k}\right)$$

conditioned on the previous $< k$ choices

C a clause in the unrestricted refutation π

r # of pigeons mentioned in C

$$r \geq k \log n + k$$

$$\Pr[C \text{ is not satisfied}] \leq \left(\frac{1}{2}\right)^{k \log n} \leq n^{-k}$$

$$r < k \log n + k$$

$$\Pr[C \text{ contains all } k \text{ picked pigeons}] \lessapprox \binom{k \log n + k}{k} n^{-\Omega(k)}$$

Restricted clause mentions k-1 pigeons with probability

$$n^{-\Omega(k)}$$

so by union bound

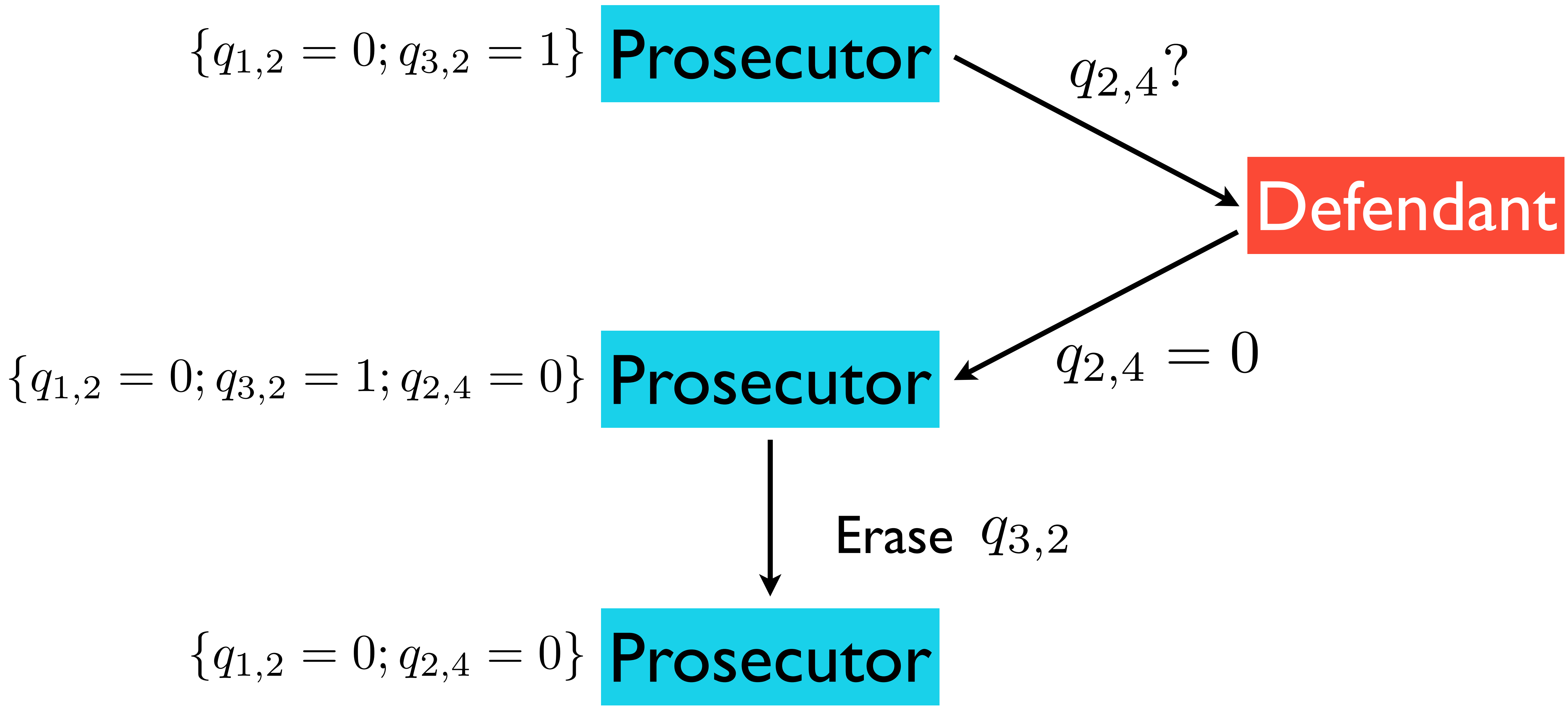
$$|\Pi| < n^{O(k)} \xrightarrow{\quad\quad\quad\rho\quad\quad\quad}$$ no clause mentions all k surviving pigeons, w.p. >0.

End of proof of Claim 1

Lemma 2: Any resolution refutation of PHP^{k}_{k-1} has a clause which mentions *k*-1 pigeons

(Proof is not hard using standard tools)

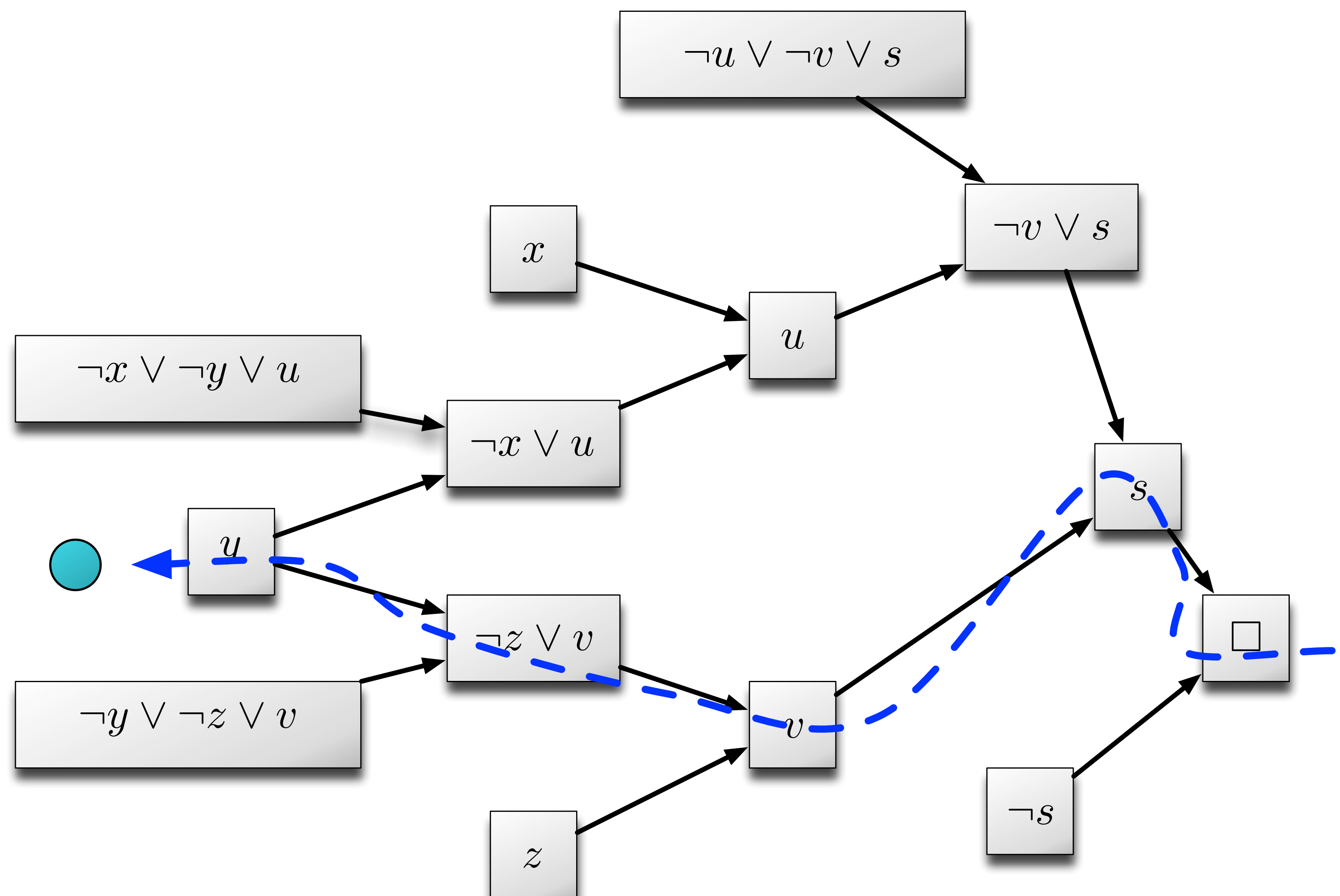
We use a 2-player game



Prosecutor wins: memory falsifies a clause of PHP^{k}_{k-1}

Defendant wins: memory mentions *k* pigeons

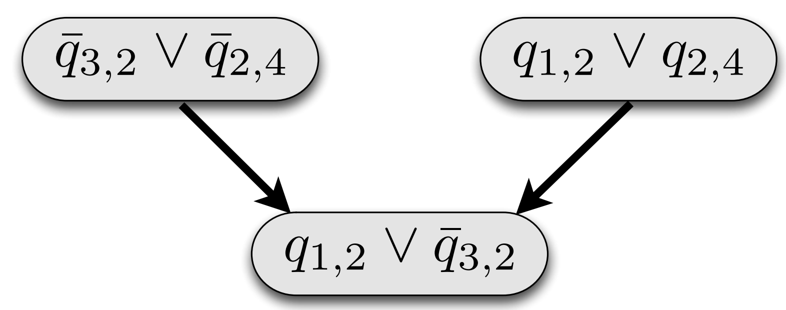
Resolution proofs which never mentions *k*-1 pigeons turns into **winning** Prosecutor strategy



How? Use the resolution proof as a rulebook
(but might need extra pigeon for resolution steps)

Prosecutor

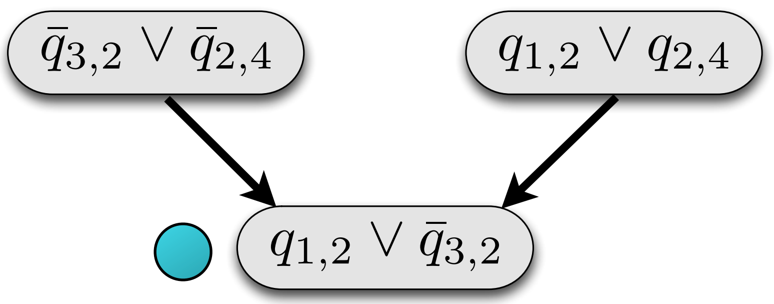
Defendant



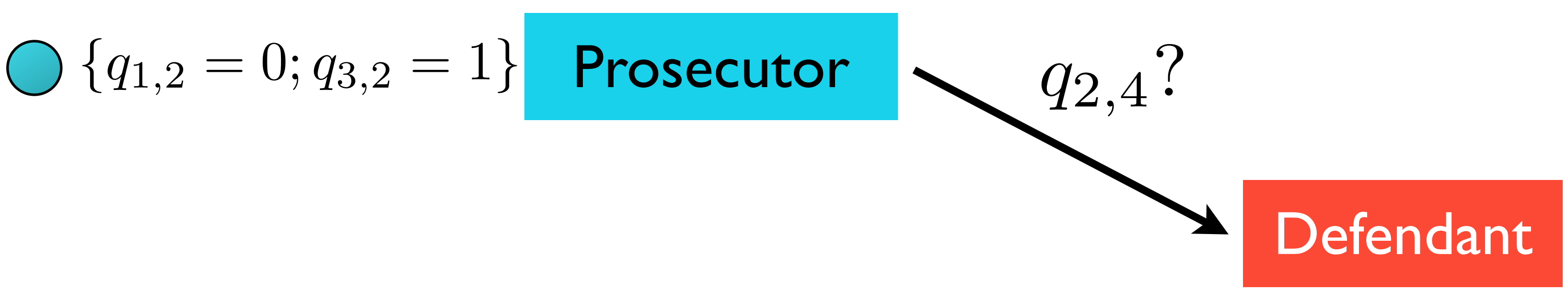
How? Use the resolution proof as a rulebook (but might need extra pigeon for resolution steps)

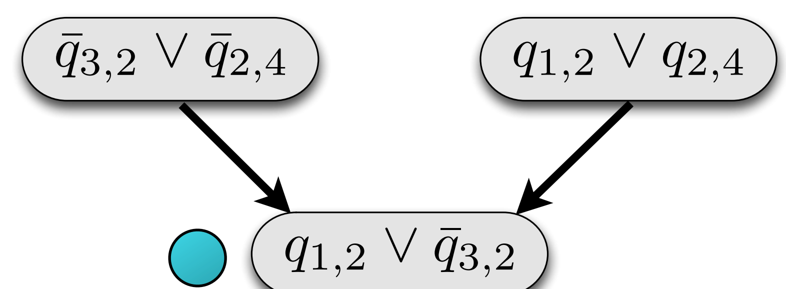
$\{q_{1,2} = 0; q_{3,2} = 1\}$ Prosecutor

Defendant

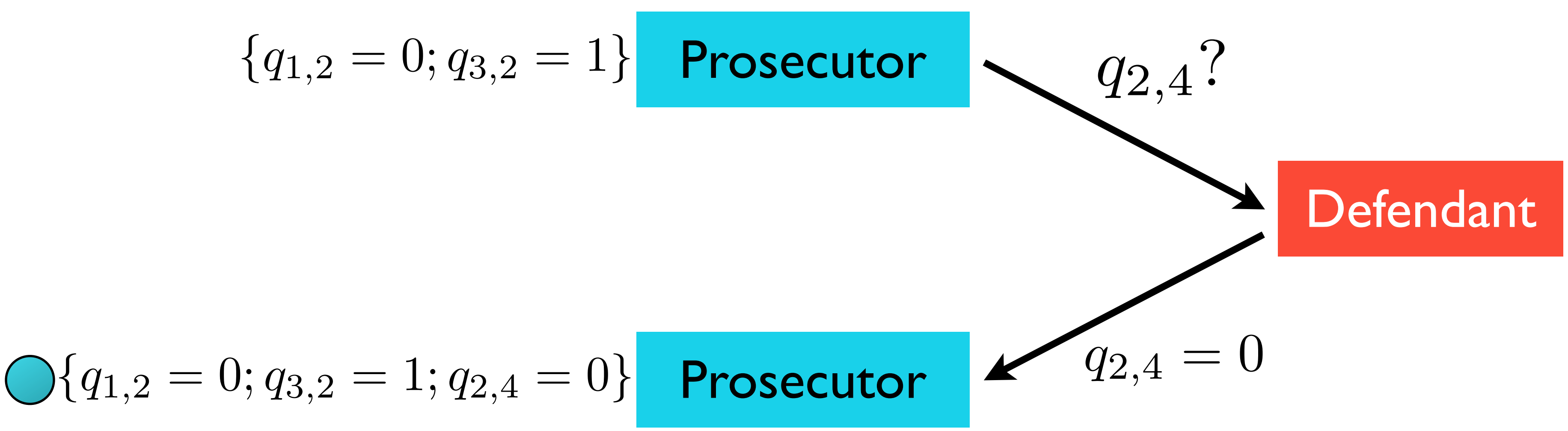


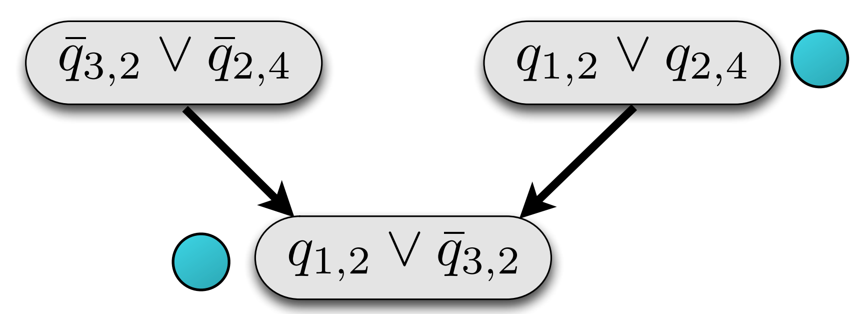
How? Use the resolution proof as a rulebook (but might need extra pigeon for resolution steps)



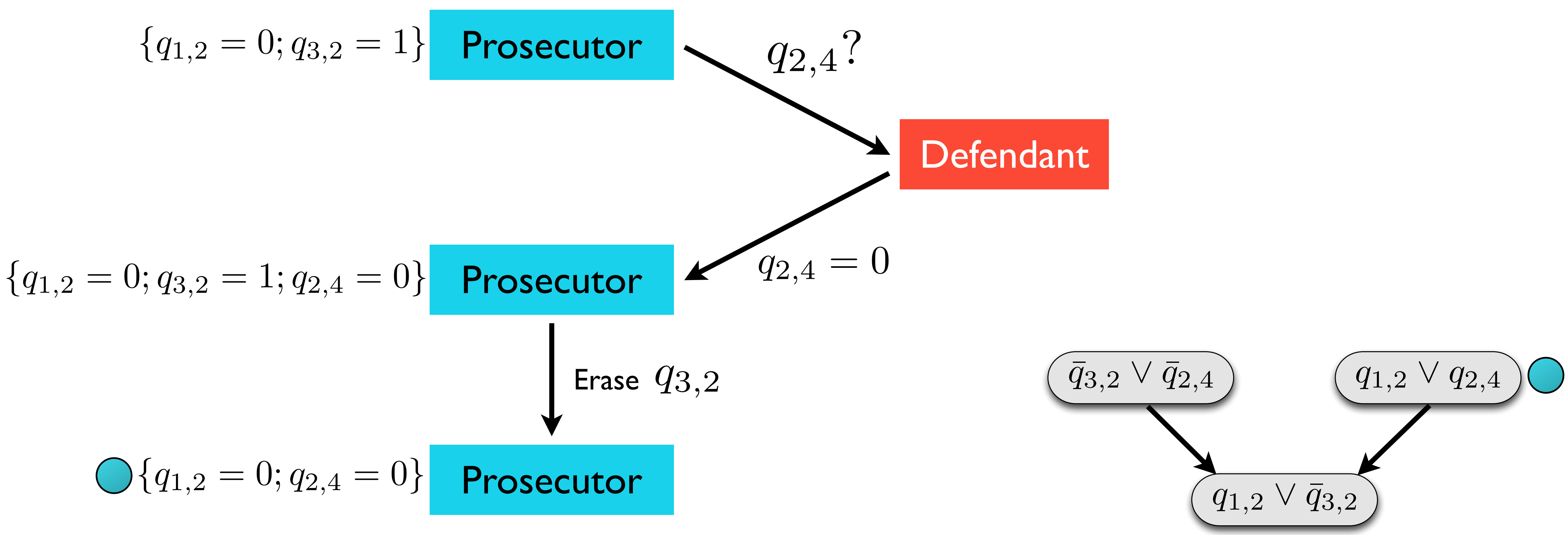


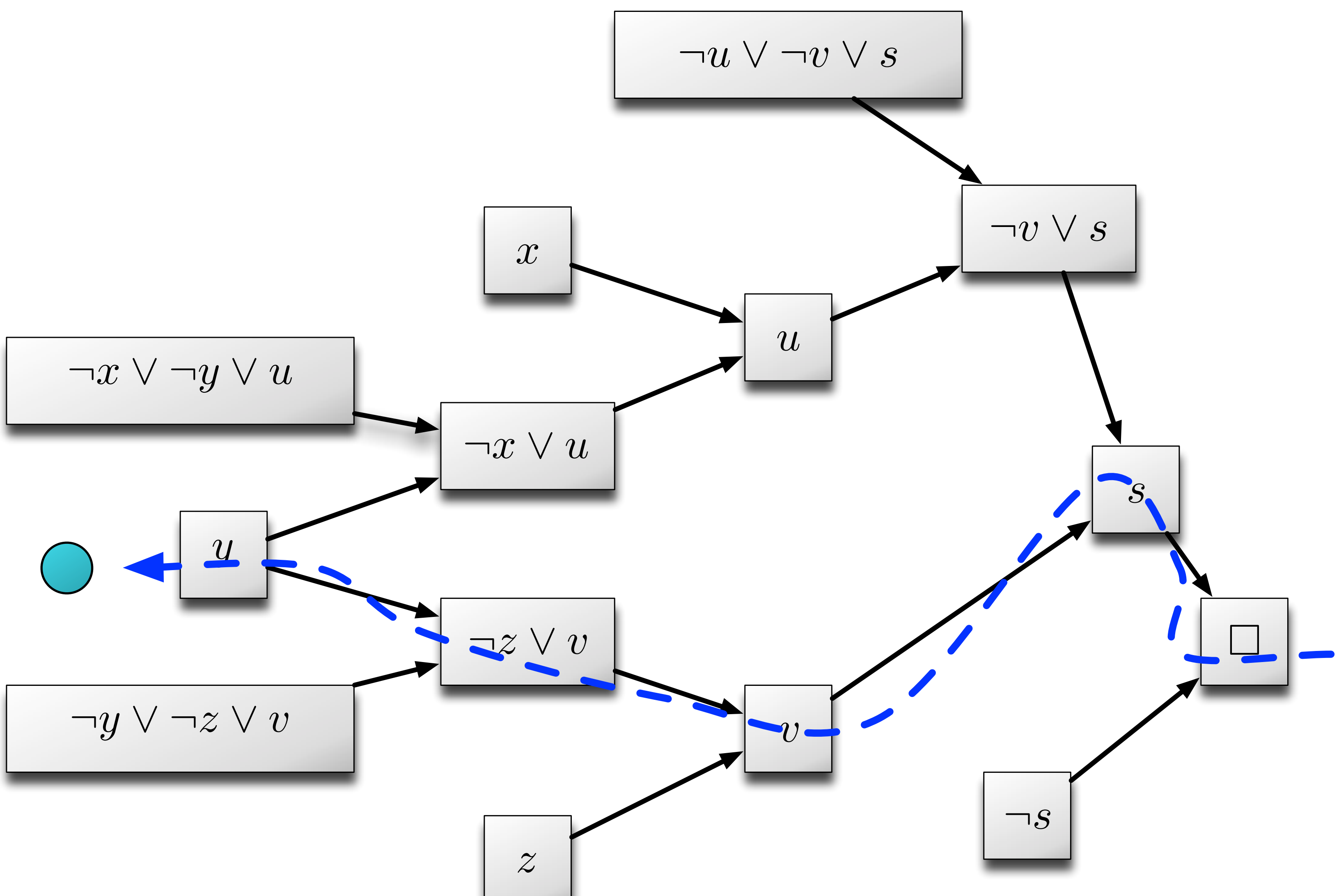
How? Use the resolution proof as a rulebook (but might need extra pigeon for resolution steps)





How? Use the resolution proof as a rulebook (but might need extra pigeon for resolution steps)





¬u ∨ ¬v ∨ s
x
¬v ∨ s
u
¬x ∨ ¬y ∨ u
¬x ∨ u
s
y
¬z ∨ v
¬y ∨ ¬z ∨ v
v
z
¬s
□

Defendant winning strategy:

Defendant keeps a matching between the pigeons mentioned in the record and the holes.

when Prosecutor queries $q_{v,w}$ ...

v is already mentioned: Defendant answers according to matching

v is not mentioned: Defendant matches it to a free hole w_v

when Prosecutor erases last occurrence of v...

Defendant removes (v, w_v) from the matching

Defendant matches the mentioned pigeons

$\longrightarrow$

Prosecutor must mention all k pigeons

$\longrightarrow$

Any proof of PHP^{k}_{k-1} has clause mentioning k-1 pigeons

End of proof of Claim 2

Proof recap

Consider a proof π of formula $F_{n,k}$ with $|\pi| < n^{O(k)}$

By random restriction, we get $\pi \restriction_{\rho}$ refutation of PHP^{k}_{k-1}

by Lemma 1, there is a restriction such that

$\pi \restriction_{\rho}$ does not mention *k*-1 pigeons in any clause

by Lemma 2

$\pi \restriction_{\rho}$ must mention *k*-1 pigeons in some clause

Our results

There are 3-CNF formulas $F_{n,k}$

- n^2 variables, kn^2 clauses,

with narrow tree-like resolution proof of

- width $2k+1$

Requires proof of size $n^{\Omega(k)}$ in proof systems

- resolution
- polynomial calculus
- Sherali-Adams

iii.

open problems

Lasserre/Sum of squares proof system

Is the counting argument $n^{\Omega(k)}$ tight for Lasserre?

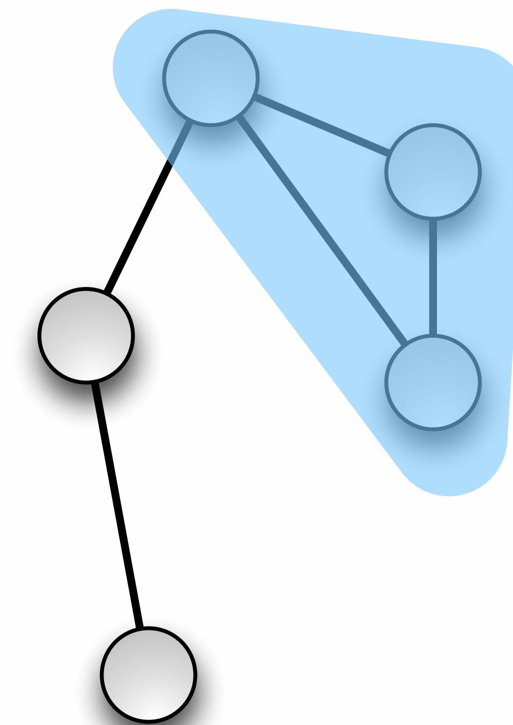
Our formula has *polynomial size* Lasserre proofs

k-Clique formula

Fix $G = (V, E)$ with no k-clique

$\bigvee_{v \in V} x_{iv}$ for $i \in [k]$

$\neg x_{iv} \vee \neg x_{jw}$ for $i \neq j, \{v, w\} \notin E$



[Beyersdorff, Galesi, Lauria, Razborov '12] conjecture size $|V|^{\Omega(k)}$

[Beyersdorff, Galesi, Lauria '13] prove it for treelike resolution

[Lauria, Pudlák, Rödl, Thapen '13] prove it for binary encoding

[Beame, Impagliazzo, Sabharwal '07] size $2^{\Omega(|V|)}$ for $k=O(|V|)$

Still open for **general** resolution and k much smaller than $|V|$

Parameterized Proof Complexity

[Dantchev, Martin, Szeider '11] discuss resolution proofs for the claim:

"CNF formula F has no SAT assignment with at most k ones"

and ask for formulas that require proof length $|Vars|^{\Omega(k)}$

Size-width trade-offs for resolution?

[Ben-Sasson & Wigderson ’99]:
Short resolution proof can be transformed into narrow one

However, transformation incurs exponential size blowup
So narrow proof is no longer short...

Can the proof be made narrow without exploding the size?
Or is there a trade-off between size and width so that the two measures cannot be optimized simultaneously?

Strong trade-offs known for
- width vs. space [Ben-Sasson ’02]
- size vs. space [Ben-Sasson & Nordström ’11, Beame, Beck, Impagliazzo ’12]

Thank you for your attention!