Extremely Deep Proofs

Noah Fleming, **Toniann Pitassi** and **Robert Robere**

UCSD

Columbia University
IAS
University of Toronto

McGill University

A New Kind of Tradeoff

Recently, several works exhibited an extremely strong type of tradeoff

A New Kind of Tradeoff

Recently, several works exhibited an extremely strong type of tradeoff

Supercritical Tradeoff

When one parameter is restricted, the other is pushed beyond worst-case.

A New Kind of Tradeoff

Recently, several works exhibited an extremely strong type of tradeoff

Supercritical Tradeoff

When one parameter is restricted, the other is pushed beyond worst-case.

Phenomenon observed primarily in proof complexity

A New Kind of Tradeoff

Recently, several works exhibited an extremely strong type of tradeoff

Supercritical Tradeoff

When one parameter is restricted, the other is pushed beyond worst-case.

Phenomenon observed primarily in proof complexity

- First observed by [BBI16] — supercritical size/space tradeoff for Resolution

A New Kind of Tradeoff

Recently, several works exhibited an extremely strong type of tradeoff

Supercritical Tradeoff

When one parameter is restricted, the other is pushed beyond worst-case.

Phenomenon observed primarily in proof complexity

- First observed by [BBI16] — supercritical size/space tradeoff for Resolution
- [Razborov16] proved a particularly strong tradeoff for tree-Resolution — there is an unsatisfiable CNF F such that any low width proof requires doubly exponential size

A New Kind of Tradeoff

Recently, several works exhibited an extremely strong type of tradeoff

Supercritical Tradeoff

When one parameter is restricted, the other is pushed beyond worst-case.

Phenomenon observed primarily in proof complexity

- First observed by [BBI16] — supercritical size/space tradeoff for Resolution
- [Razborov16] proved a particularly strong tradeoff for tree-Resolution — there is an unsatisfiable CNF F such that any low width proof requires doubly exponential size
- Several other size/space tradeoffs for various proof systems [R17,BN20,R18]

This Work

Supercritical Tradeoff

When one parameter is restricted, the other is pushed beyond worst-case.

This work: The first supercritical tradeoff between size and depth.

This Work

Supercritical Tradeoff

When one parameter is restricted, the other is pushed beyond worst-case.

This work: The first supercritical tradeoff between size and depth. For

- Resolution
- k-DNF Resolution
- Cutting Planes

This Work

Supercritical Tradeoff

When one parameter is restricted, the other is pushed beyond worst-case.

This work: The first supercritical tradeoff between size and depth. For

- Resolution — Focus on for today
- k-DNF Resolution
- Cutting Planes

Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

Given an unsatisfiable CNF formula F as a set of clauses

$$F = (x_2 \vee x_3)\,(\bar{x}_1 \vee \bar{x}_3)\,(\bar{x}_2)\,(x_1 \vee \neg x_3)$$

Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

Given an unsatisfiable CNF formula F as a set of clauses

Derive new clauses from old ones using:

Resolution rule:

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

$$(x_2 \vee x_3)\ (\bar{x}_1 \vee \bar{x}_3)\ (\bar{x}_2)\ (x_1 \vee \neg x_3)$$

Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

Given an unsatisfiable CNF formula F as a set of clauses

Derive new clauses from old ones using:

Resolution rule:

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

Π

$(x_2 \vee x_3)\ (\bar{x}_1 \vee \bar{x}_3)\ (\bar{x}_2)\ (x_1 \vee \bar{x}_3)$

Resolution

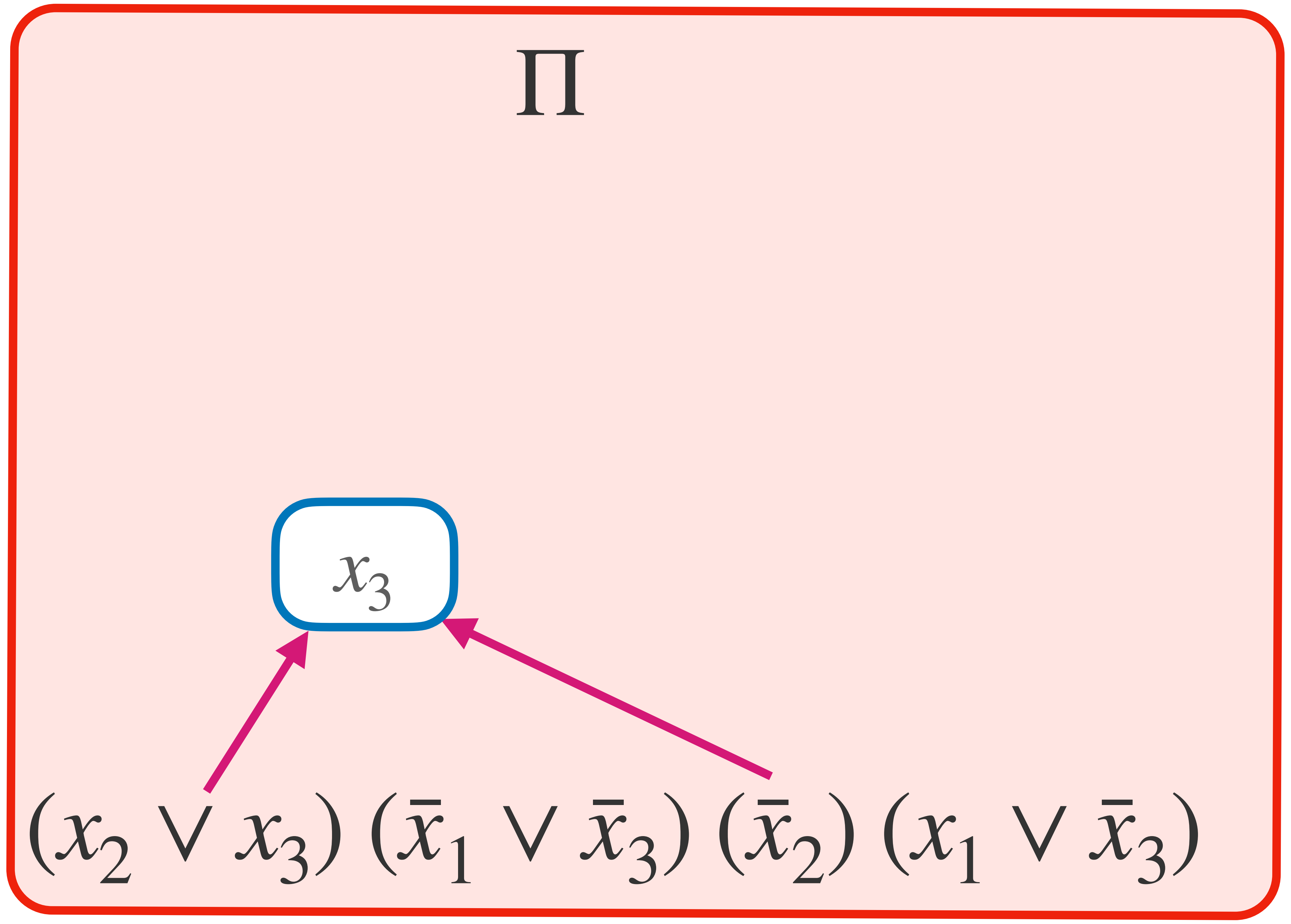
Resolution: A method for proving that a CNF formula is **unsatisfiable**

Given an unsatisfiable CNF formula F as a set of clauses

Derive new clauses from old ones using:

Resolution rule:

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$



Resolution

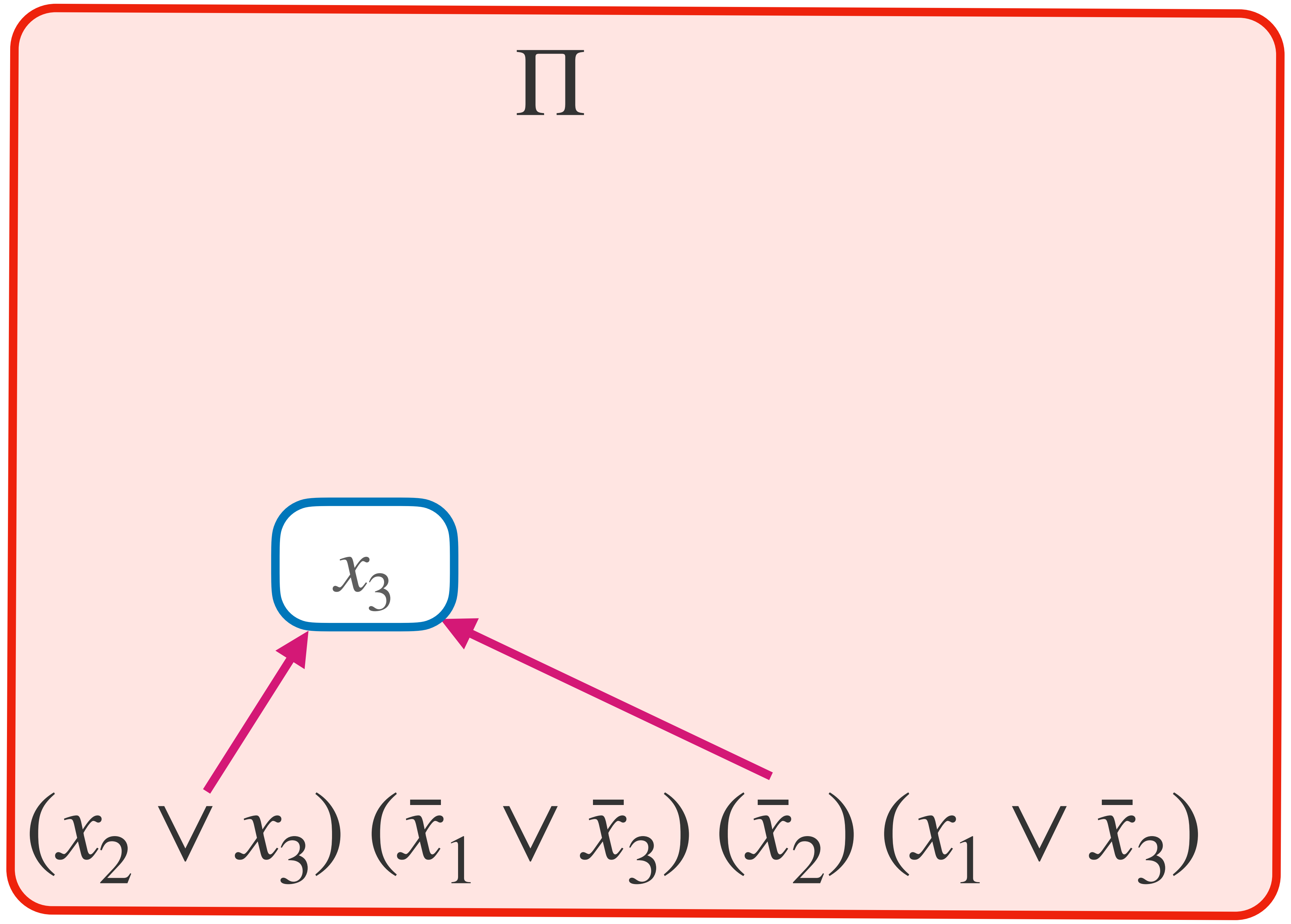
Resolution: A method for proving that a CNF formula is **unsatisfiable**

Given an unsatisfiable CNF formula F as a set of clauses

Derive new clauses from old ones using:

Resolution rule:

$$\frac{C_1 \lor x, \quad C_2 \lor \neg x}{C_1 \lor C_2}$$



Goal: Derive empty clause Λ

Resolution is sound $\implies$ F is unsatisfiable

Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

Given an unsatisfiable CNF formula F as a set of clauses

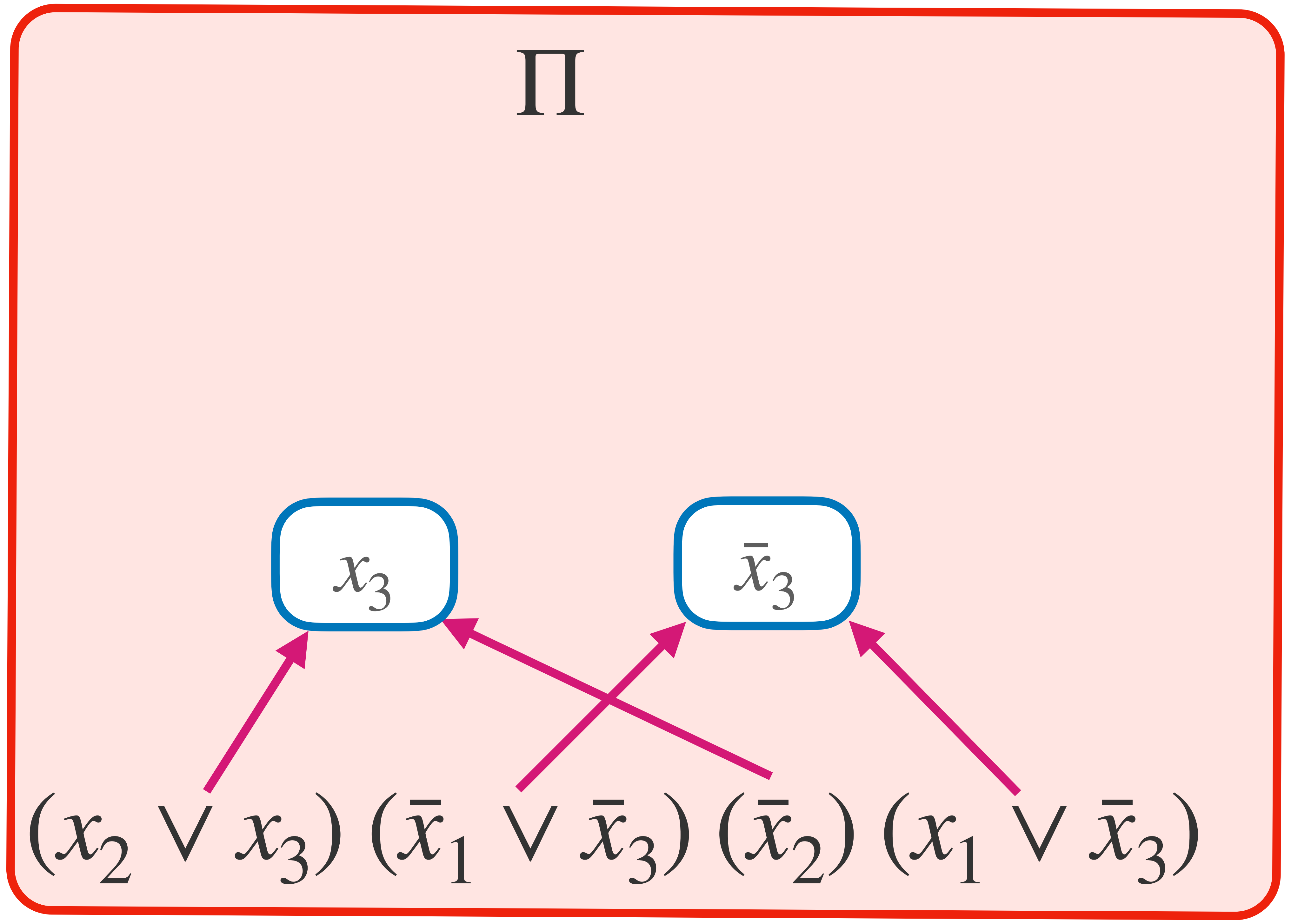
Derive new clauses from old ones using:

Resolution rule:

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

Goal: Derive empty clause Λ

Resolution is sound $\implies F$ is unsatisfiable



Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

Given an unsatisfiable CNF formula F as a set of clauses

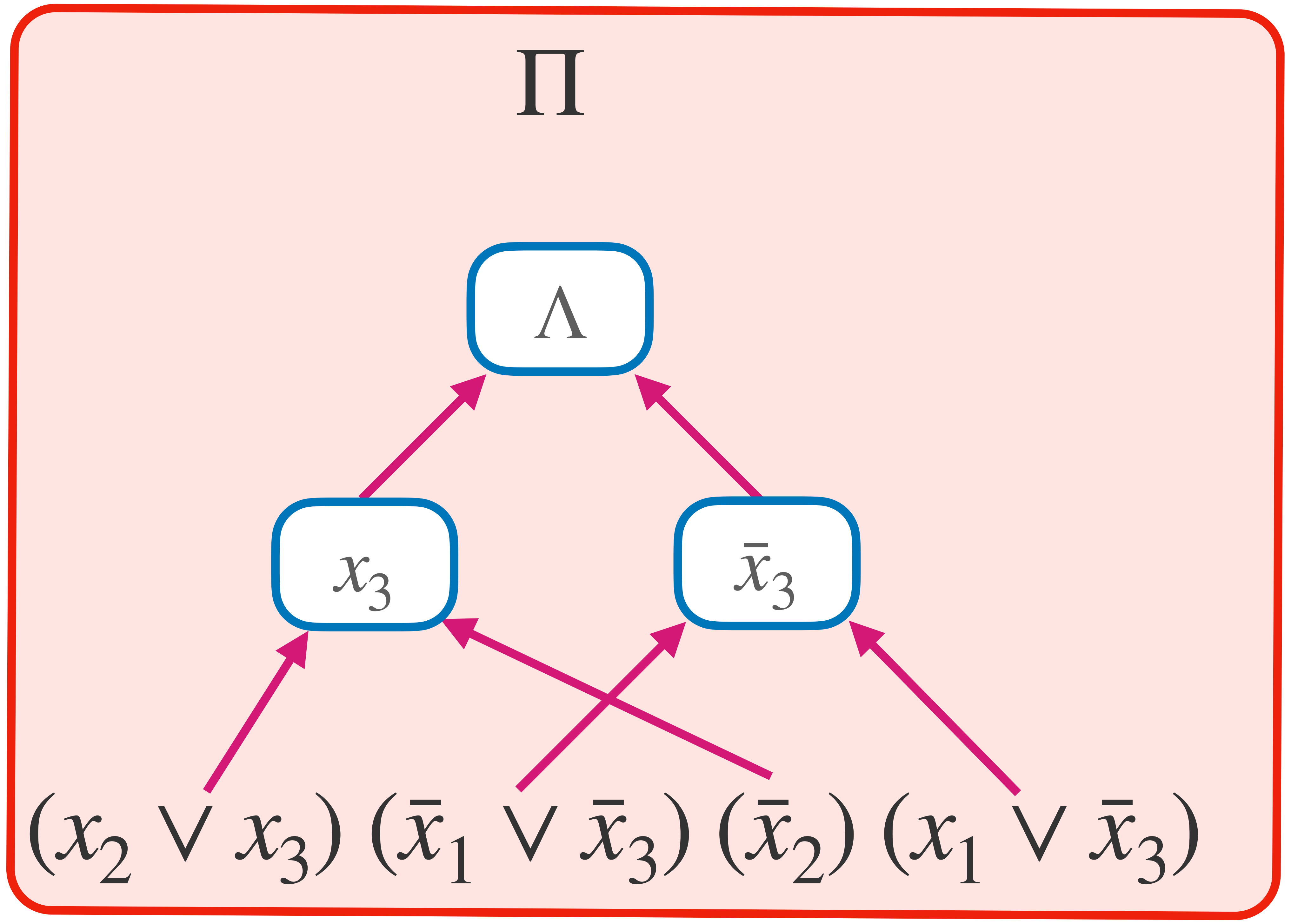
Derive new clauses from old ones using:

Resolution rule:

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

Goal: Derive empty clause Λ

Resolution is sound $\implies$ F is unsatisfiable



Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

Given an unsatisfiable CNF formula F as a set of clauses
Derive new clauses from old ones using:

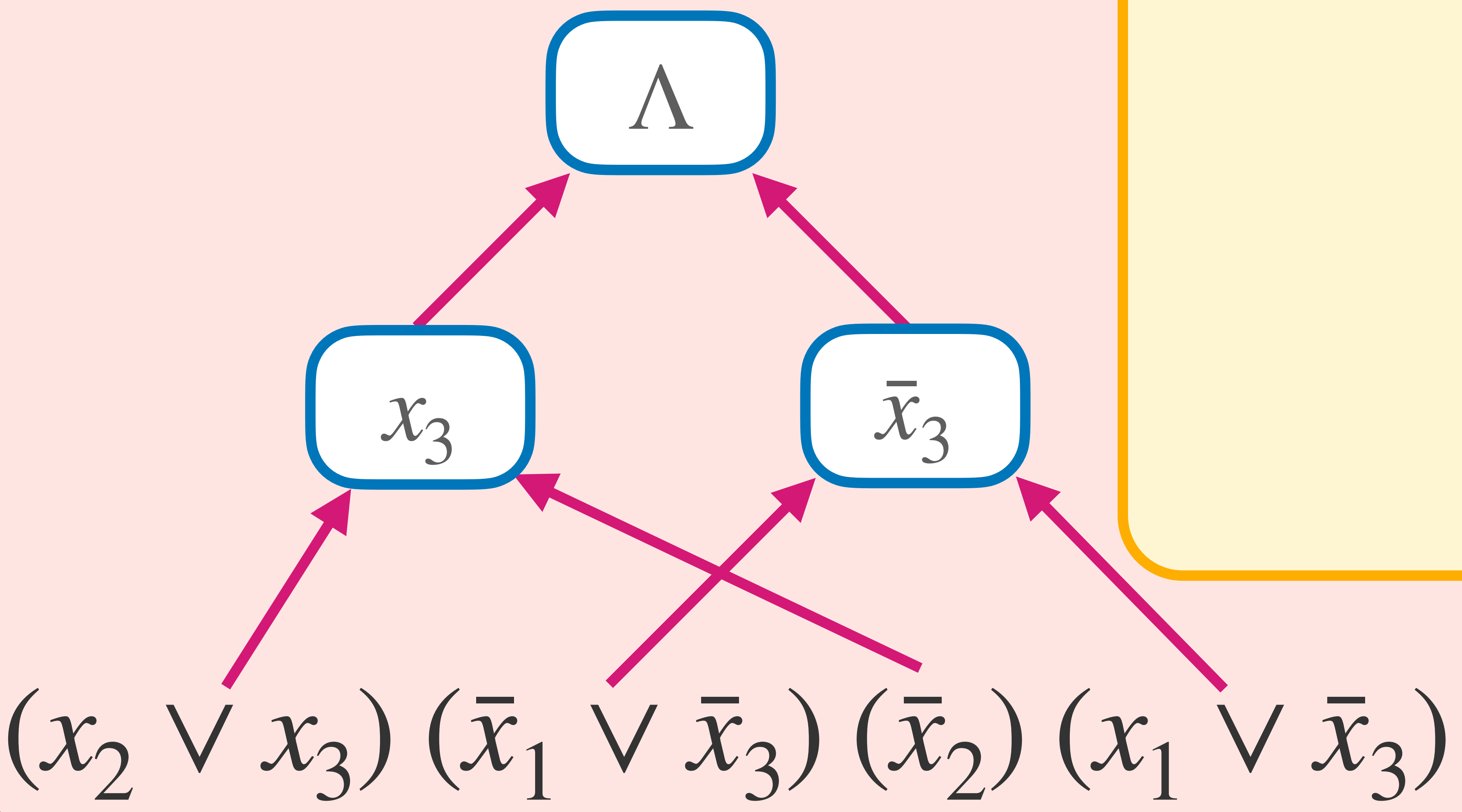
Resolution rule:

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

Goal: Derive empty clause Λ

Resolution is sound $\Longrightarrow$ F is unsatisfiable

Π



Parameters of proofs

size(Π): # of clauses (7)

Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

Given an unsatisfiable CNF formula F as a set of clauses

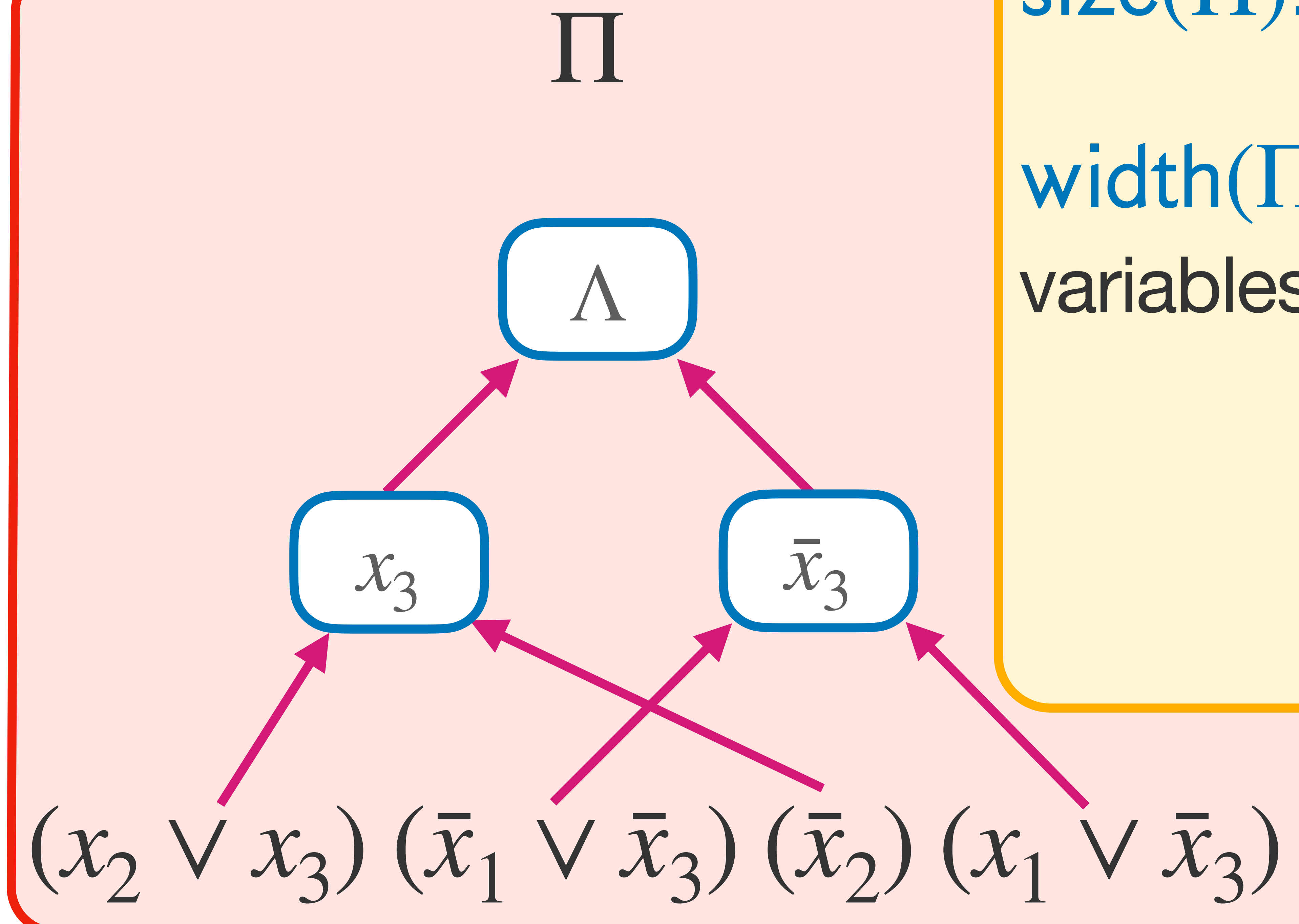
Derive new clauses from old ones using:

Resolution rule:

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

Goal: Derive empty clause Λ

Resolution is sound $\Longrightarrow$ F is unsatisfiable



Parameters of proofs

size(Π): # of clauses (7)

width(Π): max # of variables in any clause (2)

Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

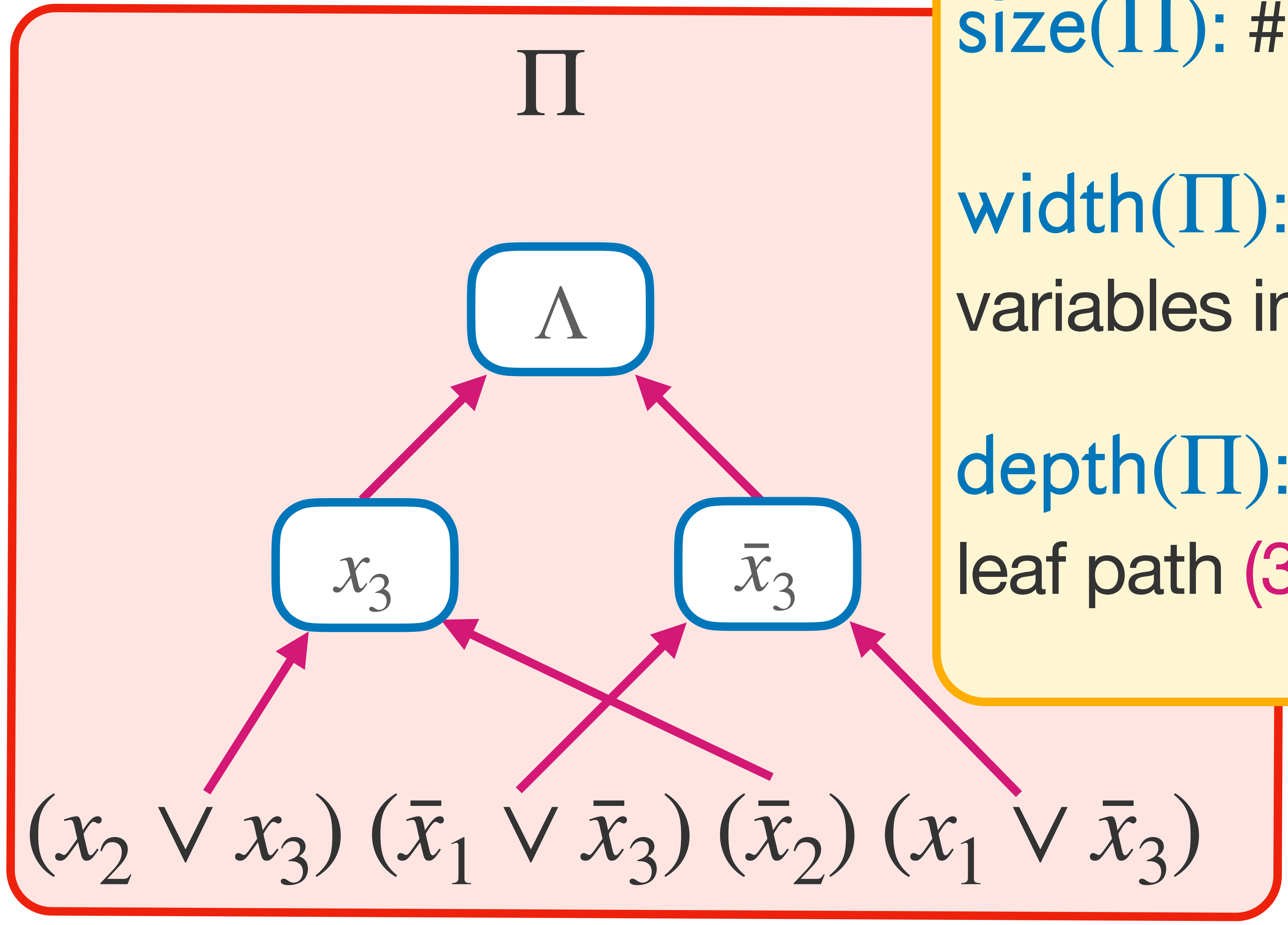
Given an unsatisfiable CNF formula F as a set of clauses
Derive new clauses from old ones using:

Resolution rule:

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

Goal: Derive empty clause Λ

Resolution is sound $\implies F$ is unsatisfiable



Parameters of proofs

size(Π): # of clauses (7)

width(Π): max # of variables in any clause (2)

depth(Π): longest root-to-leaf path (3)

Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

Given an unsatisfiable CNF formula F as a set of clauses

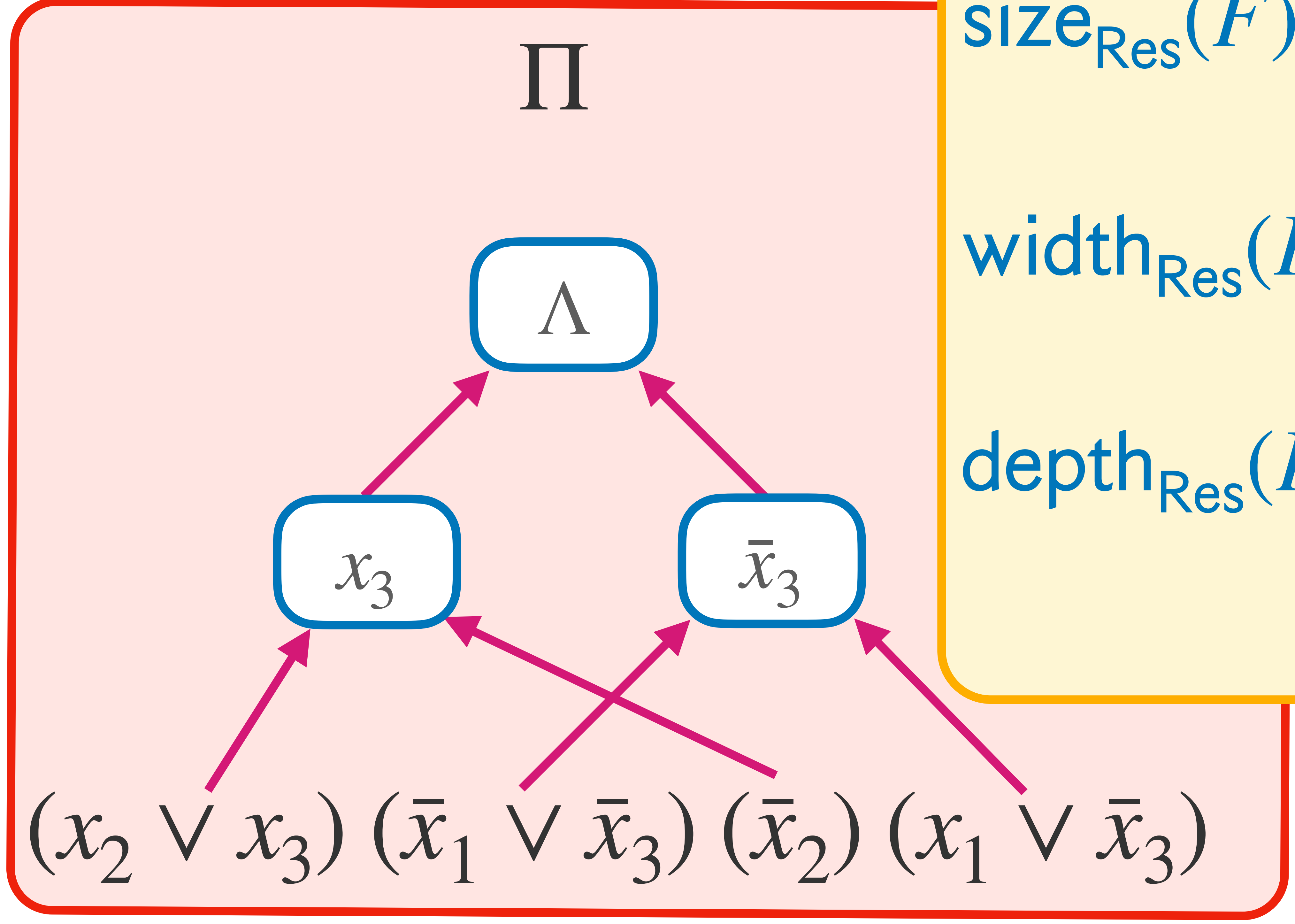
Derive new clauses from old ones using:

Resolution rule:

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

Goal: Derive empty clause Λ

Resolution is sound $\Longrightarrow$ F is unsatisfiable



Parameters of proofs

$\mathrm{size}_{\mathrm{Res}}(F) = \min_{\Pi} \mathrm{size}(\Pi)$

$\mathrm{width}_{\mathrm{Res}}(F) = \min_{\Pi} \mathrm{width}(\Pi)$

$\mathrm{depth}_{\mathrm{Res}}(F) = \min_{\Pi} \mathrm{depth}(\Pi)$

Depth

Like circuit depth, proof depth captures a notion of “parallelism” of a proof

Depth

Like circuit depth, proof depth captures a notion of “parallelism” of a proof

Resolution proofs capture the complexity of modern algorithms for SAT

Depth

Like circuit depth, proof depth captures a notion of "parallelism" of a proof

Resolution proofs capture the complexity of modern algorithms for SAT

→ Size lower bounds runtime

Depth

Like circuit depth, proof depth captures a notion of "parallelism" of a proof

Resolution proofs capture the complexity of modern algorithms for SAT

→ Size lower bounds runtime

→ Depth lower bounds parallelizability

Depth

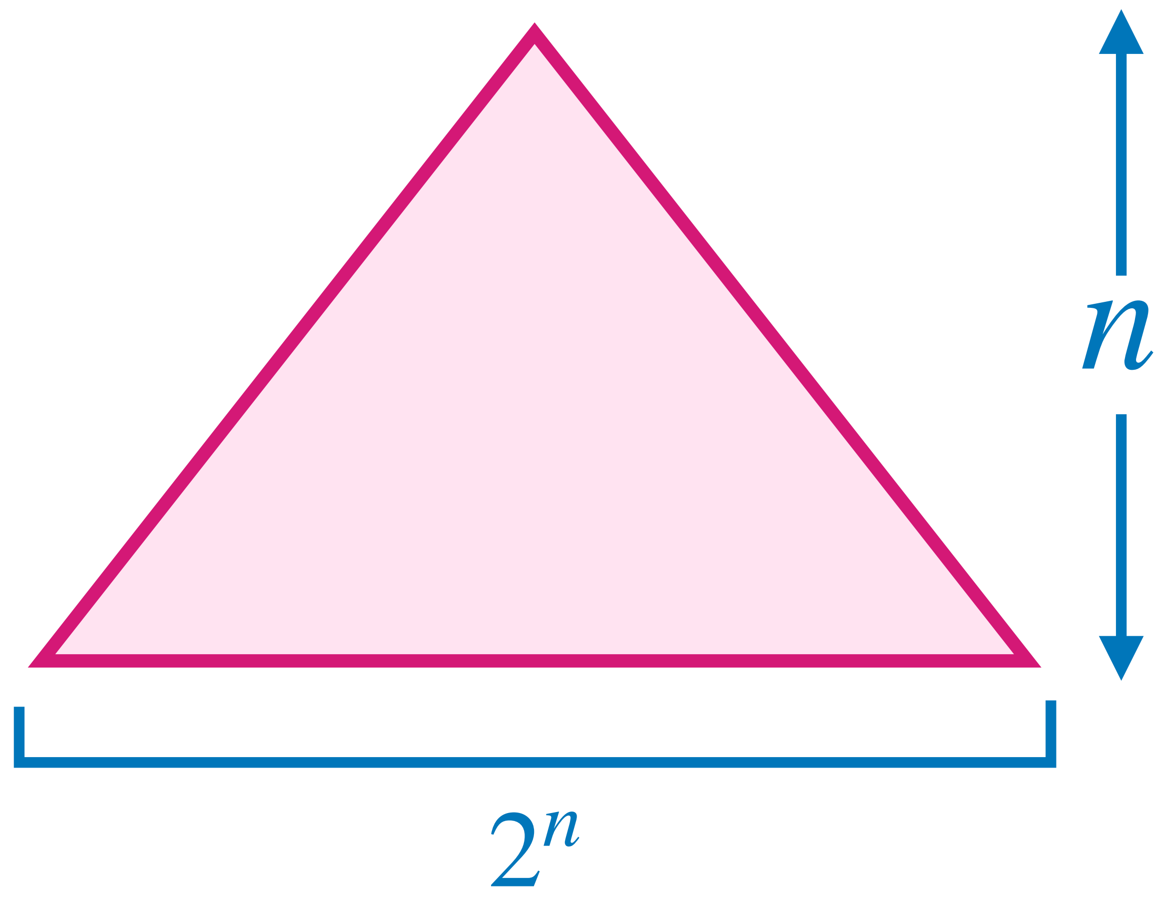
Like circuit depth, proof depth captures a notion of "parallelism" of a proof

Resolution proofs capture the complexity of modern algorithms for SAT

→ Size lower bounds runtime

→ Depth lower bounds parallelizability

There is always a depth n Resolution proof (but may have size 2^n)



Depth

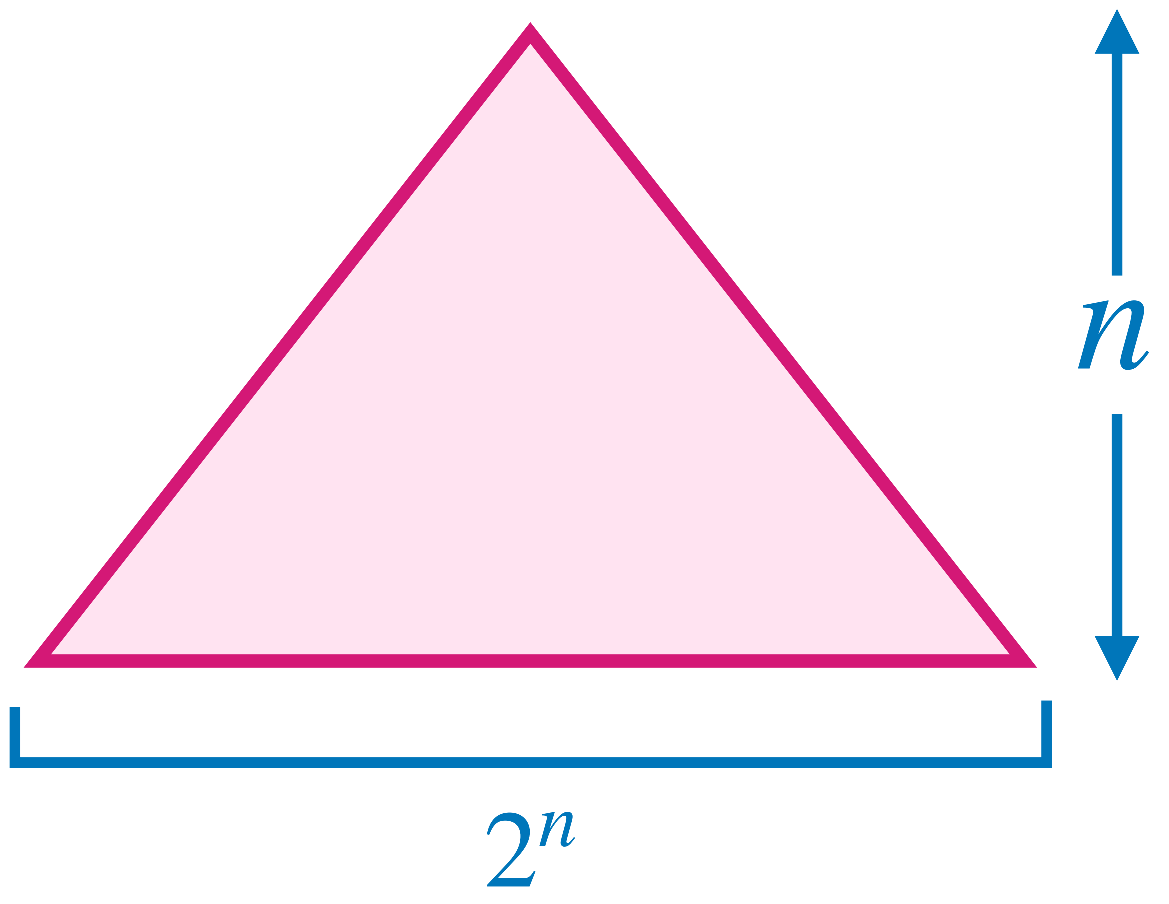
Like circuit depth, proof depth captures a notion of "parallelism" of a proof

Resolution proofs capture the complexity of modern algorithms for SAT

→ Size lower bounds runtime

→ Depth lower bounds parallelizability

There is always a depth n Resolution proof (but may have size 2^n)



Many strong proof systems can be balanced — depth is always at most log of the size

Depth

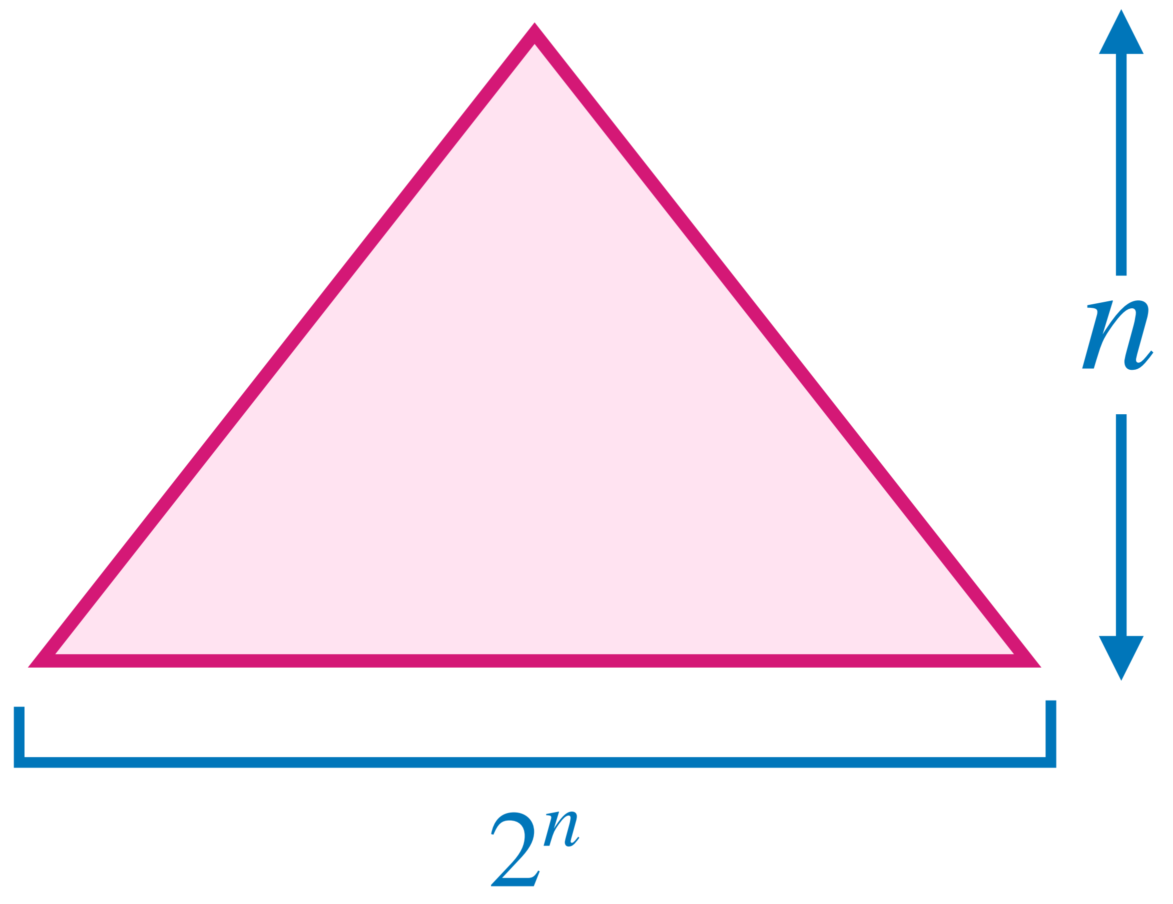
Like circuit depth, proof depth captures a notion of "parallelism" of a proof

Resolution proofs capture the complexity of modern algorithms for SAT

→ Size lower bounds runtime

→ Depth lower bounds parallelizability

There is always a depth n Resolution proof (but may have size 2^n)



Many strong proof systems can be balanced — depth is always at most log of the size

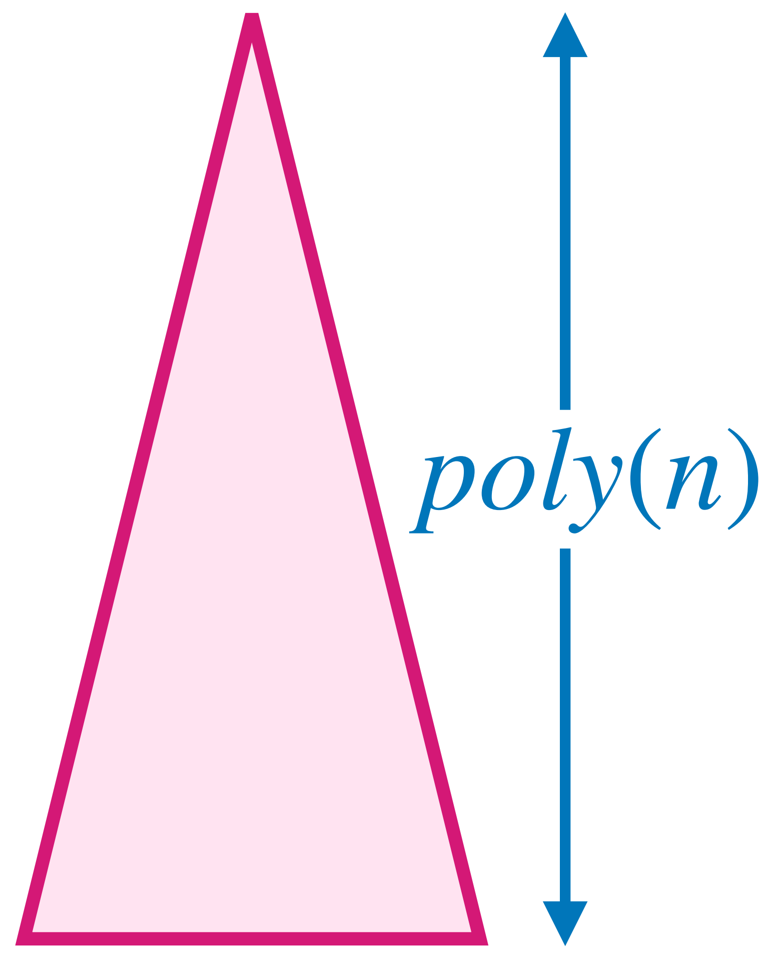
→ Resolution (Res(k), Cutting Planes) cannot always be balanced

This Work

For any $P \in \{$Resolution, Res(k), Cutting Planes$\}$

There is a CNF formula F on n variables such that

- There is a polynomial size P-proof of F
- Any subexponential-size P-proof of F must have poly$(n) > n$ depth

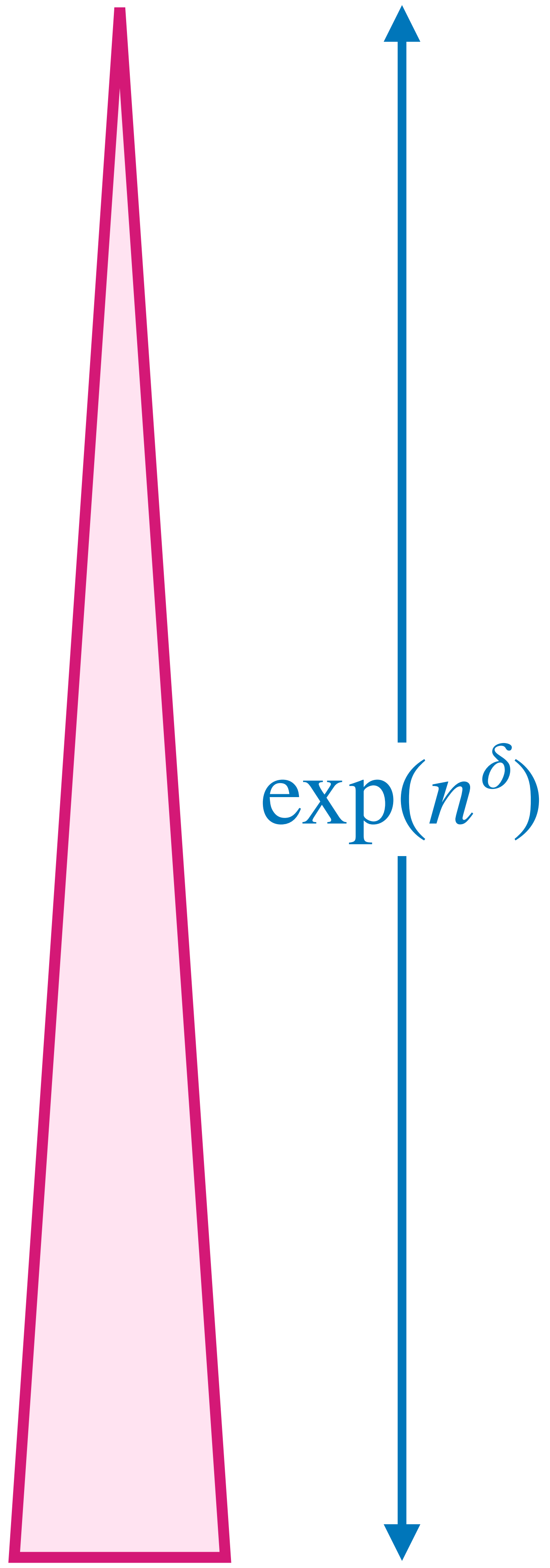


This Work

For any $P \in \{$Resolution, Res(k), Cutting Planes$\}$

There is a CNF formula F on n variables such that

- There is a weakly exponential size P-proof of F
- Any subexponential-size P-proof of F must have weakly exponential depth



This Work

Let $\varepsilon > 0$, let $c \geq 1$ be real-valued parameter that will control our tradeoff

Main Theorem (Res): There is a CNF formula F on n variables s.t.

This Work

Let $\varepsilon > 0$, let $c \geq 1$ be real-valued parameter that will control our tradeoff

Main Theorem (Res): There is a CNF formula F on n variables s.t.

1. There is a Resolution-proof of size $n^c \cdot 2^{O(c)}$

This Work

Let $\varepsilon > 0$, let $c \geq 1$ be real-valued parameter that will control our tradeoff

Main Theorem (Res): There is a CNF formula F on n variables s.t.

1. There is a Resolution-proof of size $n^c \cdot 2^{O(c)}$
2. If Π is a Resolution-proof with $\mathsf{size}(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ then

$$\mathsf{depth}(\Pi) \cdot \log \mathsf{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$$

This Work

Let $\varepsilon > 0$, let $c \geq 1$ be real-valued parameter that will control our tradeoff

Main Theorem (Res): There is a CNF formula F on n variables s.t.

1. There is a Resolution-proof of size $n^c \cdot 2^{O(c)}$
2. If Π is a Resolution-proof with $\mathsf{size}(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ then

$$\mathsf{depth}(\Pi) \cdot \log \mathsf{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$$

A tradeoff between runtime and parallelizability for CDCL

This Work

Let $\varepsilon > 0$, let $c \geq 1$ be real-valued parameter that will control our tradeoff

Main Theorem (Res): There is a CNF formula F on n variables s.t.

1. There is a Resolution-proof of size $n^c \cdot 2^{O(c)}$
2. If Π is a Resolution-proof with $\mathsf{size}(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ then

$$\mathsf{depth}(\Pi) \cdot \log \mathsf{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$$

A tradeoff between runtime and parallelizability for CDCL

* Caveat: F has $n^{O(c)}$ many clauses — we'll come back to this!

Proof Technique

Hardness Condensation

1. Find CNF formula F on N variables such that
   (a) F has small size proofs
   (b) F requires deep proofs

Proof Technique

Hardness Condensation

1. Find CNF formula F on N variables such that (e.g. pebbling formulas)
   (a) F has small size proofs — N
   (b) F requires deep proofs — $\Omega(N/\log N)$

Proof Technique

Hardness Condensation

1. Find CNF formula F on N variables such that (e.g. pebbling formulas)
   (a) F has small size proofs — N
   (b) F requires deep proofs — $\Omega(N/\log N)$

2. Compress the number of variables of F to $n \ll N$ while maintaining that (a) and (b) hold for any small size proof

Proof Technique

Hardness Condensation

1. Find CNF formula F on N variables such that (e.g. pebbling formulas)
   (a) F has small size proofs — N
   (b) F requires deep proofs — $\Omega(N/\log N)$

2. Compress the number of variables of F to $n \ll N$ while maintaining that (a) and (b) hold for any small size proof

Upshot: New F requires depth $\Omega(N/\log N)$ but only has n variables!
$\rightarrow$ If $n = o(N/\log N)$ we get supercritical depth lower bounds for small proofs!

Proof Technique

Hardness Condensation

1. Find CNF formula F on N variables such that (e.g. pebbling formulas)
   (a) F has small size proofs — N
   (b) F requires deep proofs — $\Omega(N/\log N)$

2. Compress the number of variables of F to $n \ll N$ while maintaining that (a) and (b) hold for any small size proof

Upshot: New F requires depth $\Omega(N/\log N)$ but only has n variables!
$\rightarrow$ If $n = o(N/\log N)$ we get supercritical depth lower bounds for small proofs!

How do we do compression?

Proof Technique

Hardness Condensation

1. Find CNF formula F on N variables such that (e.g. pebbling formulas)
   (a) F has small size proofs — N
   (b) F requires deep proofs — $\Omega(N/\log N)$

2. Compress the number of variables of F to $n \ll N$ while maintaining that (a) and (b) hold for any small size proof

Upshot: New F requires depth $\Omega(N/\log N)$ but only has n variables!
$\rightarrow$ If $n = o(N/\log N)$ we get supercritical depth lower bounds for small proofs!

How do we do compression? Lifting!

Lifting (Composition)

Composition is one of our most powerful tools for proving lower bounds

Lifting (Composition)

Composition is one of our most powerful tools for proving lower bounds

- Let $F(z_1, \dots, z_N) = C_1 \wedge \dots \wedge C_m$ be a CNF formula

Lifting (Composition)

Composition is one of our most powerful tools for proving lower bounds

- Let $F(z_1, \ldots, z_N) = C_1 \wedge \ldots \wedge C_m$ be a CNF formula
- Let $g : \{0,1\}^t \to \{0,1\}$ be a "gadget" function

Lifting (Composition)

Composition is one of our most powerful tools for proving lower bounds

- Let $F(z_1, \ldots, z_N) = C_1 \wedge \ldots \wedge C_m$ be a CNF formula
- Let $g : \{0,1\}^t \to \{0,1\}$ be a "gadget" function

The composed function is $F \circ g := F(g(\overrightarrow{x}_1), \ldots, g(\overrightarrow{x}_N))$

Lifting (Composition)

Composition is one of our most powerful tools for proving lower bounds

- Let $F(z_1, \ldots, z_N) = C_1 \wedge \ldots \wedge C_m$ be a CNF formula
- Let $g : \{0,1\}^t \to \{0,1\}$ be a "gadget" function

The composed function is $F \circ g := F(g(\vec{x}_1), \ldots, g(\vec{x}_N))$

Typically $\vec{x}_1, \ldots, \vec{x}_N$ are disjoint sets of variables

Lifting (Composition)

Composition is one of our most powerful tools for proving lower bounds

- Let $F(z_1, \ldots, z_N) = C_1 \wedge \ldots \wedge C_m$ be a CNF formula
- Let $g : \{0,1\}^t \to \{0,1\}$ be a "gadget" function

The composed function is $F \circ g := F(g(\overrightarrow{x}_1), \ldots, g(\overrightarrow{x}_N))$

Typically $\overrightarrow{x}_1, \ldots, \overrightarrow{x}_N$ are disjoint sets of variables

Let P, Q be two proof systems

A lifting theorem relates the complexity of

- P-proofs of F
- Q-proofs of $F \circ g$

Lifting (Composition)

Simple Example: $g = \mathsf{XOR}_2$ then $F \circ \mathsf{XOR}_2 := F(x_1 \oplus x'_1, \ldots, x_N \oplus x'_N)$

Lifting (Composition)

Simple Example: $g = \mathrm{XOR}_2$ then $F \circ \mathrm{XOR}_2 := F(x_1 \oplus x_1', \ldots, x_N \oplus x_N')$

Width-to-Size Lifting Theorem: Let F be any unsatisfiable formula. If Π is a resolution proof of $F \circ \mathrm{XOR}_2$ then

$$\mathrm{size}(\Pi) \geq 2^{\Omega(\mathrm{width}_{\mathrm{Res}}(F))}$$

Lifting (Composition)

Simple Example: $g = \text{XOR}_2$ then $F \circ \text{XOR}_2 := F(x_1 \oplus x_1', \ldots, x_N \oplus x_N')$

Width-to-Size Lifting Theorem: Let F be any unsatisfiable formula. If Π is a resolution proof of $F \circ \text{XOR}_2$ then

$$\text{size}(\Pi) \geq 2^{\Omega(\text{width}_{\text{Res}}(F))}$$
$$\text{depth}(\Pi) \geq \text{depth}_{\text{Res}}(F)$$

Lifting (Composition)

Simple Example: $g = \mathsf{XOR}_2$ then $F \circ \mathsf{XOR}_2 := F(x_1 \oplus x_1', \ldots, x_N \oplus x_N')$

Width-to-Size Lifting Theorem: Let F be any unsatisfiable formula. If Π is a resolution proof of $F \circ \mathsf{XOR}_2$ then

$$\mathsf{size}(\Pi) \geq 2^{\Omega(\mathsf{width}_{\mathsf{Res}}(F))}$$

$$\mathsf{depth}(\Pi) \geq \mathsf{depth}_{\mathsf{Res}}(F)$$

- $P =$ Resolution, $Q =$ Resolution

Lifting (Composition)

Simple Example: $g = \mathsf{XOR}_2$ then $F \circ \mathsf{XOR}_2 := F(x_1 \oplus x_1', \ldots, x_N \oplus x_N')$

Width-to-Size Lifting Theorem: Let F be any unsatisfiable formula. If Π is a resolution proof of $F \circ \mathsf{XOR}_2$ then

$$\mathsf{size}(\Pi) \geq 2^{\Omega(\mathsf{width}_{\mathsf{Res}}(F))}$$

$$\mathsf{depth}(\Pi) \geq \mathsf{depth}_{\mathsf{Res}}(F)$$

- $P =$ Resolution, $Q =$ Resolution

If F has a proof of size s and width $w \Longrightarrow F \circ \mathsf{XOR}_2$ has a proof of size $O(s2^w)$

Lifting (Composition)

Simple Example: $g = \mathrm{XOR}_2$ then $F \circ \mathrm{XOR}_2 := F(x_1 \oplus x_1', \ldots, x_N \oplus x_N')$

Width-to-Size Lifting Theorem: Let F be any unsatisfiable formula. If Π is a resolution proof of $F \circ \mathrm{XOR}_2$ then

$$\mathsf{size}(\Pi) \geq 2^{\Omega(\mathsf{width}_{\mathsf{Res}}(F))}$$

$$\mathsf{depth}(\Pi) \geq \mathsf{depth}_{\mathsf{Res}}(F)$$

- $P =$ Resolution, $Q =$ Resolution

If F has a proof of size s and width $w \Longrightarrow F \circ \mathrm{XOR}_2$ has a proof of size $O(s2^w)$

$\rightarrow$ Locally simulate the XOR in every step of the proof of F

Lifting (Composition)

Simple Example: $g = \text{XOR}_2$ then $F \circ \text{XOR}_2 := F(x_1 \oplus x_1', \ldots, x_N \oplus x_N')$

Width-to-Size Lifting Theorem: Let F be any unsatisfiable formula. If Π is a resolution proof of $F \circ \text{XOR}_2$ then

$$\text{size}(\Pi) \geq 2^{\Omega(\text{width}_{\text{Res}}(F))}$$

$$\text{depth}(\Pi) \geq \text{depth}_{\text{Res}}(F)$$

- $P =$ Resolution, $Q =$ Resolution

If F has a proof of size s and width $w \implies F \circ \text{XOR}_2$ has a proof of size $O(s2^w)$

$\rightarrow$ Locally simulate the XOR in every step of the proof of F

$\implies$ Naively simulation is essentially the best! (A theme of lifting theorems)

Lifting (Composition)

Typically

- P is a “weak” proof system
- Q is a “strong” proof system

A lifting theorem shows that the most efficient Q-proof of $F \circ g$ is to simulate the most efficient P-proof of F (with some extra overhead to handle g)

Our Lifting

Does the opposite!

Our Lifting

Does the opposite! — Lifts depth lower bounds on a strong proof system to (much stronger) depth lower bounds on weak proof system

Our Lifting

Does the opposite! — Lifts depth lower bounds on a strong proof system to (much stronger) depth lower bounds on weak proof system

- P is Resolution
- Q is size-bounded Resolution

Our Lifting

Does the opposite! — Lifts depth lower bounds on a strong proof system to (much stronger) depth lower bounds on weak proof system

- P is Resolution
- Q is size-bounded Resolution

Proof Idea:

Find a gadget g such that

Our Lifting

Does the opposite! — Lifts depth lower bounds on a strong proof system to (much stronger) depth lower bounds on weak proof system

- P is Resolution
- Q is size-bounded Resolution

Proof Idea:

Find a gadget g such that

1. The number of variables n of $F \circ g$ will be much smaller than N

Our Lifting

Does the opposite! — Lifts depth lower bounds on a strong proof system to (much stronger) depth lower bounds on weak proof system

- P is Resolution
- Q is size-bounded Resolution

Proof Idea:

Find a gadget g such that

1. The number of variables n of $F \circ g$ will be much smaller than N
2. Any small-size Resolution proof of $F \circ g$ will require the same depth as proving F

The Gadget

Our gadget will be the XOR function

$F(\mathrm{XOR}(\overrightarrow{x}_1), \ldots, \mathrm{XOR}(\overrightarrow{x}_N))$

The Gadget

Our gadget will be the XOR function

$F(\text{XOR}(\overrightarrow{x}_1), \ldots, \text{XOR}(\overrightarrow{x}_N))$ … With a **twist**!

The variable sets $\overrightarrow{x}_1, \ldots, \overrightarrow{x}_N$ will no longer be disjoint!

The Gadget

Our gadget will be the XOR function

$F(\text{XOR}(\overrightarrow{x}_1), \ldots, \text{XOR}(\overrightarrow{x}_N))$ … With a **twist**!

The variable sets $\overrightarrow{x}_1, \ldots, \overrightarrow{x}_N$ will no longer be disjoint!

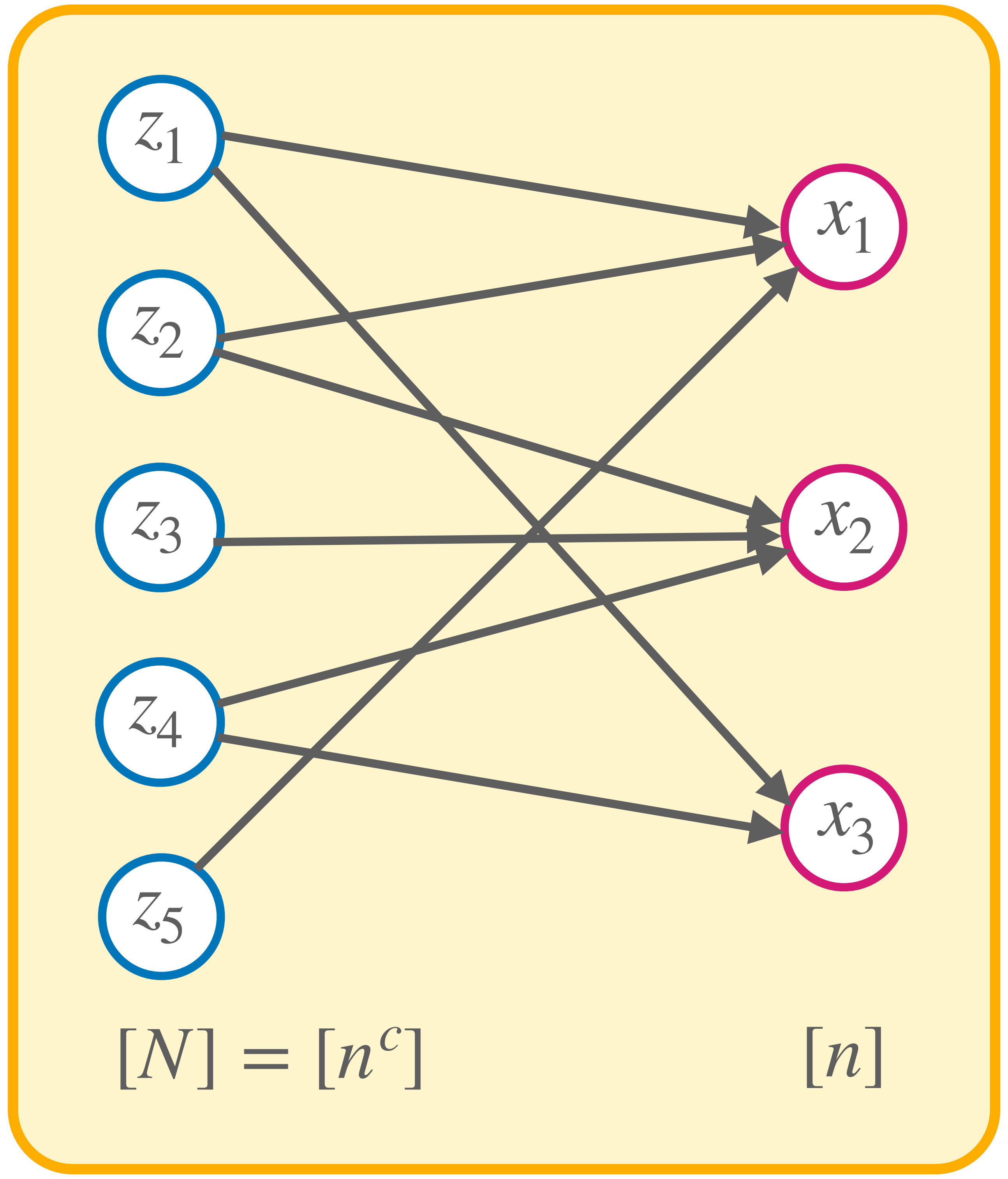
$\rightarrow$ Composing will reduce the total number of variables to $n \ll N$

The Gadget

Let G be an $N \times n$ bipartite graph

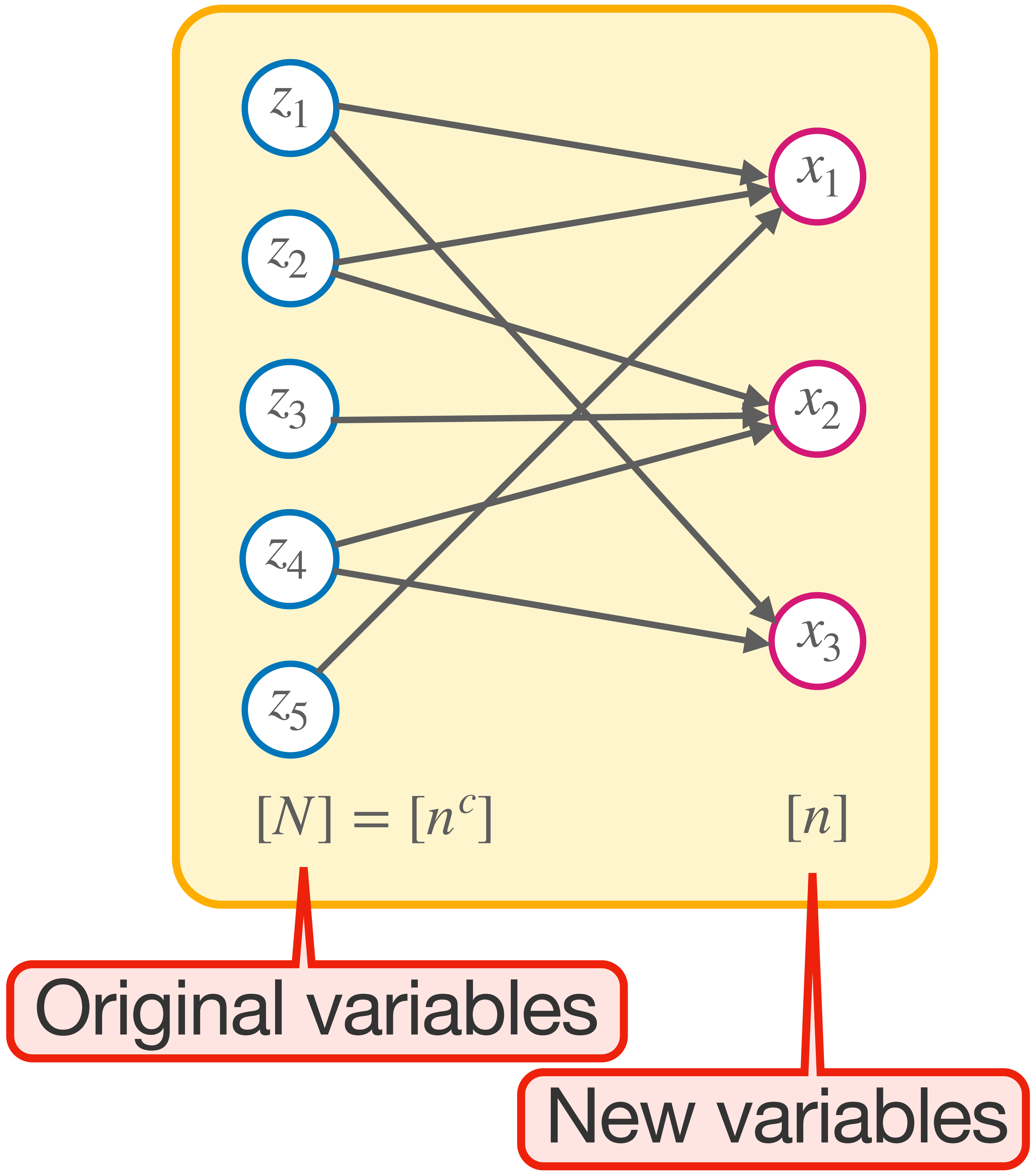
The Gadget

Let G be an $N \times n$ bipartite graph



The Gadget

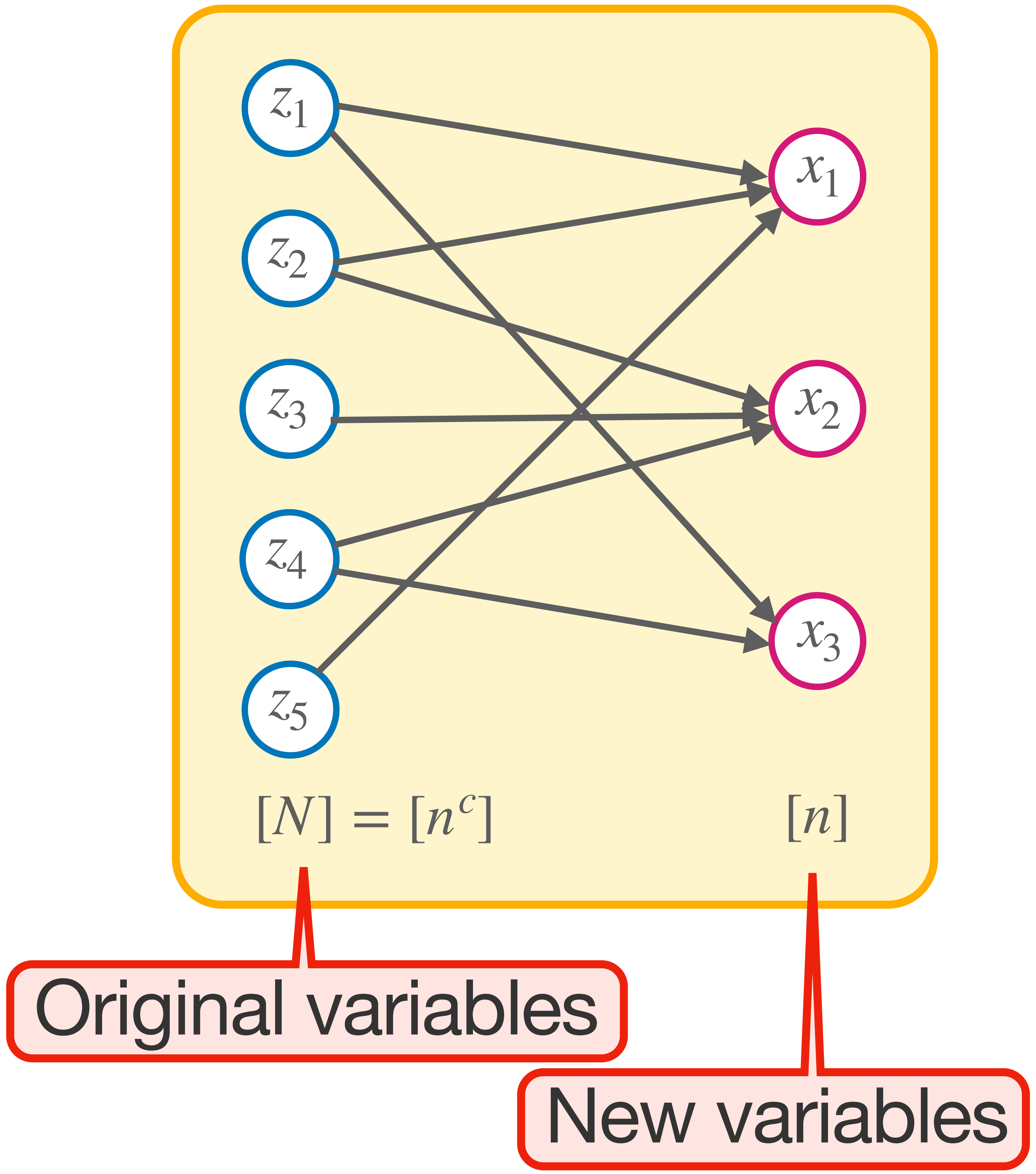
Let G be an $N \times n$ bipartite graph



The Gadget

Let G be an $N \times n$ bipartite graph

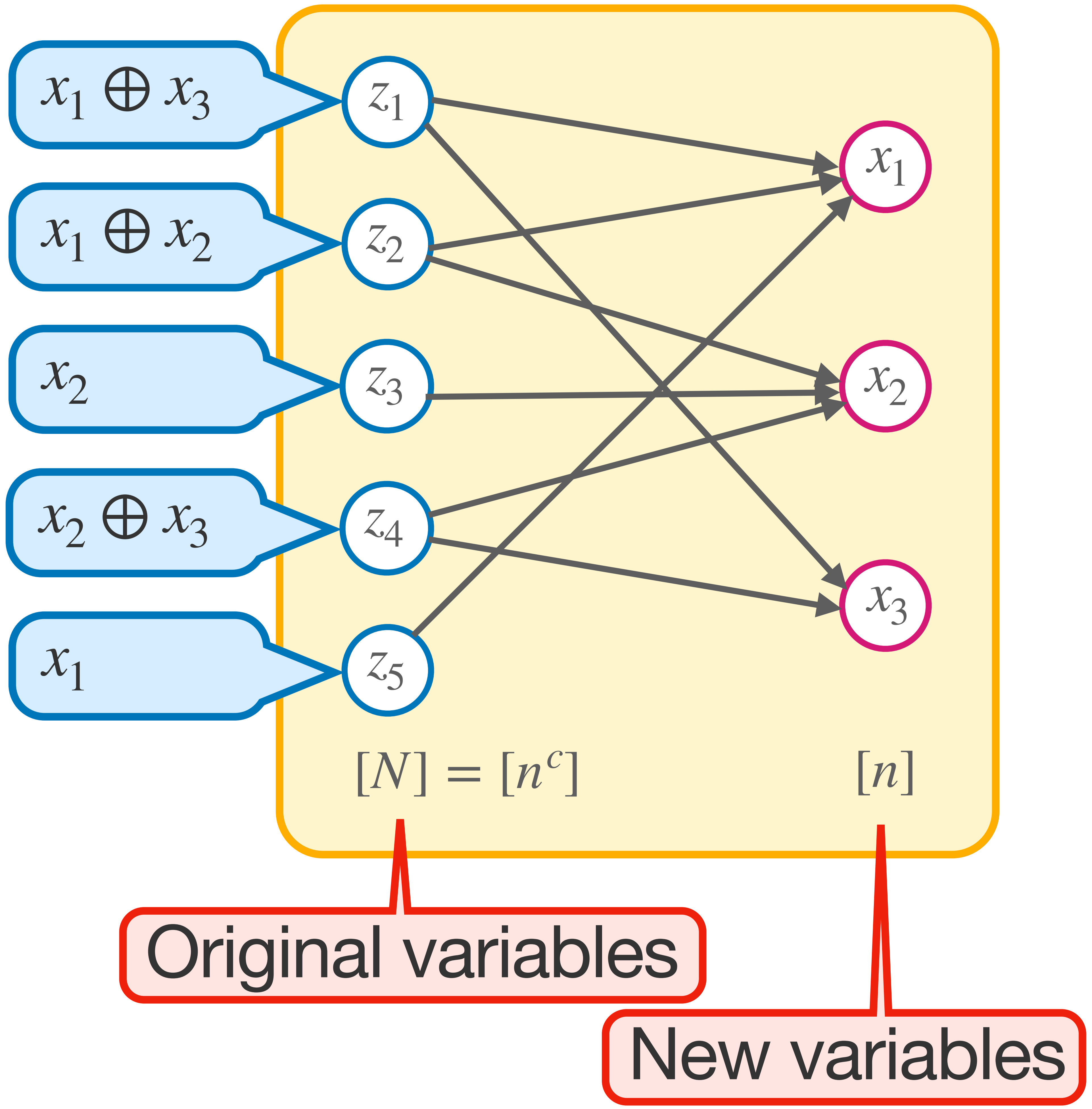
$F \circ \mathsf{XOR}_G$ replaces $z_i \mapsto \bigoplus_{x_j \in \mathsf{N}(z_i)} x_j$



The Gadget

Let G be an $N \times n$ bipartite graph

$F \circ \mathrm{XOR}_G$ replaces $z_i \mapsto \bigoplus_{x_j \in N(z_i)} x_j$

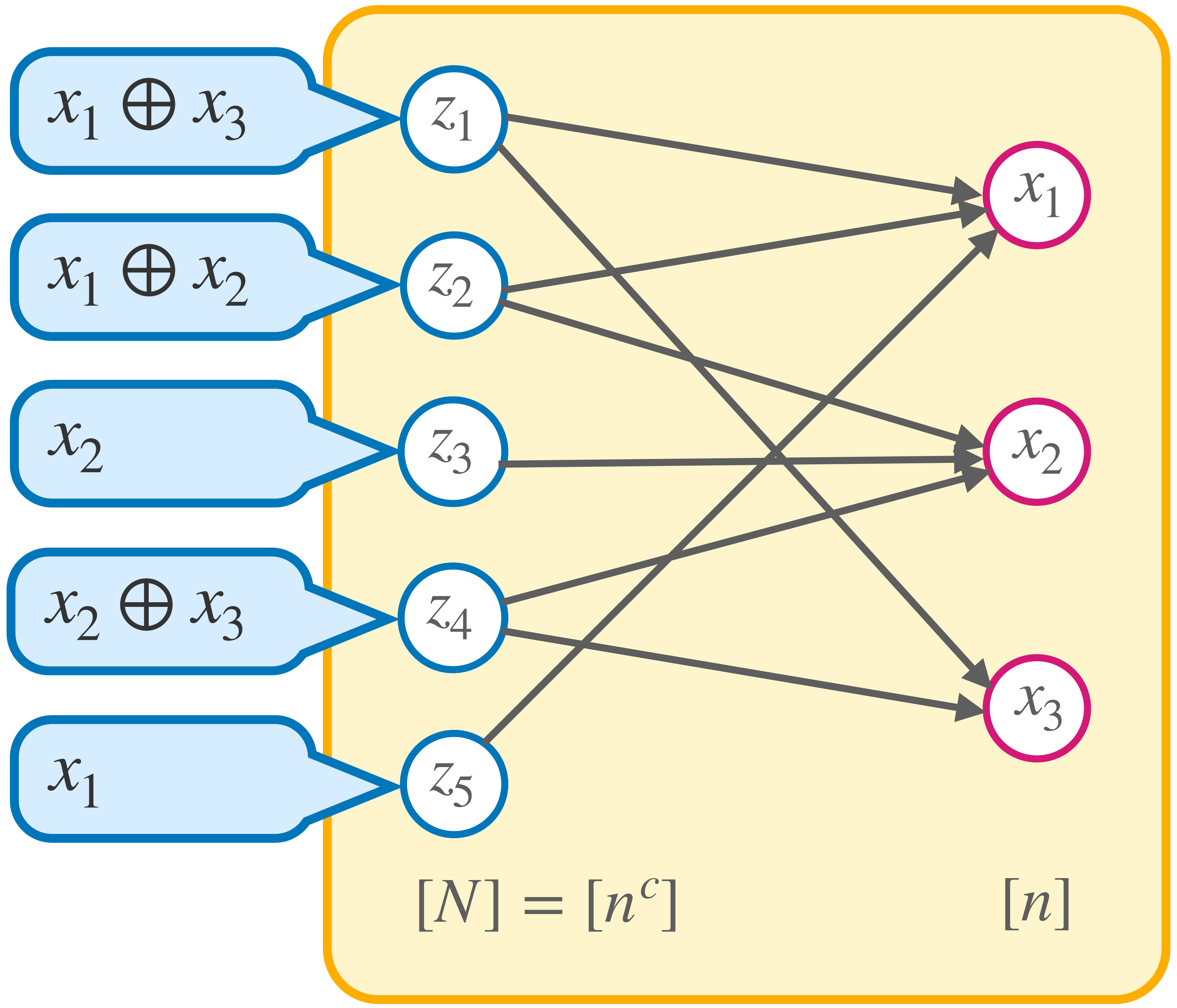


The Gadget

Let G be an $N \times n$ bipartite graph

$F \circ \mathsf{XOR}_G$ replaces $z_i \mapsto \bigoplus_{x_j \in \mathsf{N}(z_i)} x_j$

E.g. $((z_1 \vee \neg z_2) \wedge z_5) \circ \mathsf{XOR}_G$



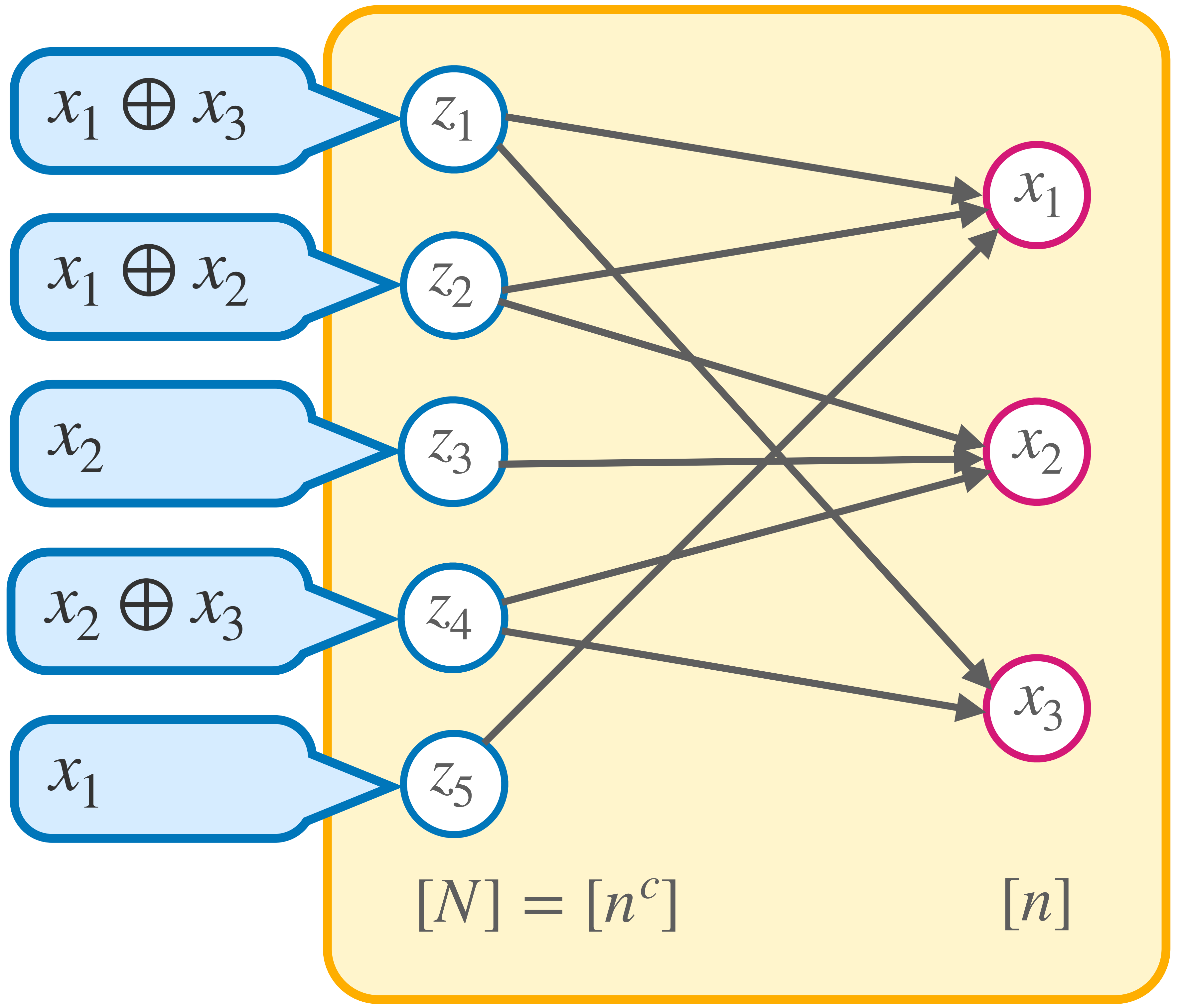
The Gadget

Let G be an $N \times n$ bipartite graph

$F \circ \mathrm{XOR}_G$ replaces $z_i \mapsto \bigoplus_{x_j \in \mathrm{N}(z_i)} x_j$

E.g. $((z_1 \vee \neg z_2) \wedge z_5) \circ \mathrm{XOR}_G$

$$\big((x_1 \oplus x_3) \vee \neg(x_1 \oplus x_2)\big) \wedge x_1$$

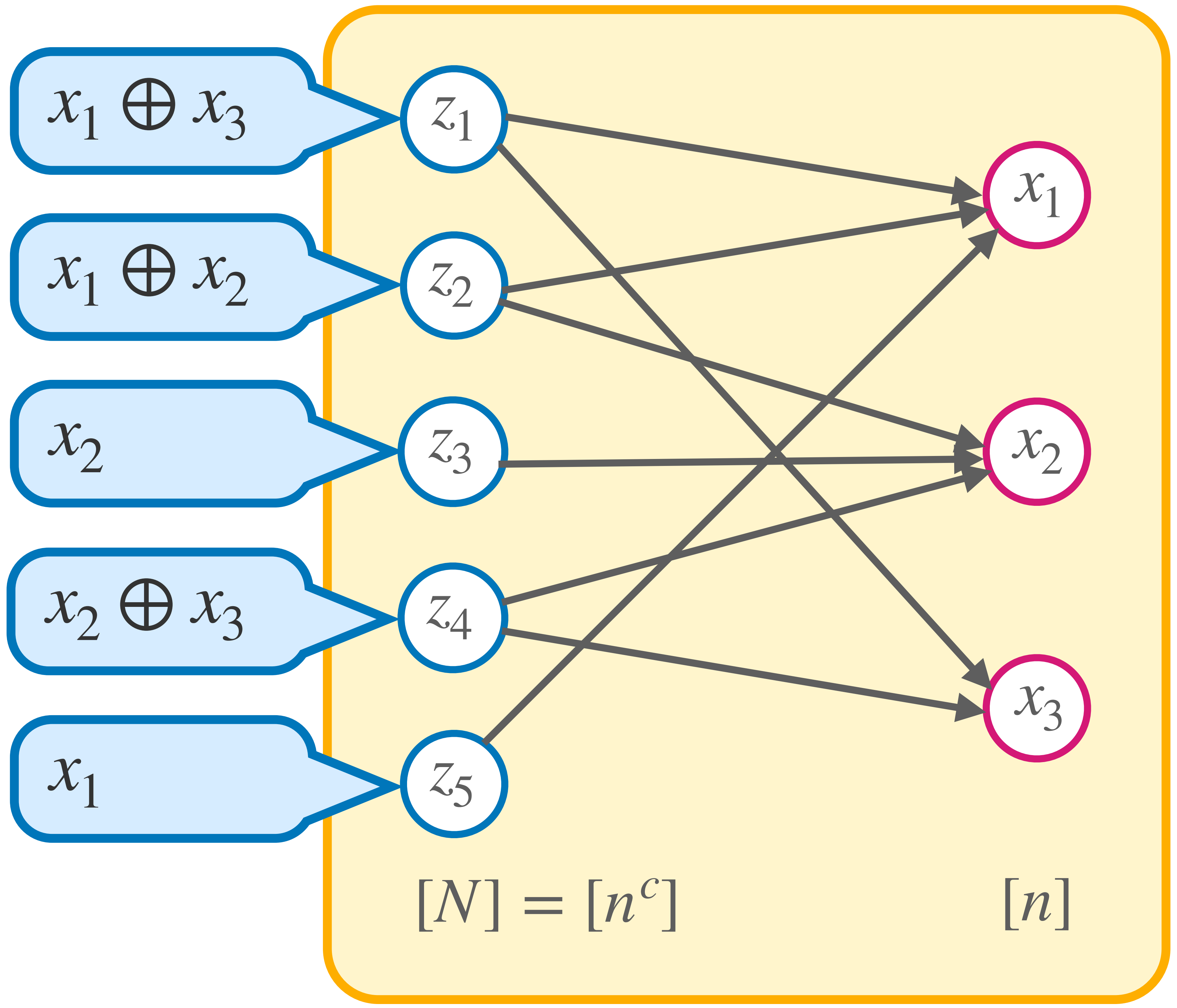


The Gadget

Let G be an $N \times n$ bipartite graph

$F \circ \mathrm{XOR}_G$ replaces $z_i \mapsto \bigoplus_{x_j \in N(z_i)} x_j$

Idea: If the edges of G are sufficiently "spread out"



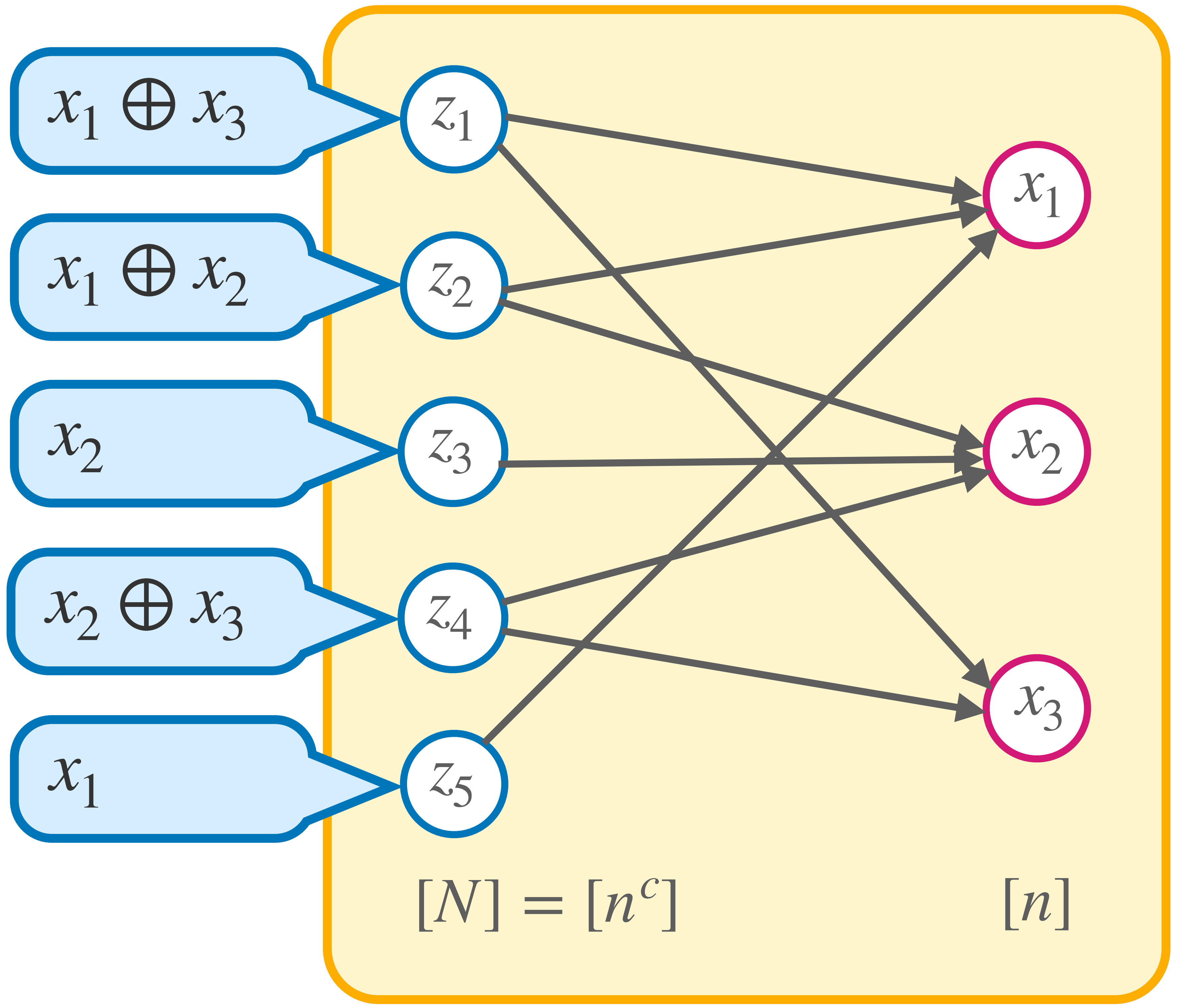
The Gadget

Let G be an $N \times n$ bipartite graph

$F \circ \mathsf{XOR}_G$ replaces $z_i \mapsto \bigoplus_{x_j \in \mathsf{N}(z_i)} x_j$

Idea: If the edges of G are sufficiently "spread out"

$\rightarrow$ learning the value of one XOR won't reveal much information about any other XOR

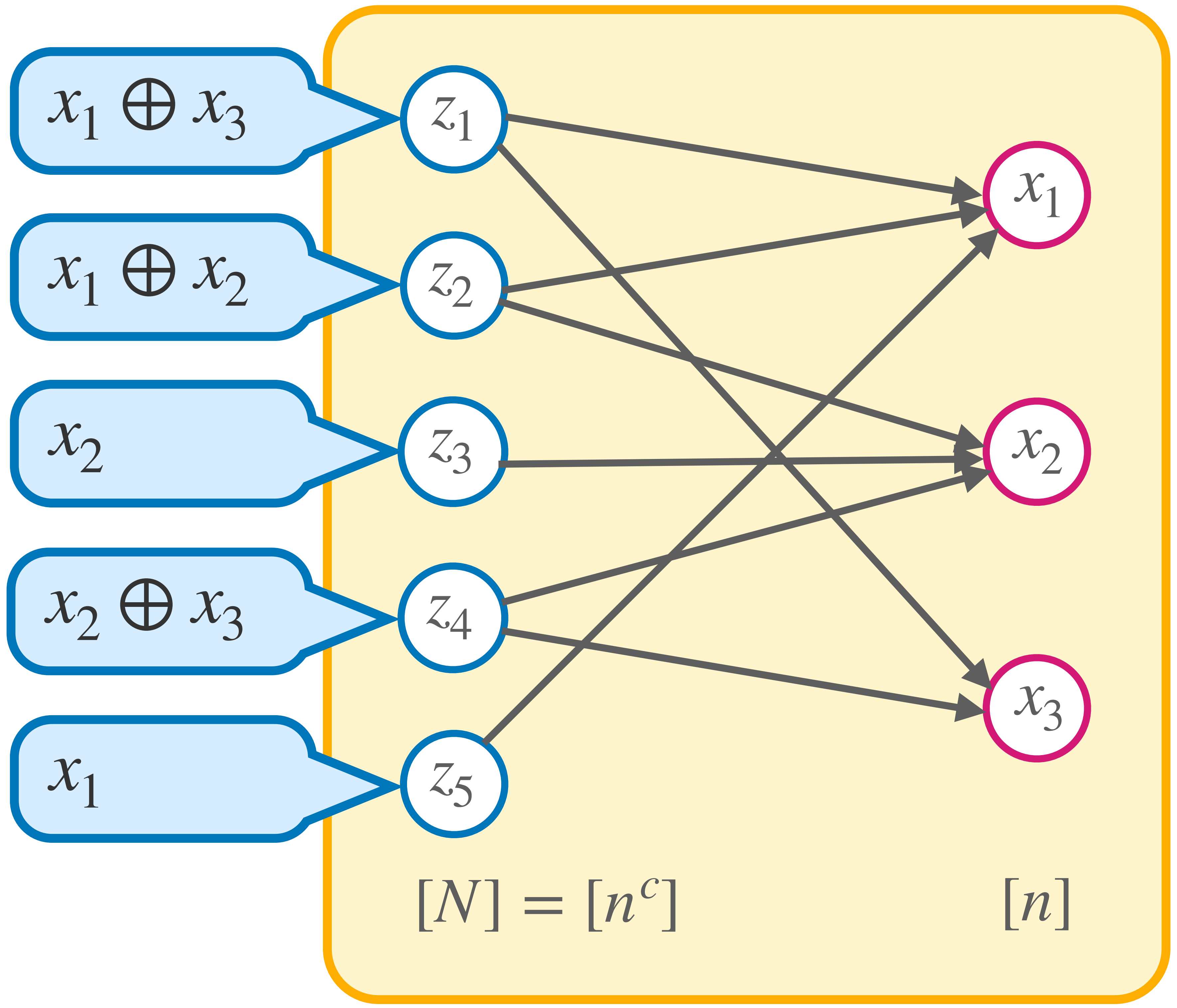


The Gadget

Let G be an $N \times n$ bipartite graph

$F \circ \mathrm{XOR}_G$ replaces $z_i \mapsto \bigoplus_{x_j \in N(z_i)} x_j$

Idea: If the edges of G are sufficiently "spread out"
$\rightarrow$ learning the value of one XOR won't reveal much information about any other XOR
$\rightarrow$ The best Resolution proof of $F \circ \mathrm{XOR}_G$ should essentially be to simulate the best proof of F



The Gadget

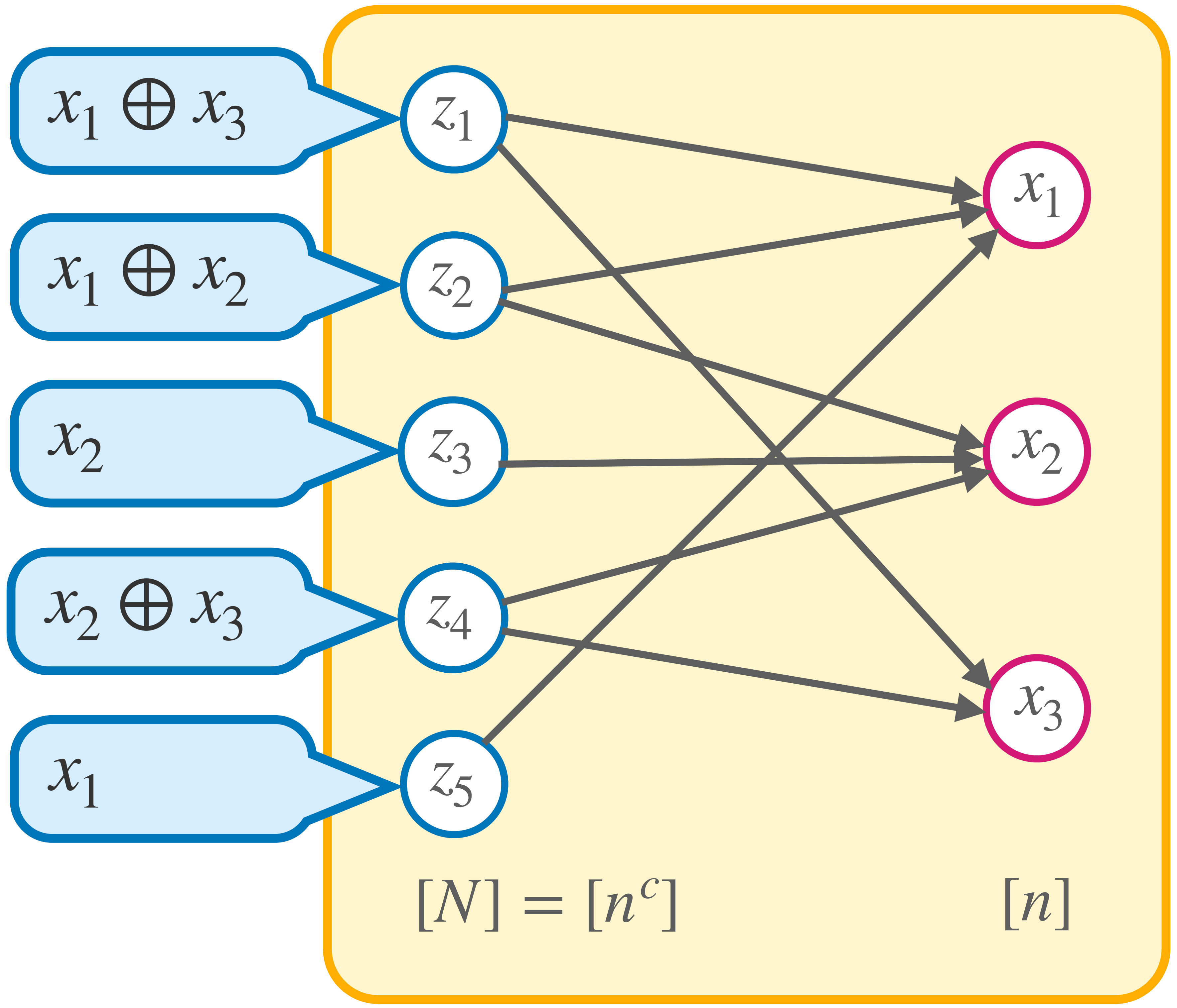
Let G be an $N \times n$ bipartite graph

$F \circ \mathrm{XOR}_G$ replaces $z_i \mapsto \bigoplus_{x_j \in N(z_i)} x_j$

Idea: If G is sufficiently expanding:

→ learning the value of one XOR won't reveal much information about any other XOR

→ The best Resolution proof of $F \circ \mathrm{XOR}_G$ should essentially be to simulate the best proof of F

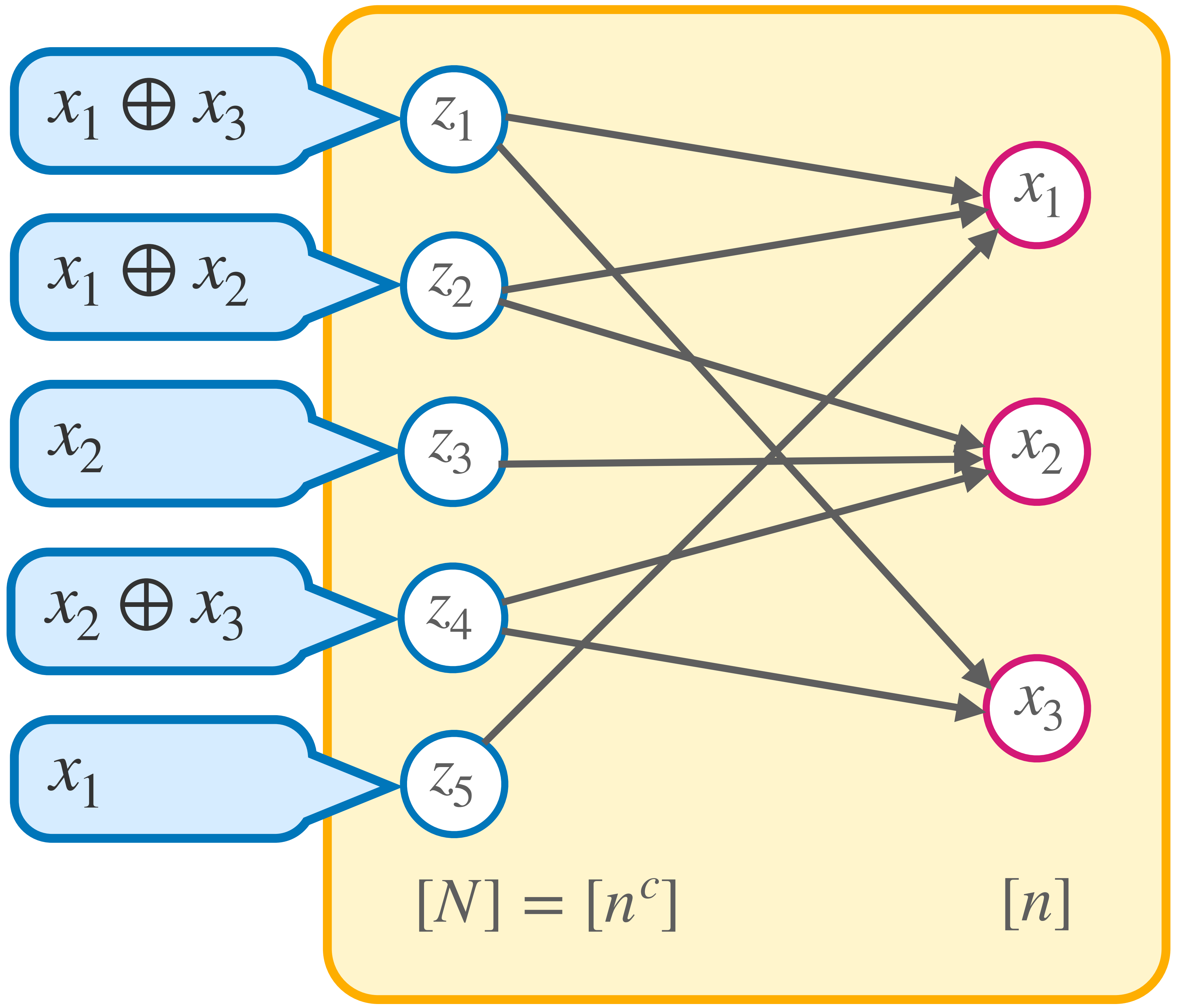


r-Expanding: For any set $U \subseteq [N]$ with $|U| \leq r$ the number of unique neighbours is at least $2|U|$

The Gadget

Let G be an $N \times n$ bipartite graph

$F \circ \mathrm{XOR}_G$ replaces $z_i \mapsto \bigoplus_{x_j \in \mathrm{N}(z_i)} x_j$



Idea: If G is sufficiently expanding:

→ learning the value of one XOR won't reveal much information about any other XOR

→ The best Resolution proof of $F \circ \mathrm{XOR}_G$ should essentially be to simulate the best proof of F

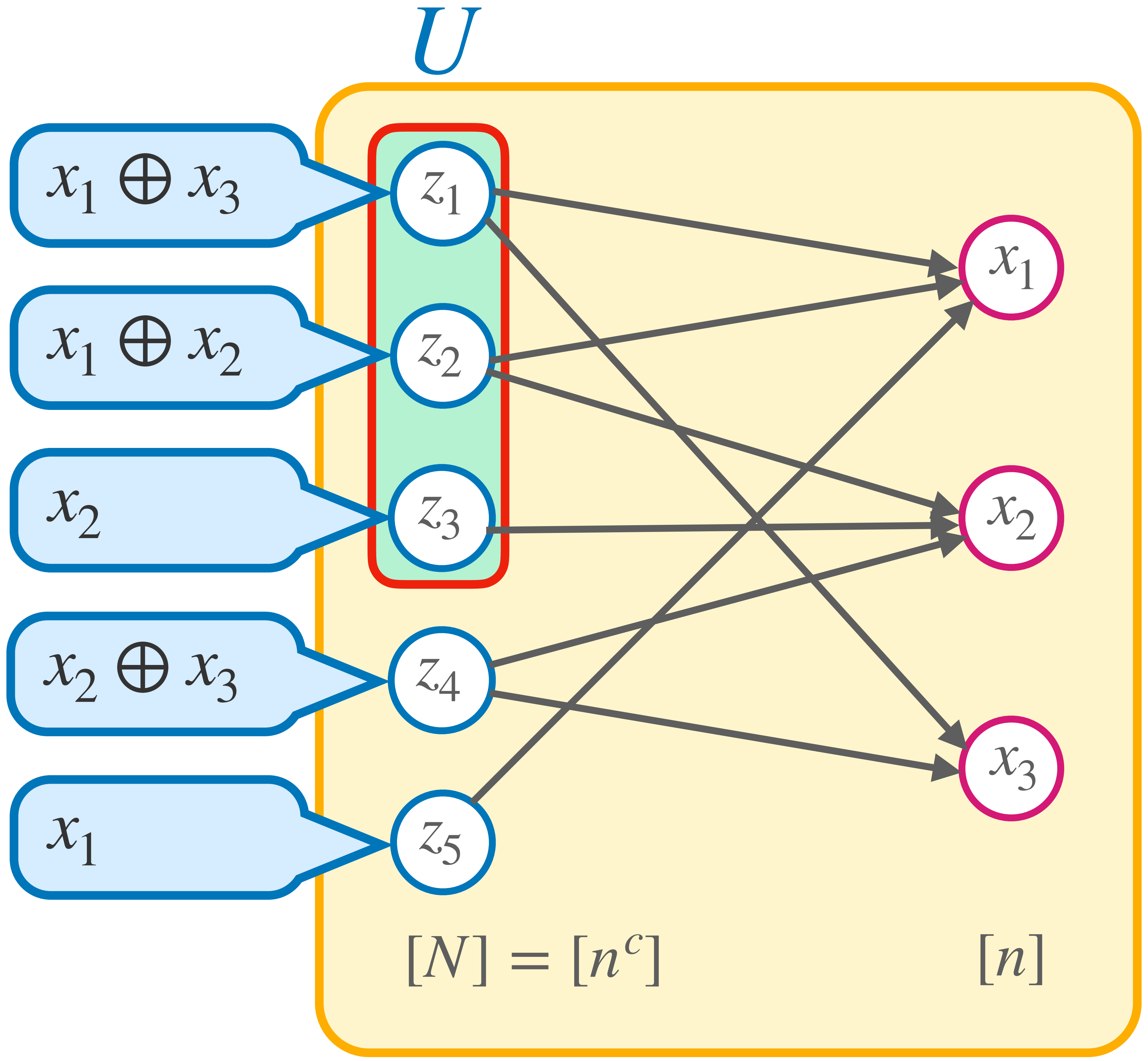
Number of x-variables that occur in exactly one XOR in U

r-Expanding: For any set $U \subseteq [N]$ with $|U| \leq r$ the number of unique neighbours is at least $2|U|$

The Gadget

Let G be an $N \times n$ bipartite graph

$F \circ \mathrm{XOR}_G$ replaces $z_i \mapsto \bigoplus_{x_j \in \mathrm{N}(z_i)} x_j$



Idea: If G is sufficiently expanding:
→ learning the value of one XOR won't reveal much information about any other XOR
→ The best Resolution proof of $F \circ \mathrm{XOR}_G$ should essentially be to simulate the best proof of F

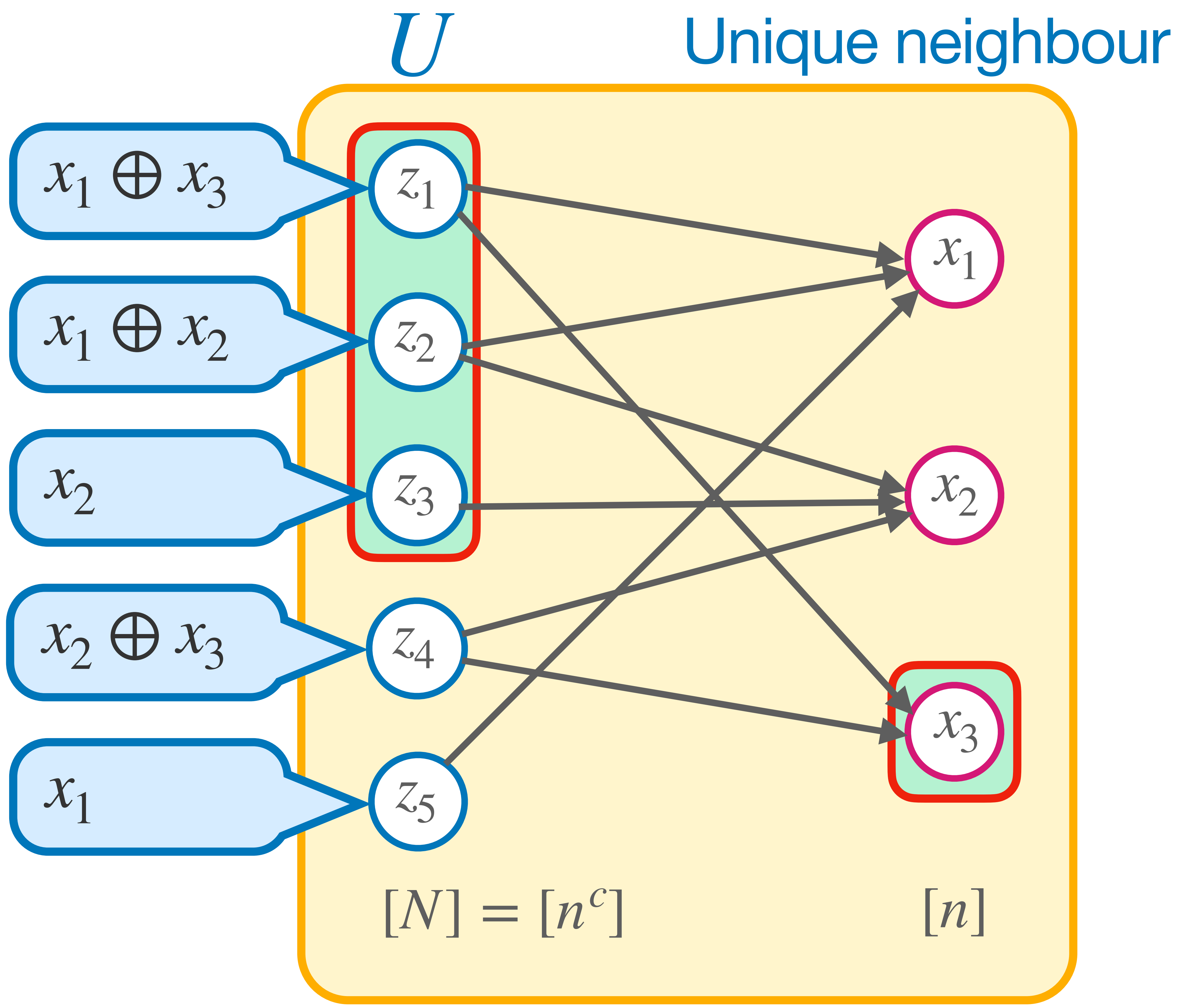
Number of x-variables that occur in exactly one XOR in U

r-Expanding: For any set $U \subseteq [N]$ with $|U| \leq r$ the number of unique neighbours is at least $2|U|$

The Gadget

Let G be an $N \times n$ bipartite graph

$F \circ \mathrm{XOR}_G$ replaces $z_i \mapsto \bigoplus_{x_j \in \mathrm{N}(z_i)} x_j$



Idea: If G is sufficiently expanding:
→ learning the value of one XOR won't reveal much information about any other XOR
→ The best Resolution proof of $F \circ \mathrm{XOR}_G$ should essentially be to simulate the best proof of F

Number of x-variables that occur in exactly one XOR in U

r-Expanding: For any set $U \subseteq [N]$ with $|U| \leq r$ the number of unique neighbours is at least $2|U|$

→ Our gadget g will be XOR_G for expanding G

Depth Condensation

Main workhorse behind our tradeoff:

Depth Condensation Theorem: ([Razborov16] stated for tree-resolution)

Let G be r-expanding, F any unsatisfiable formula.

If Π is a Resolution proof of $F \circ \mathsf{XOR}_G$ with $\mathsf{width}(\Pi) \leq r/4$ then

$$\mathsf{depth}(\Pi)\mathsf{width}(\Pi) = \Omega(\mathsf{depth}_{\mathsf{Res}}(F))$$

Depth Condensation

Main workhorse behind our tradeoff:

Depth Condensation Theorem: ([Razborov16] stated for tree-resolution)

Let G be r-expanding, F any unsatisfiable formula.

If Π is a Resolution proof of $F \circ \mathsf{XOR}_G$ with $\mathsf{width}(\Pi) \leq r/4$ then

$$\mathsf{depth}(\Pi)\mathsf{width}(\Pi) = \Omega(\mathsf{depth}_{\mathsf{Res}}(F))$$

$\rightarrow$ We give a simple proof

Depth Condensation

Main workhorse behind our tradeoff:

Depth Condensation Theorem: ([Razborov16] stated for tree-resolution)

Let G be r-expanding, F any unsatisfiable formula.

If Π is a Resolution proof of $F \circ \mathsf{XOR}_G$ with $\mathsf{width}(\Pi) \leq r/4$ then

$$\mathsf{depth}(\Pi)\mathsf{width}(\Pi) = \Omega(N/\log N)$$

→ We give a simple proof

→ Take $F = Peb$

Depth Condensation

Main workhorse behind our tradeoff:

Depth Condensation Theorem: ([Razborov16] stated for tree-resolution)

Let G be r-expanding, F any unsatisfiable formula.

If Π is a Resolution proof of $F \circ \mathsf{XOR}_G$ with $\mathsf{width}(\Pi) \leq r/4$ then

$$\mathsf{depth}(\Pi)\mathsf{width}(\Pi) = \Omega(N/\log N) = \Omega(n^c/c \log n)$$

→ We give a simple proof

→ Take $F = Peb$

Depth Condensation

Main workhorse behind our tradeoff:

Depth Condensation Theorem: ([Razborov16] stated for tree-resolution)

Let G be r-expanding, F any unsatisfiable formula.

If Π is a Resolution proof of $F \circ \mathsf{XOR}_G$ with $\mathsf{width}(\Pi) \leq r/4$ then

$$\mathsf{depth}(\Pi)\mathsf{width}(\Pi) = \Omega(N/\log N) = \Omega(n^c/c\log n)$$

→ We give a simple proof

→ Take $F = Peb$, combine with **width-to-size lifting theorem** proves our tradeoff!

Depth Condensation

Main workhorse behind our tradeoff:

Depth Condensation Theorem: ([Razborov16] stated for tree-resolution)

Let G be r-expanding, F any unsatisfiable formula.

If Π is a Resolution proof of $F \circ \mathsf{XOR}_G$ with $\mathsf{width}(\Pi) \leq r/4$ then

$$\mathsf{depth}(\Pi)\mathsf{width}(\Pi) = \Omega(N/\log N) = \Omega(n^c/c\log n)$$

→ We give a simple proof

→ Take $F = Peb$, combine with **width-to-size lifting theorem** proves our tradeoff!

Width-to-Size Lifting Theorem: If Π is a resolution proof of $F \circ \mathsf{XOR}_2$ then

$$\mathsf{size}(\Pi) \geq 2^{\Omega(\mathsf{width}_{\mathsf{Res}}(F))}$$

$$\mathsf{depth}(\Pi) \geq \mathsf{depth}_{\mathsf{Res}}(F)$$

Depth Condensation

Main workhorse behind our tradeoff:

Depth Condensation Theorem: ([Razborov16] stated for tree-resolution)

Let G be r-expanding, F any unsatisfiable formula.

If Π is a Resolution proof of $F \circ \mathsf{XOR}_G$ with $\mathsf{width}(\Pi) \leq r/4$ then

$$\mathsf{depth}(\Pi) \log \mathsf{size}(\Pi) = \Omega(N/\log N) = \Omega(n^c/c \log n)$$

→ We give a simple proof

→ Take $F = Peb$, combine with **width-to-size lifting theorem** proves our tradeoff!

Width-to-Size Lifting Theorem: If Π is a resolution proof of $F \circ \mathsf{XOR}_2$ then

$$\mathsf{size}(\Pi) \geq 2^{\Omega(\mathsf{width}_{\mathsf{Res}}(F))}$$

$$\mathsf{depth}(\Pi) \geq \mathsf{depth}_{\mathsf{Res}}(F)$$

Main Tradeoff (For Resolution)

Let $\varepsilon > 0$, let $c \geq 1$ be real-valued parameter

Main Theorem: There is a CNF formula F on n variables such that

1. There is a P-proof of F of size $n^c \cdot 2^{O(c)}$
2. If Π is a P-proof of F with $\mathsf{size}(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ then

$$\mathsf{depth}(\Pi) \cdot \log \mathsf{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$$

Tradeoffs for other proof systems are obtained by an extra step of lifting!

Main Tradeoff (For Resolution)

Let $\varepsilon > 0$, let $c \geq 1$ be real-valued parameter

Main Theorem: There is a CNF formula F on n variables such that

1. There is a P-proof of F of size $n^c \cdot 2^{O(c)}$
2. If Π is a P-proof of F with $\text{size}(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ then

$$\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$$

Tradeoffs for other proof systems are obtained by an extra step of lifting!

- For Cutting Planes we use the lifting theorem of **[GGKS18]**

Main Tradeoff (For Resolution)

Let $\varepsilon > 0$, let $c \geq 1$ be real-valued parameter

Main Theorem: There is a CNF formula F on n variables such that

1. There is a P-proof of F of size $n^c \cdot 2^{O(c)}$
2. If Π is a P-proof of F with $\text{size}(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ then

$$\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$$

Tradeoffs for other proof systems are obtained by an extra step of lifting!

- For Cutting Planes we use the lifting theorem of **[GGKS18]**
- For Res(k) we prove a Resolution width $\rightarrow$ Res(k) size lifting theorem with $g = XOR_2$, which uses the switching lemma of **[SBI04]**

(New) Proof of Depth Condensation

Depth Condensation Theorem:

Let G be r-expanding, F any unsatisfiable formula.

If Π is a resolution proof of $F \circ \mathsf{XOR}_G$ with $\mathsf{width}(\Pi) \leq r/4$ then

$$\mathsf{depth}(\Pi)\mathsf{width}(\Pi) = \Omega(\mathsf{depth}_{\mathsf{Res}}(F))$$

Our proof uses a characterization of resolution depth by Prover-Adversary games

Prover Adversary Games

Prover Adversary Games: Characterizes Resolution depth of proving F

Prover Adversary Games

Prover Adversary Games: Characterizes Resolution depth of proving F

Two players Prover, Adversary share a state $\rho \in \{0,1,*\}^n$, initially $\rho = *^n$

Prover Adversary Games

Prover Adversary Games: Characterizes Resolution depth of proving F

Two players Prover, Adversary share a state $\rho \in \{0,1,*\}^n$, initially $\rho = *^n$

- **Prover** wants to construct a state ρ falsifying a clause of F ($\exists C \in F$, $C(\rho) = 0$)

Prover Adversary Games

Prover Adversary Games: Characterizes Resolution depth of proving F

Two players Prover, Adversary share a state $\rho \in \{0,1,*\}^n$, initially $\rho = *^n$

- **Prover** wants to construct a state ρ falsifying a clause of F ($\exists C \in F, C(\rho) = 0$)
- **Adversary** wants to prolong the game

Prover Adversary Games

Prover Adversary Games: Characterizes Resolution depth of proving F

Two players Prover, Adversary share a state $\rho \in \{0,1,*\}^n$, initially $\rho = *^n$

- **Prover** wants to construct a state ρ falsifying a clause of F ($\exists C \in F, C(\rho) = 0$)
- **Adversary** wants to prolong the game

Each round:

Prover Adversary Games

Prover Adversary Games: Characterizes Resolution depth of proving F

Two players Prover, Adversary share a state $\rho \in \{0,1,*\}^n$, initially $\rho = *^n$

- **Prover** wants to construct a state ρ falsifying a clause of F ($\exists C \in F, C(\rho) = 0$)
- **Adversary** wants to prolong the game

Each round:

- Prover chooses $i \in [n]$ such that $\rho_i = *$

Prover Adversary Games

Prover Adversary Games: Characterizes Resolution depth of proving F

Two players Prover, Adversary share a state $\rho \in \{0,1,*\}^n$, initially $\rho = *^n$

- **Prover** wants to construct a state ρ falsifying a clause of F ($\exists C \in F, C(\rho) = 0$)
- **Adversary** wants to prolong the game

Each round:

- Prover chooses $i \in [n]$ such that $\rho_i = *$
- Adversary chooses $b \in \{0,1\}$ and sets $\rho_i = b$

Prover Adversary Games

Prover Adversary Games: Characterizes Resolution depth of proving F

Two players Prover, Adversary share a state $\rho \in \{0,1,*\}^n$, initially $\rho = *^n$

- **Prover** wants to construct a state ρ falsifying a clause of F ($\exists C \in F$, $C(\rho) = 0$)
- **Adversary** wants to prolong the game

Each round:

- Prover chooses $i \in [n]$ such that $\rho_i = *$
- Adversary chooses $b \in \{0,1\}$ and sets $\rho_i = b$

Claim: If there is a strategy for the Adversary such that the game always continues for at least d rounds, then any resolution proof of F requires depth $\geq d$

Prover Adversary Games

Prover Adversary Games: Characterizes Resolution depth of proving F

Two players Prover, Adversary share a state $\rho \in \{0,1,*\}^n$, initially $\rho = *^n$

- **Prover** wants to construct a state ρ falsifying a clause of F ($\exists C \in F, C(\rho) = 0$)
- **Adversary** wants to prolong the game

Each round:

- Prover chooses $i \in [n]$ such that $\rho_i = *$
- Adversary chooses $b \in \{0,1\}$ and sets $\rho_i = b$

w-**Bounded Game:** ρ remembers at most w variables every round. ($|\rho| \leq w$)

Prover Adversary Games

Prover Adversary Games: Characterizes Resolution depth of proving F

Two players Prover, Adversary share a state $\rho \in \{0,1,*\}^n$, initially $\rho = *^n$

- **Prover** wants to construct a state ρ falsifying a clause of F ($\exists C \in F, C(\rho) = 0$)
- **Adversary** wants to prolong the game

Each round:

- Prover chooses $i \in [n]$ such that $\rho_i = *$
- Adversary chooses $b \in \{0,1\}$ and sets $\rho_i = b$
- Prover chooses $S \subseteq [n]$ and sets $\rho_i = *$ for all $i \in S$ (Forgetting)

w**-Bounded Game:** ρ remembers at most w variables every round. ($|\rho| \leq w$)

Prover Adversary Games

Prover Adversary Games: Characterizes Resolution depth of proving F

Two players Prover, Adversary share a state $\rho \in \{0,1,*\}^n$, initially $\rho = *^n$

- **Prover** wants to construct a state ρ falsifying a clause of F ($\exists C \in F, C(\rho) = 0$)
- **Adversary** wants to prolong the game

Each round:

- Prover chooses $i \in [n]$ such that $\rho_i = *$
- Adversary chooses $b \in \{0,1\}$ and sets $\rho_i = b$
- Prover chooses $S \subseteq [n]$ and sets $\rho_i = *$ for all $i \in S$ (Forgetting)

w**-Bounded Game:** ρ remembers at most w variables every round. ($|\rho| \leq w$)

Unbounded Game: No bound on $|\rho|$

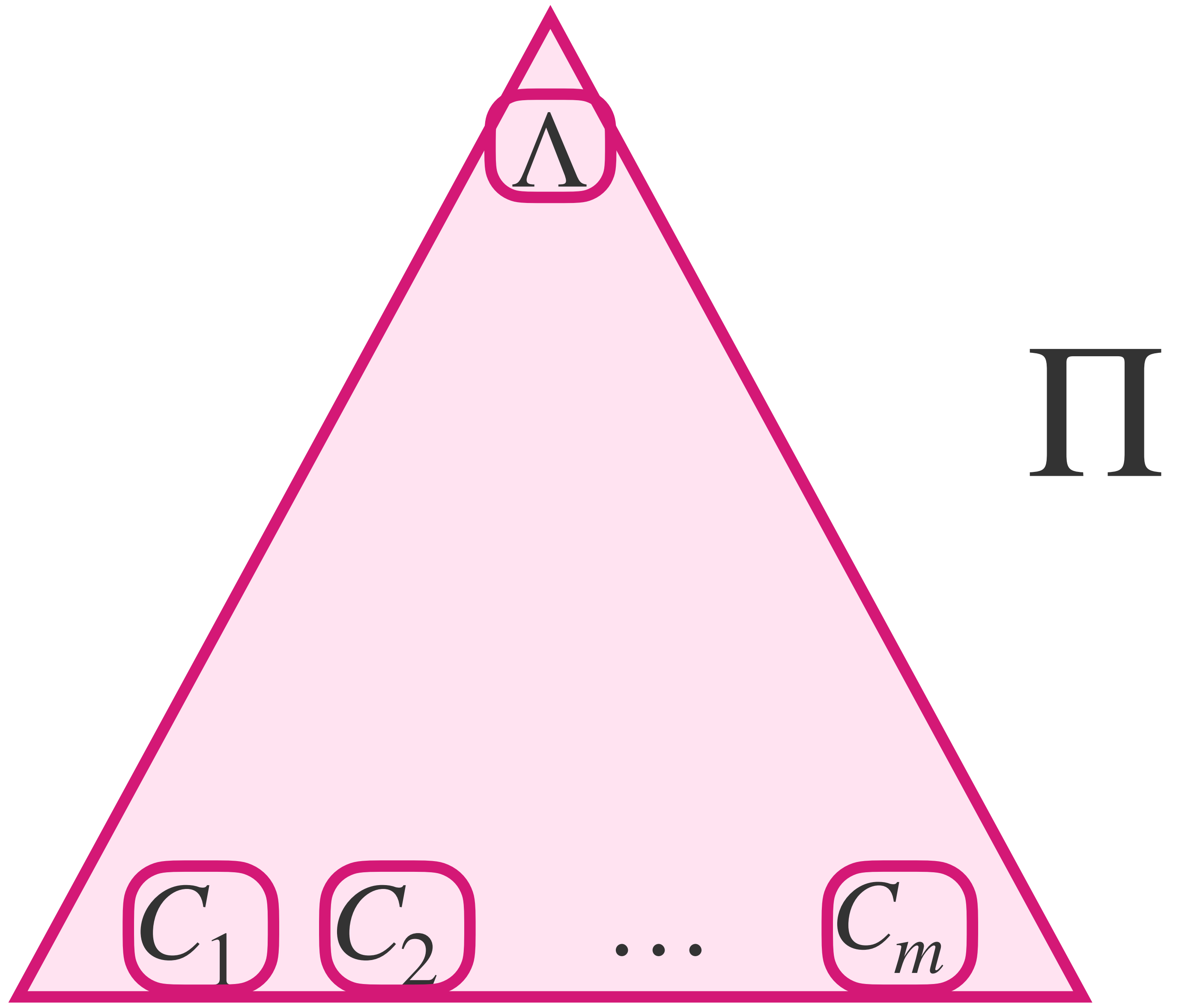
Prover Adversary Games

Claim: For any F, if there is a Resolution proof Π of F of width $\leq w$ and depth $\leq d$ then there is a strategy for the Prover to win the $(w+1)$-bounded game in d rounds.

Prover Adversary Games

Claim: For any F, if there is a Resolution proof Π of F of width $\leq w$ and depth $\leq d$ then there is a strategy for the Prover to win the $(w+1)$-bounded game in d rounds.

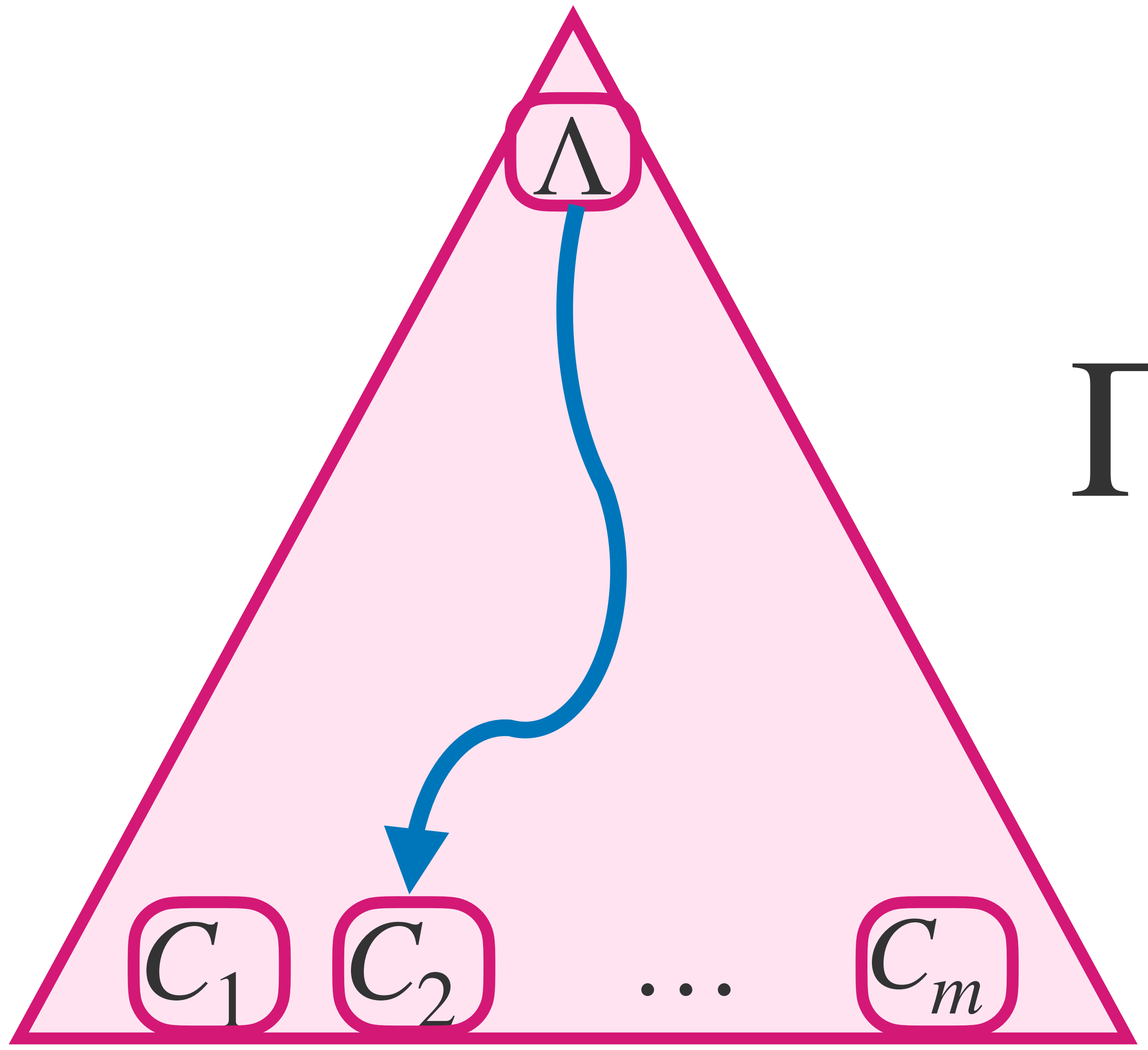
Pf:



Prover Adversary Games

Claim: For any F, if there is a Resolution proof Π of F of width $\leq w$ and depth $\leq d$ then there is a strategy for the Prover to win the $(w+1)$-bounded game in d rounds.

Pf: Prover will walk from the root of Π to a leaf



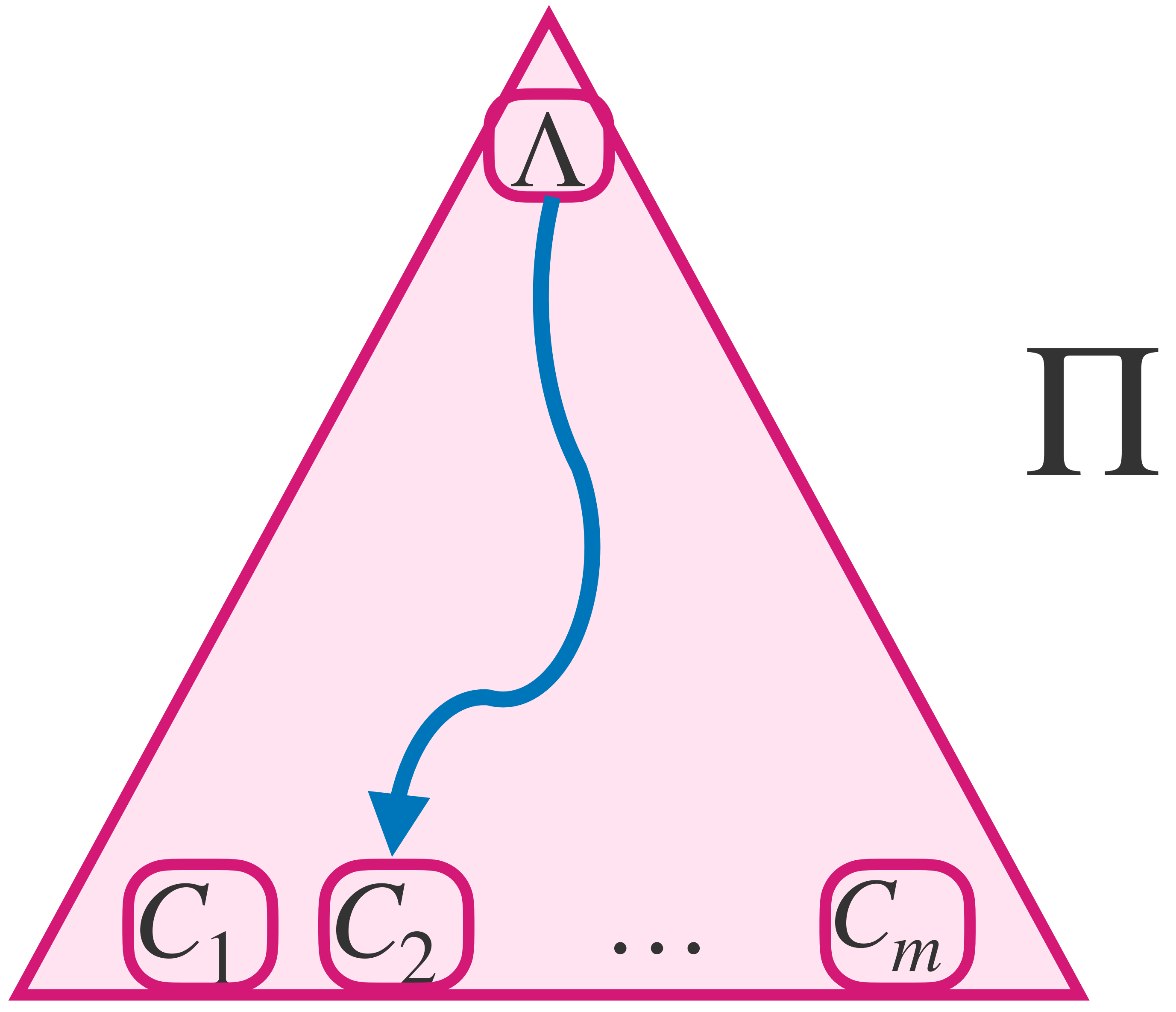
Π

Prover Adversary Games

Claim: For any F, if there is a Resolution proof Π of F of width $\leq w$ and depth $\leq d$ then there is a strategy for the Prover to win the $(w+1)$-bounded game in d rounds.

Pf: Prover will walk from the root of Π to a leaf

Invariant: If current clause is C then $C(\rho) = 0$, $|\rho| \leq w$



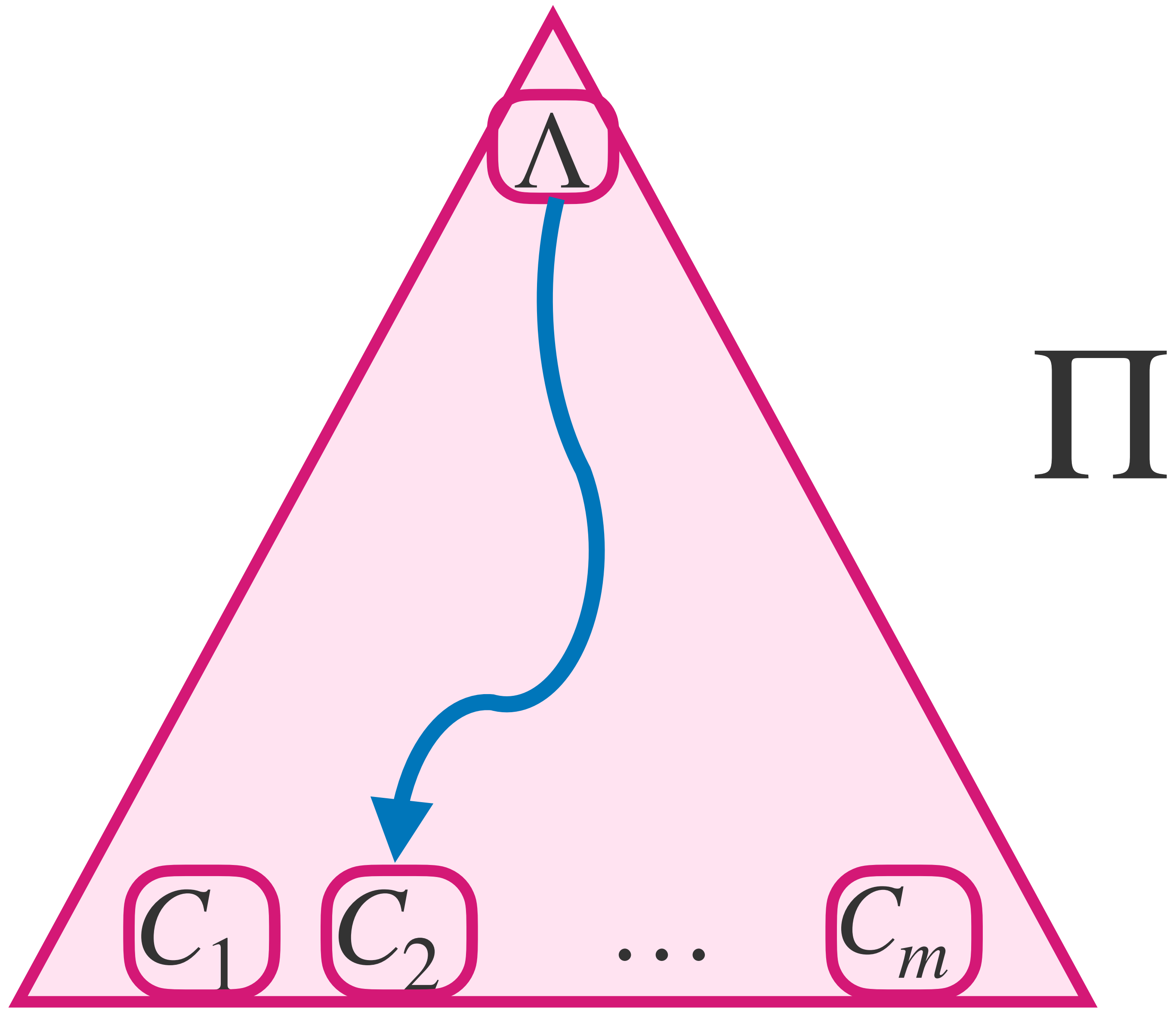
Prover Adversary Games

Claim: For any F, if there is a Resolution proof Π of F of width $\leq w$ and depth $\leq d$ then there is a strategy for the Prover to win the $(w+1)$-bounded game in d rounds.

Pf: Prover will walk from the root of Π to a leaf

Invariant: If current clause is C then $C(\rho) = 0$, $|\rho| \leq w$

$\rightarrow$ Root case is satisfied: Λ is identically false



Prover Adversary Games

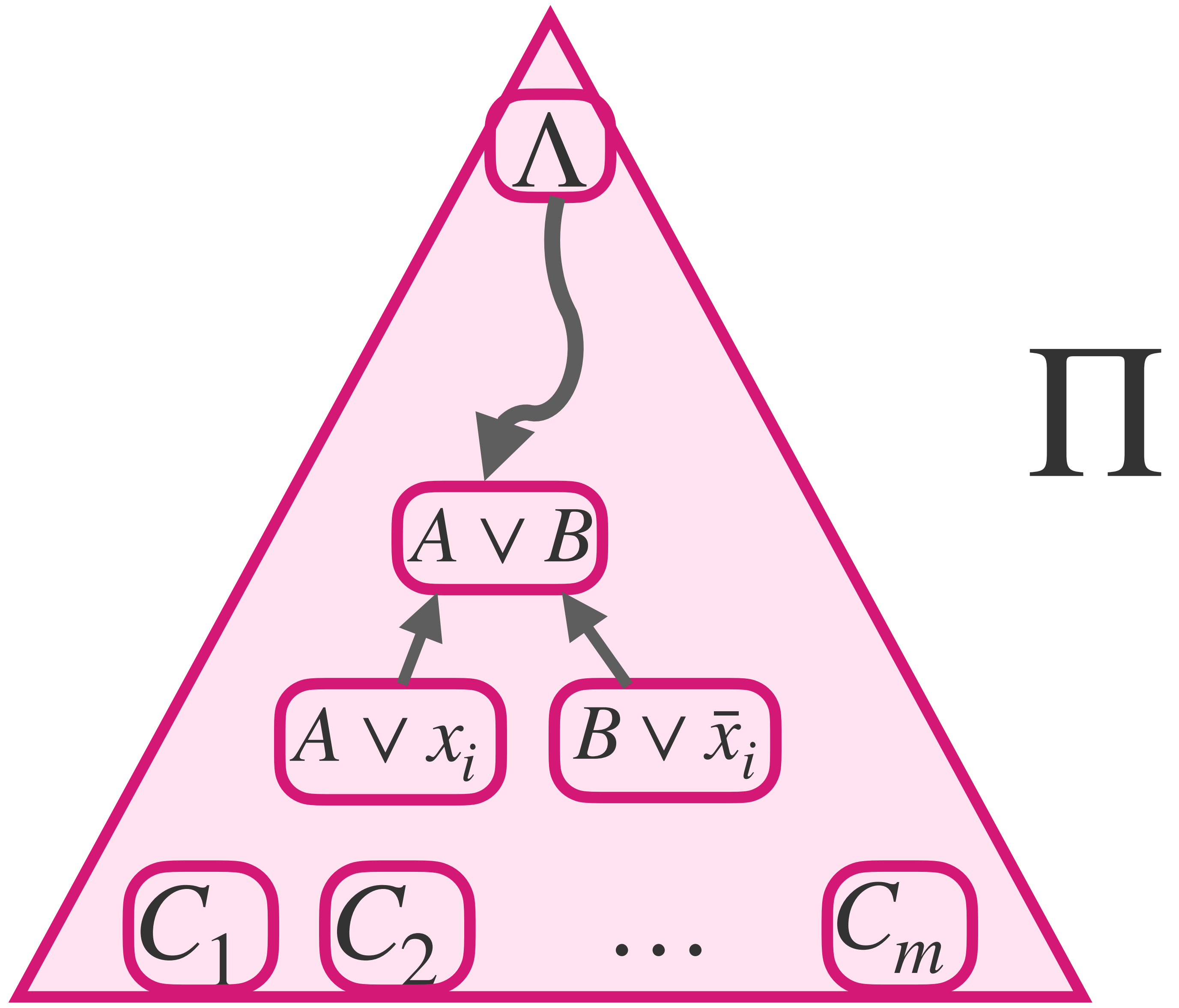
Claim: For any F, if there is a Resolution proof Π of F of width $\leq w$ and depth $\leq d$ then there is a strategy for the Prover to win the $(w+1)$-bounded game in d rounds.

Pf: Prover will walk from the root of Π to a leaf

Invariant: If current clause is C then $C(\rho) = 0$, $|\rho| \leq w$

$\rightarrow$ Root case is satisfied: Λ is identically false

Suppose current clause is $A \vee B$



Prover Adversary Games

Claim: For any F, if there is a Resolution proof Π of F of width $\leq w$ and depth $\leq d$ then there is a strategy for the Prover to win the $(w+1)$-bounded game in d rounds.

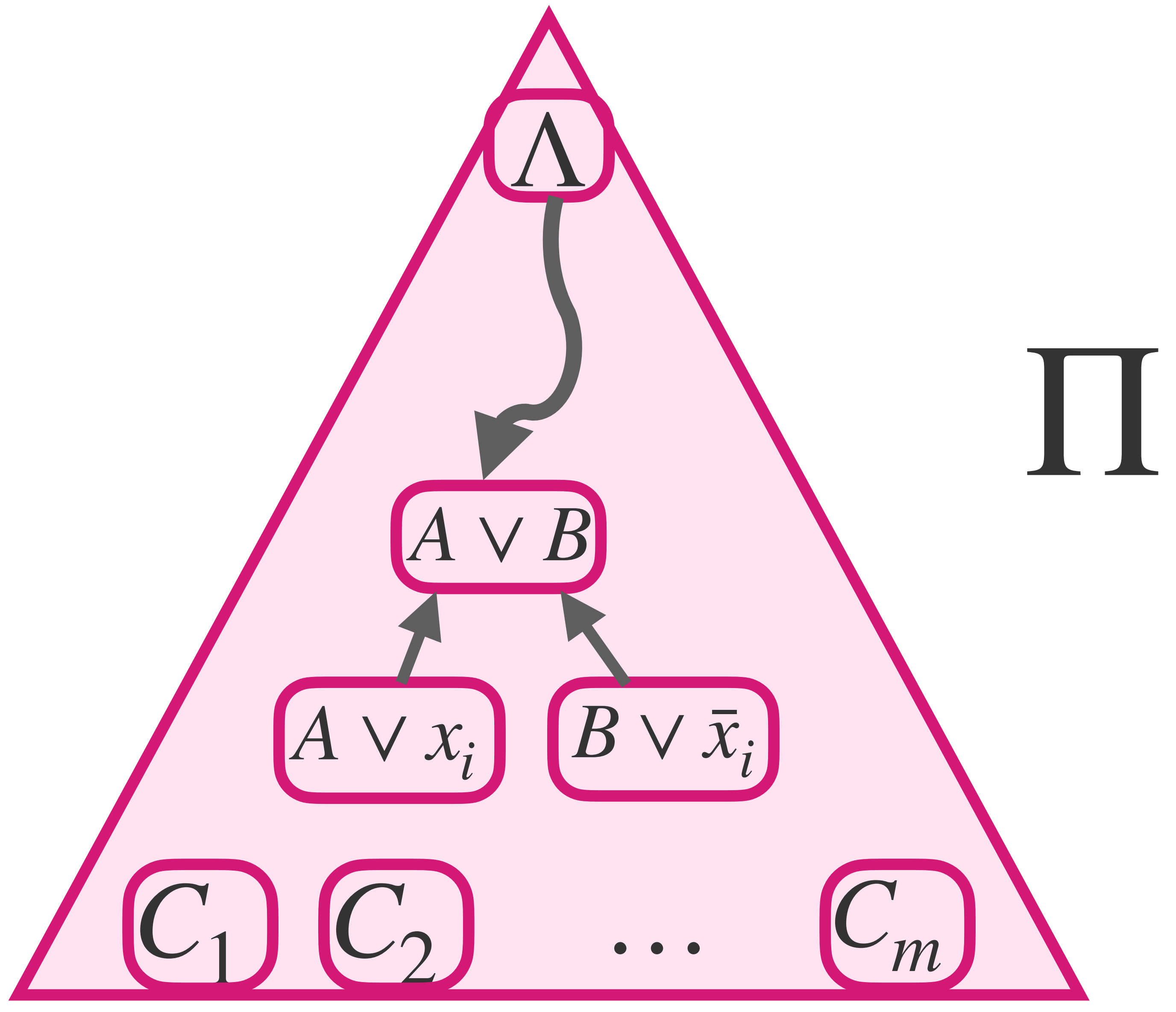
Pf: Prover will walk from the root of Π to a leaf

Invariant: If current clause is C then $C(\rho) = 0$, $|\rho| \leq w$

$\rightarrow$ Root case is satisfied: Λ is identically false

Suppose current clause is $A \vee B$

- Prover asks about x_i



Prover Adversary Games

Claim: For any F, if there is a Resolution proof Π of F of width $\leq w$ and depth $\leq d$ then there is a strategy for the Prover to win the $(w+1)$-bounded game in d rounds.

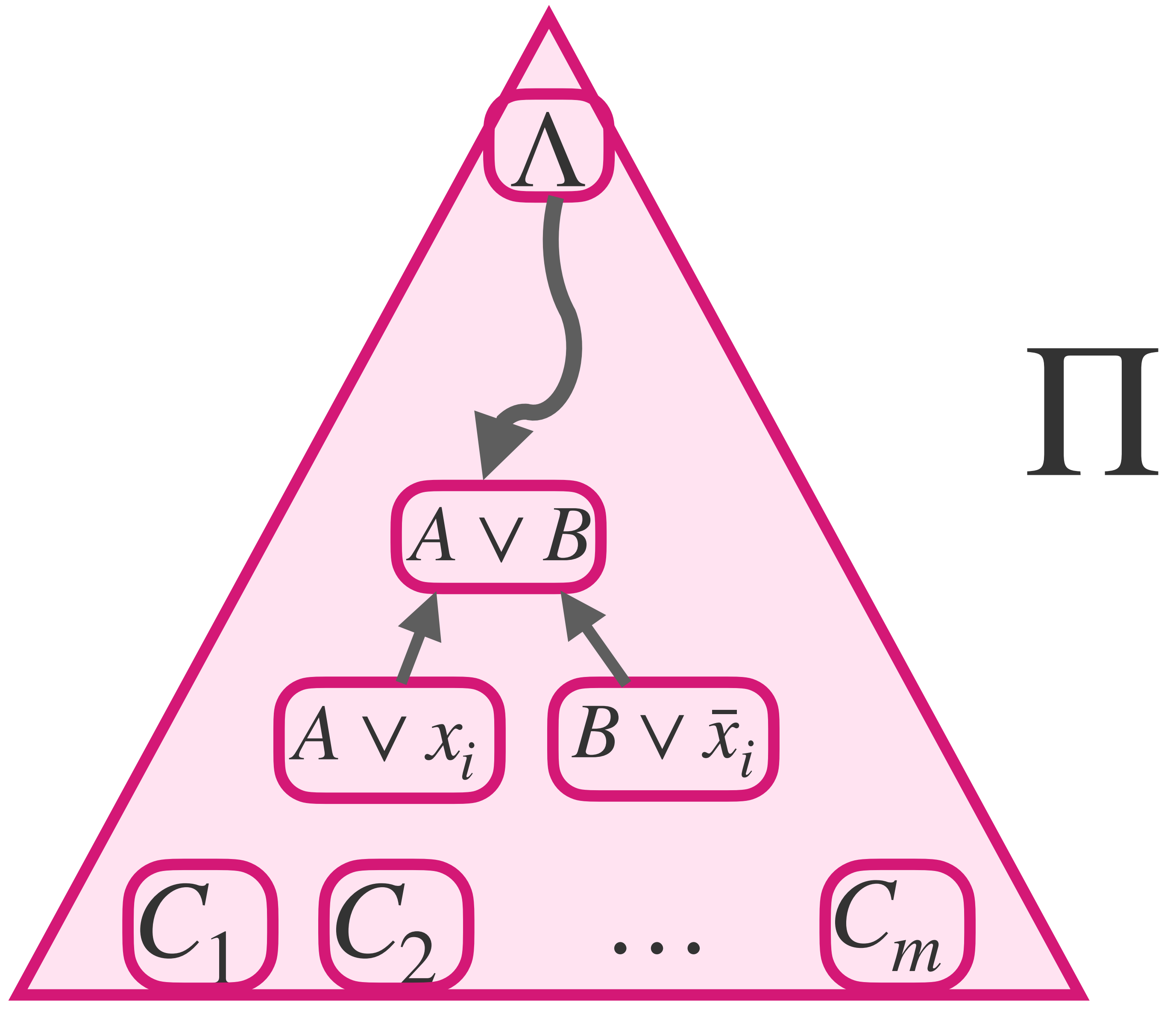
Pf: Prover will walk from the root of Π to a leaf

Invariant: If current clause is C then $C(\rho) = 0$, $|\rho| \leq w$

→ Root case is satisfied: Λ is identically false

Suppose current clause is $A \vee B$

- Prover asks about x_i
- If Adversary says $x_i = 0$ move to $A \vee x_i$. Forget $B \backslash A \cup x_i$



Prover Adversary Games

Claim: For any F, if there is a Resolution proof Π of F of width $\leq w$ and depth $\leq d$ then there is a strategy for the Prover to win the $(w+1)$-bounded game in d rounds.

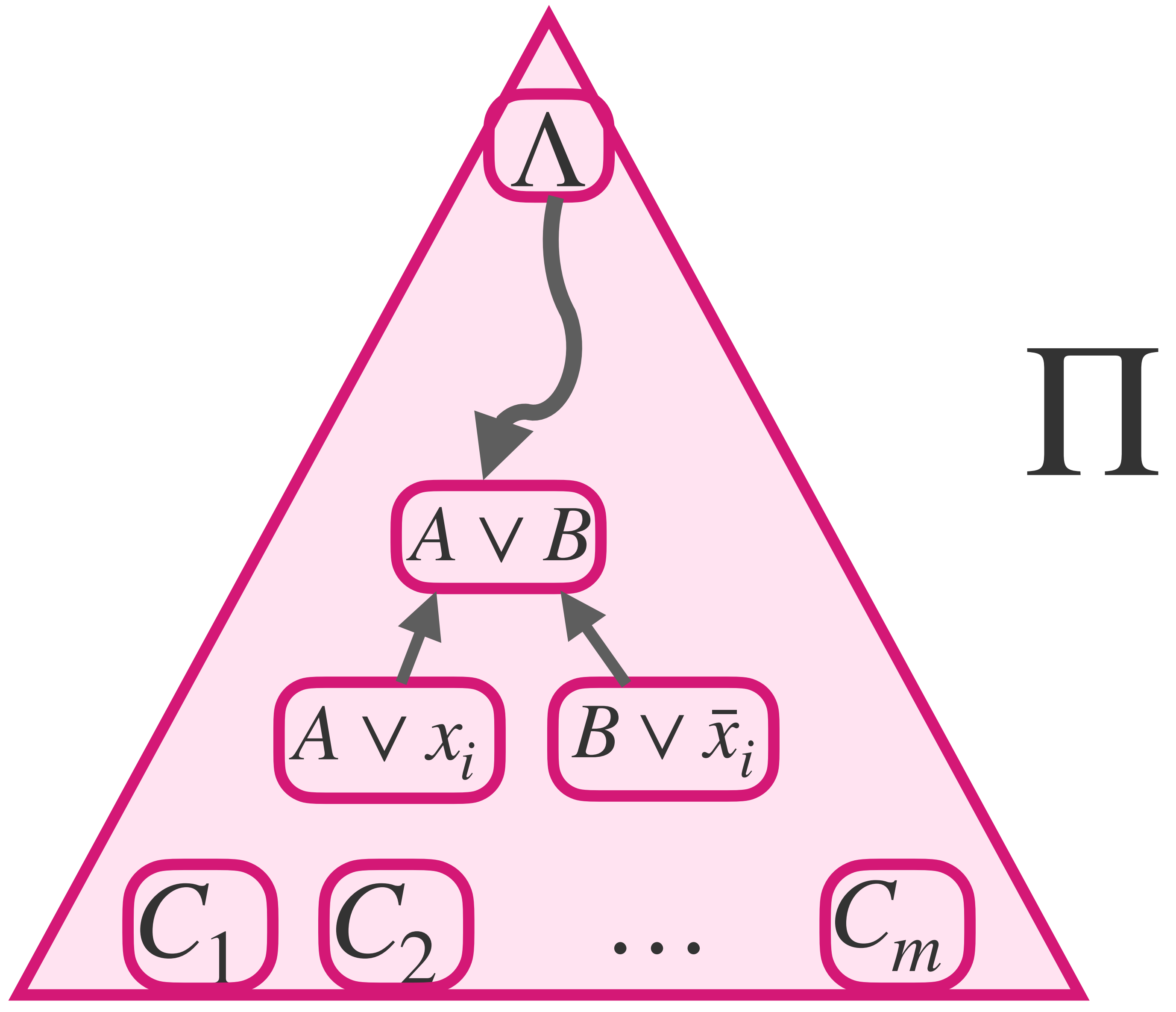
Pf: Prover will walk from the root of Π to a leaf

Invariant: If current clause is C then $C(\rho) = 0$, $|\rho| \leq w$

$\rightarrow$ Root case is satisfied: Λ is identically false

Suppose current clause is $A \vee B$

- Prover asks about x_i
- If Adversary says $x_i = 0$ move to $A \vee x_i$. Forget $B \backslash A \cup x_i$
- Otherwise, move to $B \vee \bar{x}_i$. Forget $A \backslash B$



(New) Proof of Depth Condensation

Depth Condensation Theorem:

Let G be an r-boundary expander, F any unsatisfiable formula.

If Π is a Resolution proof of $F \circ XOR_G$ with $\text{width}(\Pi) \leq r/4$ then

$$\text{depth}(\Pi)\text{width}(\Pi) = \Omega(\text{depth}_{\text{Res}}(F))$$

High Level of Proof:

(New) Proof of Depth Condensation

Depth Condensation Theorem:

Let G be an r-boundary expander, F any unsatisfiable formula.

If Π is a Resolution proof of $F \circ XOR_G$ with $\text{width}(\Pi) \leq r/4$ then

$$\text{depth}(\Pi)\text{width}(\Pi) = \Omega(\text{depth}_{\text{Res}}(F))$$

High Level of Proof:

If $\text{depth}_{\text{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F

(New) Proof of Depth Condensation

Depth Condensation Theorem:

Let G be an r-boundary expander, F any unsatisfiable formula.

If Π is a Resolution proof of $F \circ XOR_G$ with $\mathsf{width}(\Pi) \leq r/4$ then

$$\mathsf{depth}(\Pi)\mathsf{width}(\Pi) = \Omega(\mathsf{depth}_{\mathsf{Res}}(F))$$

High Level of Proof:

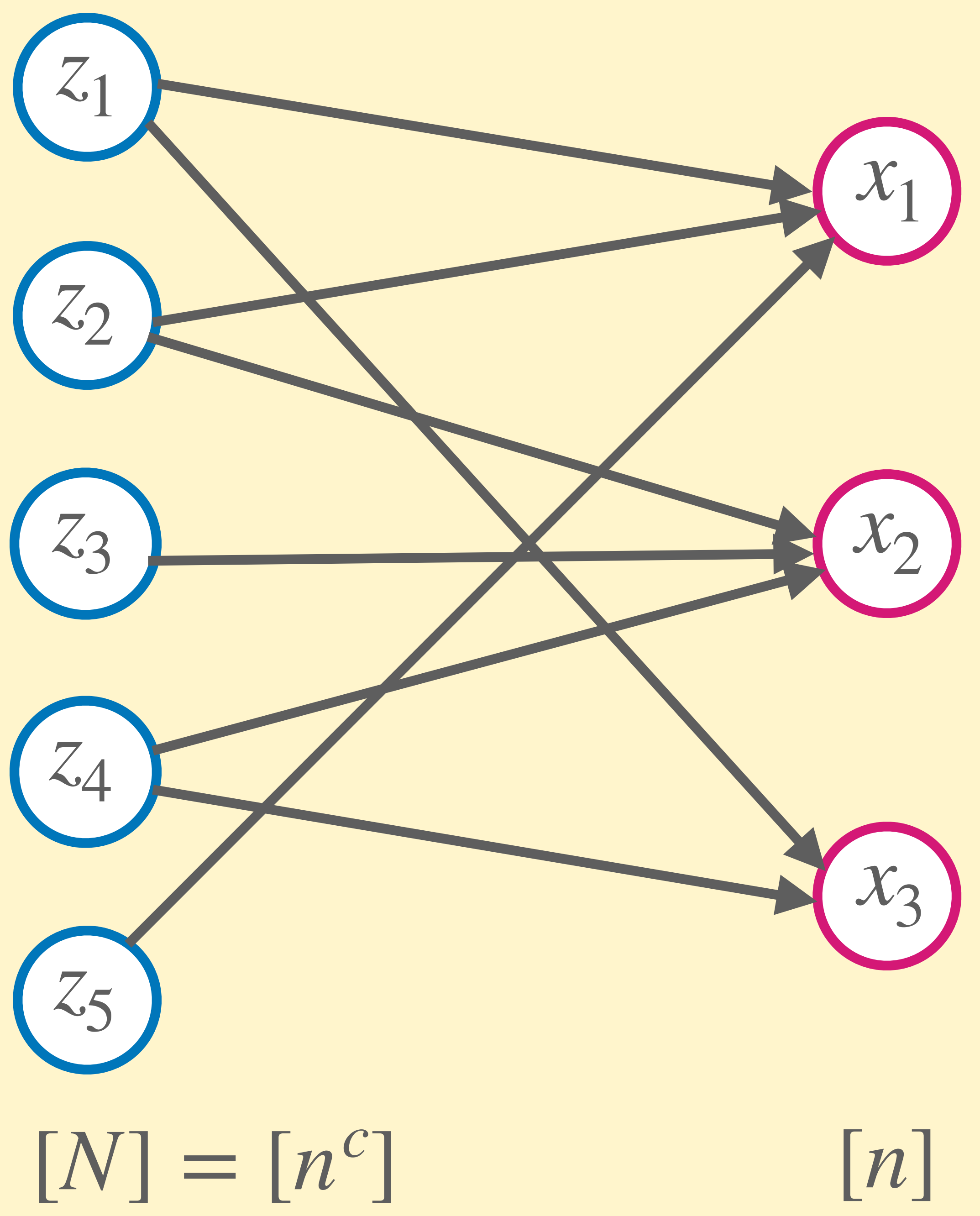
If $\mathsf{depth}_{\mathsf{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F

$\rightarrow$ Use A to construct an Adversary Strategy for the w-bounded game on $F \circ XOR_G$ to survive $\Omega(d/w)$ rounds, for any $w \leq r/4$.

(New) Proof of Depth Condensation

High Level of Proof:

If $\mathsf{depth}_{\mathsf{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F

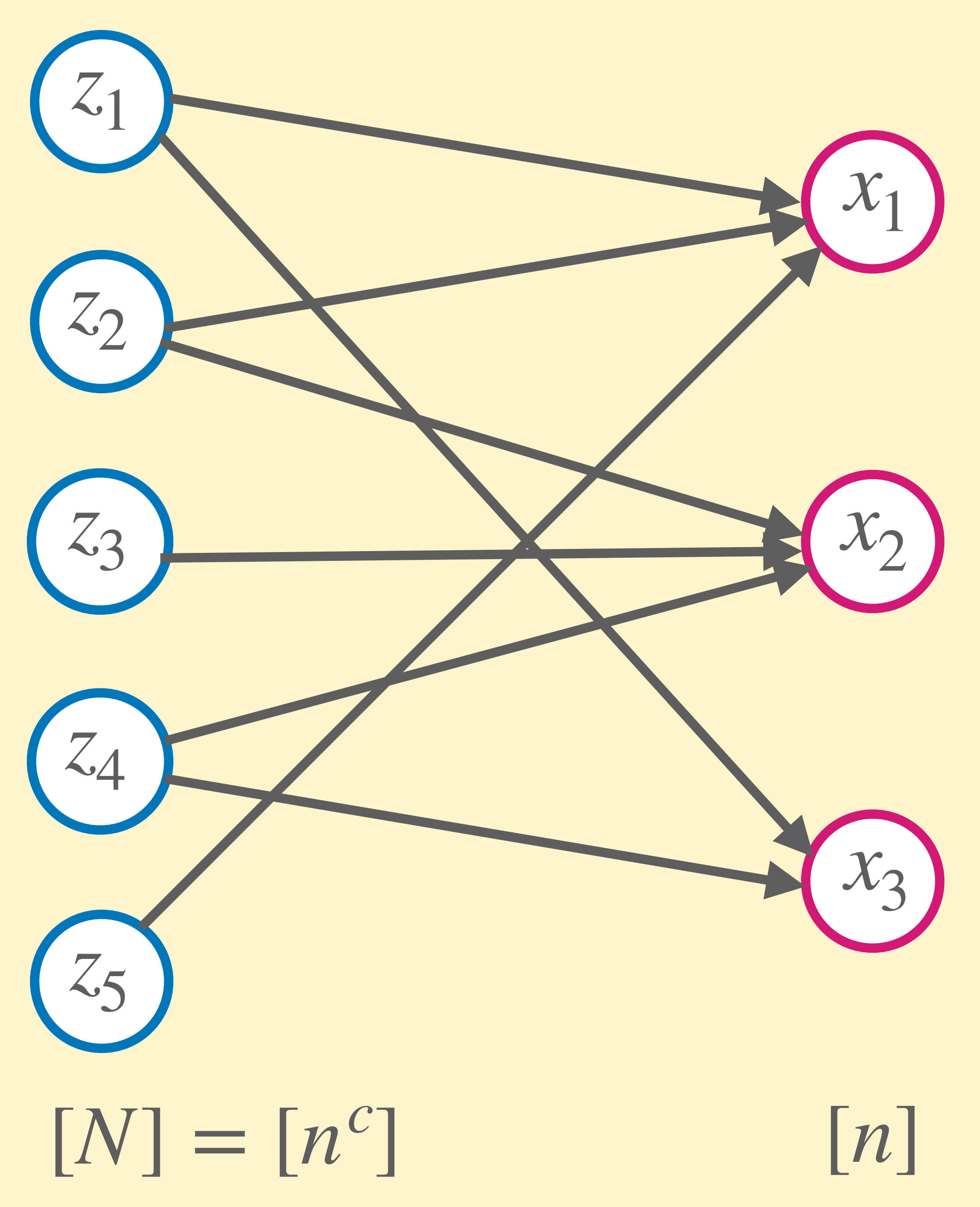


Adversary strategy for $F \circ \mathsf{XOR}_G$:

(New) Proof of Depth Condensation

High Level of Proof:

If $\mathrm{depth}_{\mathrm{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F



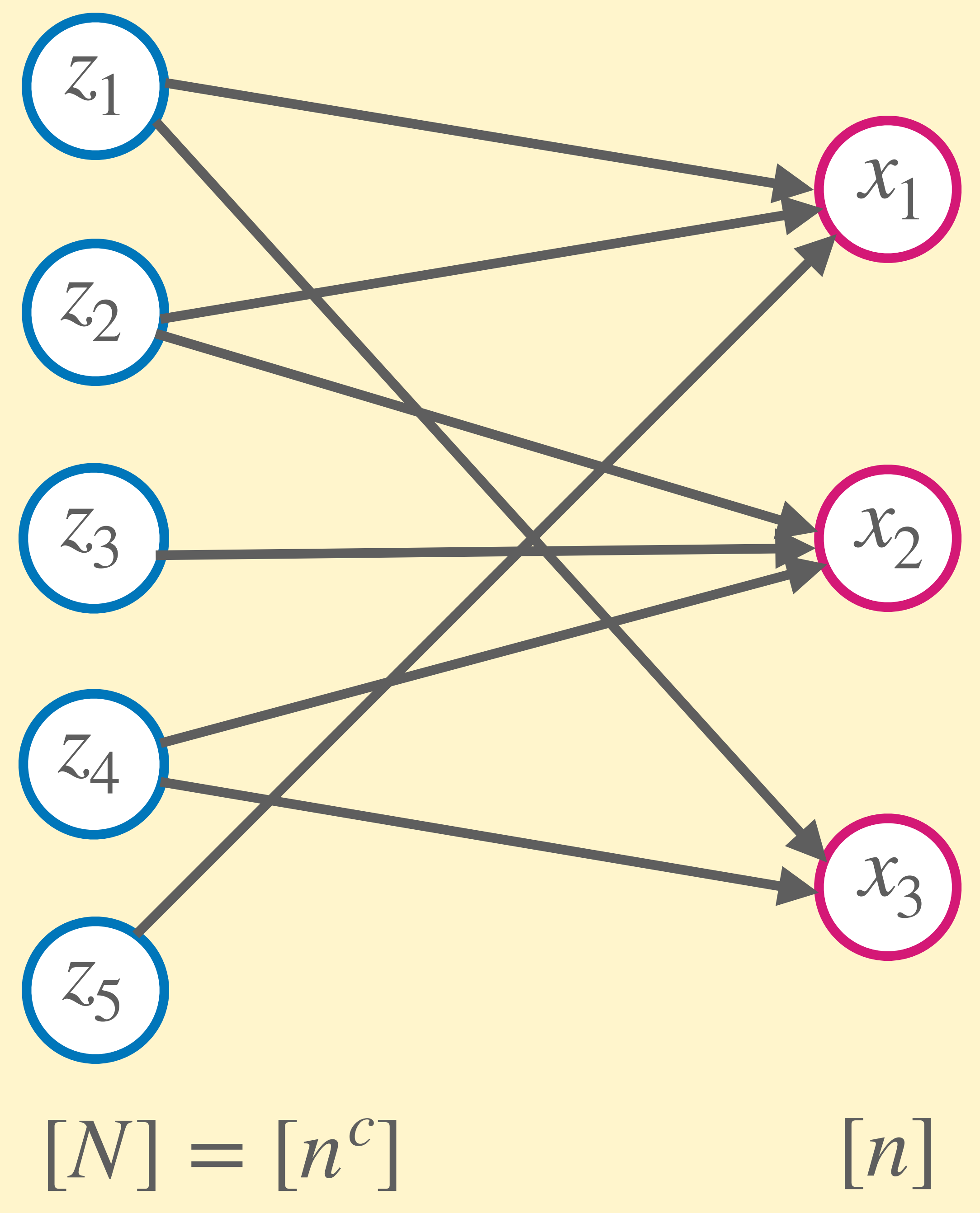
Adversary strategy for $F \circ \mathrm{XOR}_G$:

If Prover queries x_i:

(New) Proof of Depth Condensation

High Level of Proof:

If $\text{depth}_{\text{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F



Adversary strategy for $F \circ \text{XOR}_G$:

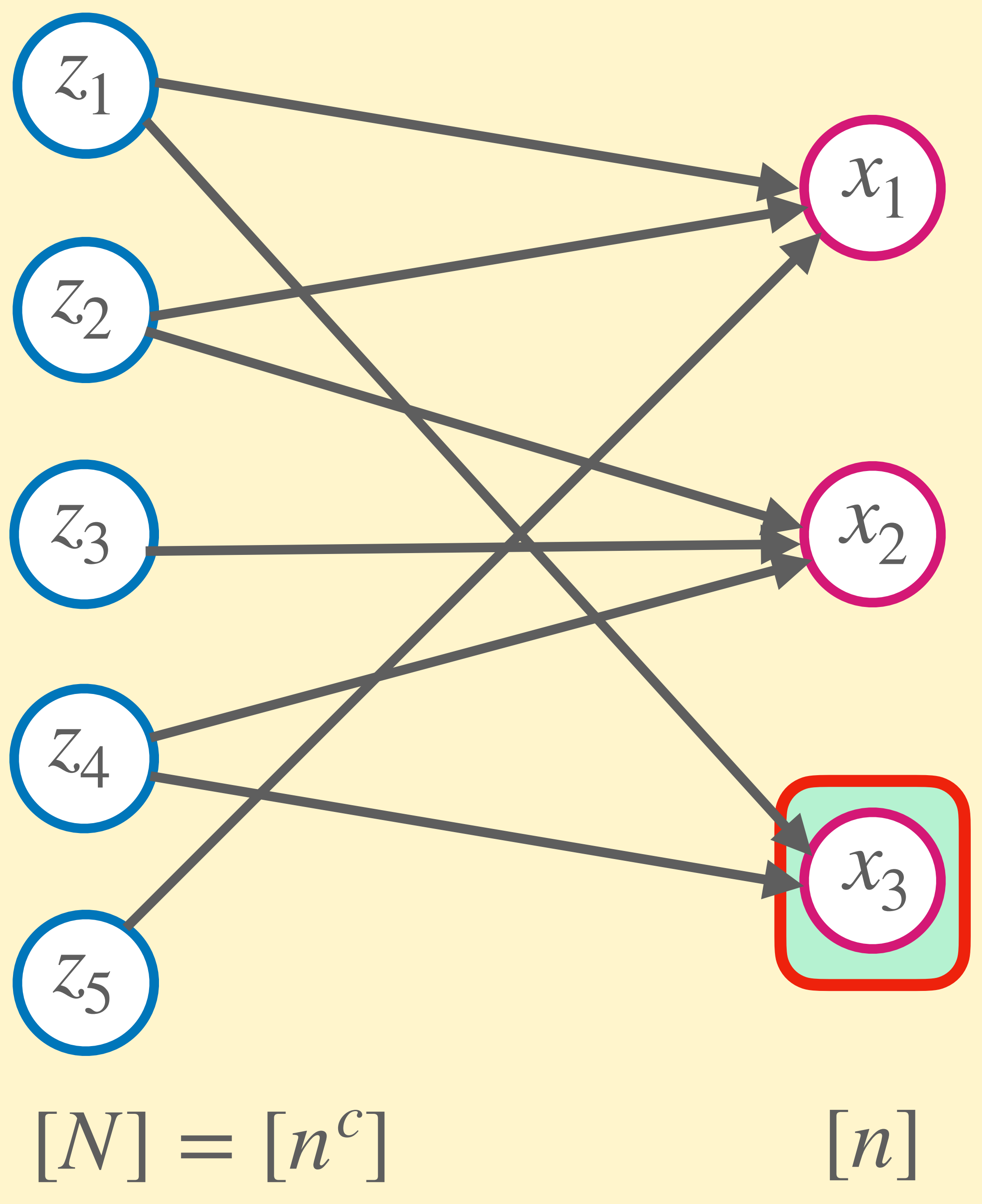
If Prover queries x_i:

- If there are ≥ 2 variables in $\text{N}(z_j)$ for every $z_j \in \text{N}(x_i)$: set x_i arbitrarily

(New) Proof of Depth Condensation

High Level of Proof:

If $\text{depth}_{\text{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F



Adversary strategy for $F \circ \text{XOR}_G$:

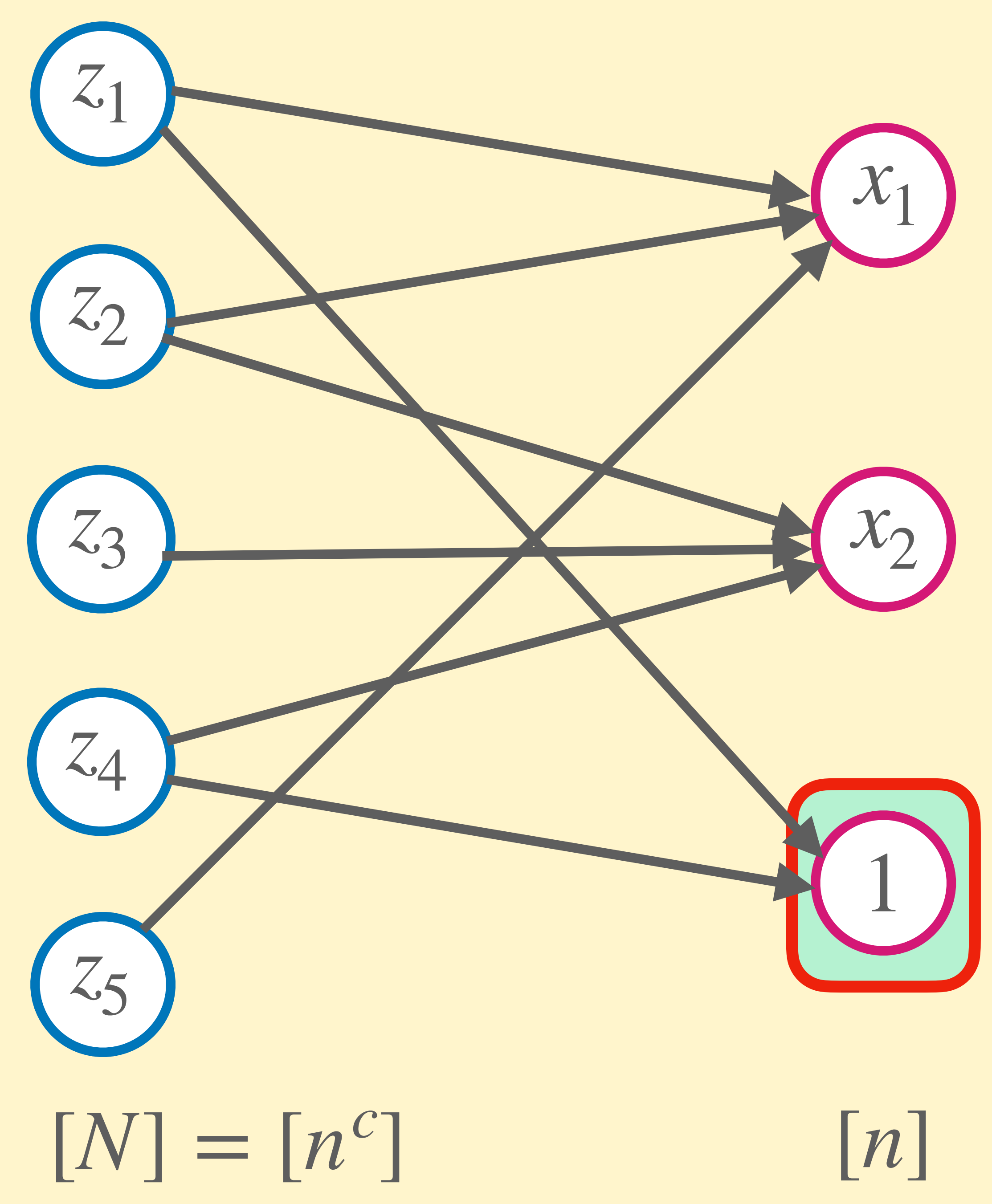
If Prover queries x_i:

- If there are ≥ 2 variables in $\text{N}(z_j)$ for every $z_j \in \text{N}(x_i)$: set x_i arbitrarily

(New) Proof of Depth Condensation

High Level of Proof:

If $\mathsf{depth}_{\mathsf{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F



Adversary strategy for $F \circ \mathsf{XOR}_G$:

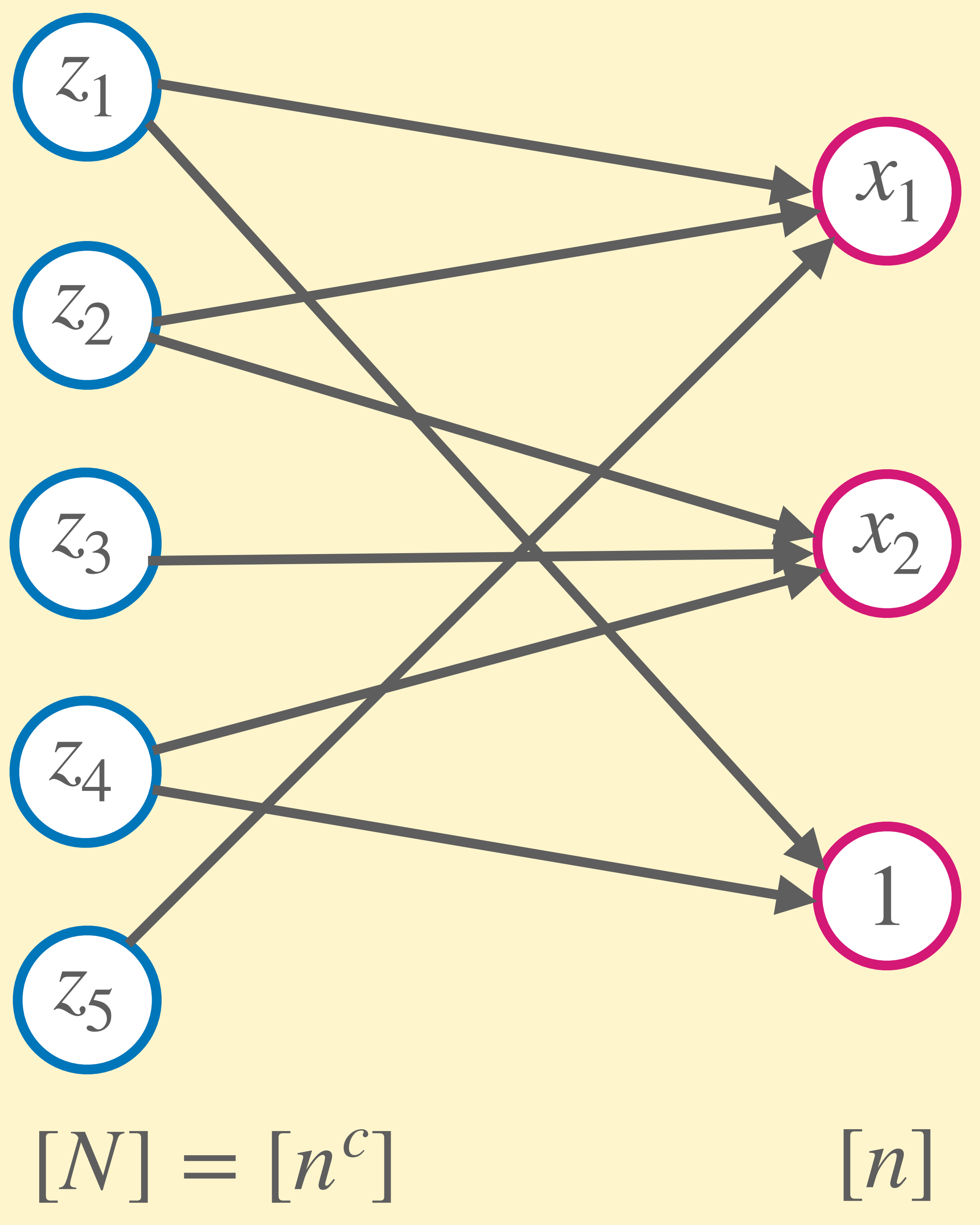
If Prover queries x_i:

- If there are ≥ 2 variables in $\mathsf{N}(z_j)$ for every $z_j \in \mathsf{N}(x_i)$: set x_i arbitrarily

(New) Proof of Depth Condensation

High Level of Proof:

If $\text{depth}_{\text{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F



Adversary strategy for $F \circ \text{XOR}_G$:

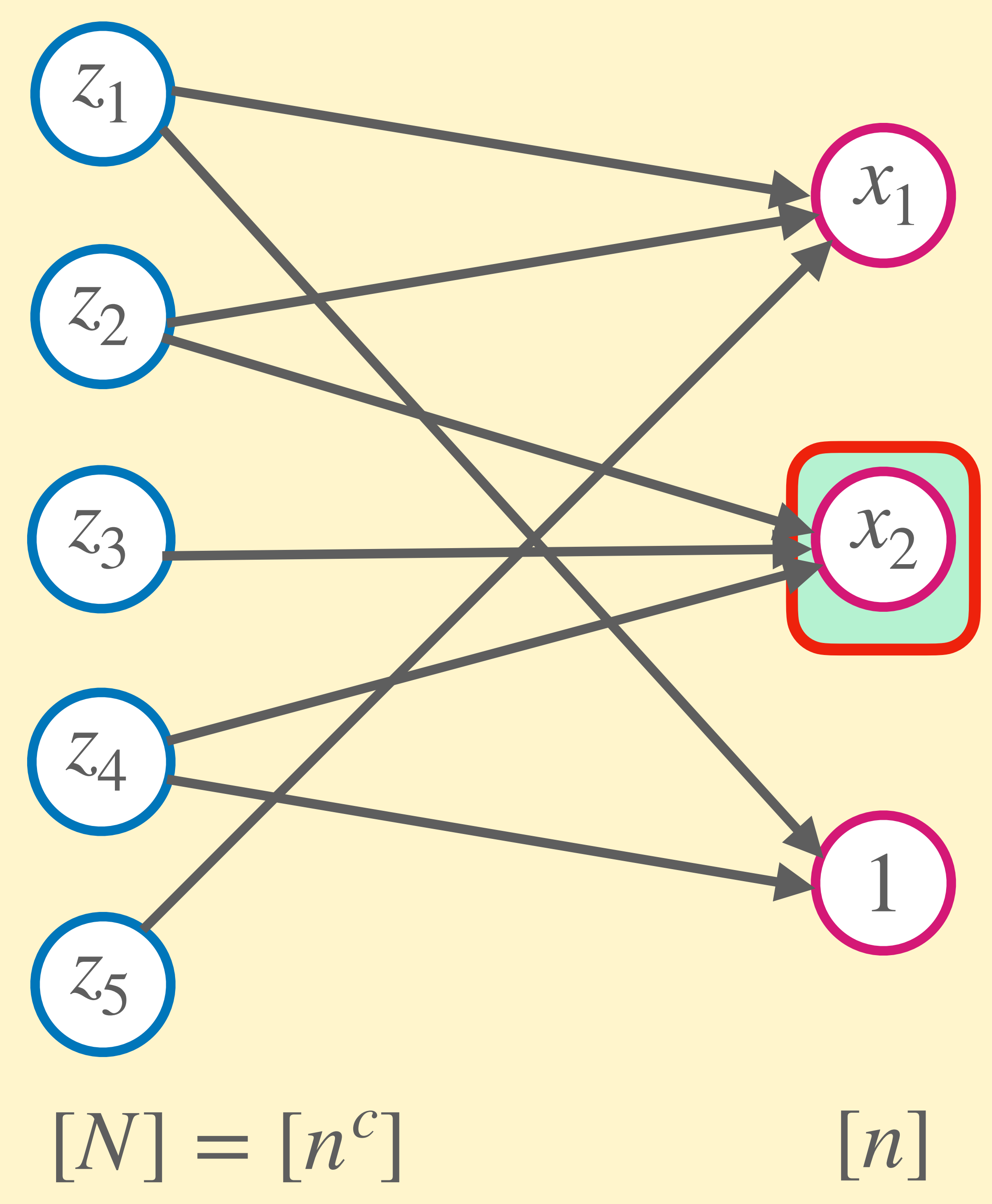
If Prover queries x_i:

- If there are ≥ 2 variables in $\text{N}(z_j)$ for every $z_j \in \text{N}(x_i)$: set x_i arbitrarily
- If x_i is the last variable in $\text{N}(z_j)$ (for some z_j) not set in ρ:

(New) Proof of Depth Condensation

High Level of Proof:

If $\text{depth}_{\text{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F



Adversary strategy for $F \circ \text{XOR}_G$:

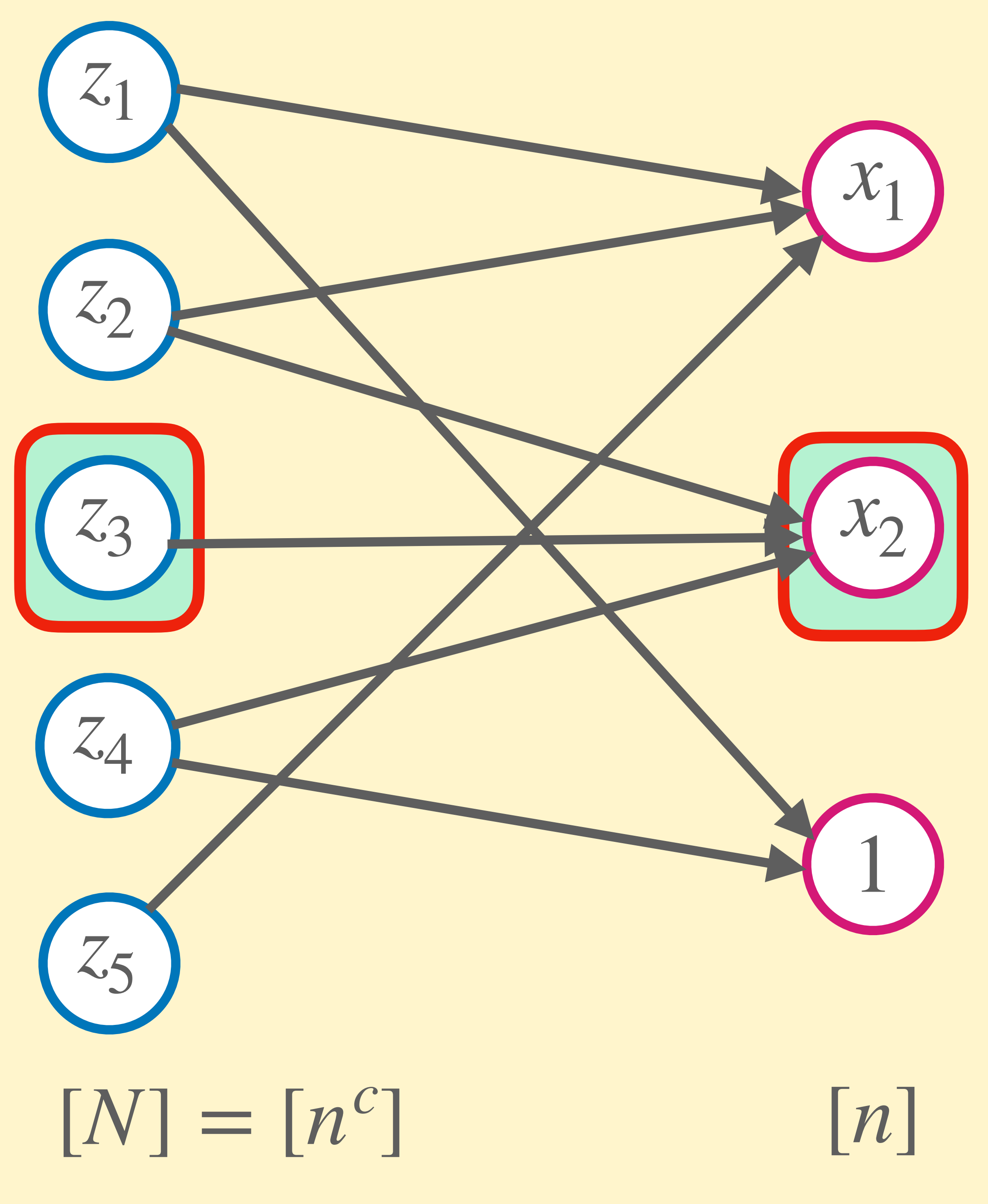
If Prover queries x_i:

- If there are ≥ 2 variables in $\text{N}(z_j)$ for every $z_j \in \text{N}(x_i)$: set x_i arbitrarily
- If x_i is the last variable in $\text{N}(z_j)$ (for some z_j) not set in ρ:

(New) Proof of Depth Condensation

High Level of Proof:

If $\text{depth}_{\text{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F



Adversary strategy for $F \circ \text{XOR}_G$:

If Prover queries x_i:

- If there are ≥ 2 variables in $\text{N}(z_j)$ for every $z_j \in \text{N}(x_i)$: set x_i arbitrarily
- If x_i is the last variable in $\text{N}(z_j)$ (for some z_j) not set in ρ:

(New) Proof of Depth Condensation

High Level of Proof:

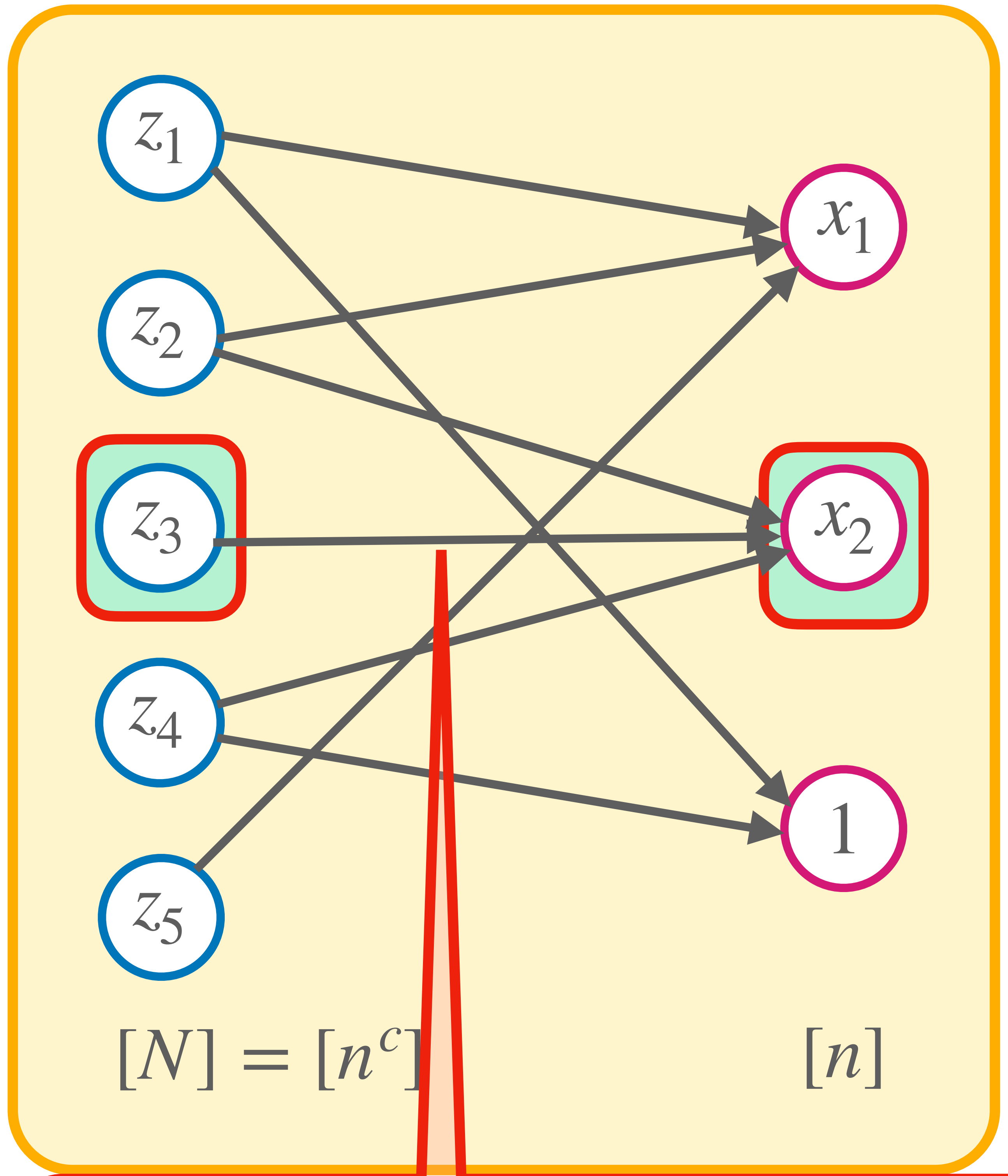
If $\text{depth}_{\text{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F

Adversary strategy for $F \circ \text{XOR}_G$:

If Prover queries x_i:

- If there are ≥ 2 variables in $\text{N}(z_j)$ for every $z_j \in \text{N}(x_i)$: set x_i arbitrarily
- If x_i is the last variable in $\text{N}(z_j)$ (for some z_j) not set in ρ:

  — Query A for the value b of z_j on state $\text{XOR}_G(\rho)$.

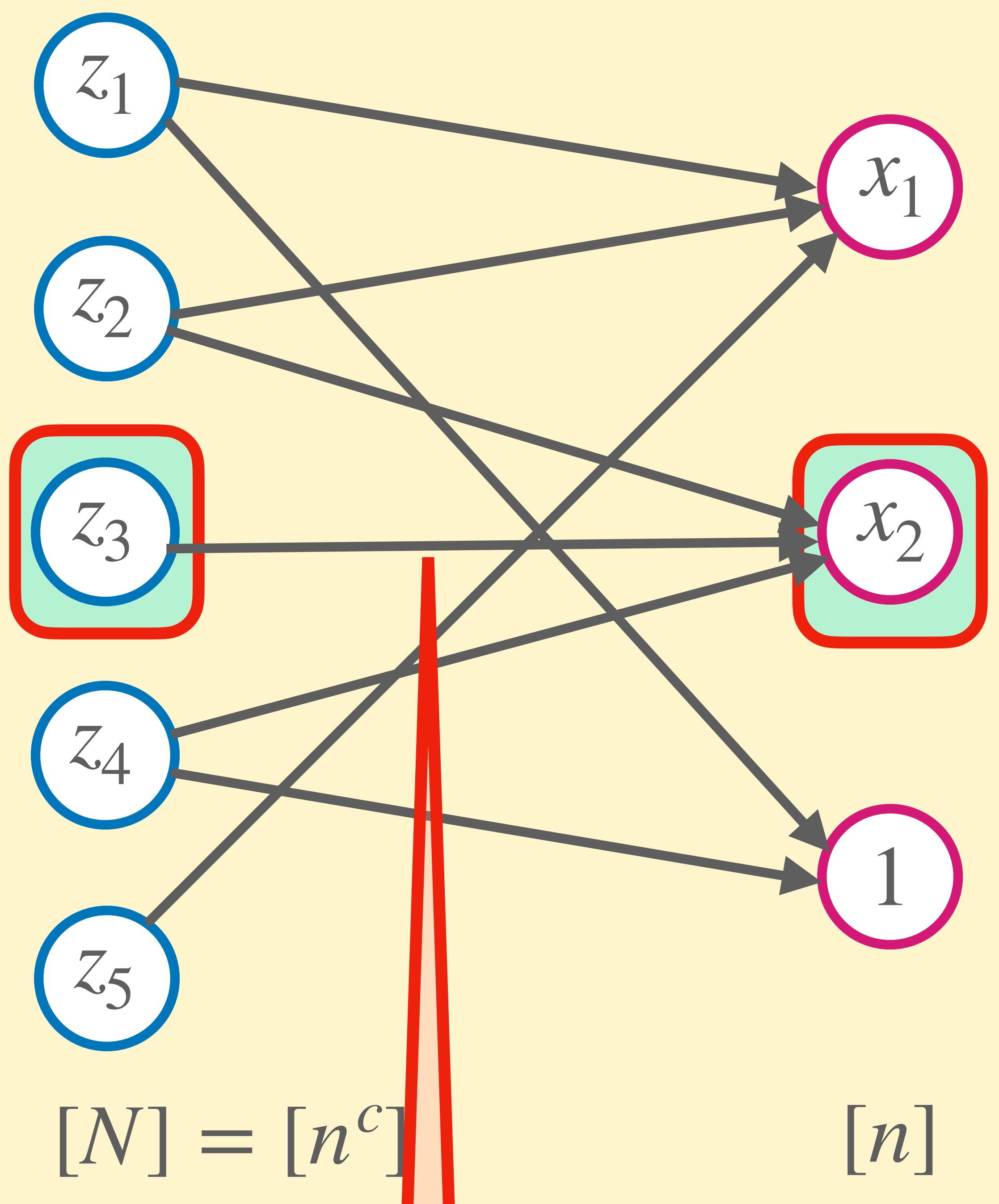


$\text{XOR}_G(\rho) = [\,*\,,*\,,*\,,*\,,*\,]$

(New) Proof of Depth Condensation

High Level of Proof:

If $\text{depth}_{\text{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F



Adversary strategy for $F \circ \text{XOR}_G$:

If Prover queries x_i:

- If there are ≥ 2 variables in $\text{N}(z_j)$ for every $z_j \in \text{N}(x_i)$: set x_i arbitrarily
- If x_i is the last variable in $\text{N}(z_j)$ (for some z_j) not set in ρ:
  — Query A for the value b of z_j on state $\text{XOR}_G(\rho)$.

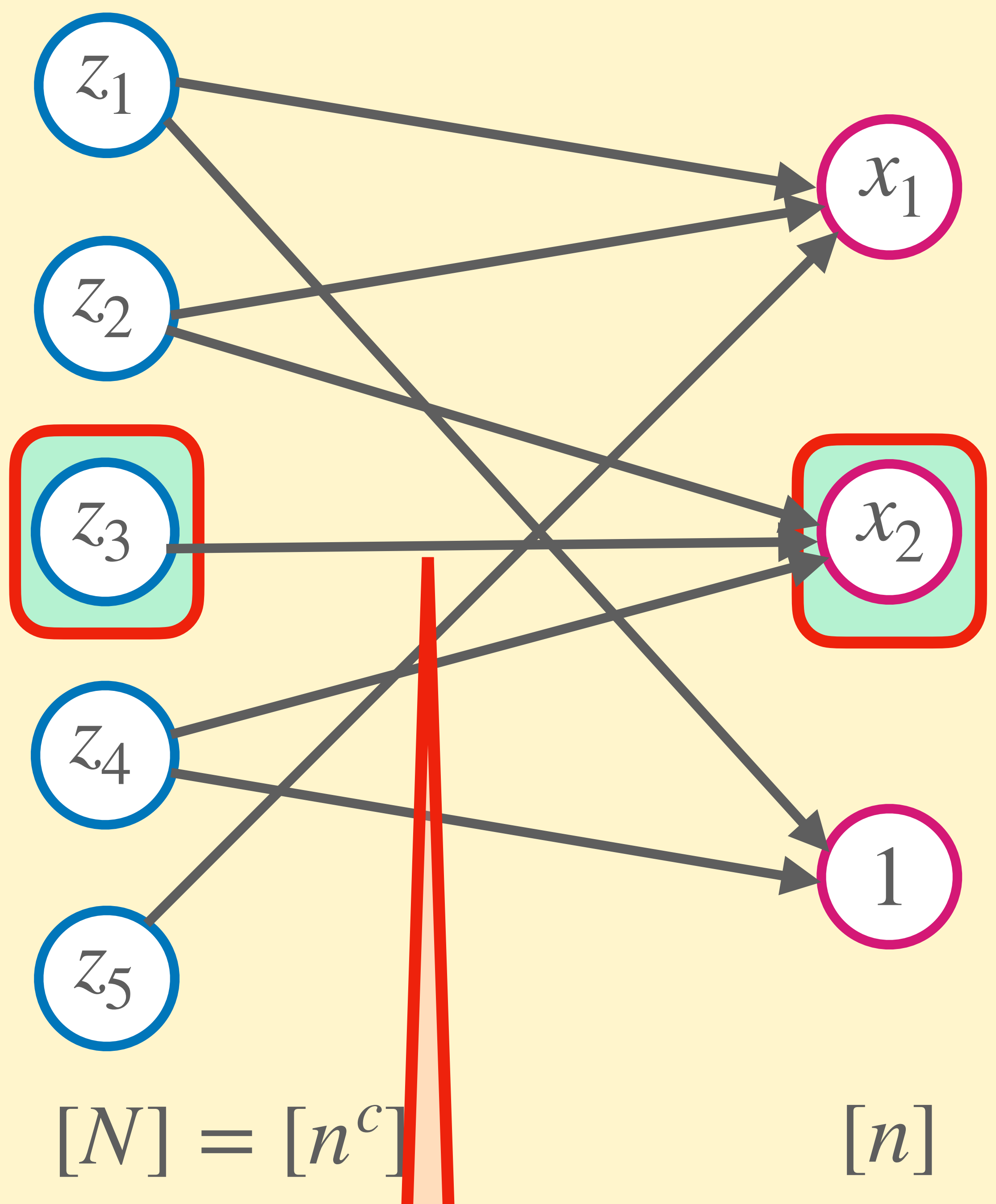
$\text{XOR}_G(\rho) = [\,*\,,*\,,*\,,*\,,*\,]$

A: set $z_3 = 1$

(New) Proof of Depth Condensation

High Level of Proof:

If $\text{depth}_{\text{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F



Adversary strategy for $F \circ \text{XOR}_G$:

If Prover queries x_i:

- If there are ≥ 2 variables in $\text{N}(z_j)$ for every $z_j \in \text{N}(x_i)$: set x_i arbitrarily
- If x_i is the last variable in $\text{N}(z_j)$ (for some z_j) not set in ρ:
  - Query A for the value b of z_j on state $\text{XOR}_G(\rho)$.
  - Set x_i so that $\bigoplus_{t: x_t \in N(z_j)} x_t = b$

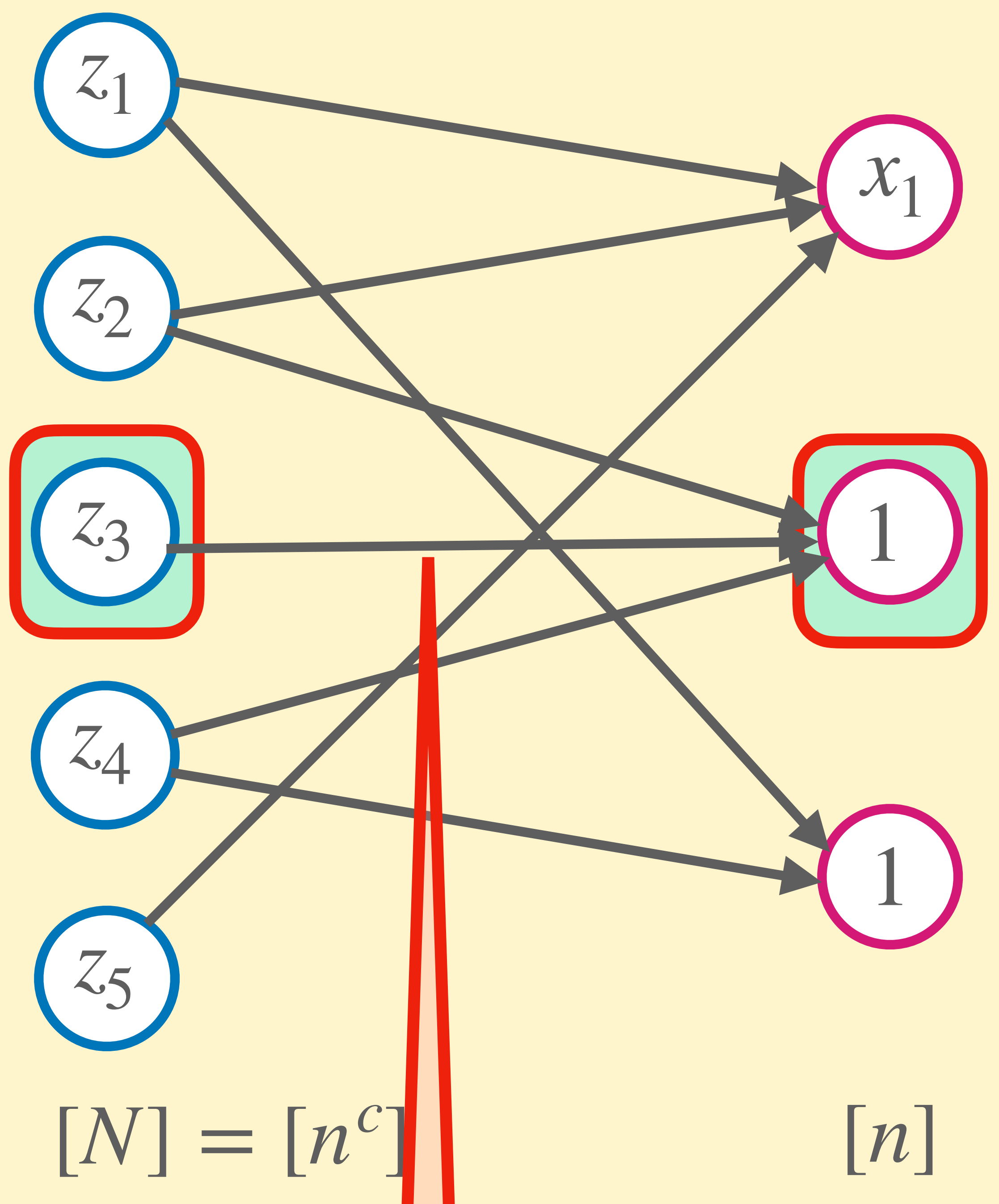
$\text{XOR}_G(\rho) = [\, *, *, *, *, * \,]$

A: set $z_3 = 1$

(New) Proof of Depth Condensation

High Level of Proof:

If $\text{depth}_{\text{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F



Adversary strategy for $F \circ \text{XOR}_G$:

If Prover queries x_i:

- If there are ≥ 2 variables in $\text{N}(z_j)$ for every $z_j \in \text{N}(x_i)$: set x_i arbitrarily
- If x_i is the last variable in $\text{N}(z_j)$ (for some z_j) not set in ρ:
  - Query A for the value b of z_j on state $\text{XOR}_G(\rho)$.
  - Set x_i so that $\bigoplus_{t:x_t \in N(z_j)} x_t = b$

$\text{XOR}_G(\rho) = [\, *\,, *\,, *\,, *\,, *\,]$

A: set $z_3 = 1$

(New) Proof of Depth Condensation

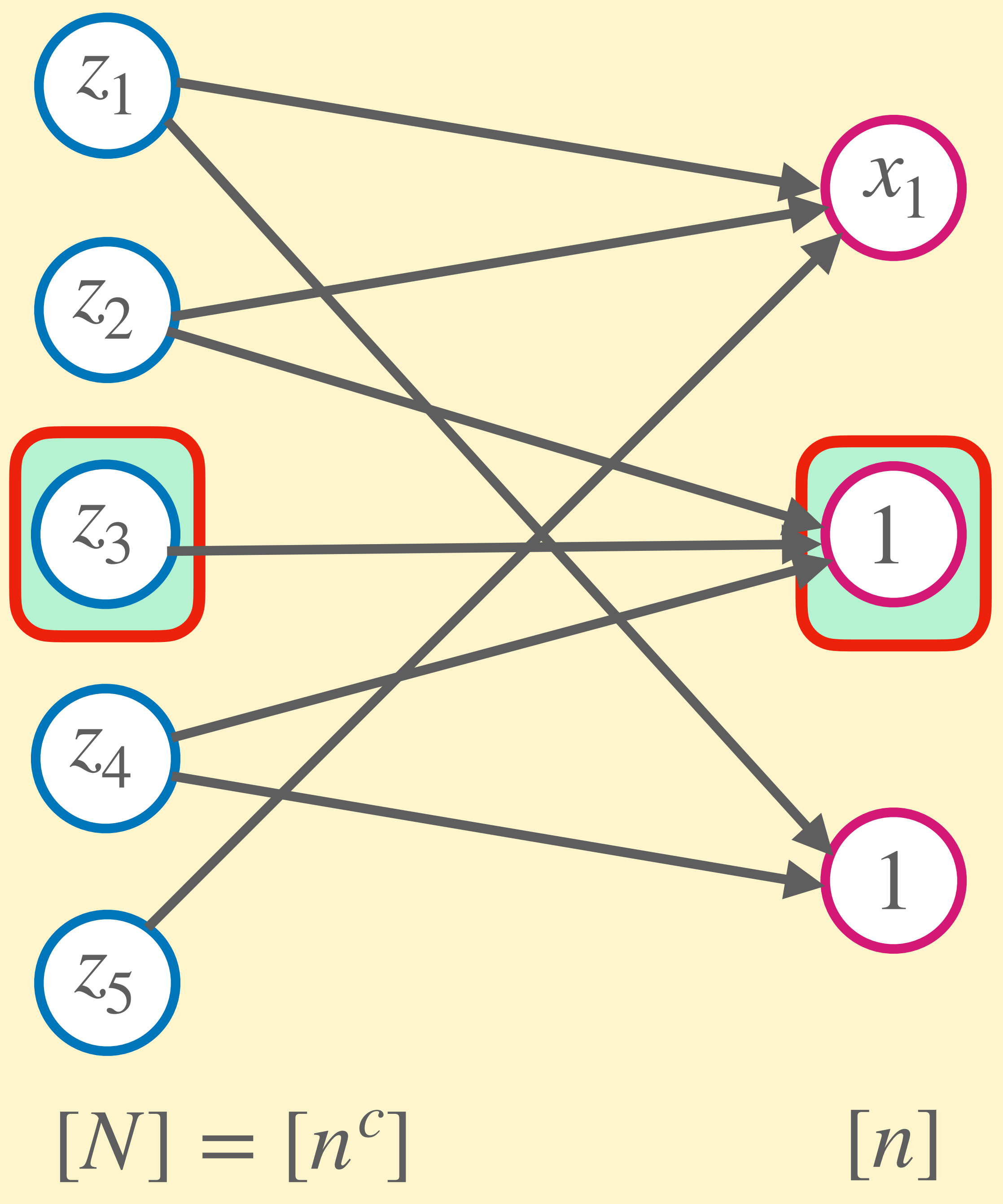
High Level of Proof:

If $\text{depth}_{\text{Res}}(F) \geq d \implies$ exists a strategy A for the Adversary to survive d rounds in the unbounded game on F

Adversary strategy for $F \circ \text{XOR}_G$:

If Prover queries x_i:

- If there are ≥ 2 variables in $\text{N}(z_j)$ for every $z_j \in \text{N}(x_i)$: set x_i arbitrarily
- If x_i is the last variable in $\text{N}(z_j)$ (for some z_j) not set in ρ:
  - — Query A for the value b of z_j on state $\text{XOR}_G(\rho)$.
  - — Set x_i so that $\bigoplus_{t: x_t \in N(z_j)} x_t = b$



Problem!

This forces $z_4 = 0$

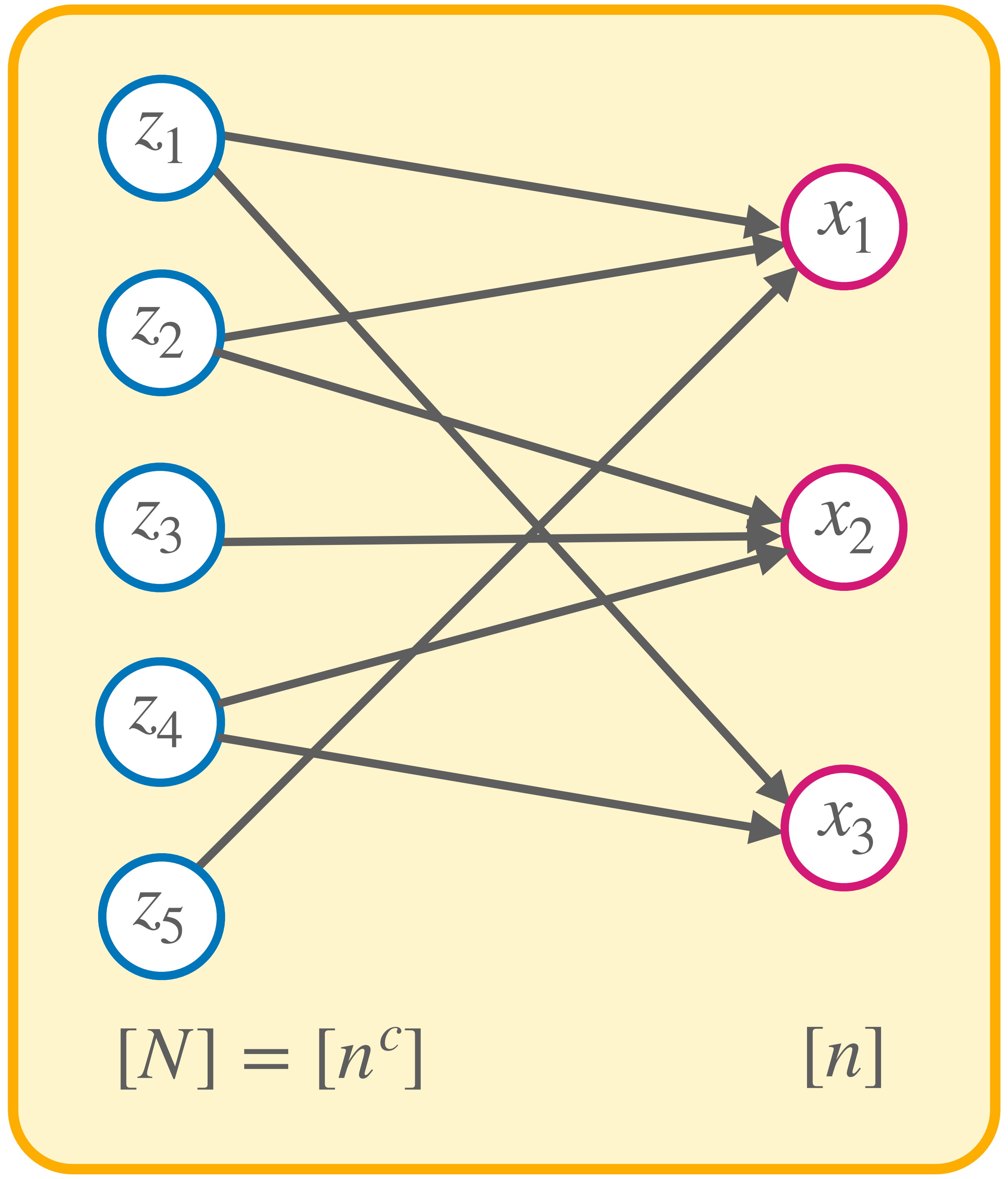
What if A sets $z_4 = 1$?

Proof Overview

Problem: z-variables are correlated

$\rightarrow$ Setting one can x-variable can force several z-variables

$\rightarrow$ Cannot follow A in this case



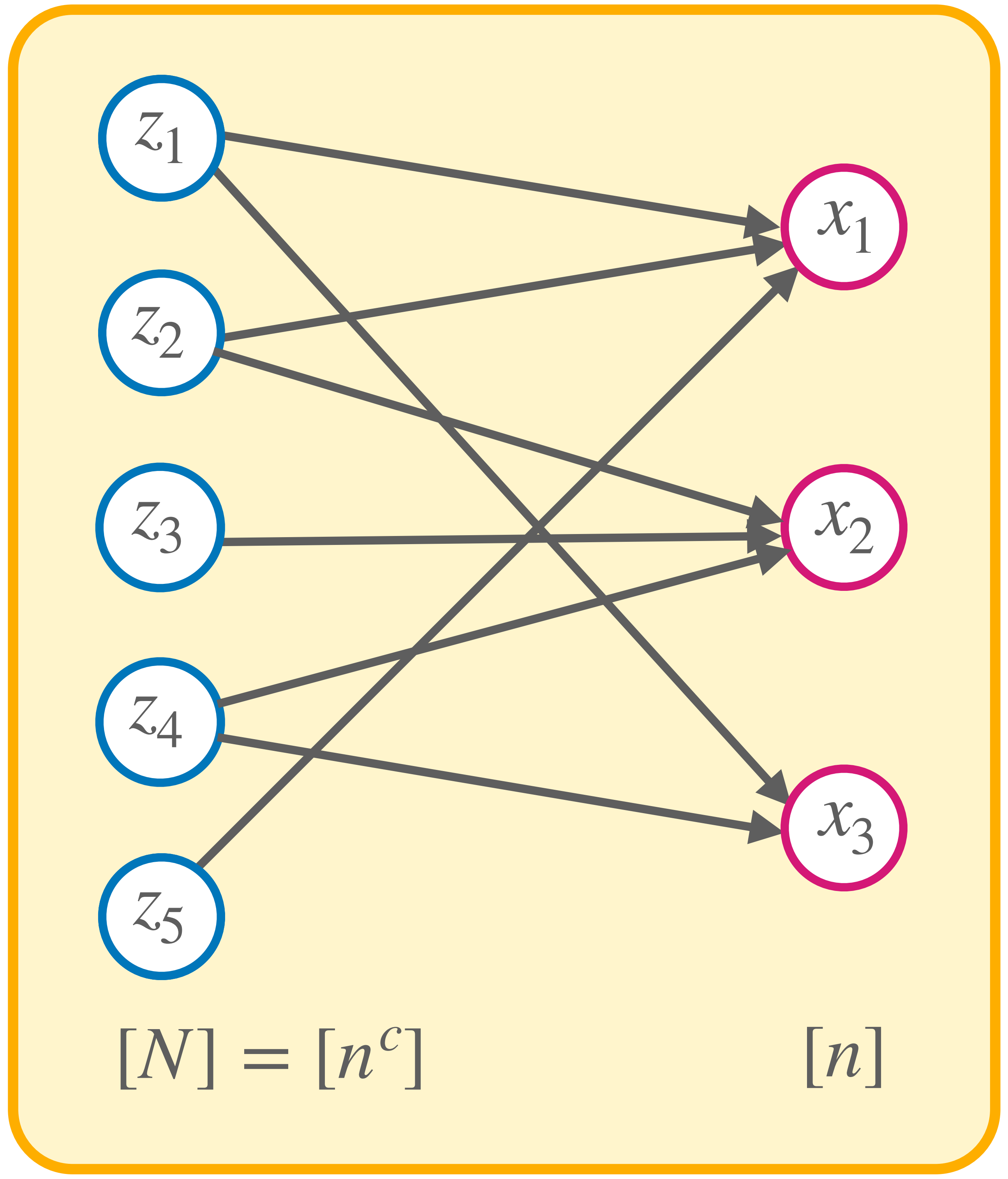
Proof Overview

Problem: z-variables are correlated

→Setting one can x-variable can force several z-variables

→Cannot follow A in this case

Use **expansion** to avoid this scenario!



Proof Overview

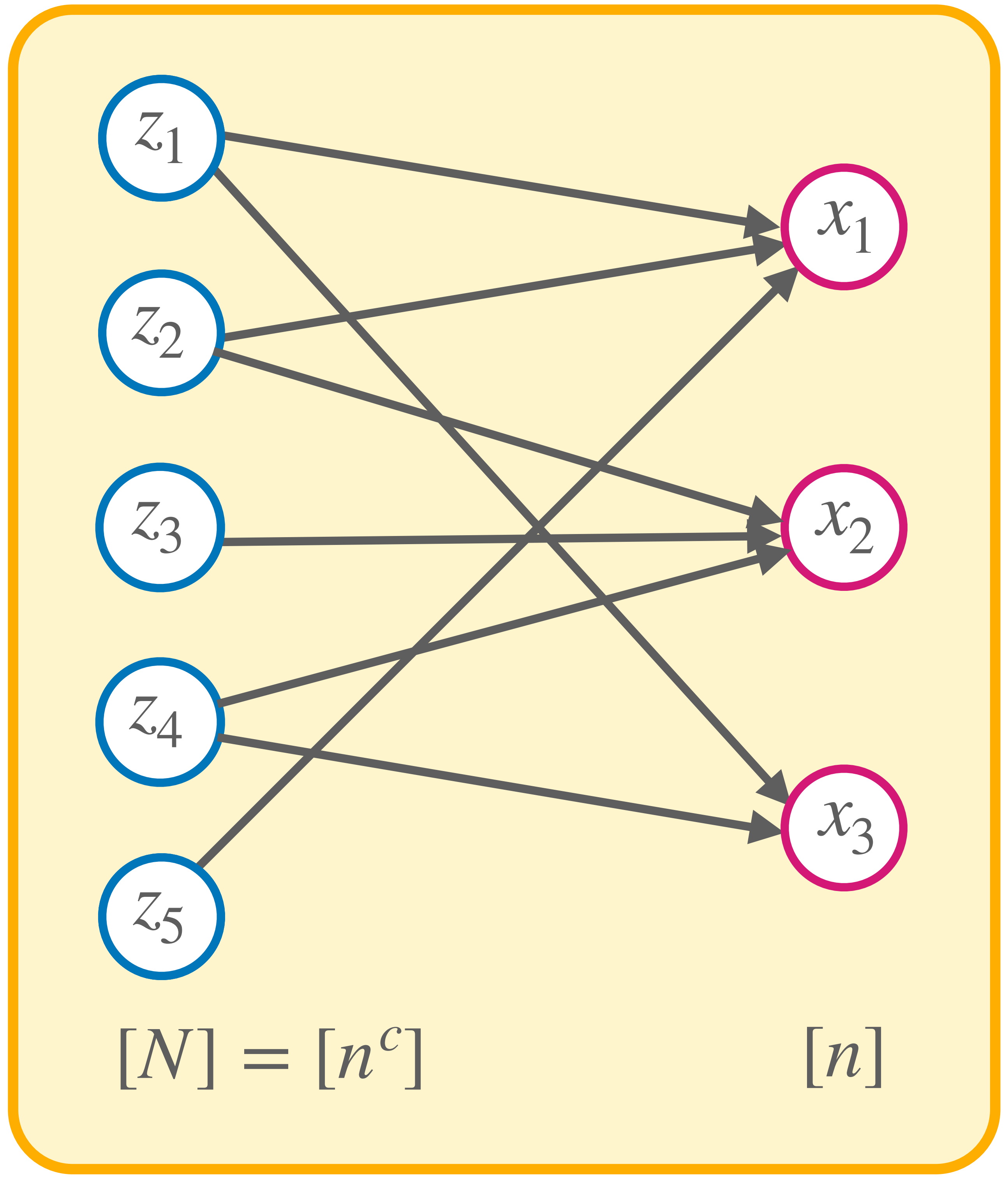
Problem: z-variables are correlated

$\rightarrow$ Setting one can x-variable can force several z-variables

$\rightarrow$ Cannot follow A in this case

Use **expansion** to avoid this scenario!

Let $G \backslash \rho$ be induced by removing the x-variables set by ρ and z-variables determined by ρ



Proof Overview

Problem: z-variables are correlated

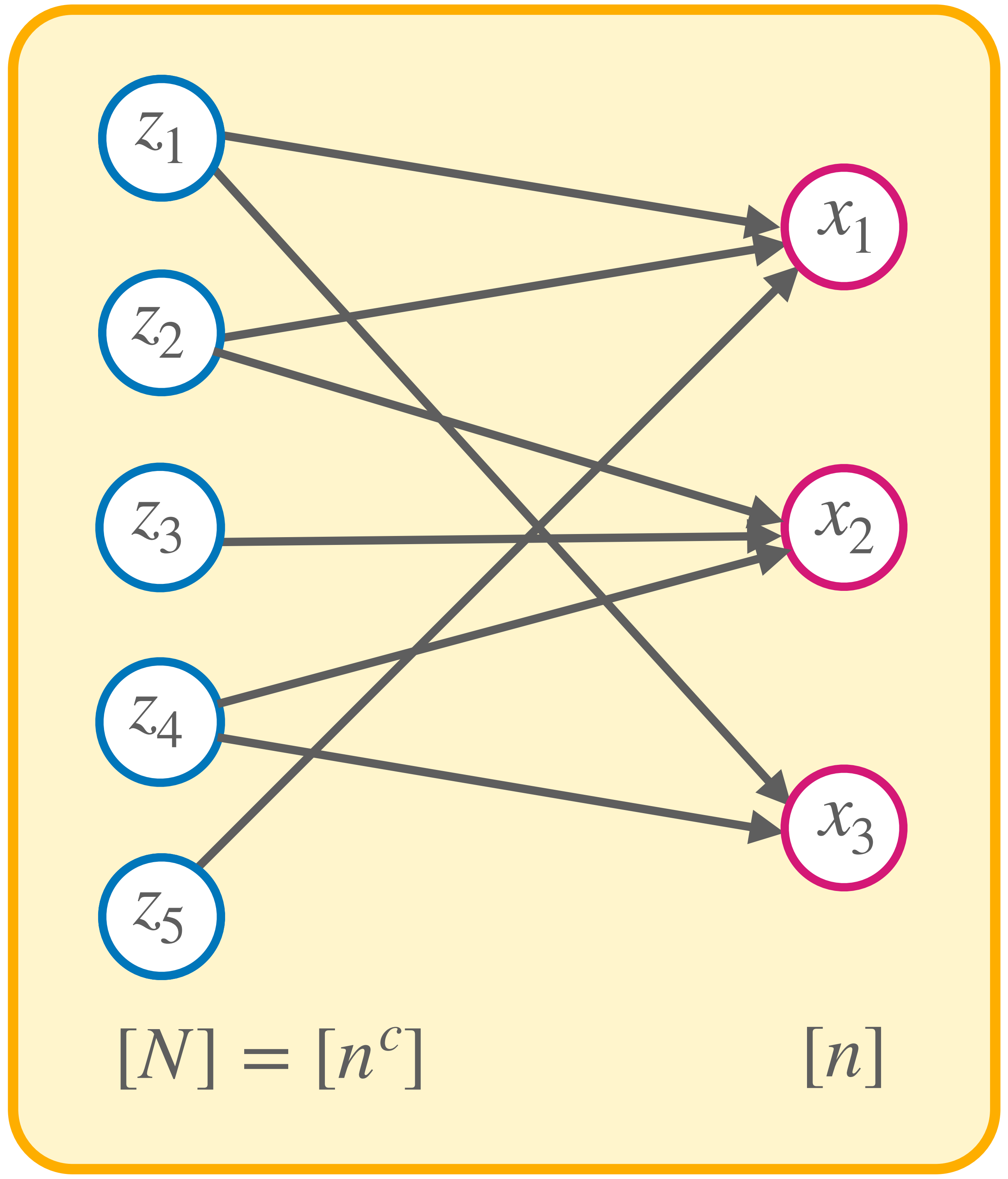
$\rightarrow$Setting one can x-variable can force several z-variables

$\rightarrow$Cannot follow A in this case

Use **expansion** to avoid this scenario!

Let $G \backslash \rho$ be induced by removing the x-variables set by ρ and z-variables determined by ρ

e.g. $\rho = [1, *, 0]$ then $G \backslash \rho$ is:



Proof Overview

Problem: z-variables are correlated

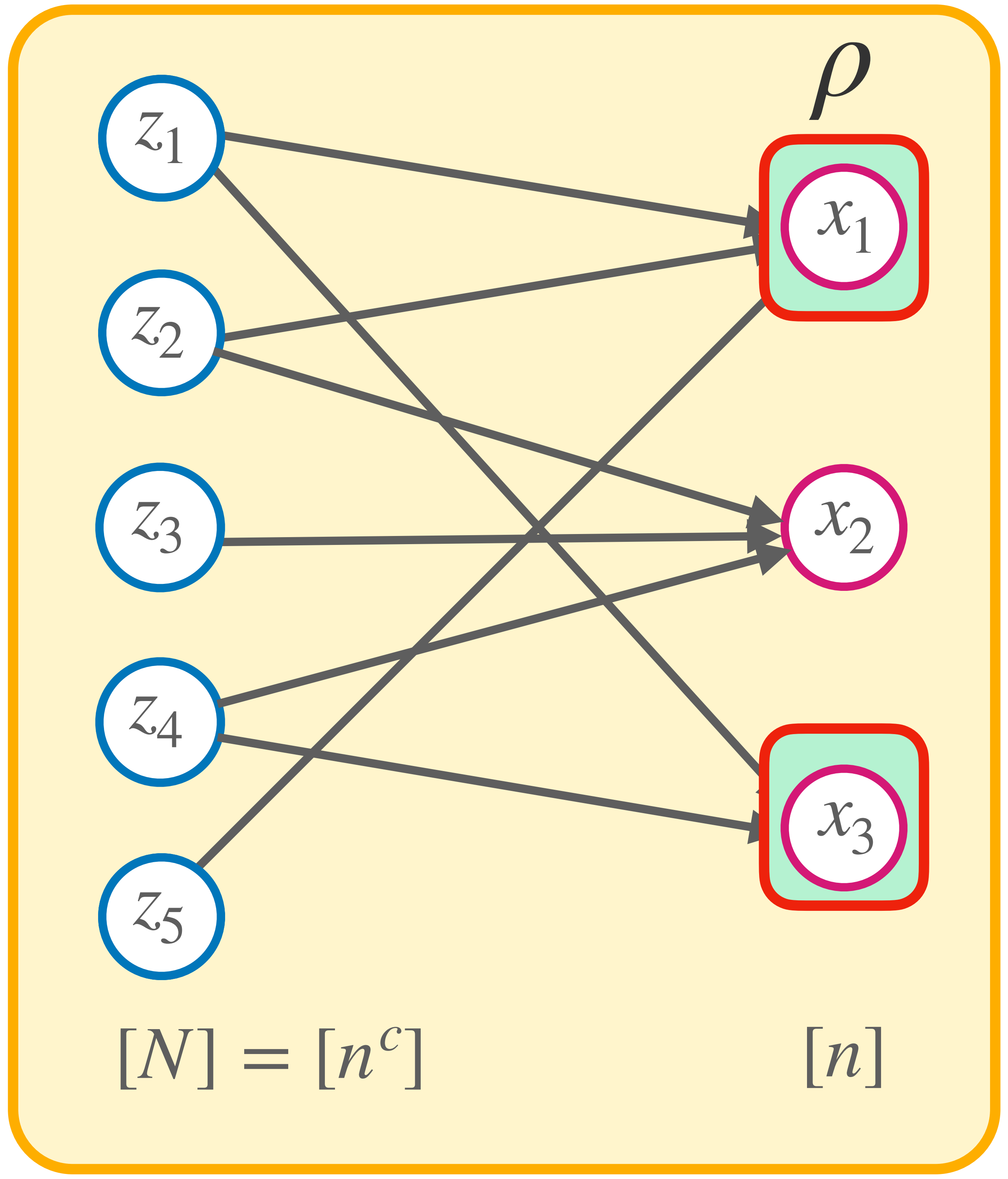
→Setting one can x-variable can force several z-variables

→Cannot follow A in this case

Use **expansion** to avoid this scenario!

Let $G\backslash\rho$ be induced by removing the x-variables set by ρ and z-variables determined by ρ

e.g. $\rho = [1, *, 0]$ then $G\backslash\rho$ is:



Proof Overview

Problem: z-variables are correlated

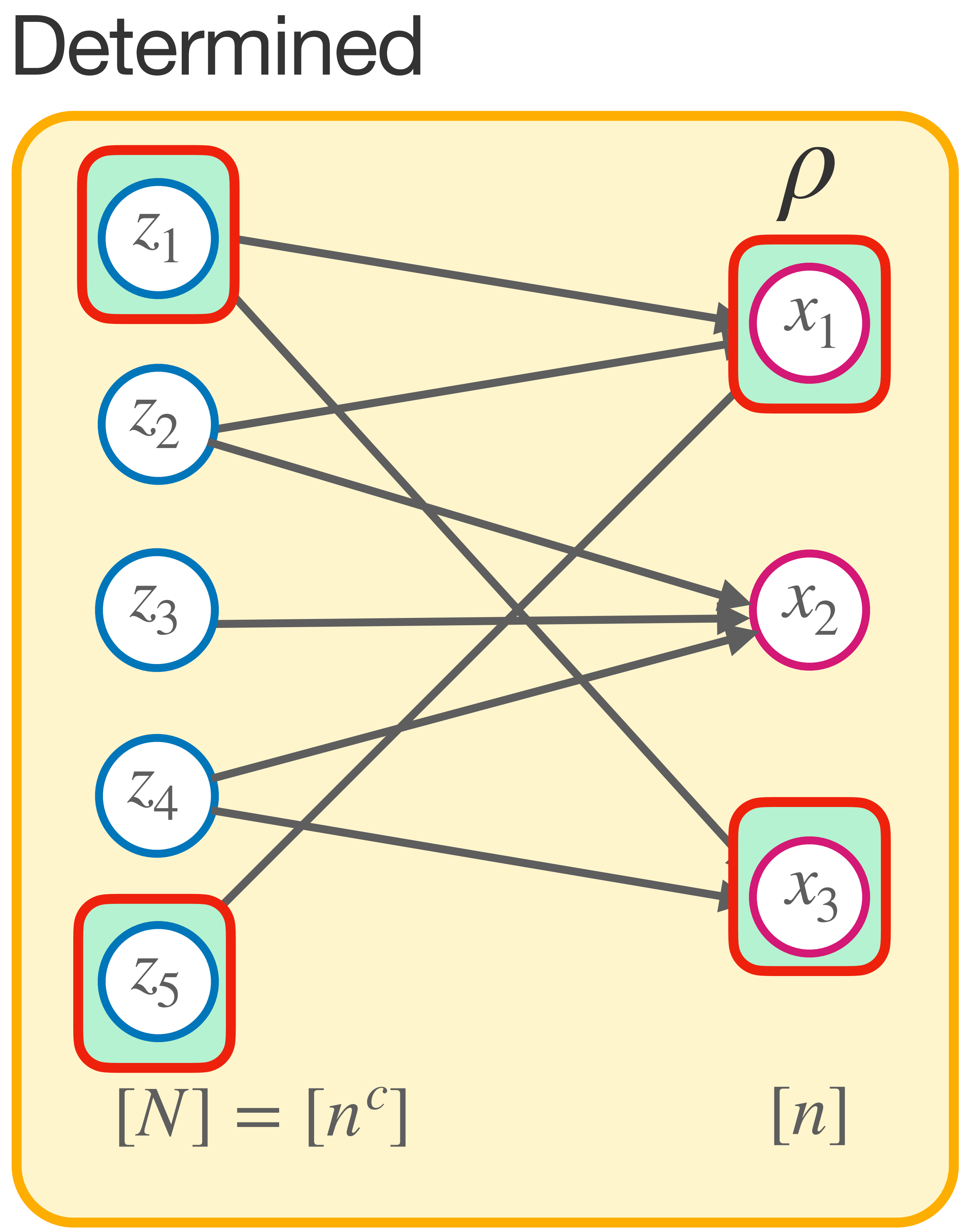
$\rightarrow$Setting one can x-variable can force several z-variables

$\rightarrow$Cannot follow A in this case

Use **expansion** to avoid this scenario!

Let $G \backslash \rho$ be induced by removing the x-variables set by ρ and z-variables determined by ρ

e.g. $\rho = [1, *, 0]$ then $G \backslash \rho$ is:



Proof Overview

Problem: z-variables are correlated

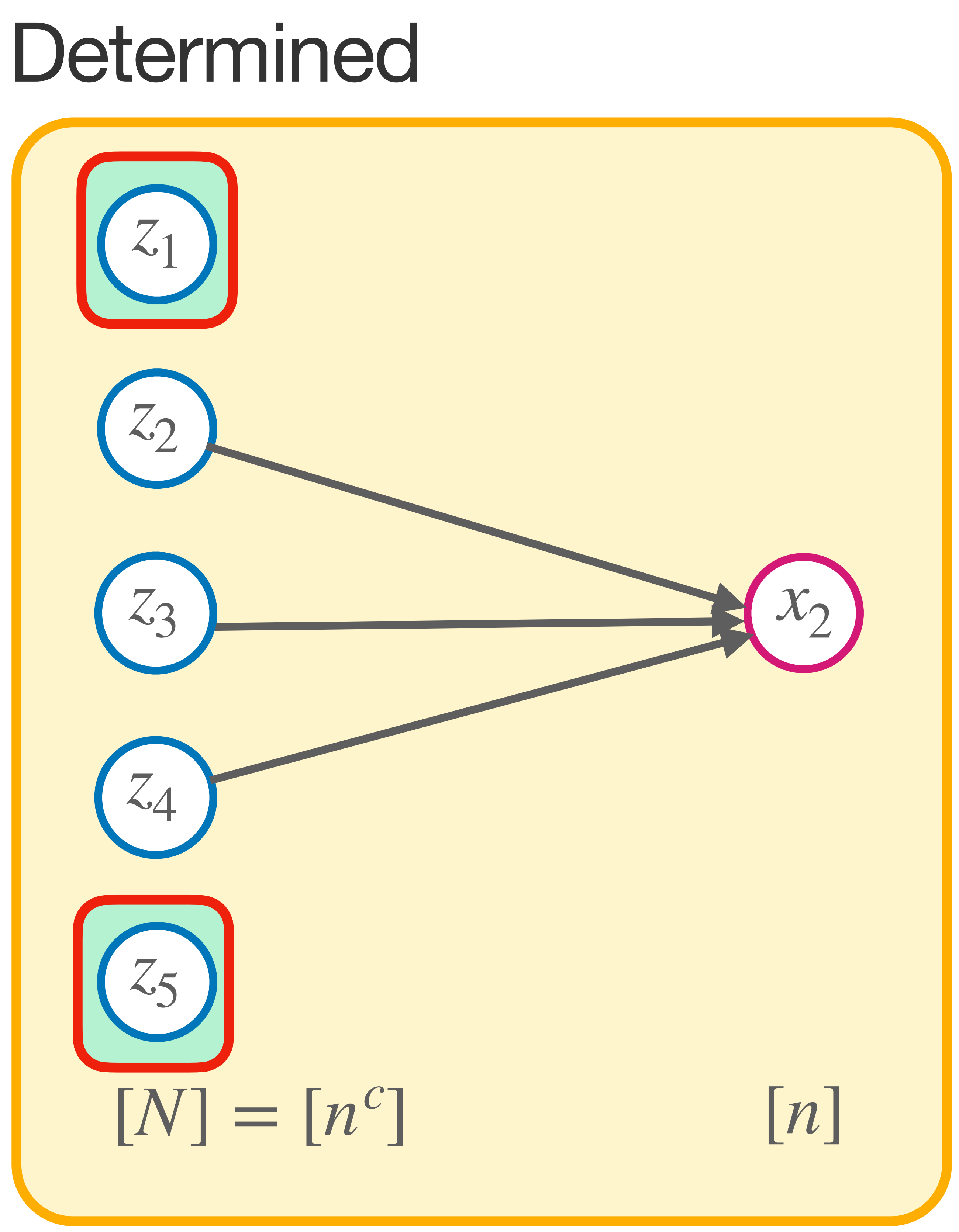
$\rightarrow$Setting one can x-variable can force several z-variables

$\rightarrow$Cannot follow A in this case

Use **expansion** to avoid this scenario!

Let $G\backslash\rho$ be induced by removing the x-variables set by ρ and z-variables determined by ρ

e.g. $\rho = [1, *, 0]$ then $G\backslash\rho$ is:



Proof Overview

Problem: z-variables are correlated

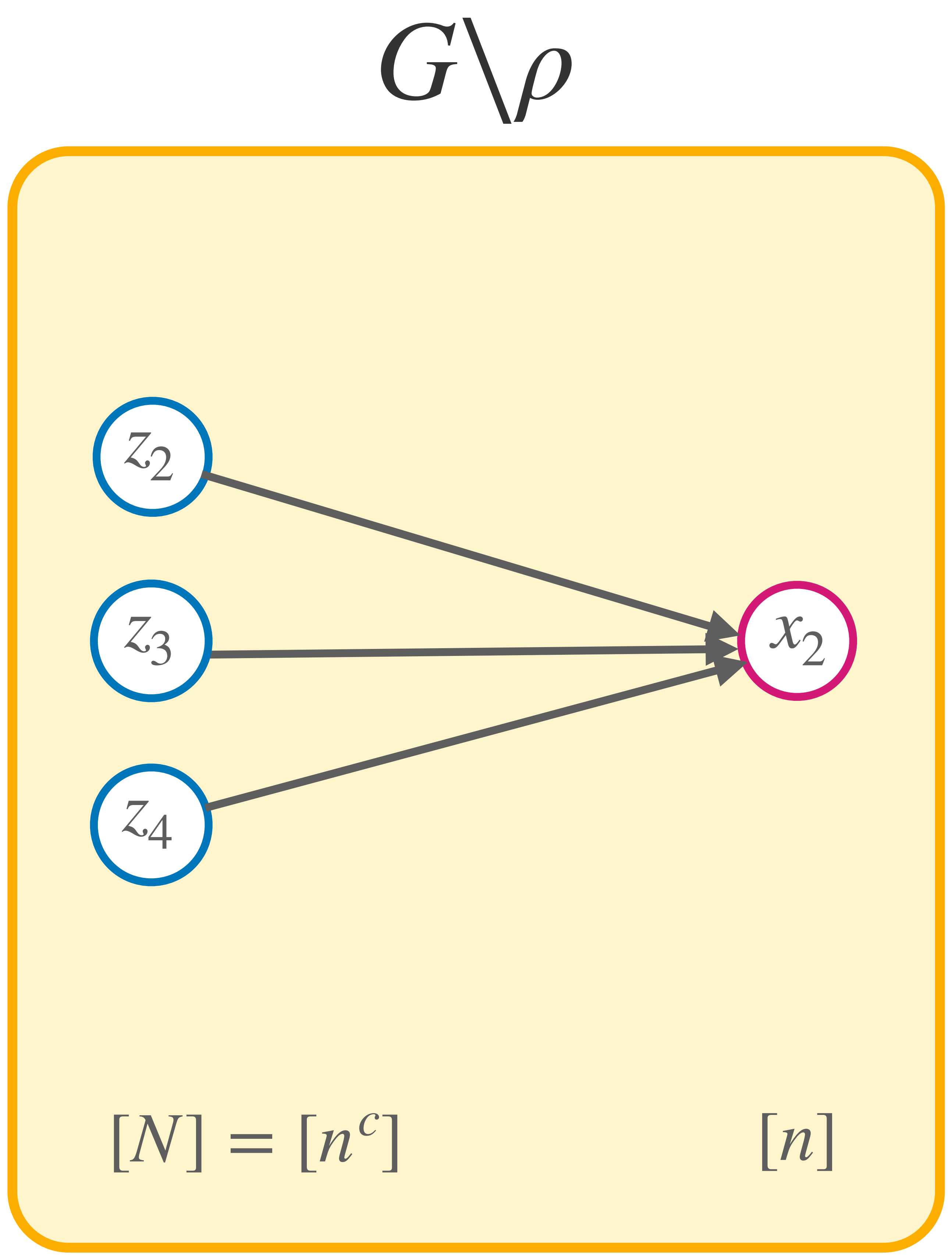
→Setting one can x-variable can force several z-variables

→Cannot follow A in this case

Use **expansion** to avoid this scenario!

Let $G\backslash\rho$ be induced by removing the x-variables set by ρ and z-variables determined by ρ

e.g. $\rho = [1, *, 0]$ then $G\backslash\rho$ is:



Proof Overview

Problem: z-variables are correlated

$\rightarrow$Setting one can x-variable can force several z-variables

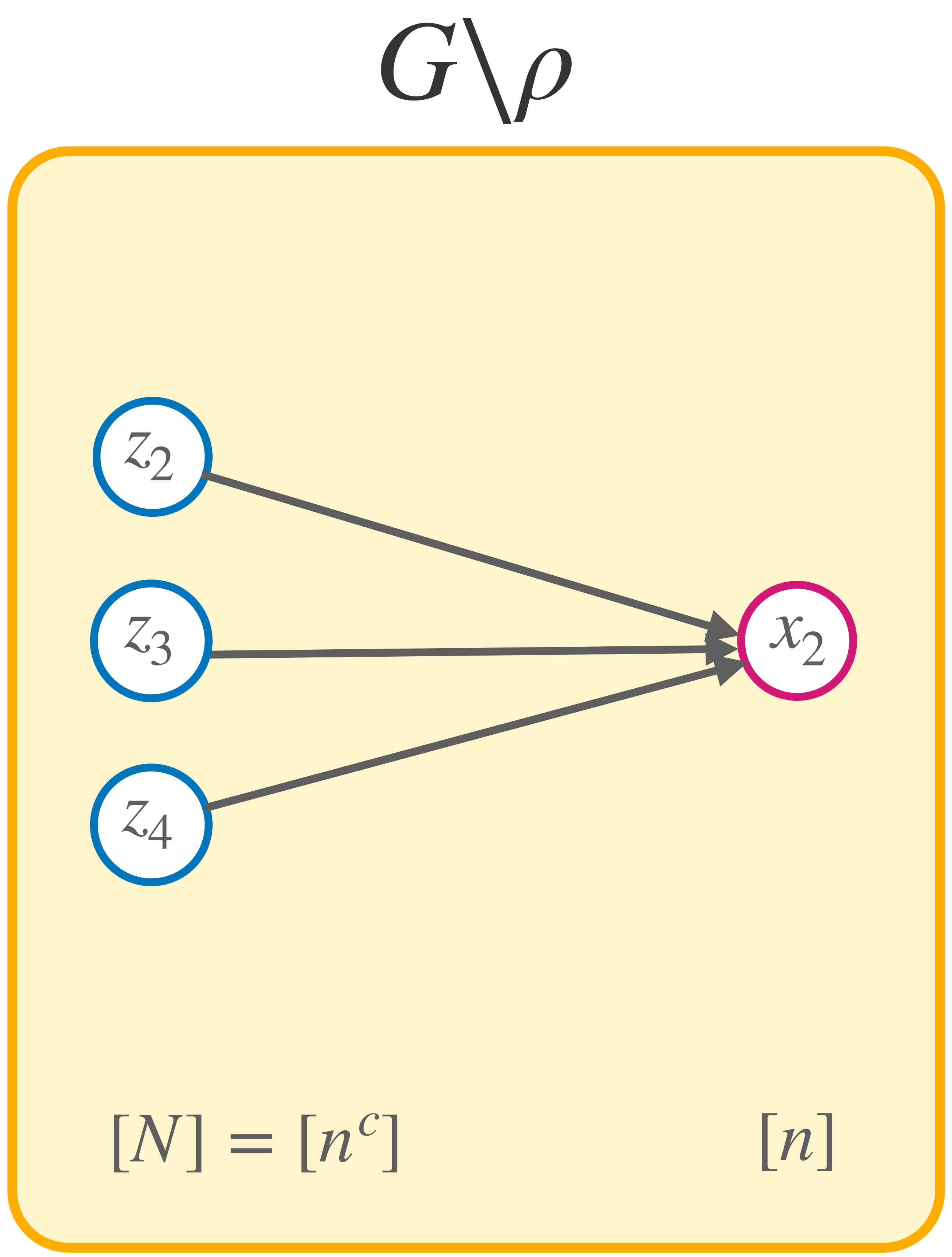
$\rightarrow$Cannot follow A in this case

Use **expansion** to avoid this scenario!

Let $G\backslash\rho$ be induced by removing the x-variables set by ρ and z-variables determined by ρ

e.g. $\rho = [1, *, 0]$ then $G\backslash\rho$ is:

Invariant: $G\backslash\rho$ is r-expanding



Proof Overview

Problem: z-variables are correlated

→Setting one can x-variable can force several z-variables

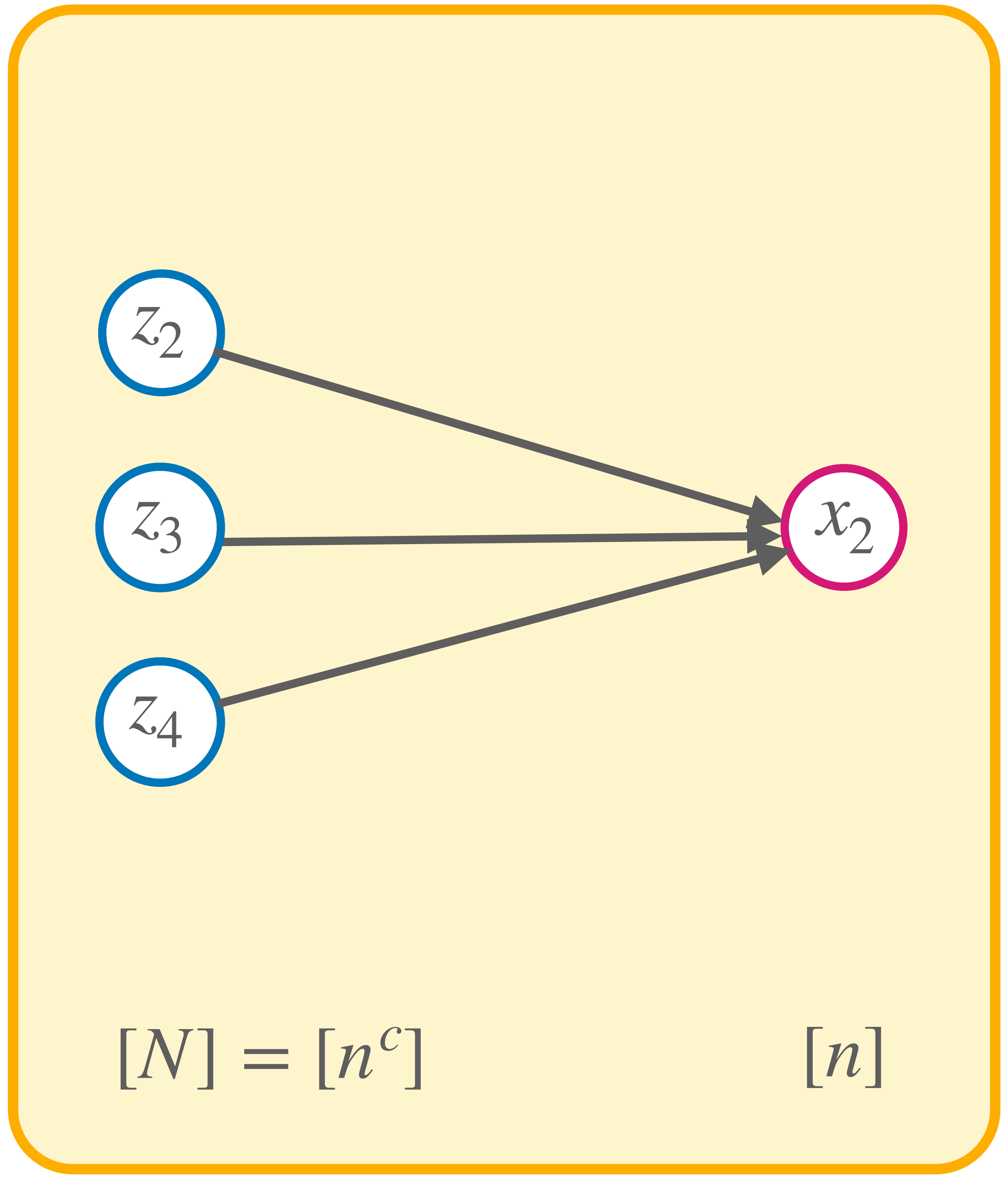
→Cannot follow A in this case

Use **expansion** to avoid this scenario!

Let $G\backslash\rho$ be induced by removing the x-variables set by ρ and z-variables determined by ρ

e.g. $\rho = [1, * ,0]$ then $G\backslash\rho$ is:

$G\backslash\rho$



Invariant: $G\backslash\rho$ is r-expanding

→ Setting any x_i doesn't determine any z-variable

Expansion Restoration

However… after setting an x_i, $G \backslash \rho$ may no longer be r-expanding

Expansion Restoration

However… after setting an x_i, $G \backslash \rho$ may no longer be r-expanding

$\rightarrow$ Query additional x-variables to restore expansion!

Expansion Restoration

However… after setting an x_i, $G\backslash\rho$ may no longer be r-expanding

$\rightarrow$ Query additional x-variables to restore expansion!

Note: Want to assign as few z-variables while doing this

— Each time we fix a z-variable we have to query A. Can only do this d times

Expansion Restoration

However… after setting an x_i, $G \backslash \rho$ may no longer be r-expanding

$\rightarrow$ Query additional x-variables to restore expansion!

Note: Want to assign as few z-variables while doing this

— Each time we fix a z-variable we have to query A. Can only do this d times

Closure Lemma: If $G \backslash \rho$ is r-expanding and ρ' is obtained by querying some x_i, then there exists $\mathsf{Cl}(\rho') \supseteq \mathsf{vars}(\rho')$ such that

Expansion Restoration

However… after setting an x_i, $G\backslash\rho$ may no longer be r-expanding

$\rightarrow$ Query additional x-variables to restore expansion!

Note: Want to assign as few z-variables while doing this

— Each time we fix a z-variable we have to query A. Can only do this d times

Closure Lemma: If $G\backslash\rho$ is r-expanding and ρ' is obtained by querying some x_i, then there exists $\mathsf{Cl}(\rho') \supseteq \mathsf{vars}(\rho')$ such that

1. $Cl(\rho')$ fixes only a few z-variables $|\mathsf{Fixed}(\mathsf{Cl}(\rho'))| \leq 2w$

Expansion Restoration

However… after setting an x_i, $G\backslash\rho$ may no longer be r-expanding

$\rightarrow$ Query additional x-variables to restore expansion!

Note: Want to assign as few z-variables while doing this

— Each time we fix a z-variable we have to query A. Can only do this d times

Closure Lemma: If $G\backslash\rho$ is r-expanding and ρ' is obtained by querying some x_i, then there exists $\mathrm{Cl}(\rho') \supseteq \mathrm{vars}(\rho')$ such that

1. $Cl(\rho')$ fixes only a few z-variables $|\mathrm{Fixed}(\mathrm{Cl}(\rho'))| \leq 2w$
2. $G\backslash\mathrm{Cl}(\rho')$ is an r-expander

Expansion Restoration

However… after setting an x_i, $G\backslash\rho$ may no longer be r-expanding

$\rightarrow$ Query additional x-variables to restore expansion!

Note: Want to assign as few z-variables while doing this

— Each time we fix a z-variable we have to query A. Can only do this d times

Closure Lemma: If $G\backslash\rho$ is r-expanding and ρ' is obtained by querying some x_i, then there exists $\mathsf{Cl}(\rho') \supseteq \mathsf{vars}(\rho')$ such that

1. $Cl(\rho')$ fixes only a few z-variables $|\mathsf{Fixed}(\mathsf{Cl}(\rho'))| \leq 2w$
2. $G\backslash\mathsf{Cl}(\rho')$ is an r-expander
3. The variables of $\mathsf{Cl}(\rho')\backslash\mathsf{vars}(\rho')$ can be set consistently with A

Expansion Restoration

However… after setting an x_i, $G\backslash\rho$ may no longer be r-expanding

→ Query additional x-variables to restore expansion!

Note: Want to assign as few z-variables while doing this

— Each time we fix a z-variable we have to query A. Can only do this d times

Closure Lemma: If $G\backslash\rho$ is r-expanding and ρ' is obtained by querying some x_i, then there exists $\mathrm{Cl}(\rho') \supseteq \mathrm{vars}(\rho')$ such that

1. $Cl(\rho')$ fixes only a few z-variables $|\mathrm{Fixed}(\mathrm{Cl}(\rho'))| \leq 2w$
2. $G\backslash\mathrm{Cl}(\rho')$ is an r-expander
3. The variables of $\mathrm{Cl}(\rho')\backslash\mathtt{vars}(\rho')$ can be set consistently with A

→ To restore expansion, set the variables of $\mathrm{Cl}(\rho')\backslash\mathtt{vars}(\rho')$!

Adversary Strategy

If $\mathrm{depth}_{\mathrm{Res}}(F) \geq d \implies \exists$ strategy A for the Adversary to survive d rounds on F

Adversary strategy for w-bounded game on $F \circ \mathrm{XOR}_G$ simulates A as follows:

Adversary Strategy

If $\mathrm{depth}_{\mathrm{Res}}(F) \geq d \implies \exists$ strategy A for the Adversary to survive d rounds on F

Adversary strategy for w-bounded game on $F \circ \mathrm{XOR}_G$ simulates A as follows:

Invariant: $G \backslash \rho$ is an r-boundary expander

Adversary Strategy

If $\text{depth}_{\text{Res}}(F) \geq d \implies \exists$ strategy A for the Adversary to survive d rounds on F

Adversary strategy for w-bounded game on $F \circ \text{XOR}_G$ simulates A as follows:

Invariant: $G \backslash \rho$ is an r-boundary expander

Query: If Prover asks for the value of x_i

$\rightarrow$ Set x_i arbitrarily

Adversary Strategy

If $\mathrm{depth}_{\mathrm{Res}}(F) \geq d \Longrightarrow \exists$ strategy A for the Adversary to survive d rounds on F

Adversary strategy for w-bounded game on $F \circ \mathrm{XOR}_G$ simulates A as follows:

Invariant: $G \setminus \rho$ is an r-boundary expander

Query: If Prover asks for the value of x_i

$\rightarrow$ Set x_i arbitrarily — Since $G \setminus \rho$ is expanding, setting x_i doesn't determine any z_j

Adversary Strategy

If $\text{depth}_{\text{Res}}(F) \geq d \implies \exists$ strategy A for the Adversary to survive d rounds on F

Adversary strategy for w-bounded game on $F \circ \text{XOR}_G$ simulates A as follows:

Invariant: $G \backslash \rho$ is an r-boundary expander

Query: If Prover asks for the value of x_i

$\rightarrow$ Set x_i arbitrarily — Since $G \backslash \rho$ is expanding, setting x_i doesn't determine any z_j

Restore Expansion: Set the variables in $\text{Cl}(\rho)$ consistent with A

Adversary Strategy

If $\text{depth}_{\text{Res}}(F) \geq d \implies \exists$ strategy A for the Adversary to survive d rounds on F

Adversary strategy for w-bounded game on $F \circ \text{XOR}_G$ simulates A as follows:

Invariant: $G \backslash \rho$ is an r-boundary expander

Query: If Prover asks for the value of x_i

$\rightarrow$ Set x_i arbitrarily — Since $G \backslash \rho$ is expanding, setting x_i doesn't determine any z_j

Restore Expansion: Set the variables in $\text{Cl}(\rho)$ consistent with A

$\rightarrow$ By Closure Lemma, A is queried at most $2w$ times.

Adversary Strategy

If $\text{depth}_{\text{Res}}(F) \geq d \implies \exists$ strategy A for the Adversary to survive d rounds on F

Adversary strategy for w-bounded game on $F \circ \text{XOR}_G$ simulates A as follows:

Invariant: $G \backslash \rho$ is an r-boundary expander

Query: If Prover asks for the value of x_i

$\rightarrow$ Set x_i arbitrarily — Since $G \backslash \rho$ is expanding, setting x_i doesn't determine any z_j

Restore Expansion: Set the variables in $\text{Cl}(\rho)$ consistent with A

$\rightarrow$ By Closure Lemma, A is queried at most $2w$ times.

Each round uses $O(w)$ queries to $A \implies$ we can continue for $\Omega(d/w)$ rounds!

Open Problems

What about supercritical size/depth tradeoffs for other models of computation?

Open Problems

What about supercritical size/depth tradeoffs for other models of computation?

$\rightarrow$ There are functions for which any poly-size monotone circuit requires depth $\Omega(n/\log^{O(1)} n)$ [deRMN+20]

Open Problems

What about supercritical size/depth tradeoffs for other models of computation?

$\rightarrow$ There are functions for which any poly-size monotone circuit requires depth $\Omega(n/\log^{O(1)} n)$ [deRMN+20]

Q. Can this be extended to supercritical?

Open Problems

What about supercritical size/depth tradeoffs for other models of computation?

$\rightarrow$There are functions for which any poly-size monotone circuit requires depth $\Omega(n/\log^{O(1)} n)$ [deRMN+20]

Q. Can this be extended to supercritical?

Interpolation: Any Cutting Planes proof of any F implies a monotone circuits computing an associated function f_F with the same topology [P96, HP17, FPPR17]

Open Problems

What about supercritical size/depth tradeoffs for other models of computation?

$\rightarrow$ There are functions for which any poly-size monotone circuit requires depth $\Omega(n/\log^{O(1)} n)$ [deRMN+20]

Q. Can this be extended to supercritical?

Interpolation: Any Cutting Planes proof of any F implies a monotone circuits computing an associated function f_F with the same topology [P96, HP17, FPPR17]

$\rightarrow$ However, the number of variables of f_F is equal to the number of clauses of F

Open Problems

What about supercritical size/depth tradeoffs for other models of computation?

$\rightarrow$ There are functions for which any poly-size monotone circuit requires depth $\Omega(n/\log^{O(1)} n)$ [deRMN+20]

Q. Can this be extended to supercritical?

Interpolation: Any Cutting Planes proof of any F implies a monotone circuits computing an associated function f_F with the same topology [P96, HP17, FPPR17]

$\rightarrow$ However, the number of variables of f_F is equal to the number of clauses of F

$\Longrightarrow$ Our tradeoffs do not imply supercritical tradeoffs for monotone circuits

Open Problems

What about supercritical size/depth tradeoffs for other models of computation?

$\rightarrow$ There are functions for which any poly-size monotone circuit requires depth $\Omega(n/\log^{O(1)} n)$ [deRMN+20]

Q. Can this be extended to supercritical?

Interpolation: Any Cutting Planes proof of any F implies a monotone circuits computing an associated function f_F with the same topology [P96, HP17, FPPR17]

Conjecture: There exist F on m clauses such that any (quasi)polynomial size Resolution proof requires depth $\Omega(mn^4)$

Open Problems

What about supercritical size/depth tradeoffs for other models of computation?

→There are functions for which any poly-size monotone circuit requires depth $\Omega(n/\log^{O(1)} n)$ [deRMN+20]

Q. Can this be extended to supercritical?

Interpolation: Any Cutting Planes proof of any F implies a monotone circuits computing an associated function f_F with the same topology [P96, HP17, FPPR17]

Conjecture: There exist F on m clauses such that any (quasi)polynomial size Resolution proof requires depth $\Omega(mn^4)$

→Implies supercritical size/depth tradeoffs for monotone circuits

Open Problems

Conjecture: There exist F on m clauses such that any (quasi)polynomial size Resolution proof requires depth $\Omega(mn^4)$

One approach…

Can the Ben-Sasson Wigderson size-width relation be balanced?

Open Problems

Conjecture: There exist F on m clauses such that any (quasi)polynomial size Resolution proof requires depth $\Omega(mn^4)$

One approach…

Can the Ben-Sasson Wigderson size-width relation be balanced?

Problem: Prove or disprove that for any k-CNF F on m clauses

a size s Resolution proof $\Longrightarrow$ a depth $O(m)$ and width $k + O(\sqrt{n \log s})$ proof

Open Problems

Conjecture: There exist F on m clauses such that any (quasi)polynomial size Resolution proof requires depth $\Omega(mn^4)$

One approach…

Can the Ben-Sasson Wigderson size-width relation be balanced?

Problem: Prove or disprove that for any k-CNF F on m clauses

a size s Resolution proof $\Longrightarrow$ a depth $O(m)$ and width $k + O(\sqrt{n \log s})$ proof

Win-win situation

Positive resolution: counter example to conjecture & surprising depth upper bound

Open Problems

Conjecture: There exist F on m clauses such that any (quasi)polynomial size Resolution proof requires depth $\Omega(mn^4)$

One approach…

Can the Ben-Sasson Wigderson size-width relation be balanced?

Problem: Prove or disprove that for any k-CNF F on m clauses

a size s Resolution proof $\Longrightarrow$ a depth $O(m)$ and width $k + O(\sqrt{n \log s})$ proof

Win-win situation

Positive resolution: counter example to conjecture & surprising depth upper bound

Negative resolution: (conditional) size/depth tradeoff for monotone circuits

Open Problems

Conjecture: There exist F on m clauses such that any (quasi)polynomial size Resolution proof requires depth $\Omega(mn^4)$

One approach…

Can the Ben-Sasson Wigderson size-width relation be balanced?

Problem: Prove or disprove that for any k-CNF F on m clauses

a size s Resolution proof $\Longrightarrow$ a depth $O(m)$ and width $k + O(\sqrt{n \log s})$ proof

Win-win situation

Positive resolution: counter example to conjecture & surprising depth upper bound

Negative resolution: (conditional) size/depth tradeoff for monotone circuits

Q. Supercritical size/depth tradeoffs for non-monotone circuits?