

# Extremely Deep Proofs

---

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

UCSD

Columbia University  
IAS

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**.

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**.

Observed primarily in proof complexity

- First 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**.

Observed primarily in proof complexity

- First 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**.

Observed primarily in proof complexity

- First 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

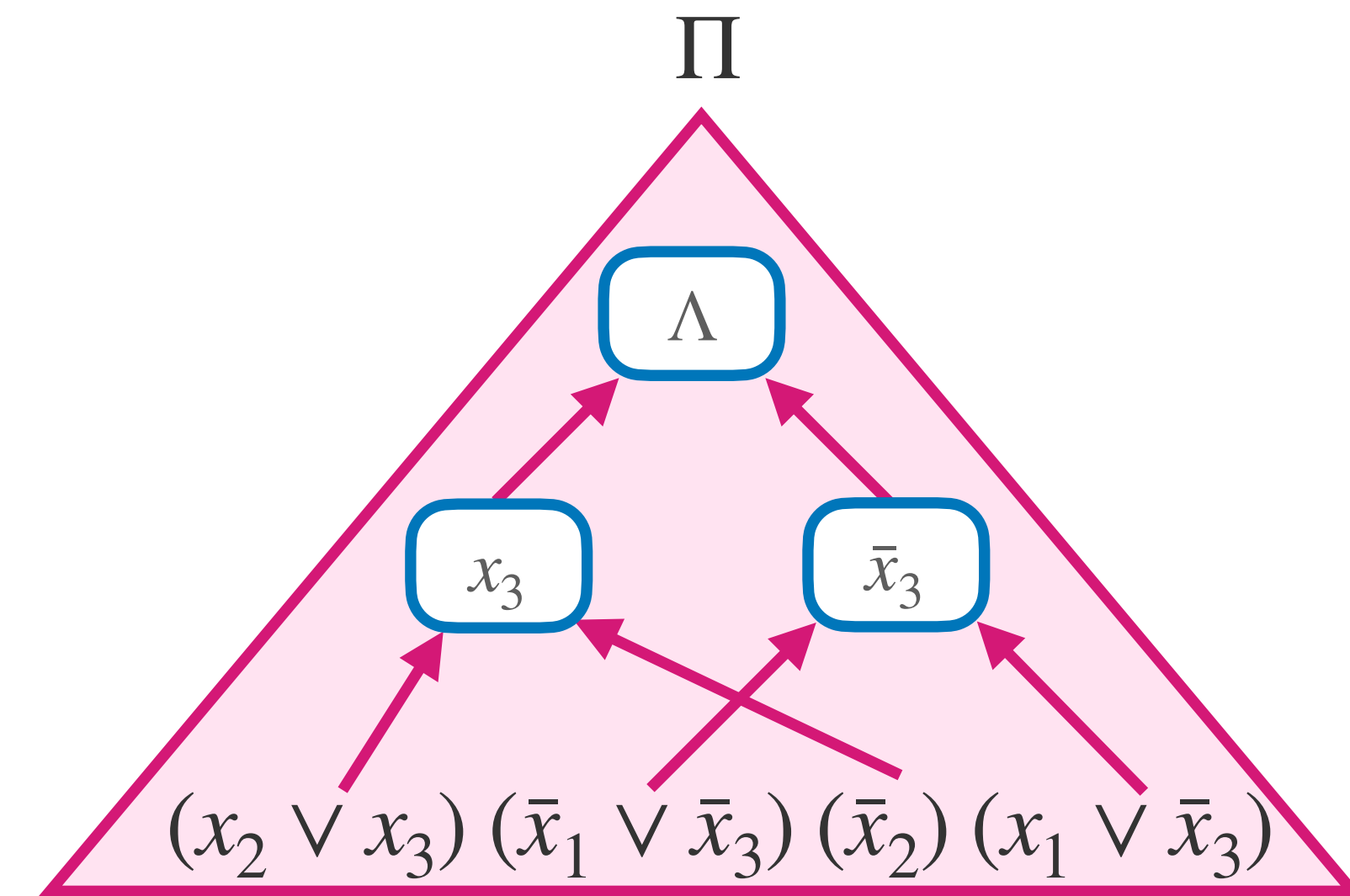
## Supercritical Tradeoff

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

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

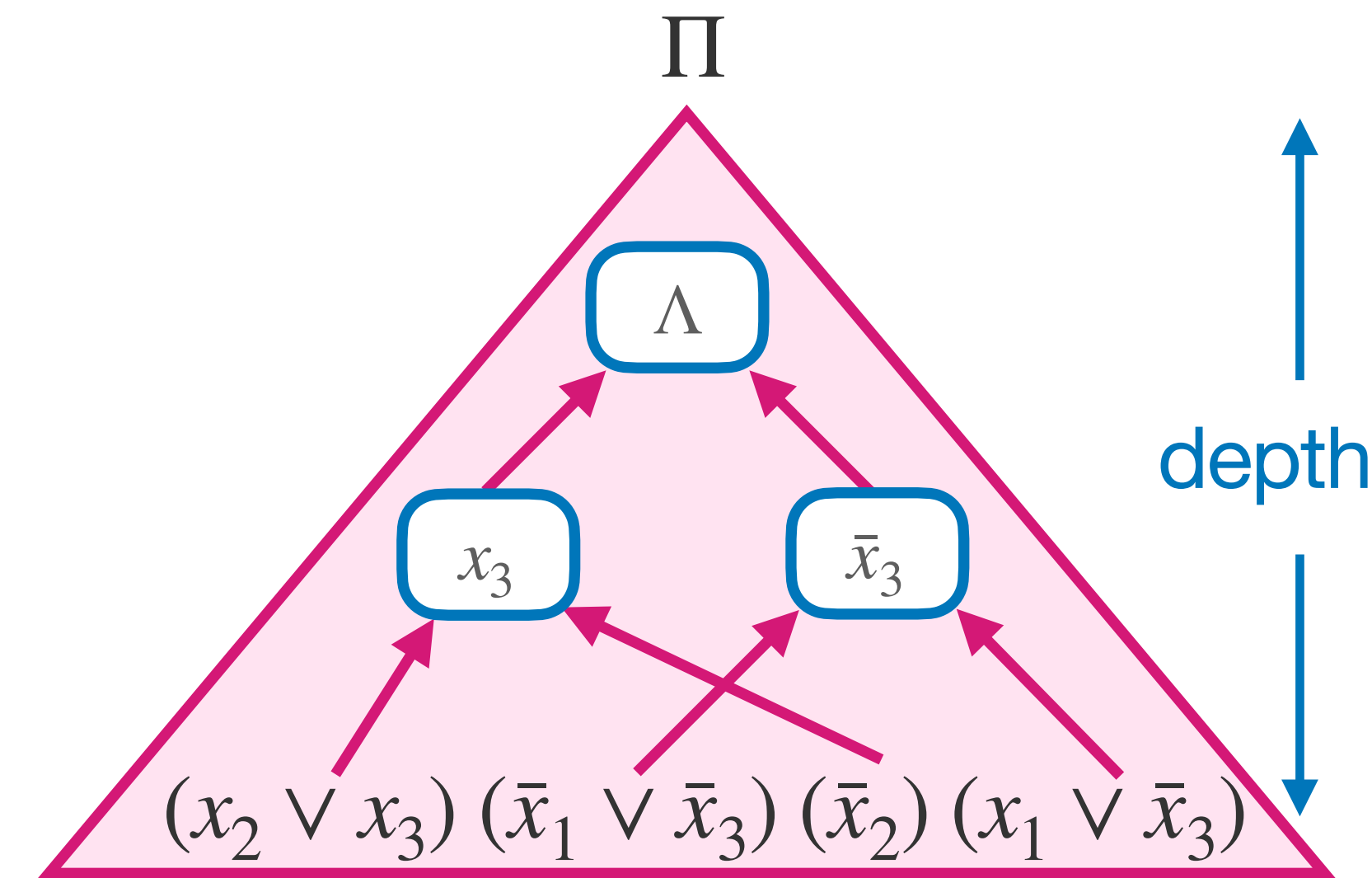
- Resolution — **Focus on for today**
- $k$ -DNF Resolution
- Cutting Planes

# Depth



# Depth

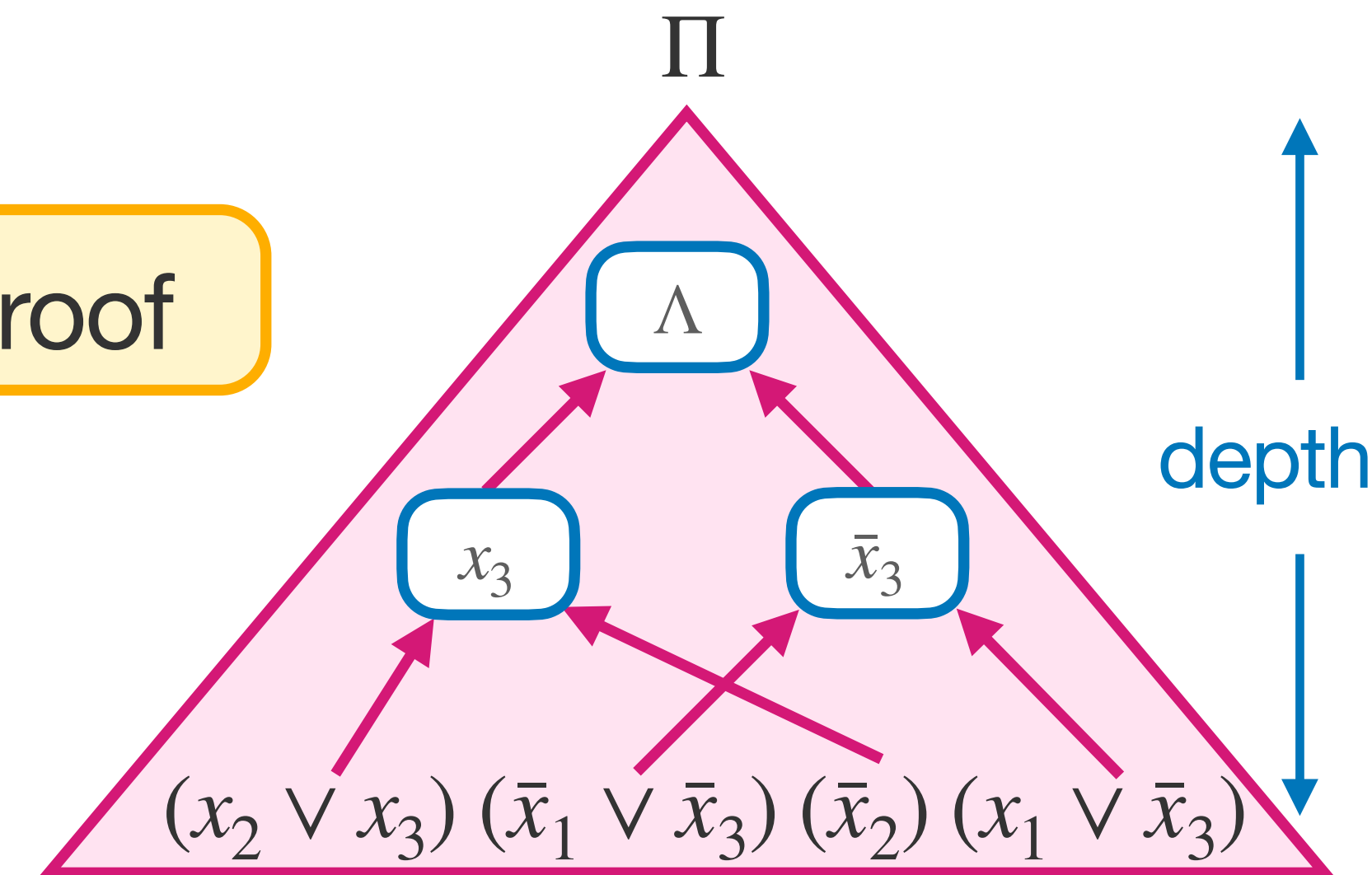
$\text{depth}(\Pi)$ : longest root-to-leaf path



# Depth

$\text{depth}(\Pi)$ : longest root-to-leaf path

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

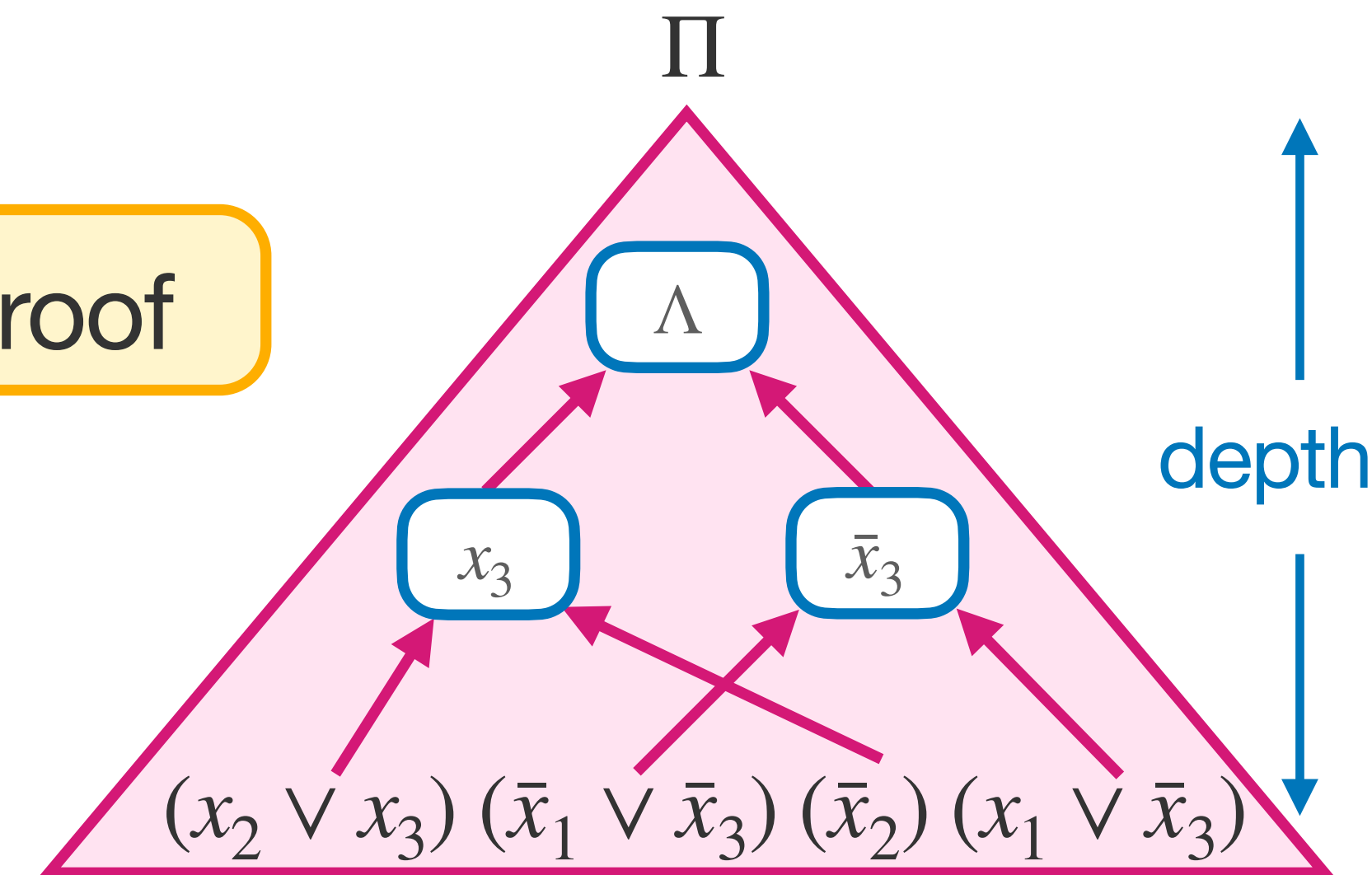


# Depth

$\text{depth}(\Pi)$ : longest root-to-leaf path

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

Resolution captures CDCL



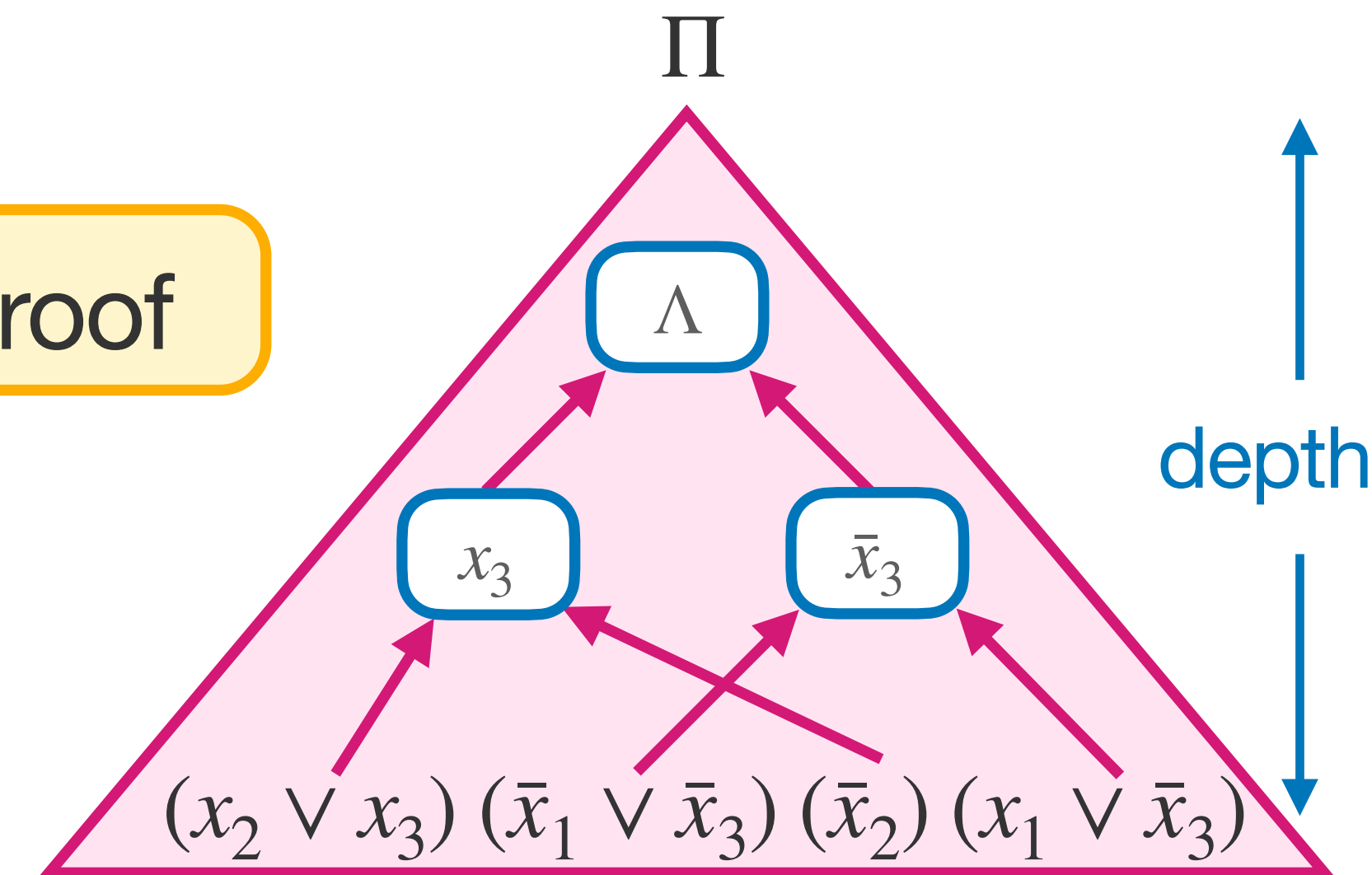
# Depth

$\text{depth}(\Pi)$ : longest root-to-leaf path

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

Resolution captures CDCL

- Size lower bounds runtime
- Depth lower bounds parallelizability



# Depth

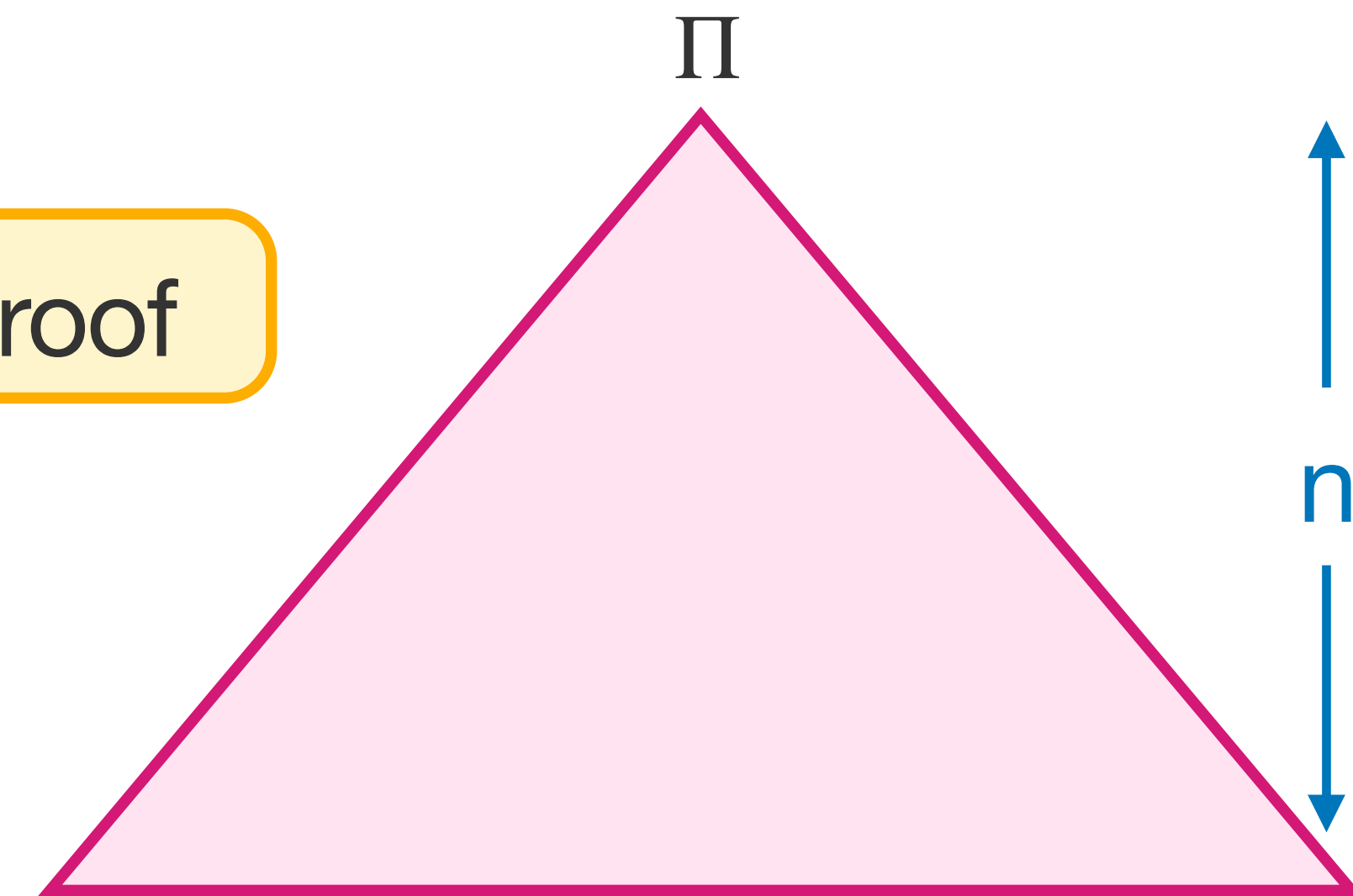
$\text{depth}(\Pi)$ : longest root-to-leaf path

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

Resolution captures CDCL

- Size lower bounds runtime
- Depth lower bounds parallelizability

Always a depth  $n$  proof





# Depth

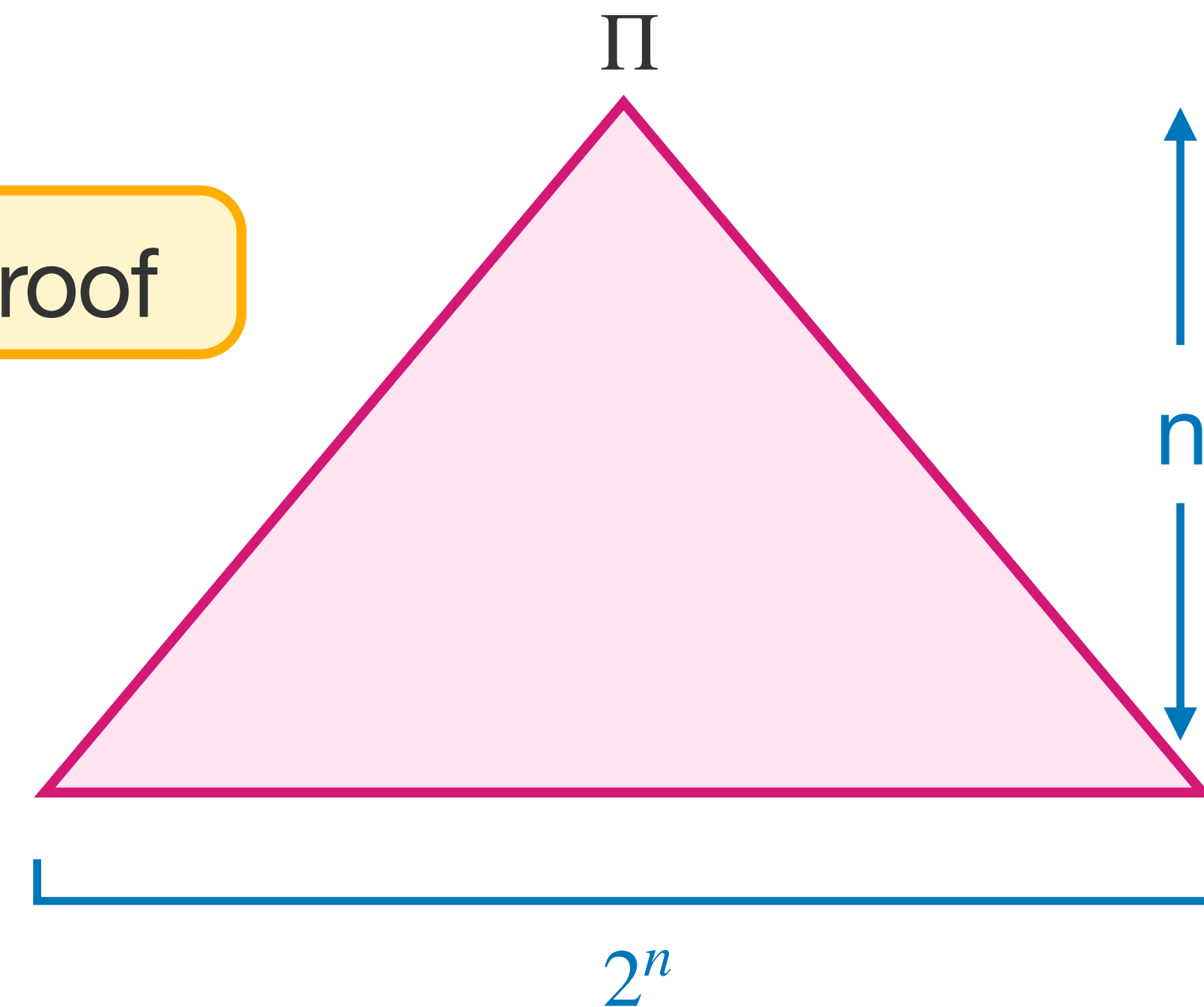
$\text{depth}(\Pi)$ : longest root-to-leaf path

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

Resolution captures CDCL

- Size lower bounds runtime
- Depth lower bounds parallelizability

Always a depth  $n$  proof — but may have size  $2^n$



# Depth

$\text{depth}(\Pi)$ : longest root-to-leaf path

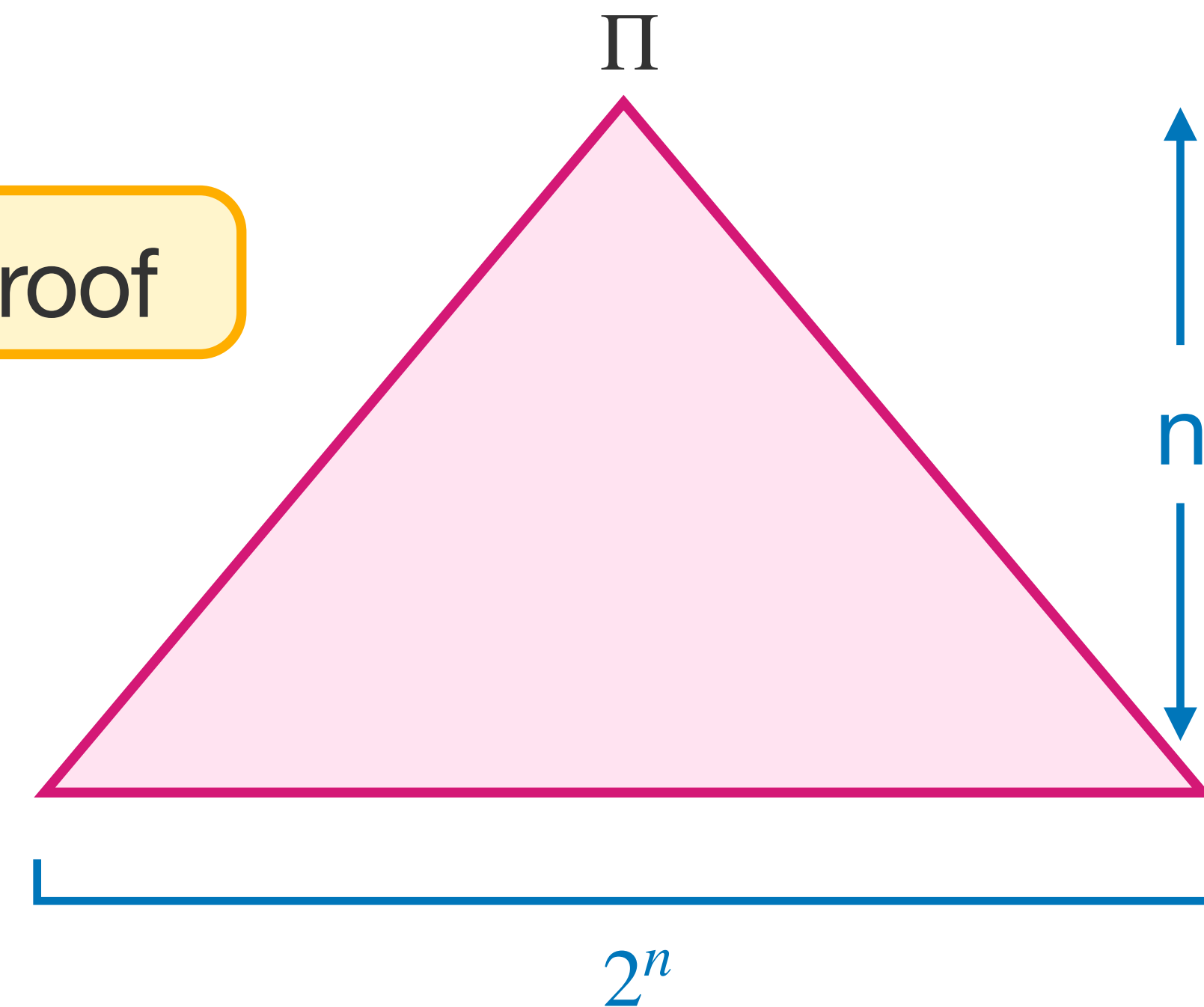
Like circuit depth, captures a notion of “parallelism” of a proof

Resolution captures CDCL

- Size lower bounds runtime
- Depth lower bounds parallelizability

Always a depth  $n$  proof — but may have size  $2^n$

Many strong proof systems can be **balanced**: depth can be assumed to be logarithmic in size.



# Depth

$\text{depth}(\Pi)$ : longest root-to-leaf path

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

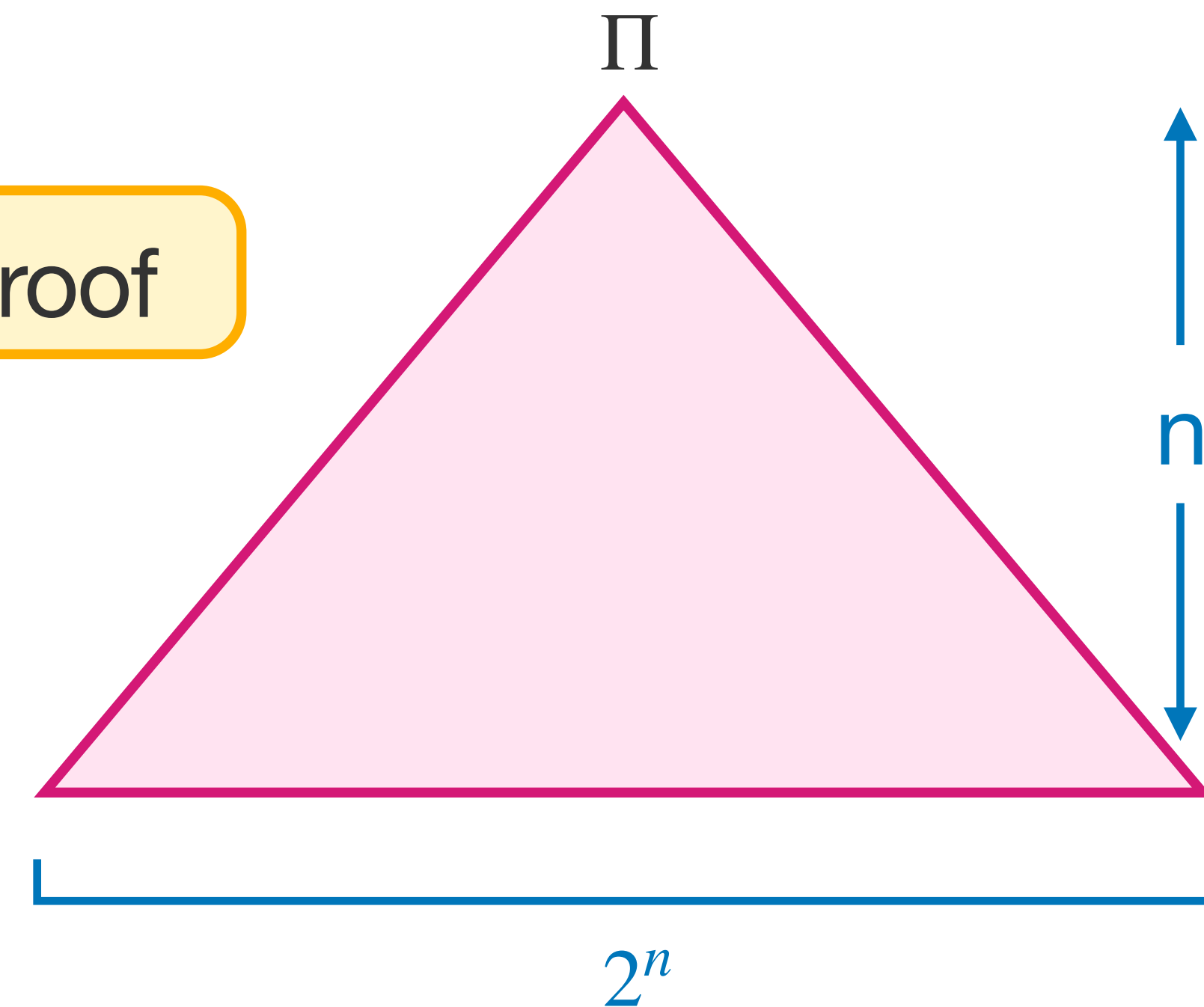
Resolution captures CDCL

- Size lower bounds runtime
- Depth lower bounds parallelizability

Always a depth  $n$  proof — but may have size  $2^n$

Many strong proof systems can be **balanced**: depth can be assumed to be logarithmic in size.

— Resolution cannot

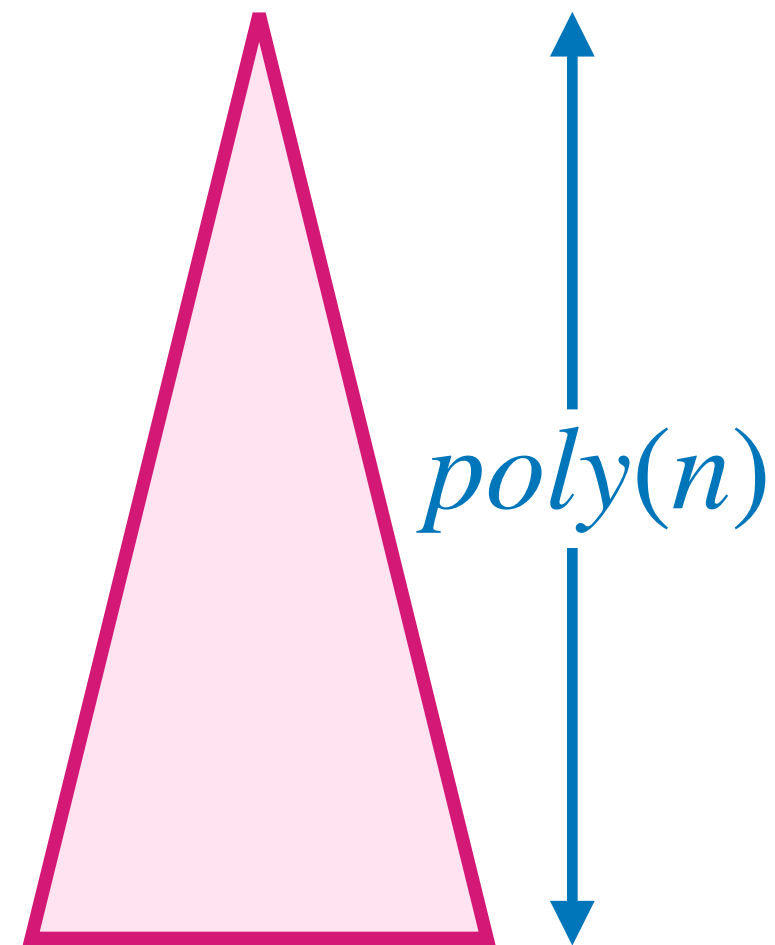


# This Work

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

There is a CNF  $F$  on  $n$  variables such that

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

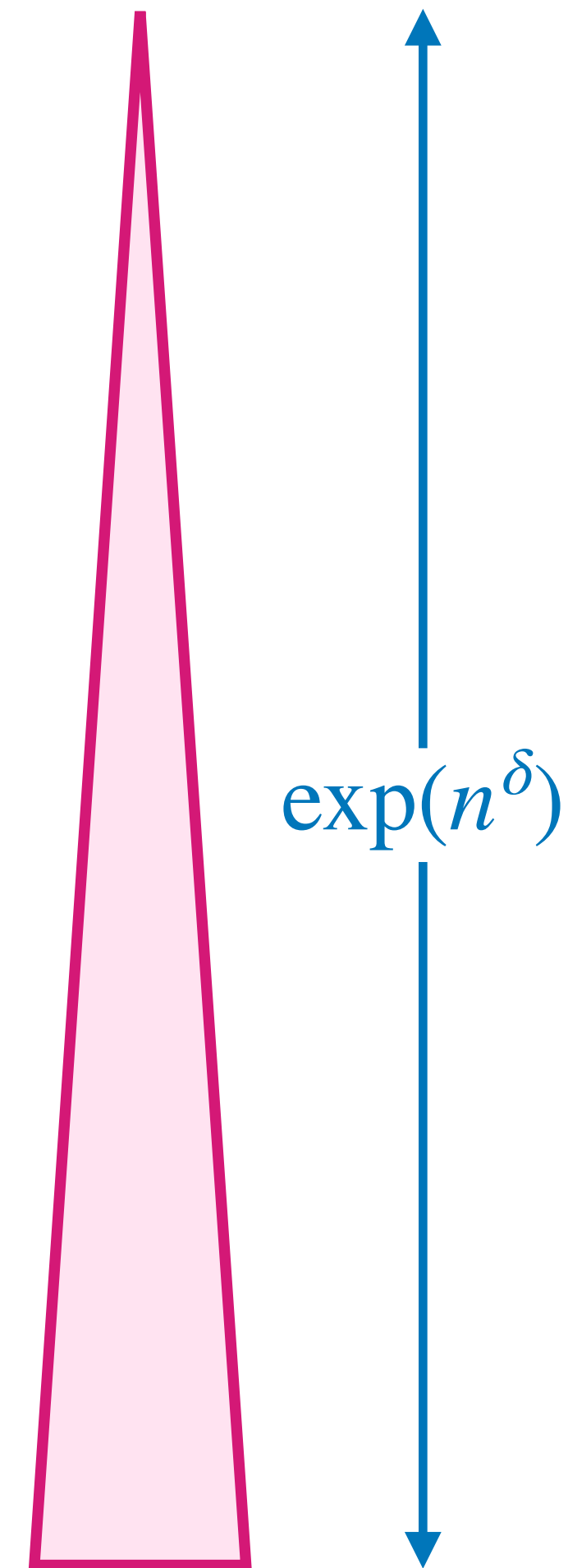


# This Work

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

There is a CNF  $F$  on  $n$  variables such that

- There is a weakly exponential size  $P$ -proof of  $F$
- Any subexponential-size  $P$ -proof of  $F$  has weakly exponential depth



# This Work

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

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

# This Work

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

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

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

# This Work

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

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

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

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



# This Work

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

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

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

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

A tradeoff between runtime and parallelizability for CDCL

# This Work

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

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

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

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

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 has only  $n$  variables!

→ 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 has only  $n$  variables!

→ 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 has only  $n$  variables!

→ 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

# 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
- Let  $g : \{0,1\}^t \rightarrow \{0,1\}$  be a function

# 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
- Let  $g : \{0,1\}^t \rightarrow \{0,1\}$  be a function

# 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
- Let  $g : \{0,1\}^t \rightarrow \{0,1\}$  be a function

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

# 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
- Let  $g : \{0,1\}^t \rightarrow \{0,1\}$  be a function

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

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

# Lifting (Composition)

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

- Let  $F(z_1, \dots, z_N) = C_1 \wedge \dots \wedge C_m$  be a CNF
- Let  $g : \{0,1\}^t \rightarrow \{0,1\}$  be a function

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

Typically  $\vec{x}_1, \dots, \vec{x}_N$  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:**  $g = \text{XOR}_2$  then  $F \circ \text{XOR}_2 := F(x_1 \oplus x'_1, \dots, x_N \oplus x'_N)$



# Lifting (Composition)

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

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

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

# Lifting (Composition)

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

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

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

A **width** lower bound on  $F$  implies a **size** lower bound on  $F \circ \text{XOR}_2$ !

# Lifting (Composition)

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

**Width-to-Size Lifting Theorem:** Let  $F$  be any unsatisfiable formula. If  $\Pi$  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:**  $g = \text{XOR}_2$  then  $F \circ \text{XOR}_2 := F(x_1 \oplus x'_1, \dots, x_N \oplus x'_N)$

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

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

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

- $P = \text{Resolution (width)}$ ,  $Q = \text{Resolution (size)}$

# Lifting (Composition)

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

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

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

- $P = \text{Resolution (width)}$ ,  $Q = \text{Resolution (size)}$

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

# Lifting (Composition)

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

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

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

- $P = \text{Resolution (width)}$ ,  $Q = \text{Resolution (size)}$

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  $F$

# Lifting (Composition)

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

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

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

- $P = \text{Resolution (width)}$ ,  $Q = \text{Resolution (size)}$

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  $F$

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

# Lifting (Composition)

## Typically in a Lifting Theorem...

→  $P$  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 extra overhead to handle  $g$ )



# Lifting (Composition)

## Typically in a Lifting Theorem...

→  $P$  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 extra overhead to handle  $g$ )

i.e., it “lifts” lower bounds on **weak** proof systems to **strong** proof systems

# 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

$$F(\text{XOR}(\vec{x}_1), \dots, \text{XOR}(\vec{x}_N))$$



# The Gadget

Our gadget will be the XOR function

$F(\text{XOR}(\vec{x}_1), \dots, \text{XOR}(\vec{x}_N))$  ... With a **twist!**

The variable sets  $\vec{x}_1, \dots, \vec{x}_N$  will no longer be **disjoint!**

# The Gadget

Our gadget will be the XOR function

$F(\text{XOR}(\vec{x}_1), \dots, \text{XOR}(\vec{x}_N))$  ... With a **twist!**

The variable sets  $\vec{x}_1, \dots, \vec{x}_N$  will no longer be **disjoint!**

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

# The Gadget

Our gadget will be the XOR function

$F(\text{XOR}(\vec{x}_1), \dots, \text{XOR}(\vec{x}_N))$  ... With a **twist!**

The variable sets  $\vec{x}_1, \dots, \vec{x}_N$  will no longer be **disjoint!**

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

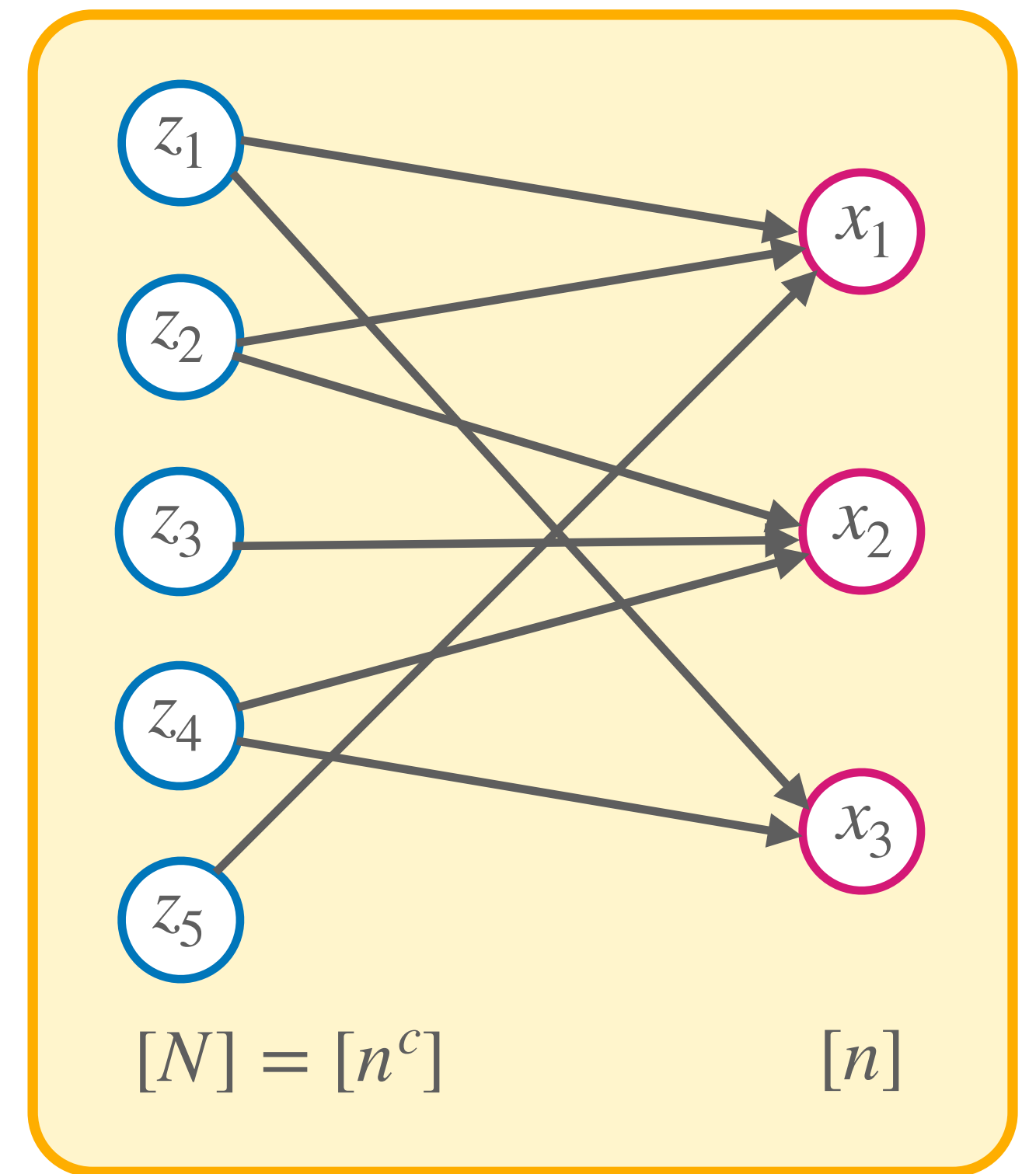
... In fact, we will compose with the **Nisan-Wigderson generator!**

# The Gadget

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

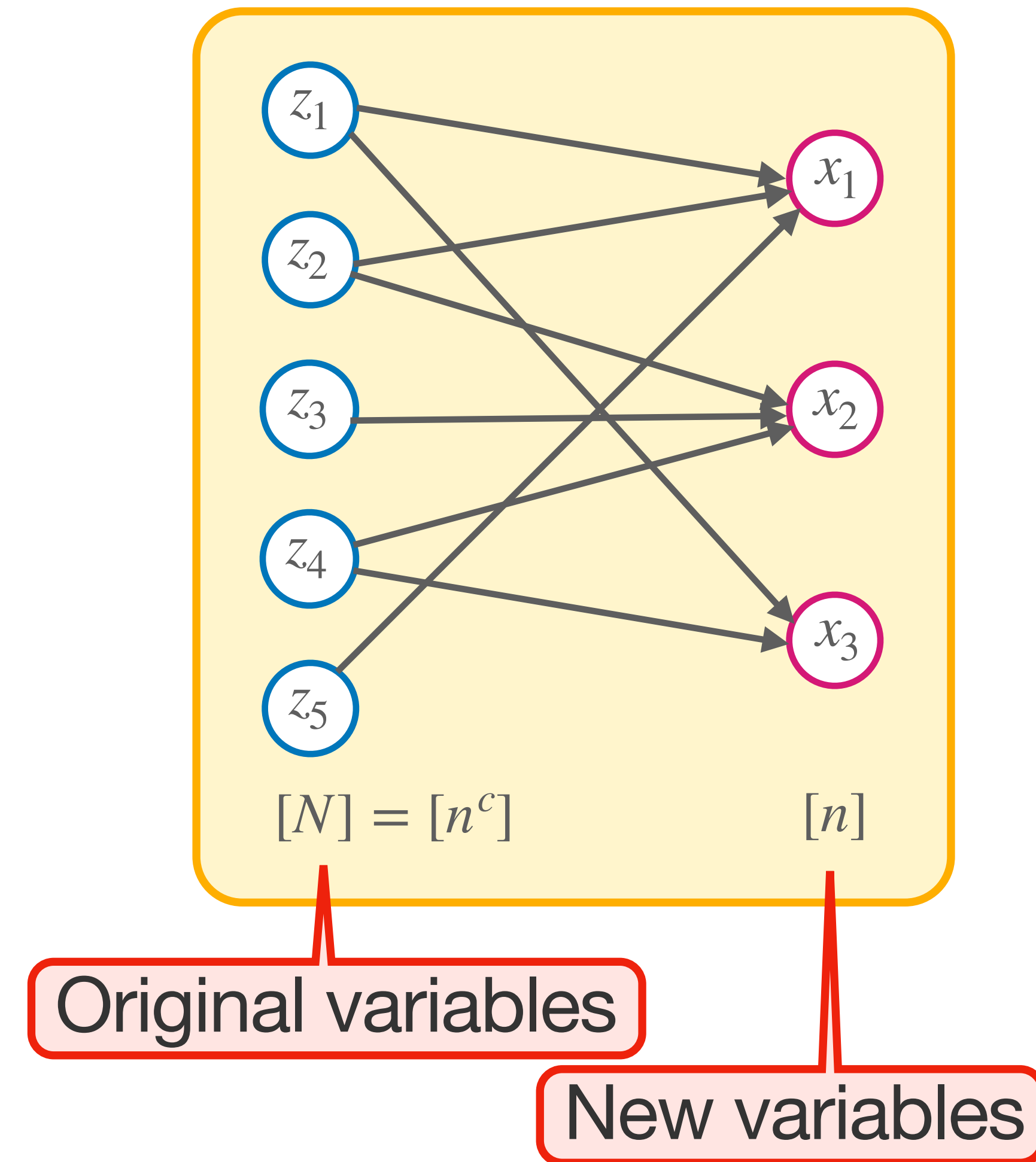
# The Gadget

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



# The Gadget

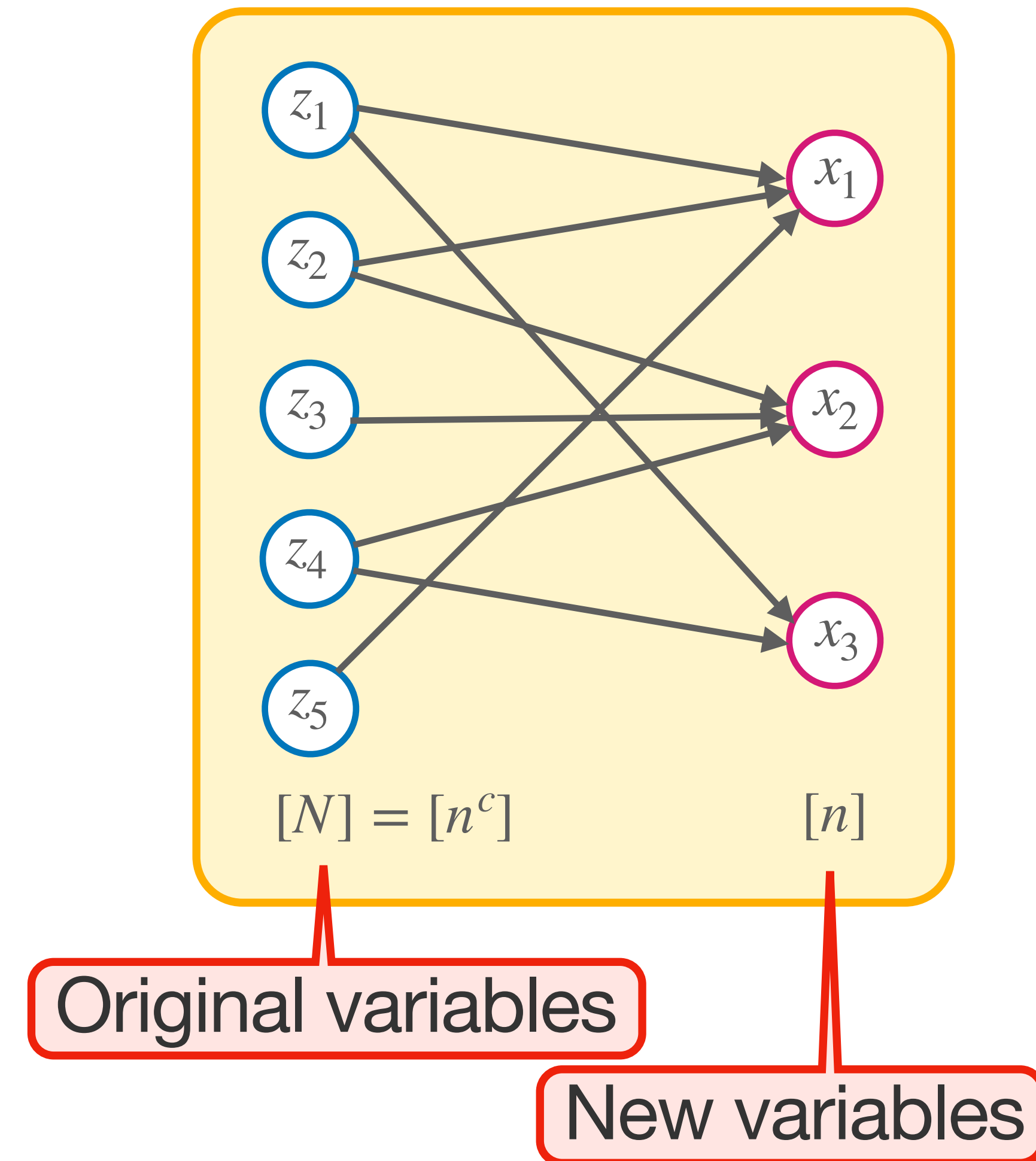
Let  $G$  be an  $N \times n$  bipartite graph



# The Gadget

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

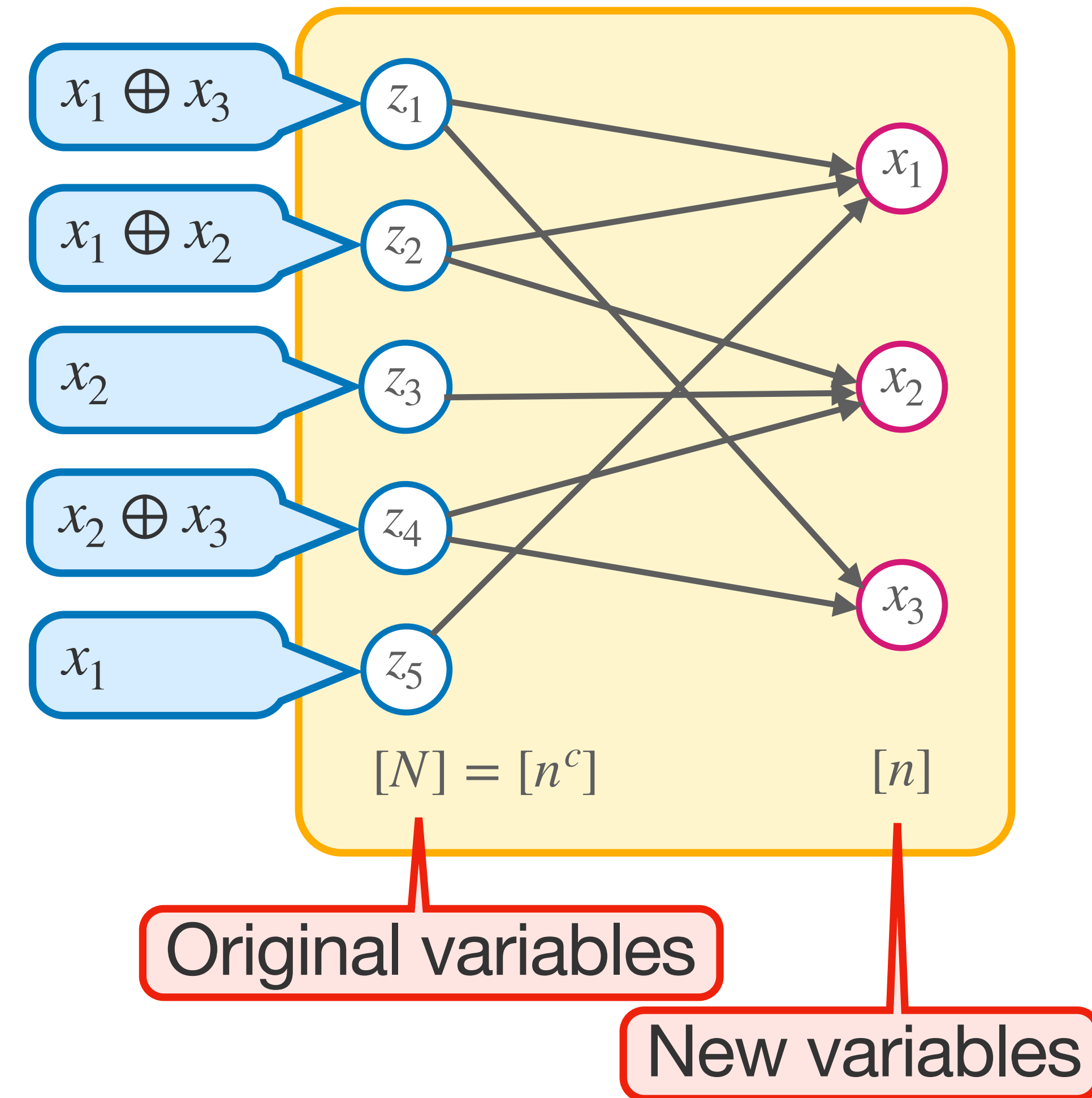
$F \circ \text{XOR}_G$  replaces  $z_i \mapsto \bigoplus_{x_j \in N(z_i)} x_j$



# The Gadget

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

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



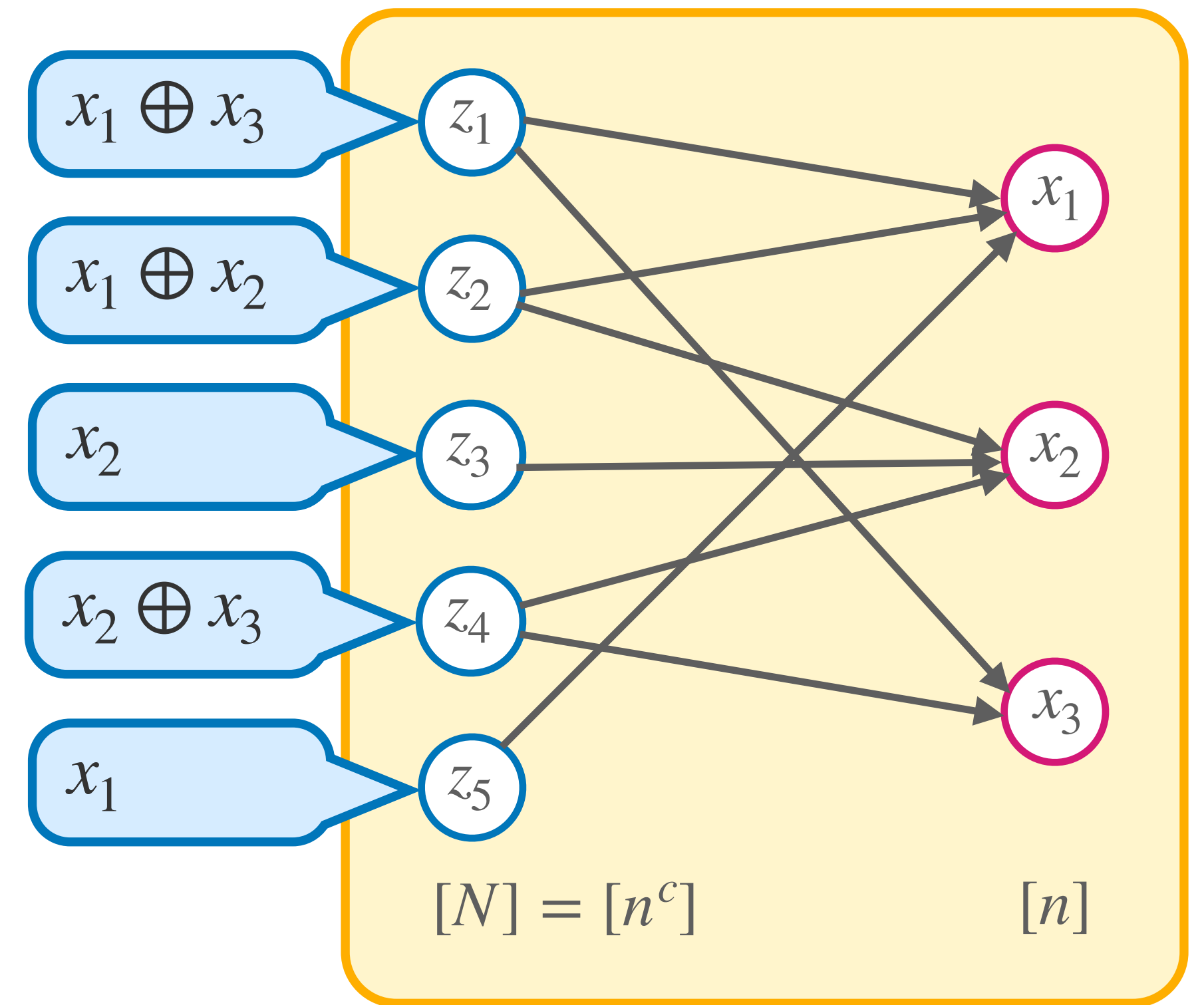


# The Gadget

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

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

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



# The Gadget

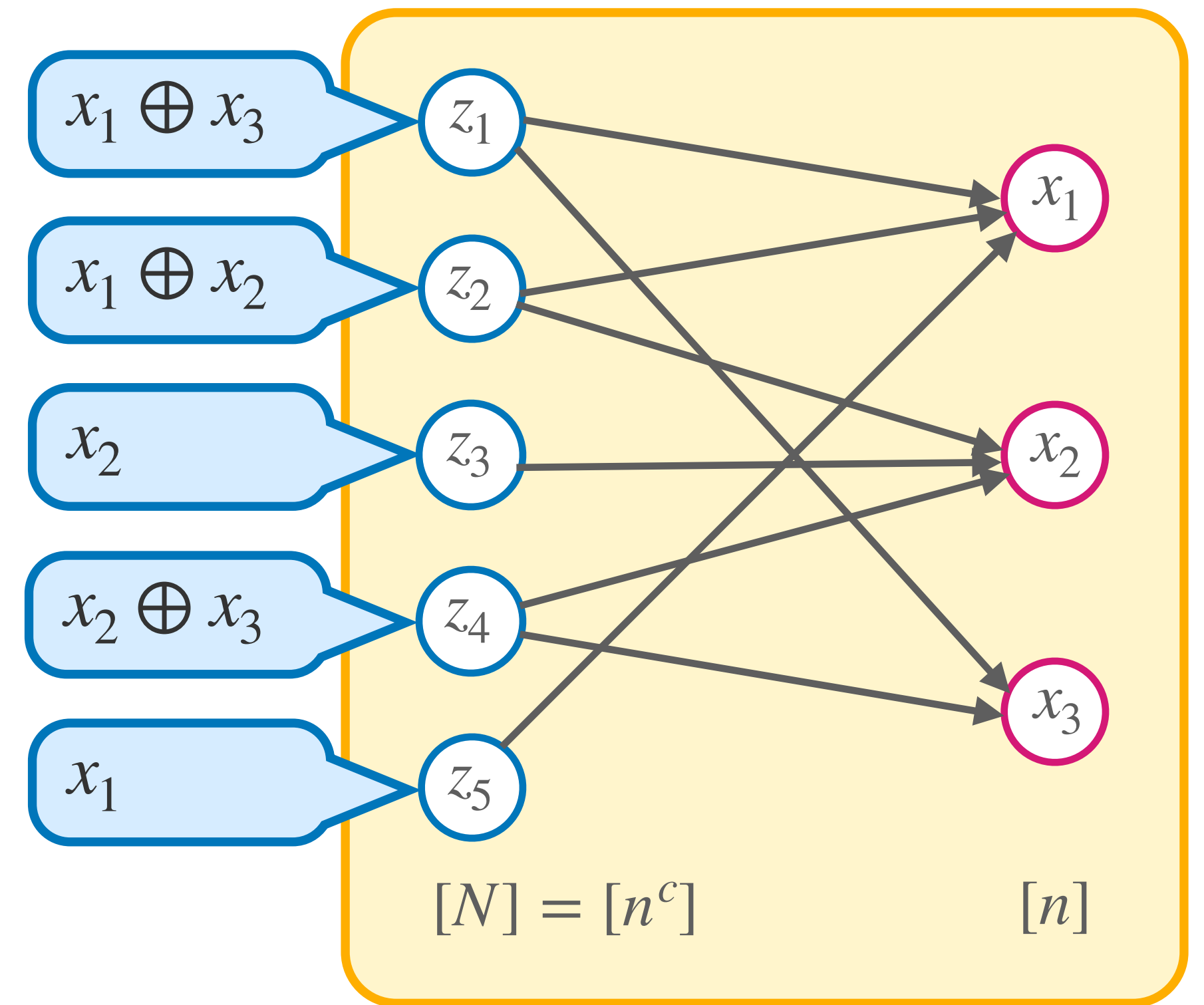
Let  $G$  be an  $N \times n$  bipartite graph

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

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

↓

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

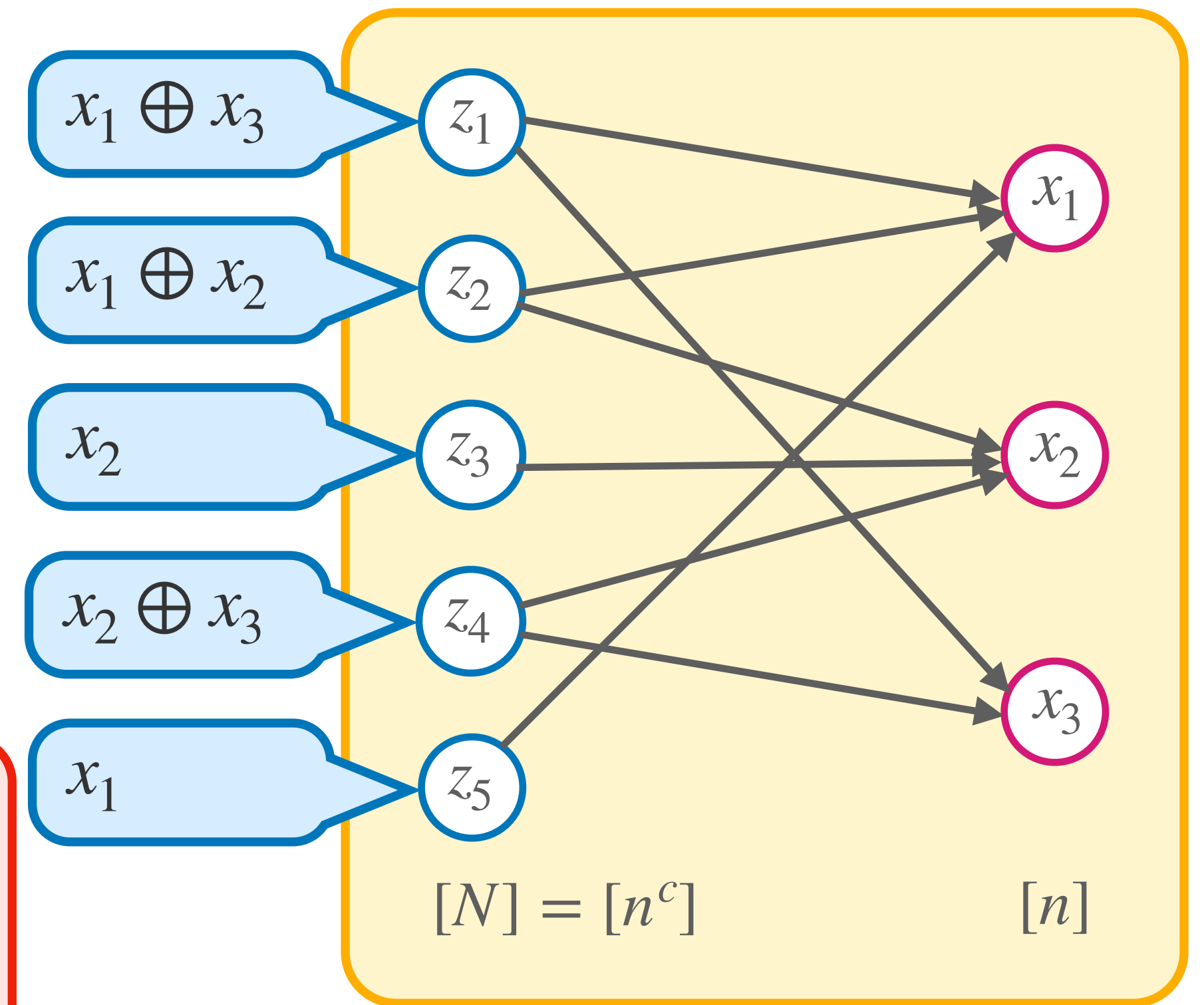


# The Gadget

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

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

**Idea:** If the edges of  $G$  are sufficiently “spread out”

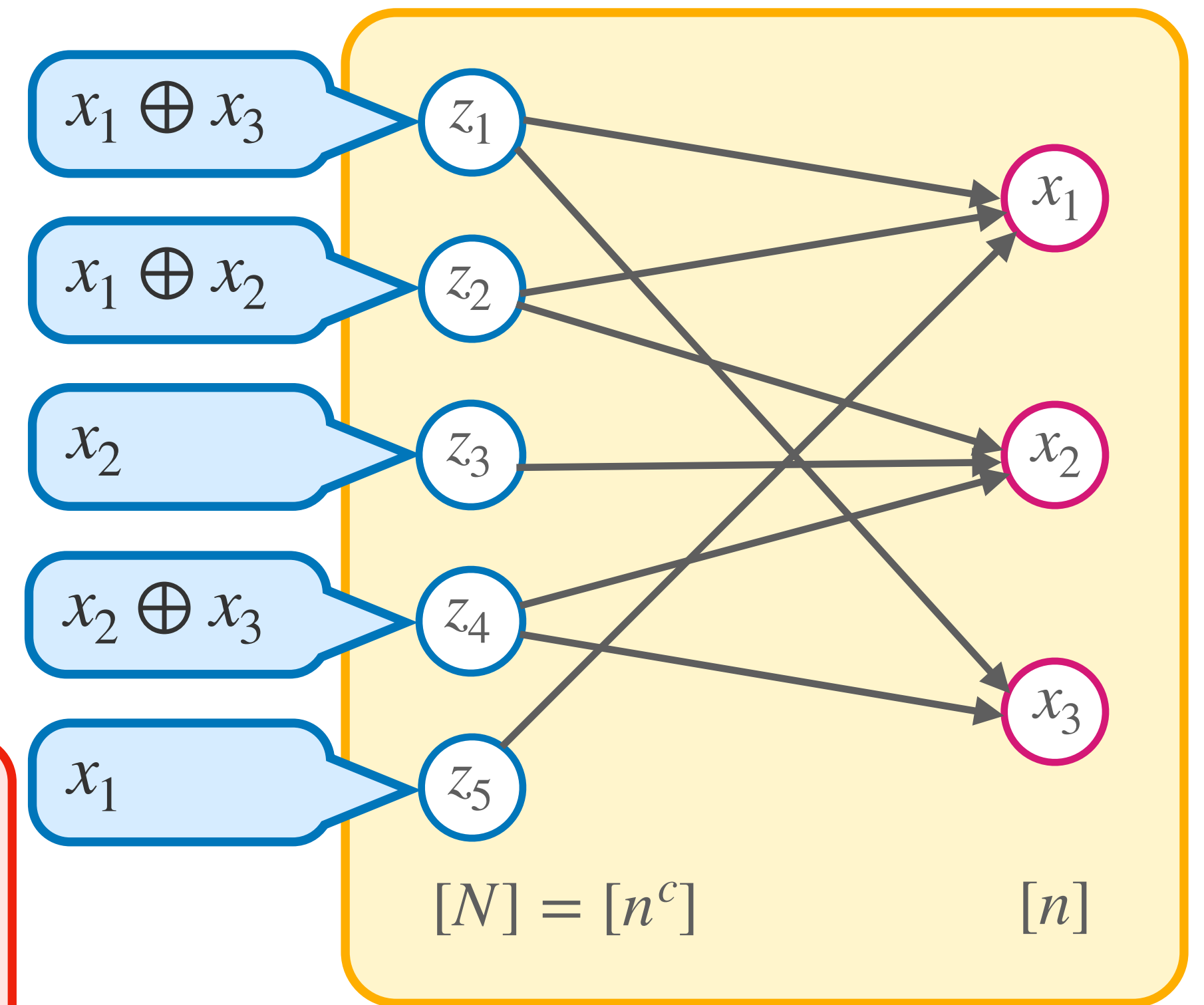


# The Gadget

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

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

**Idea:** If the edges of  $G$  are sufficiently “spread out”  
→ learning the value of one XOR won’t reveal much information about any other XOR

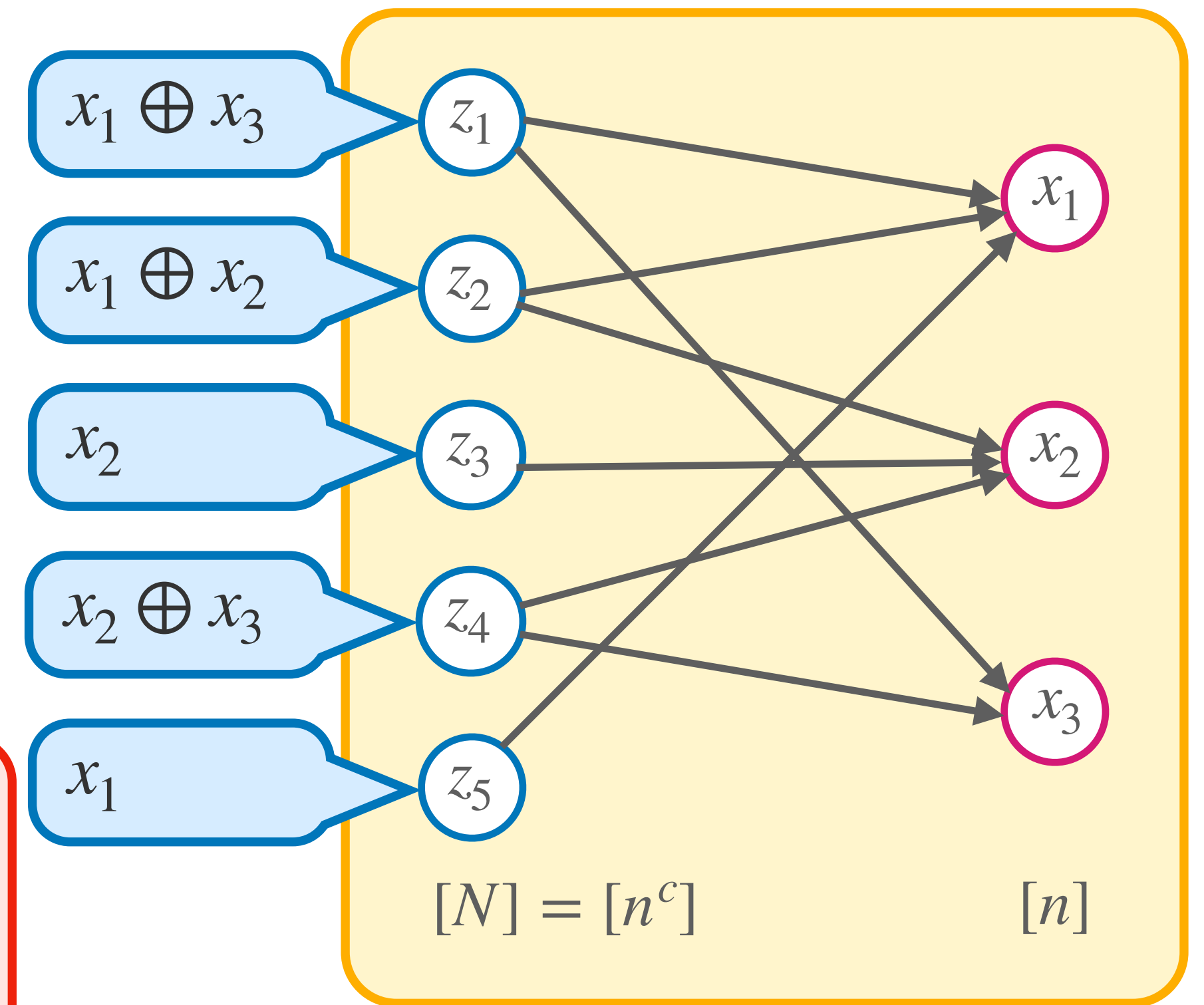


# The Gadget

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

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

**Idea:** If the edges of  $G$  are sufficiently “spread out”  
→ learning the value of one XOR won’t reveal much information about any other XOR  
→ The best Resolution proof of  $F \circ \text{XOR}_G$  should essentially be to simulate the best proof of  $F$



# The Gadget

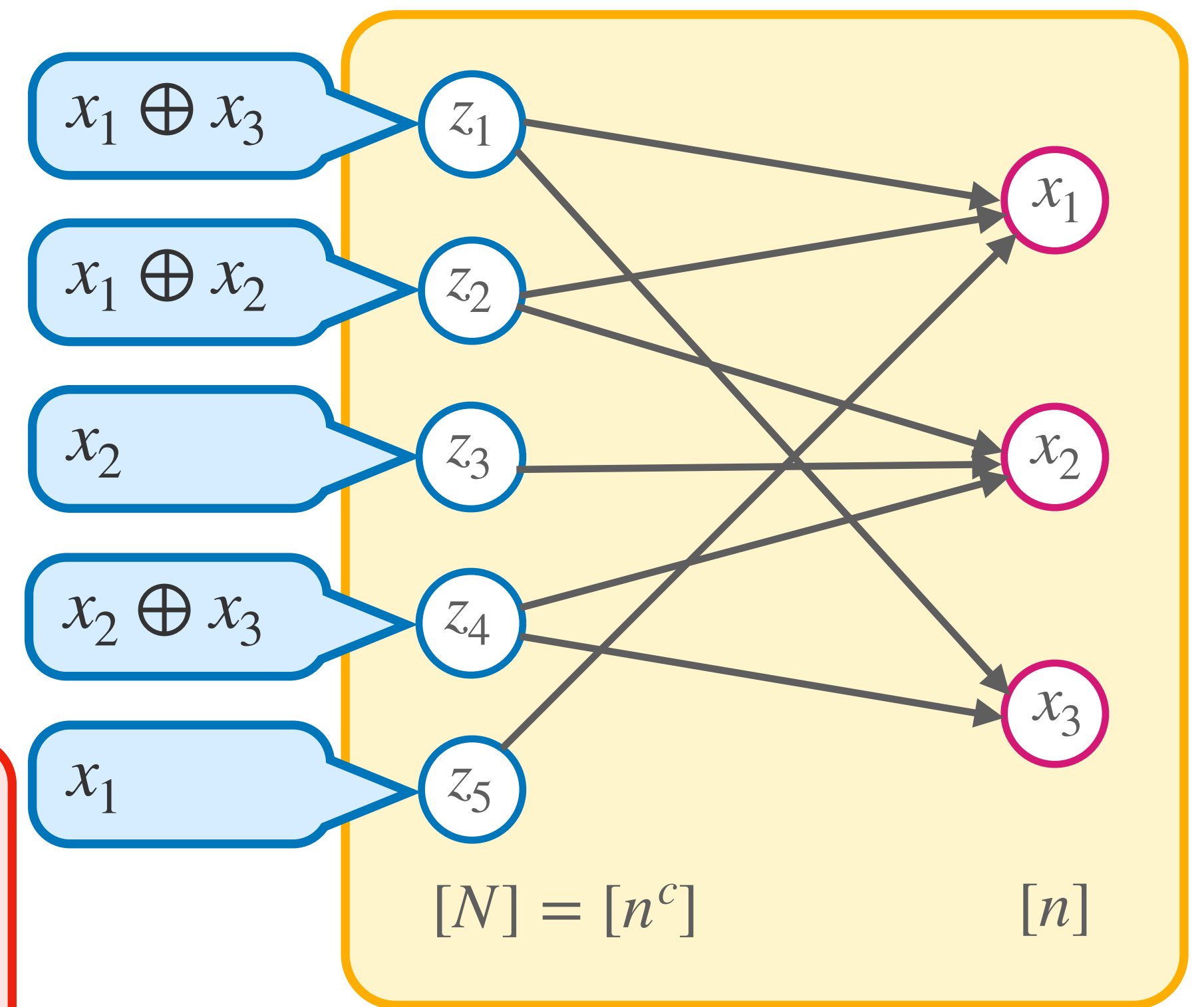
Let  $G$  be an  $N \times n$  bipartite graph

$F \circ \text{XOR}_G$  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  $F \circ \text{XOR}_G$  should essentially be to simulate the best proof of  $F$





# The Gadget

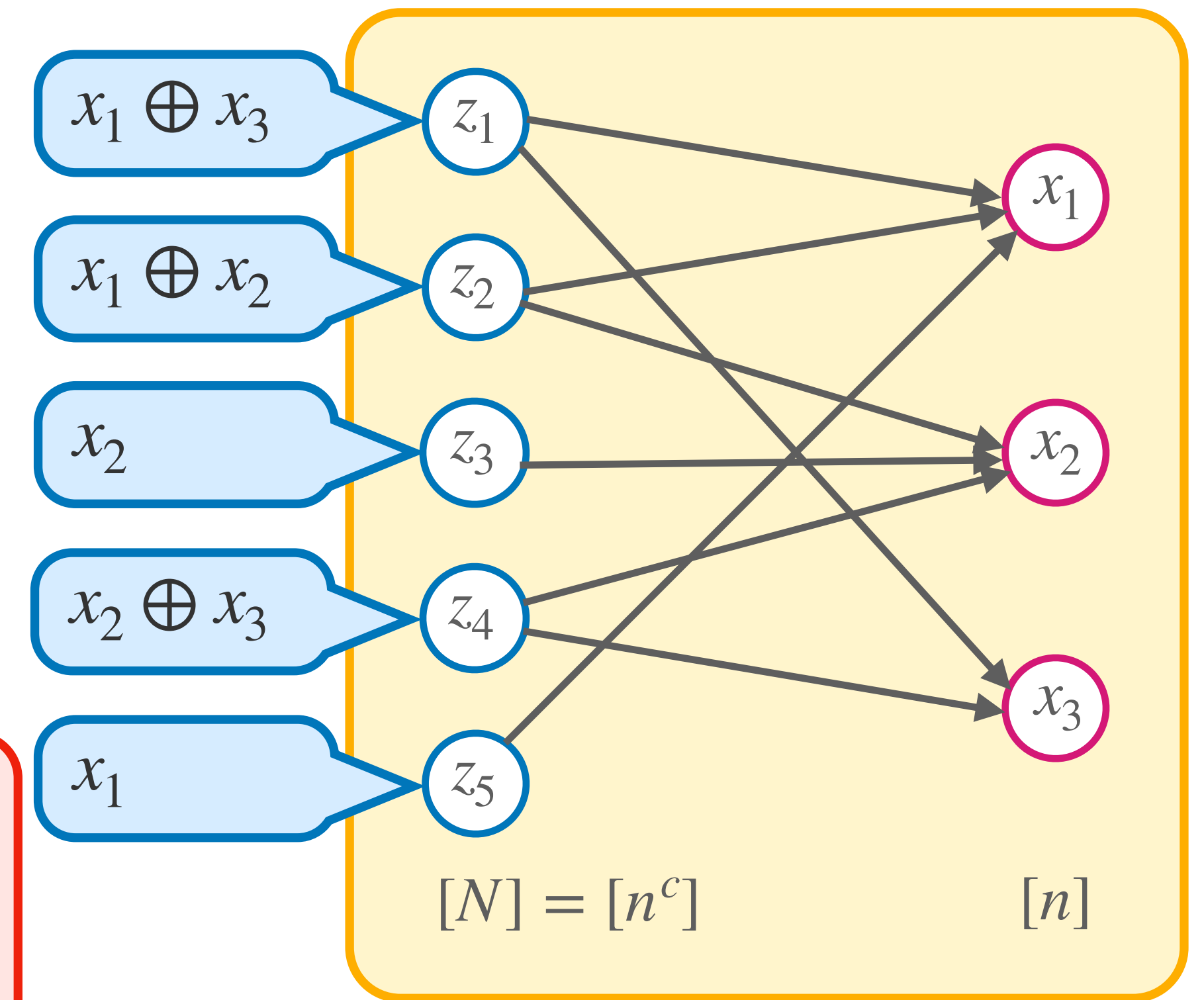
Let  $G$  be an  $N \times n$  bipartite graph

$F \circ \text{XOR}_G$  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  $F \circ \text{XOR}_G$  should essentially be to simulate the best proof of  $F$

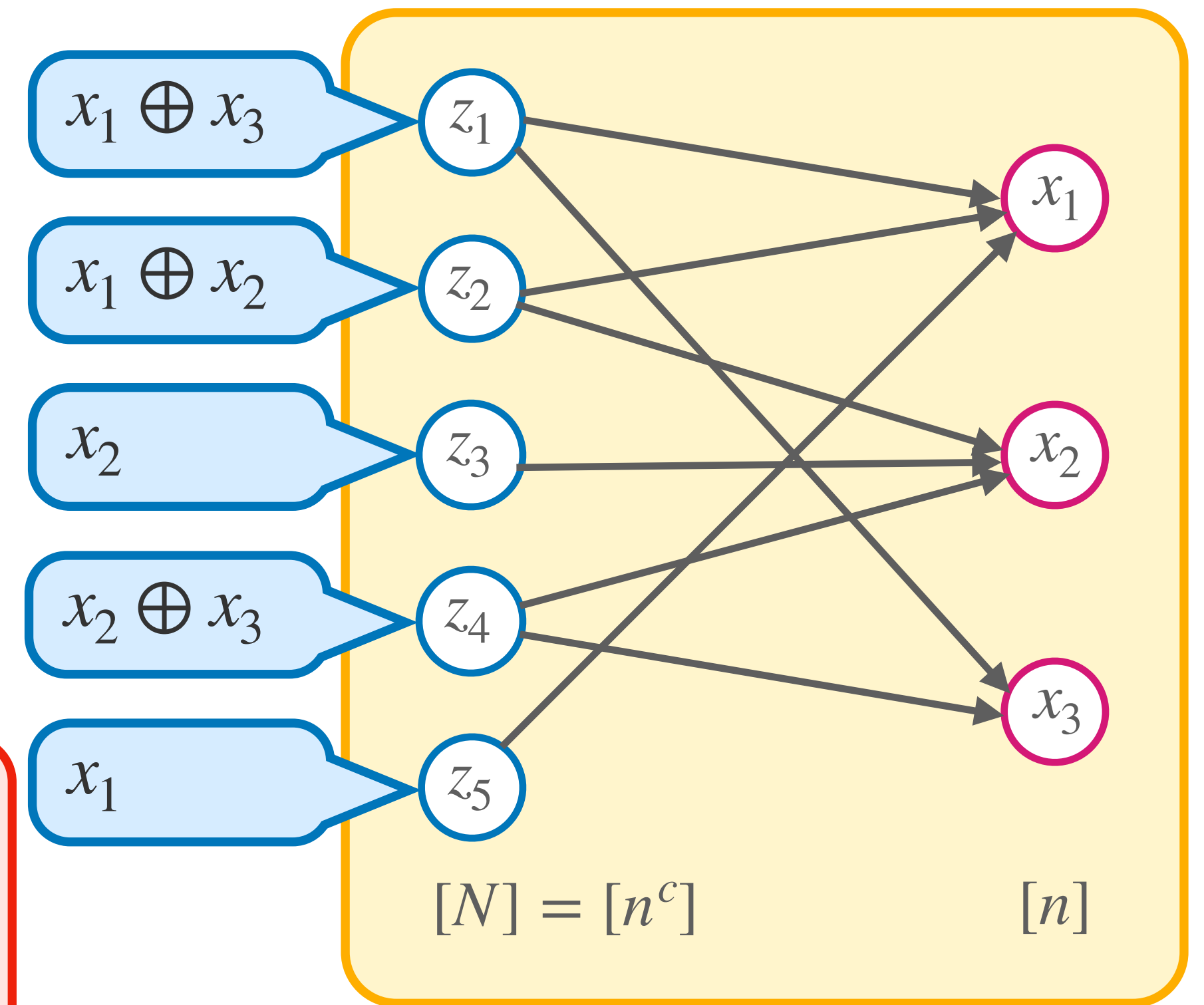


**$r$ -Expanding:** For any set  $U \subseteq [N]$  with  $|U| \leq r$  the number of **unique neighbours** is at least  $2|U|$

# The Gadget

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

$F \circ \text{XOR}_G$  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  $F \circ \text{XOR}_G$  should essentially be to simulate the best proof of  $F$

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

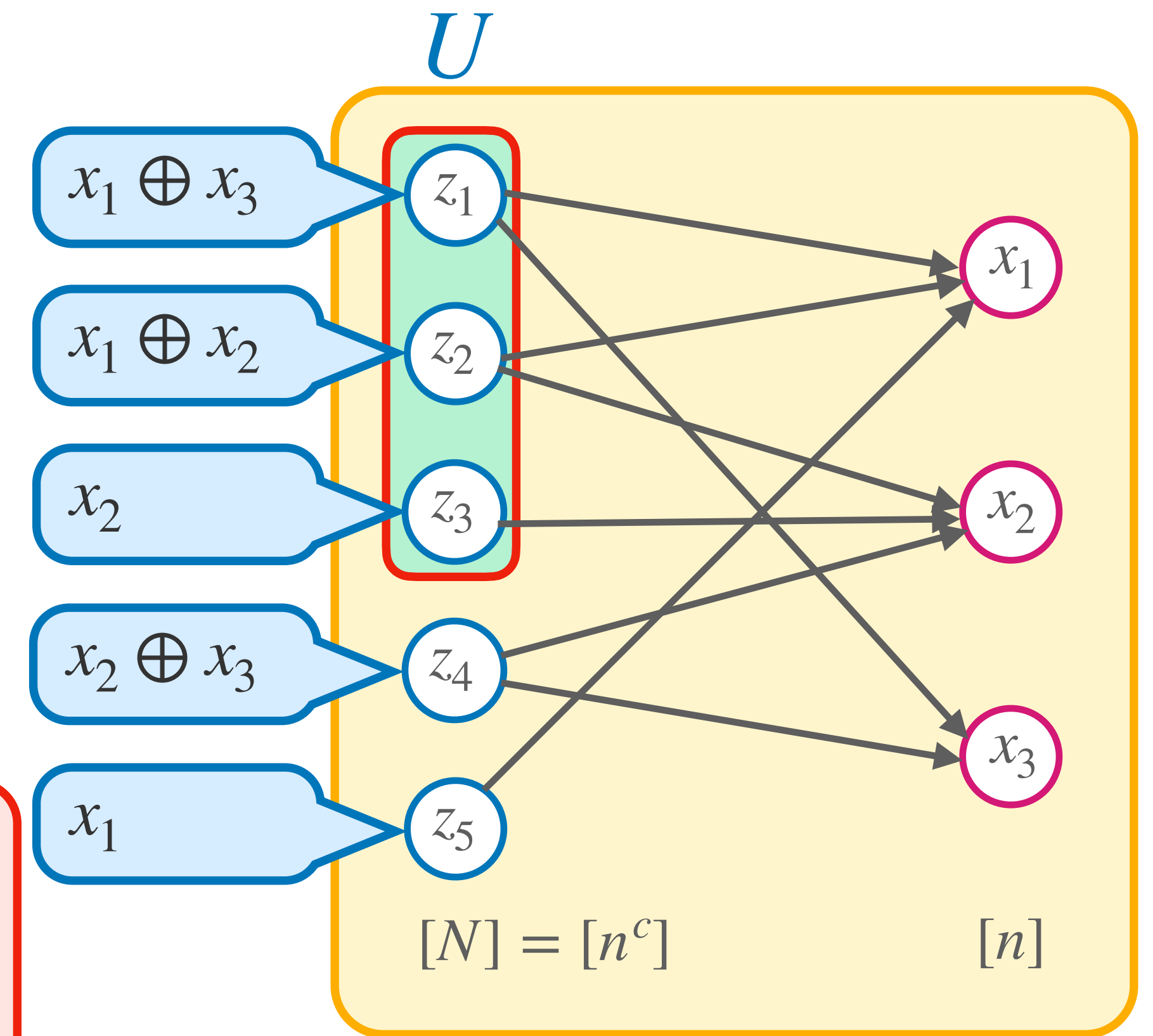
**$r$ -Expanding:** For any set  $U \subseteq [N]$  with  $|U| \leq r$  the number of **unique neighbours** is at least  $2|U|$



# The Gadget

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

$F \circ \text{XOR}_G$  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  $F \circ \text{XOR}_G$  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

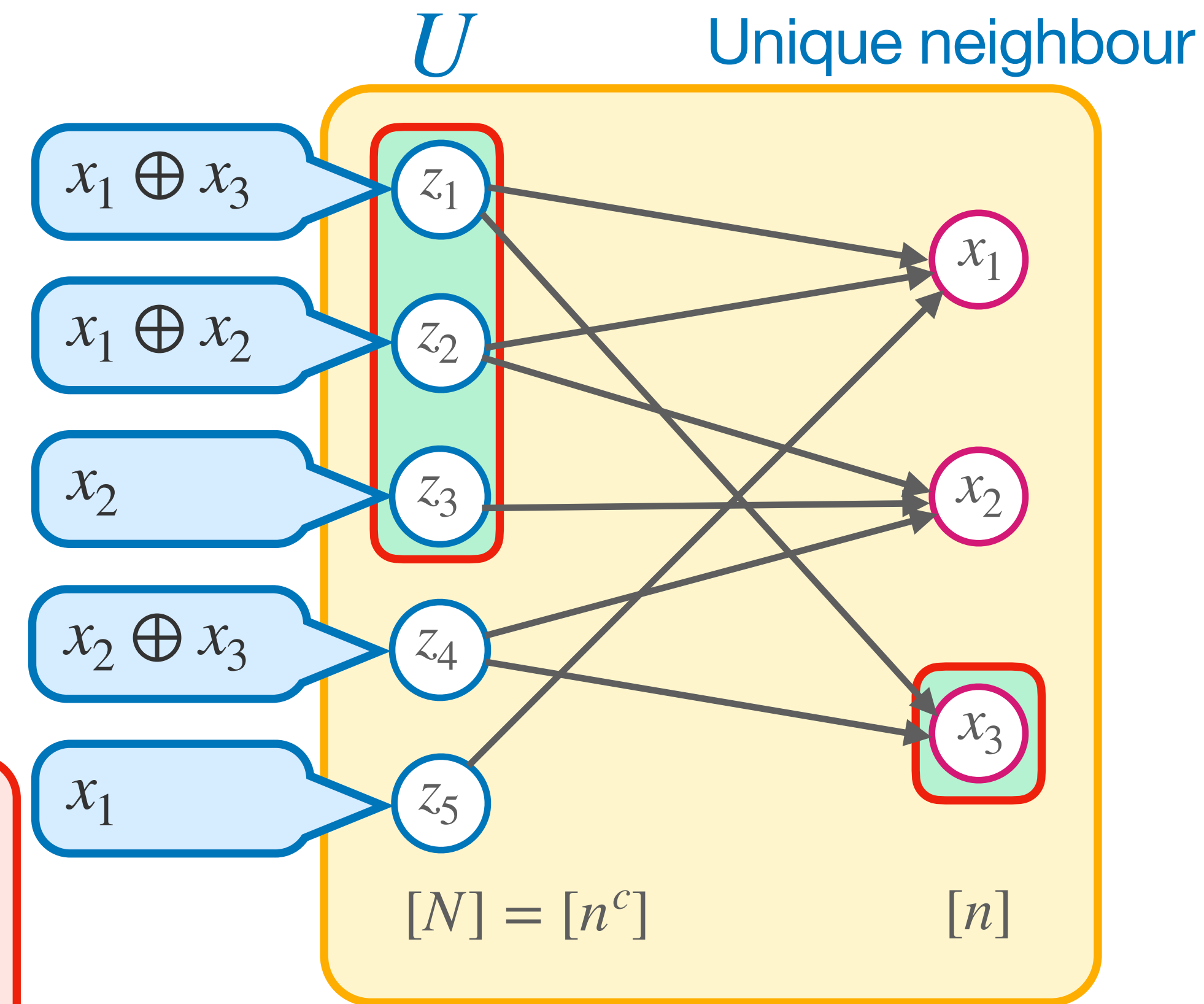
Let  $G$  be an  $N \times n$  bipartite graph

$F \circ \text{XOR}_G$  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  $F \circ \text{XOR}_G$  should essentially be to simulate the best proof of  $F$



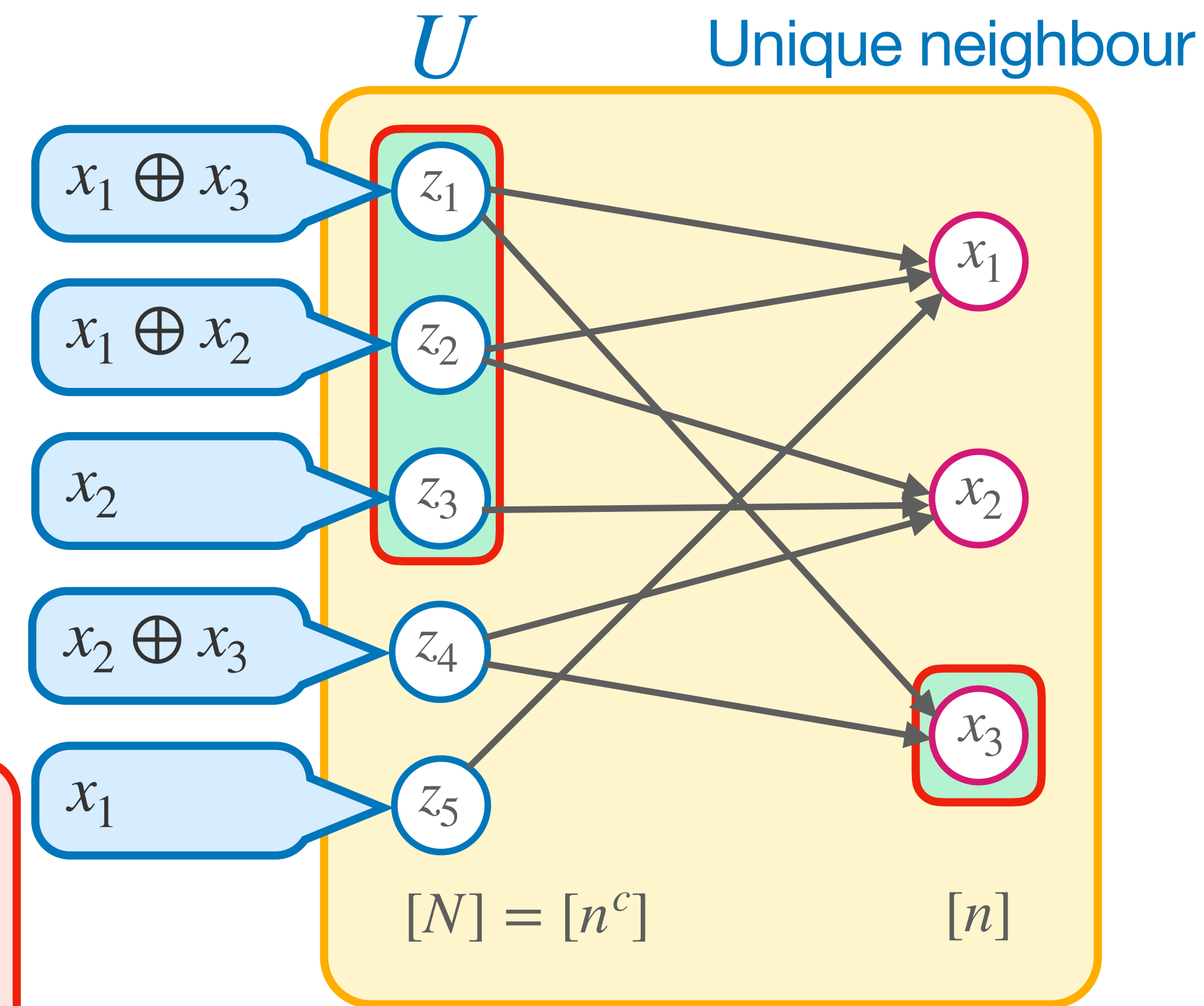
Number of  $x$ -variables that occur in **exactly one** XOR in  $U$

**$r$ -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

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

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

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

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

→ 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.

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

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

→ 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.

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

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

→ 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.

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

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

→ 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.

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

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

→ We give a **simple** proof

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

**Width-to-Size Lifting Theorem:** If  $\Pi$  is a resolution proof of  $F \circ XOR_2$  then

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

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

# Depth Condensation

Main workhorse behind our tradeoff:

## Main Theorem (Res):

Let  $G$  be  $r$ -expanding.

If  $\Pi$  is a Resolution proof of  $Peb \circ XOR_G \circ XOR_2$  with  $\log \text{size}(\Pi) \leq r/4$  then

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

→ We give a **simple** proof

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

**Width-to-Size Lifting Theorem:** If  $\Pi$  is a resolution proof of  $F \circ XOR_2$  then

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

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

# 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  $\Pi$  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!

# 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  $\Pi$  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  $\Pi$  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  $g = \text{XOR}_2$ , 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  $\Pi$  is a resolution proof of  $F \circ \text{XOR}_G$  with  $\text{width}(\Pi) \leq r/4$  then

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

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

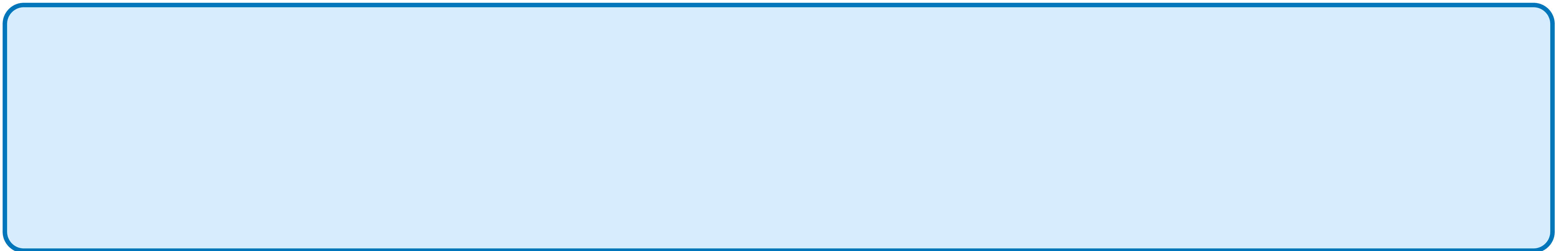
# Prover Adversary Games

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

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

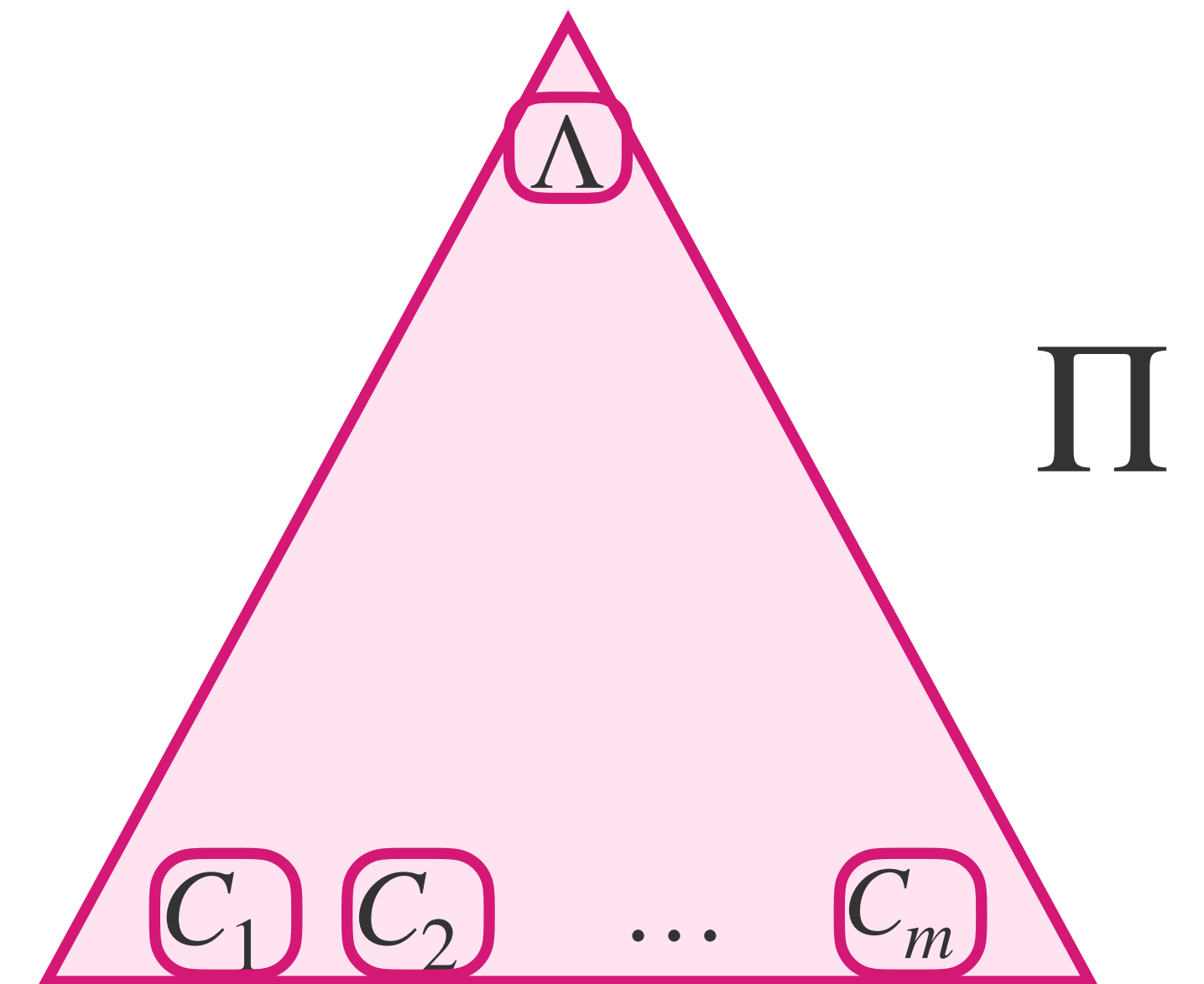
# Prover Adversary Games

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

# Prover Adversary Games

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

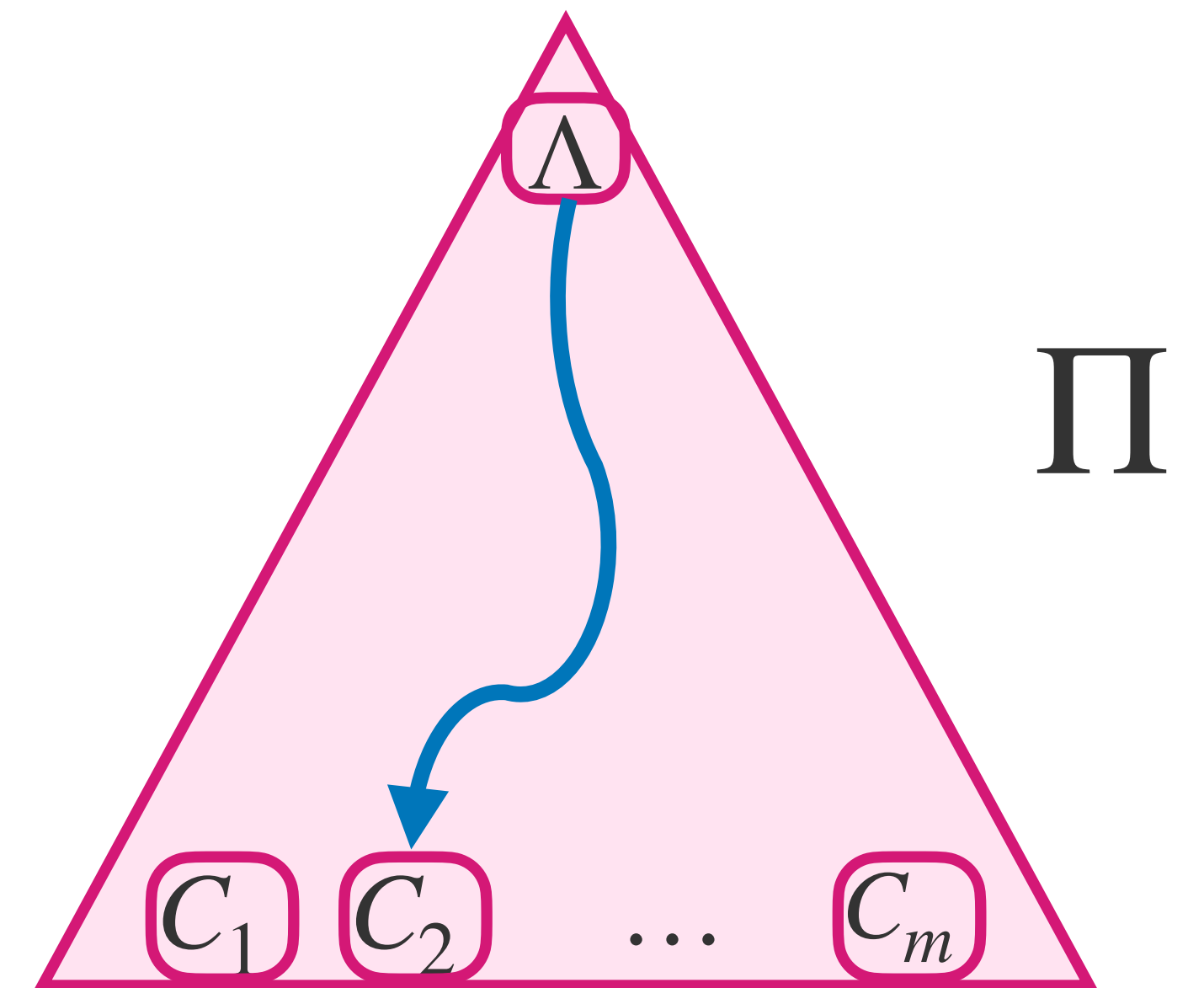
**Pf:**



# Prover Adversary Games

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

**Pf:** Prover will walk from the **root** of  $\Pi$  to **a leaf**

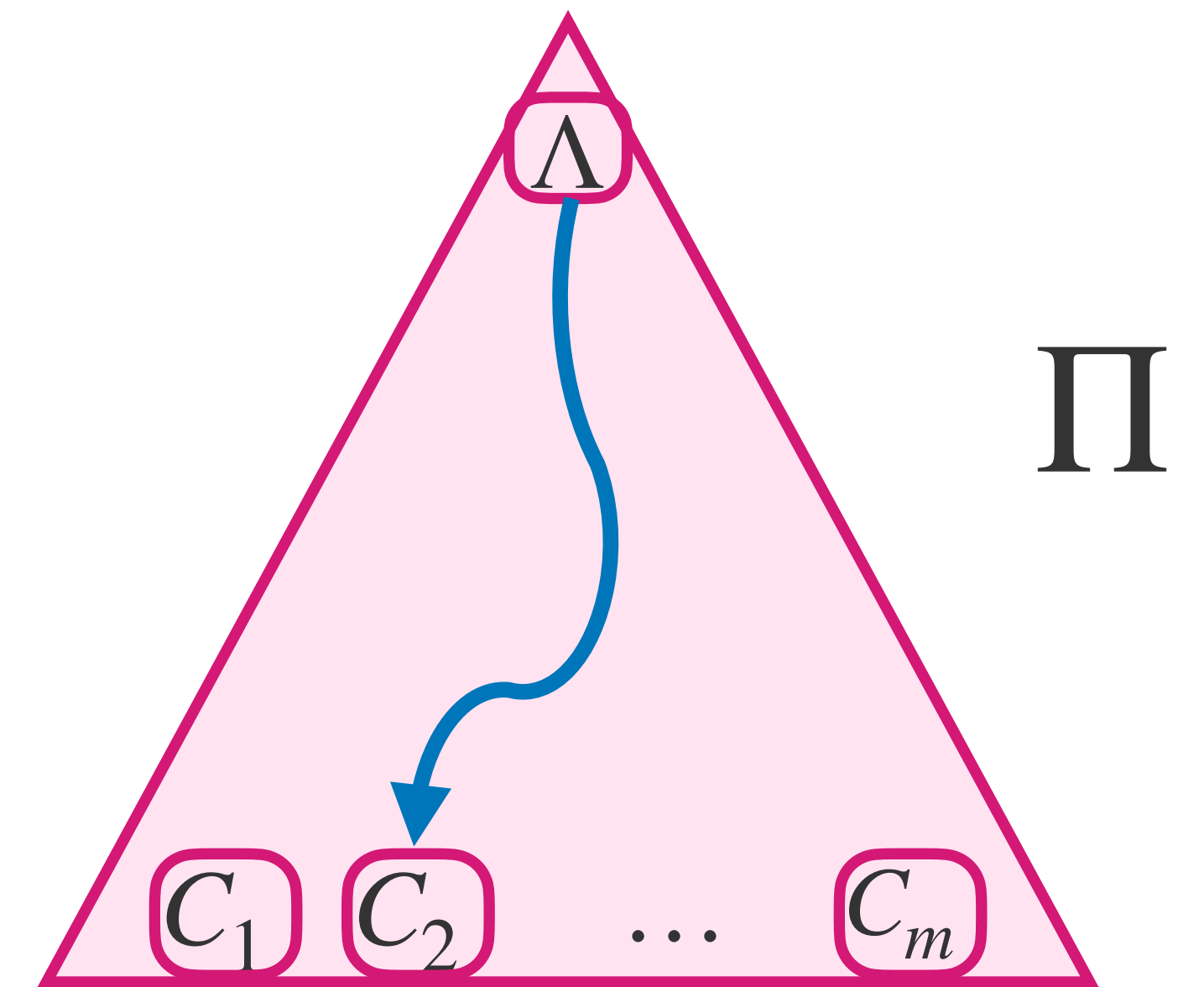


# Prover Adversary Games

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

**Pf:** Prover will walk from the root of  $\Pi$  to a leaf

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



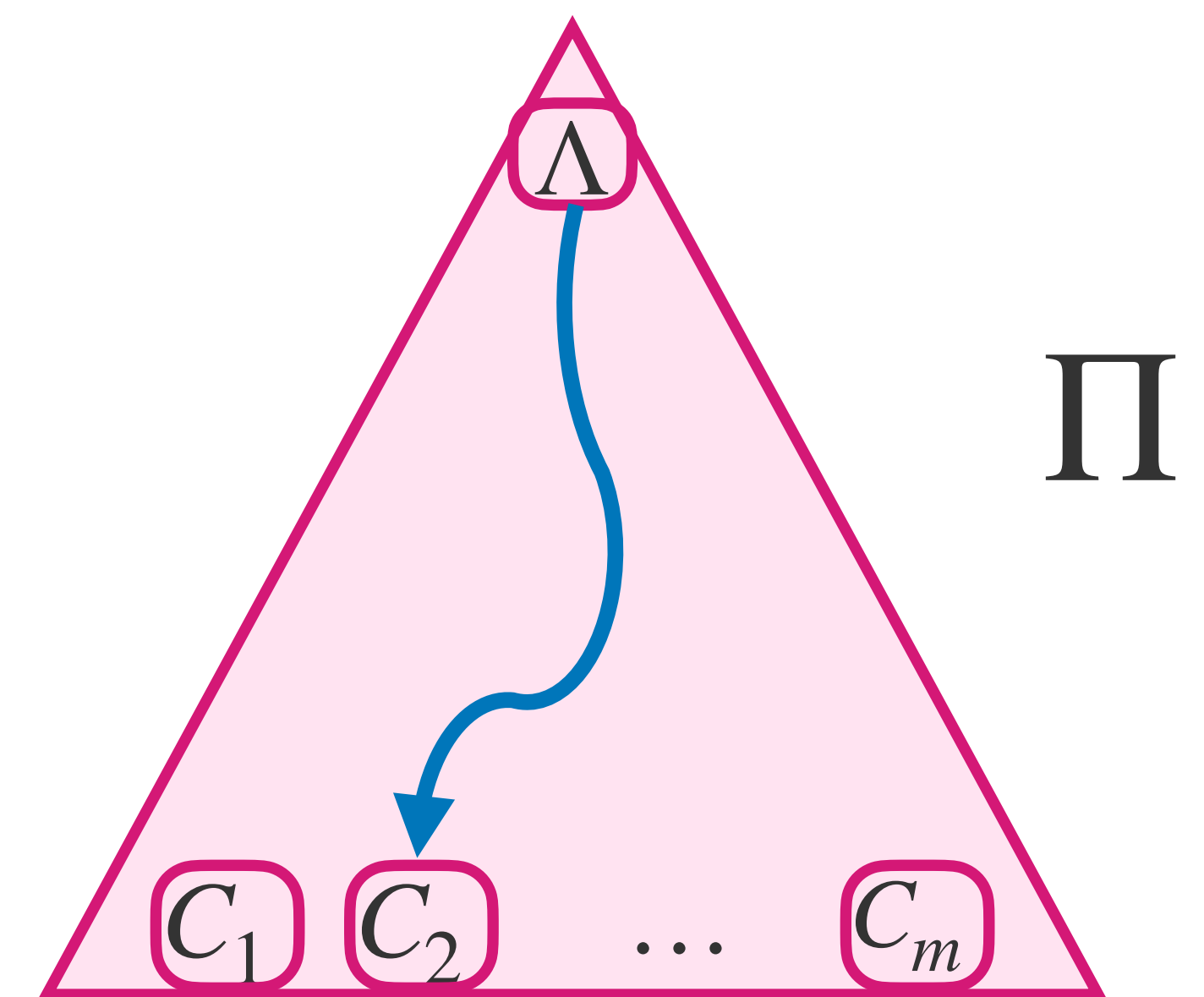
# Prover Adversary Games

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

**Pf:** Prover will walk from the **root** of  $\Pi$  to a **leaf**

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

→ **Root case is satisfied:**  $\Lambda$  is identically false



# Prover Adversary Games

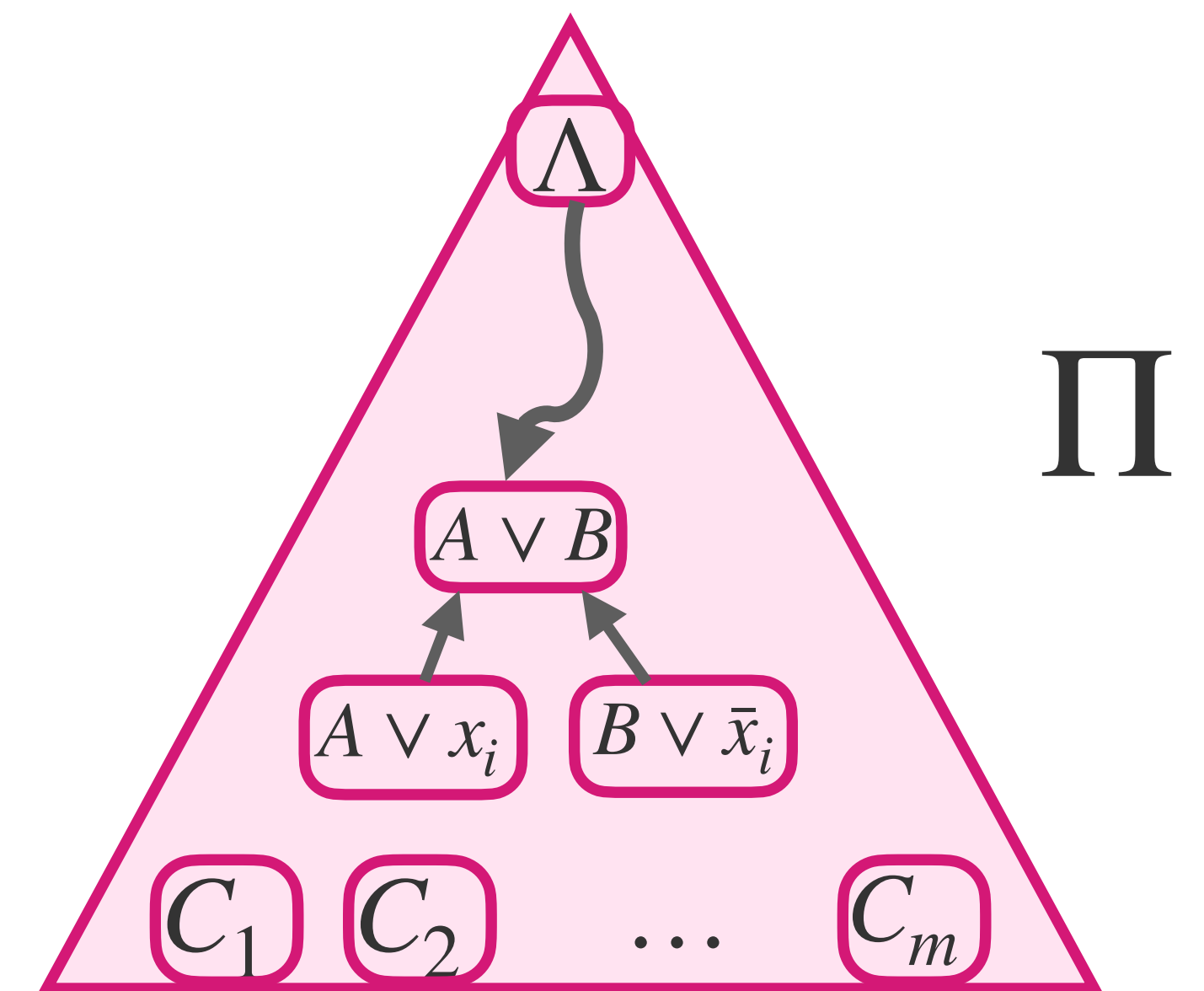
**Claim:** For any  $F$ , a Resolution proof  $\Pi$  of  $F$  of width  $\leq w$  and depth  $\leq d$  implies a strategy for the Prover to win the  $(w + 1)$ -bounded game in  $d$  rounds.

**Pf:** Prover will walk from the root of  $\Pi$  to a leaf

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

→ Root case is satisfied:  $\Lambda$  is identically false

Suppose current clause is  $A \vee B$





# Prover Adversary Games

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

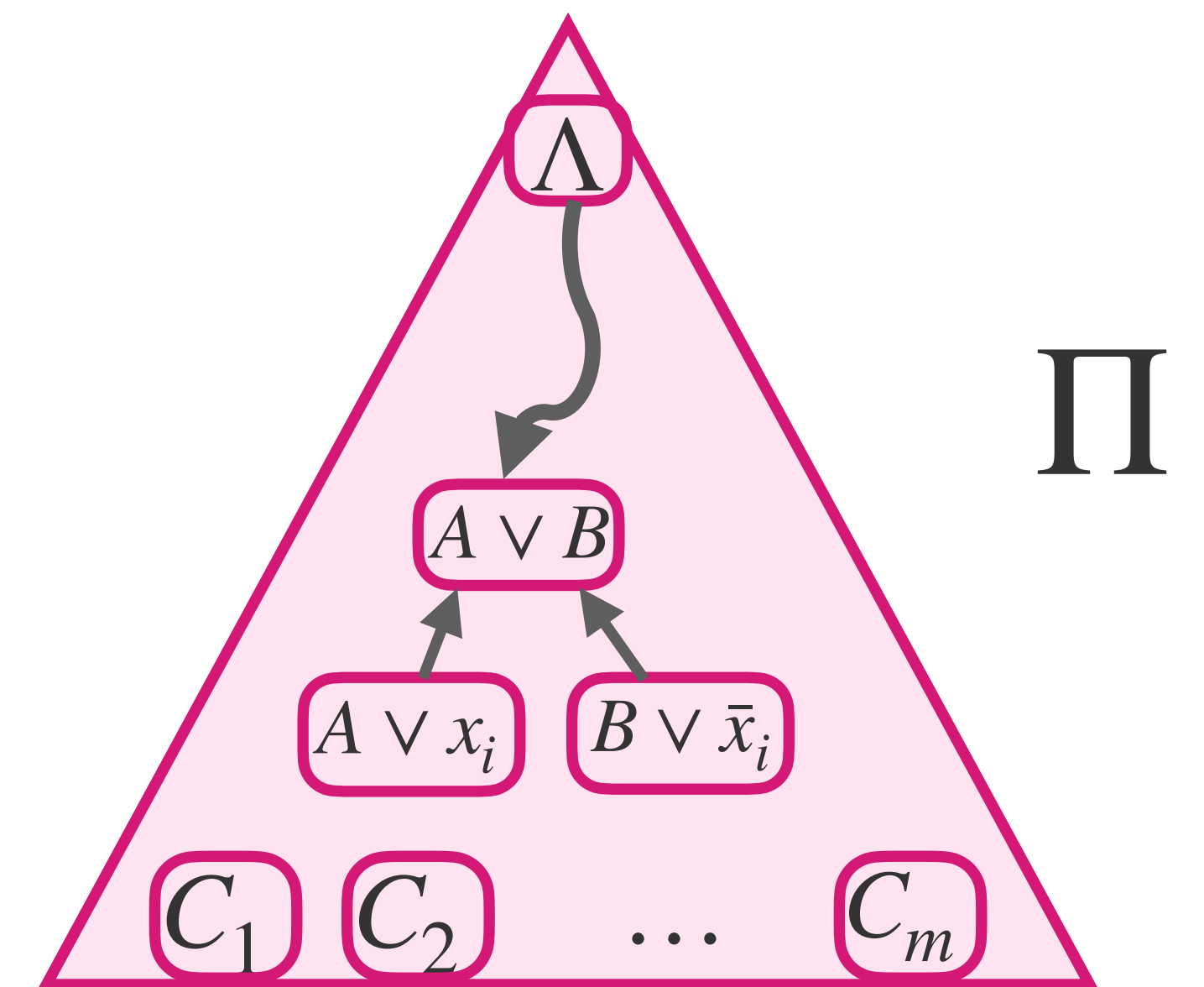
**Pf:** Prover will walk from the **root** of  $\Pi$  to a **leaf**

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

→ **Root case is satisfied:**  $\Lambda$  is identically false

Suppose current clause is  $A \vee B$

- Prover asks about  $x_i$





# Prover Adversary Games

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

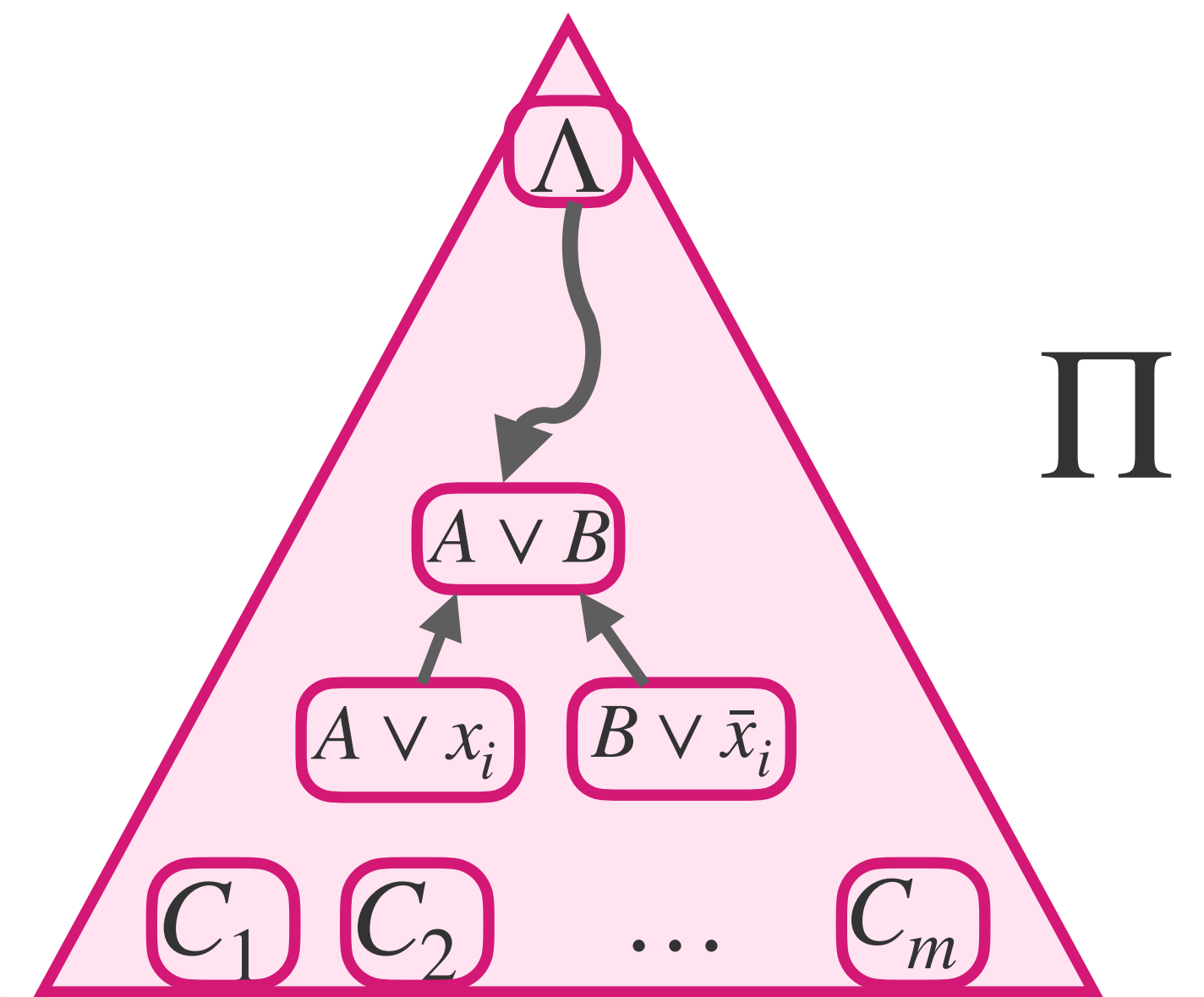
**Pf:** Prover will walk from the root of  $\Pi$  to a leaf

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

→ Root case is satisfied:  $\Lambda$  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  $B \setminus A \cup x_i$



# Prover Adversary Games

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

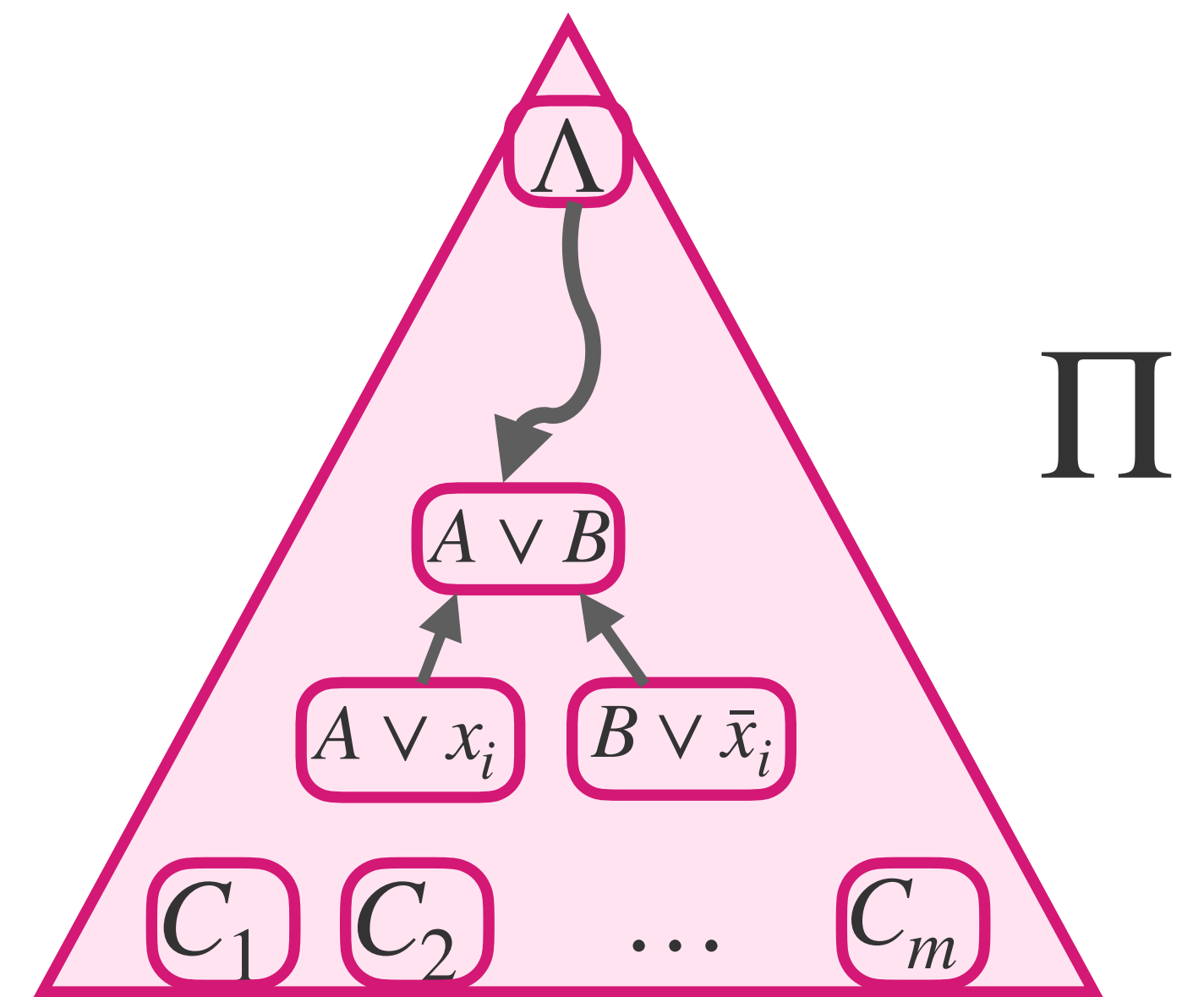
**Pf:** Prover will walk from the root of  $\Pi$  to a leaf

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

→ Root case is satisfied:  $\Lambda$  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  $B \setminus A \cup x_i$
- Otherwise, move to  $B \vee \bar{x}_i$ . Forget  $A \setminus B$



# (New) Proof of Depth Condensation

## Depth Condensation Theorem:

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

If  $\Pi$  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  $\Pi$  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  $\Pi$  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  $F$

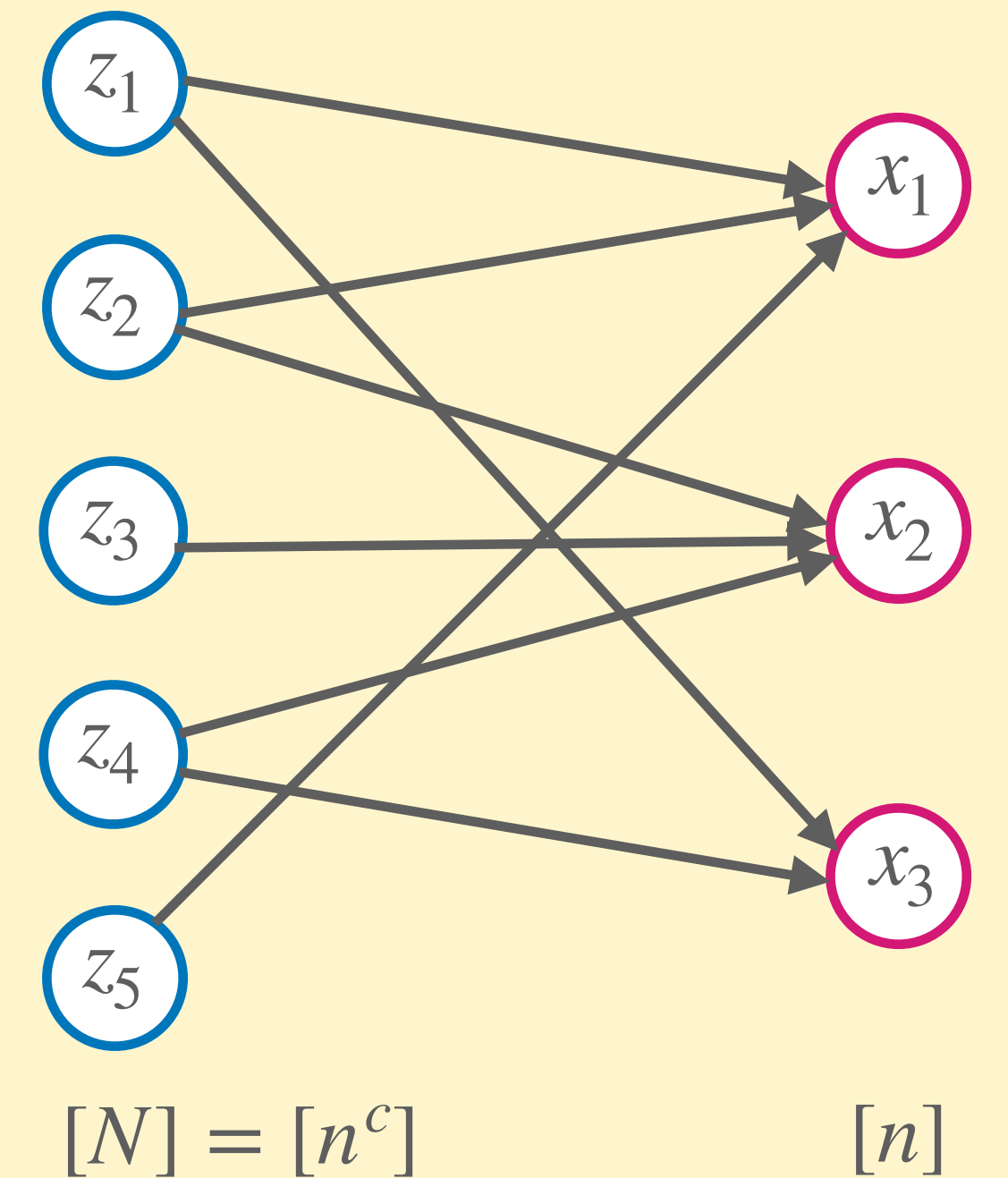
$\rightarrow$  Use  $A$  to construct an **Adversary Strategy** for the  $w$ -bounded game on  $F \circ XOR_G$  to survive  $\Omega(d/w)$  rounds, for any  $w \leq r/4$ .

# (New) Proof of Depth Condensation

## High Level of Proof:

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

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





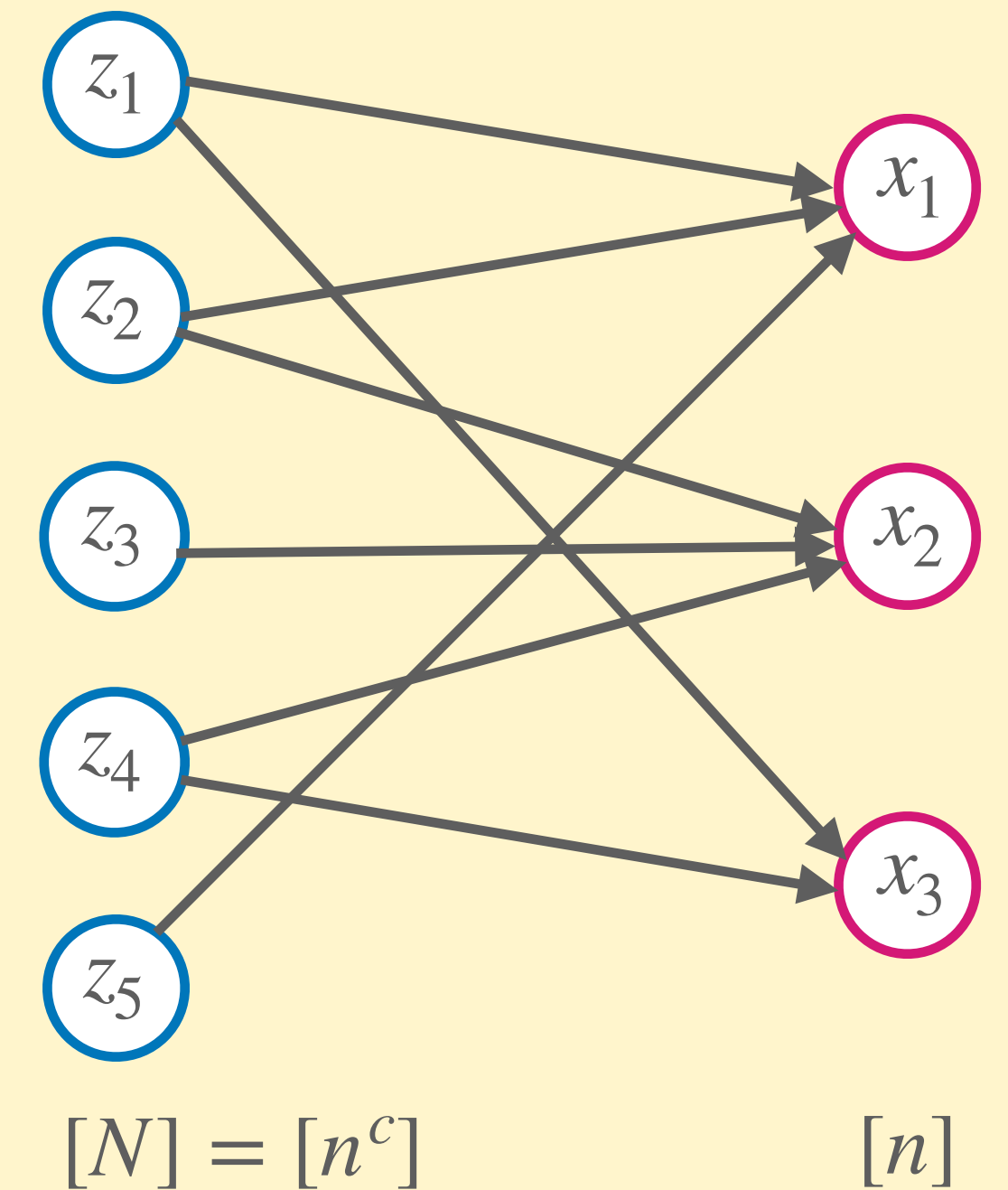
# (New) Proof of Depth Condensation

## 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  $x_i$ :



# (New) Proof of Depth Condensation

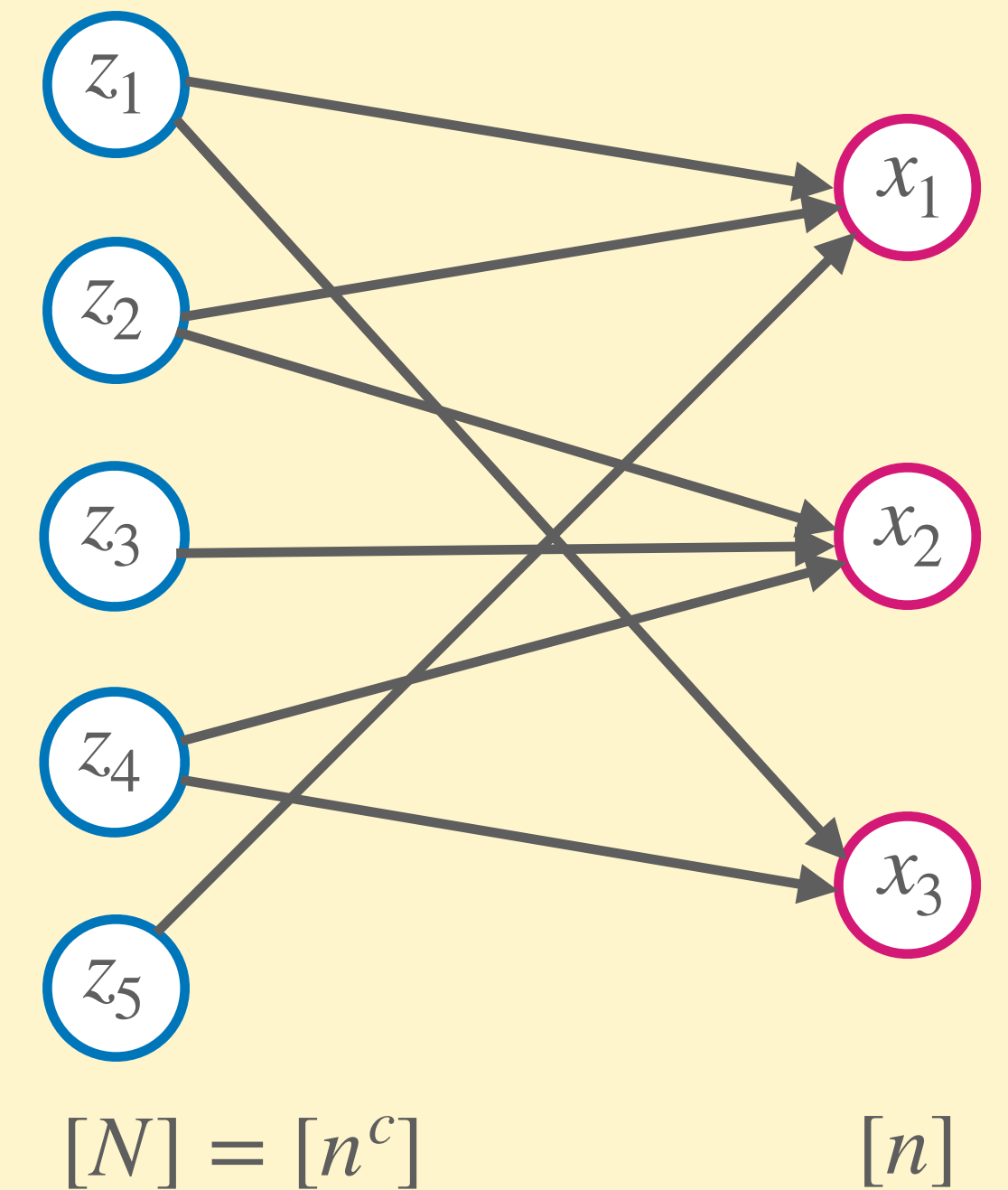
## High Level of Proof:

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  $\geq 2$  variables in  $N(z_j)$  for every  $z_j \in N(x_i)$ :  
set  $x_i$  arbitrarily





# (New) Proof of Depth Condensation

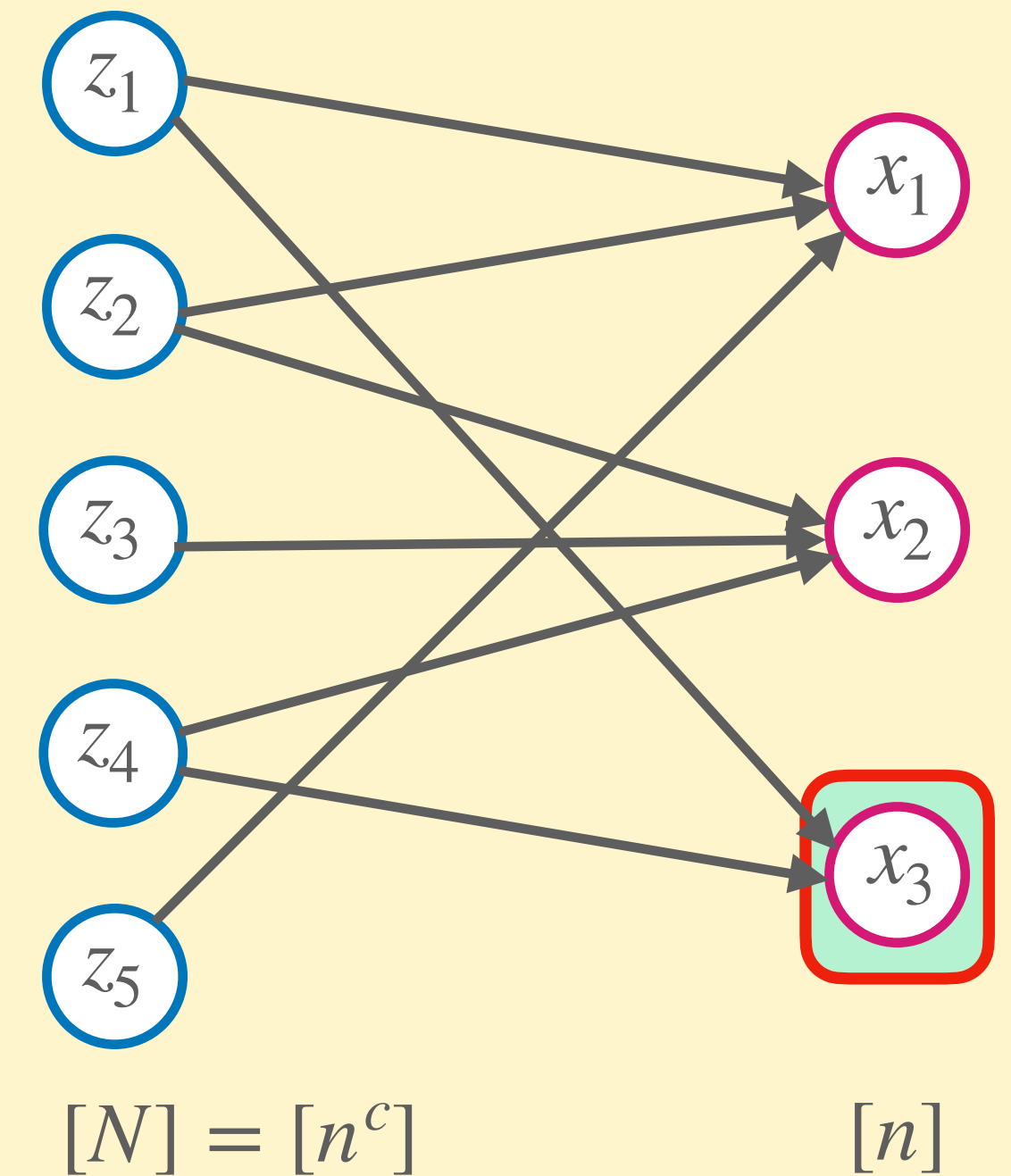
## High Level of Proof:

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  $\geq 2$  variables in  $N(z_j)$  for every  $z_j \in N(x_i)$ :  
set  $x_i$  arbitrarily



# (New) Proof of Depth Condensation

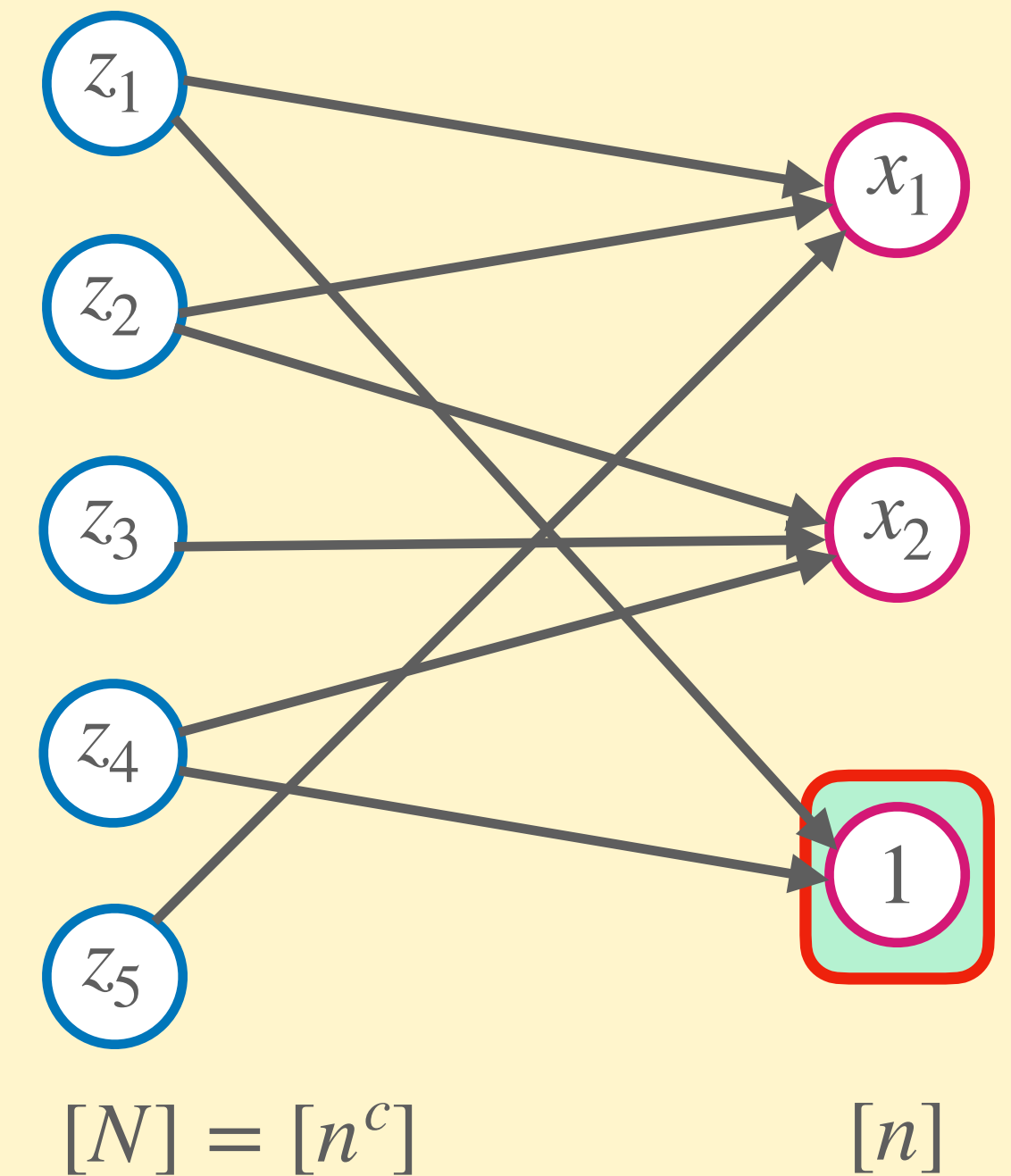
## High Level of Proof:

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  $\geq 2$  variables in  $N(z_j)$  for every  $z_j \in N(x_i)$ :  
set  $x_i$  arbitrarily



# (New) Proof of Depth Condensation

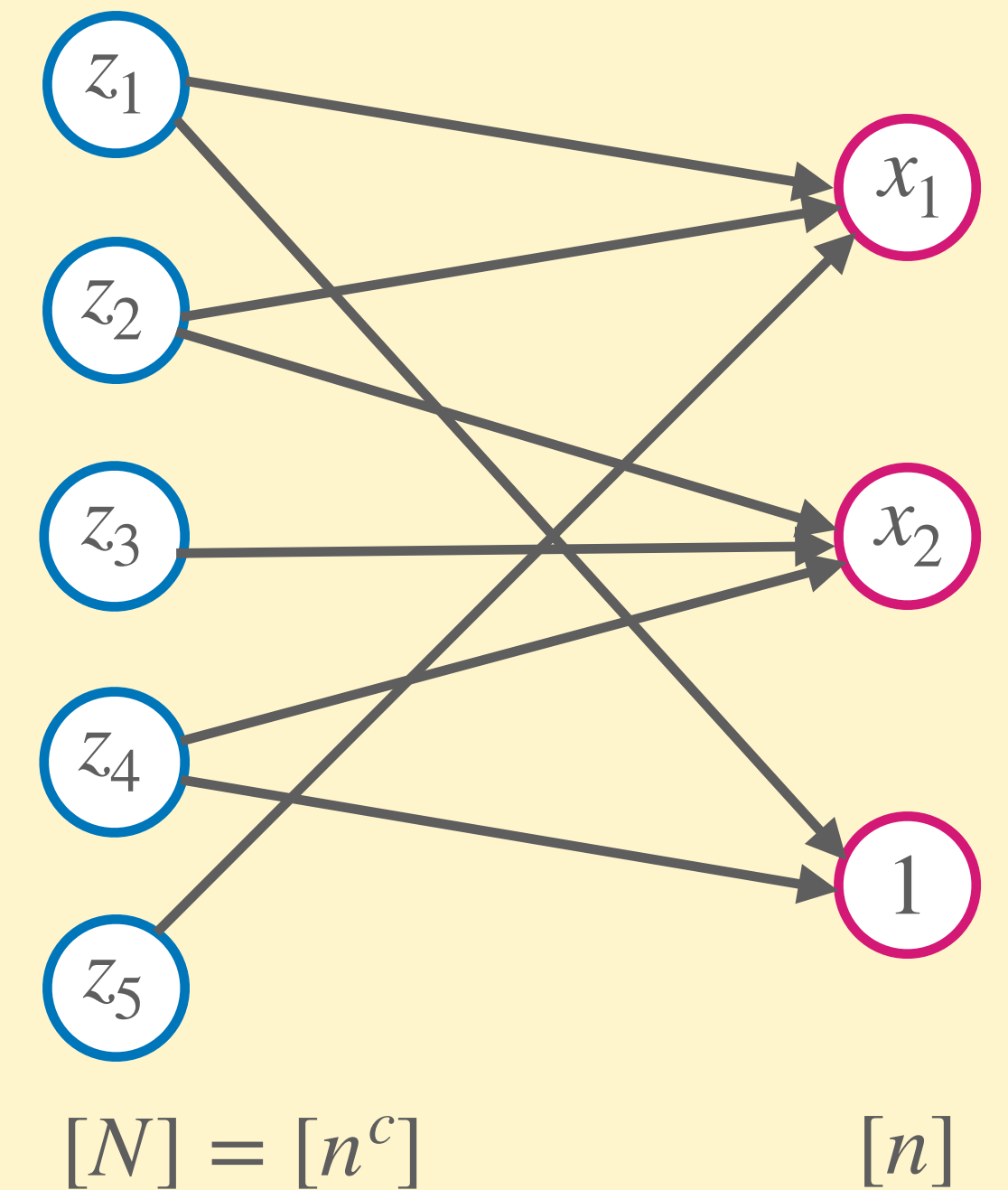
## High Level of Proof:

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  $\geq 2$  variables in  $N(z_j)$  for every  $z_j \in N(x_i)$ :  
set  $x_i$  arbitrarily
- If  $x_i$  is the last variable in  $N(z_j)$  (for some  $z_j$ ) not set in  $\rho$ :



# (New) Proof of Depth Condensation

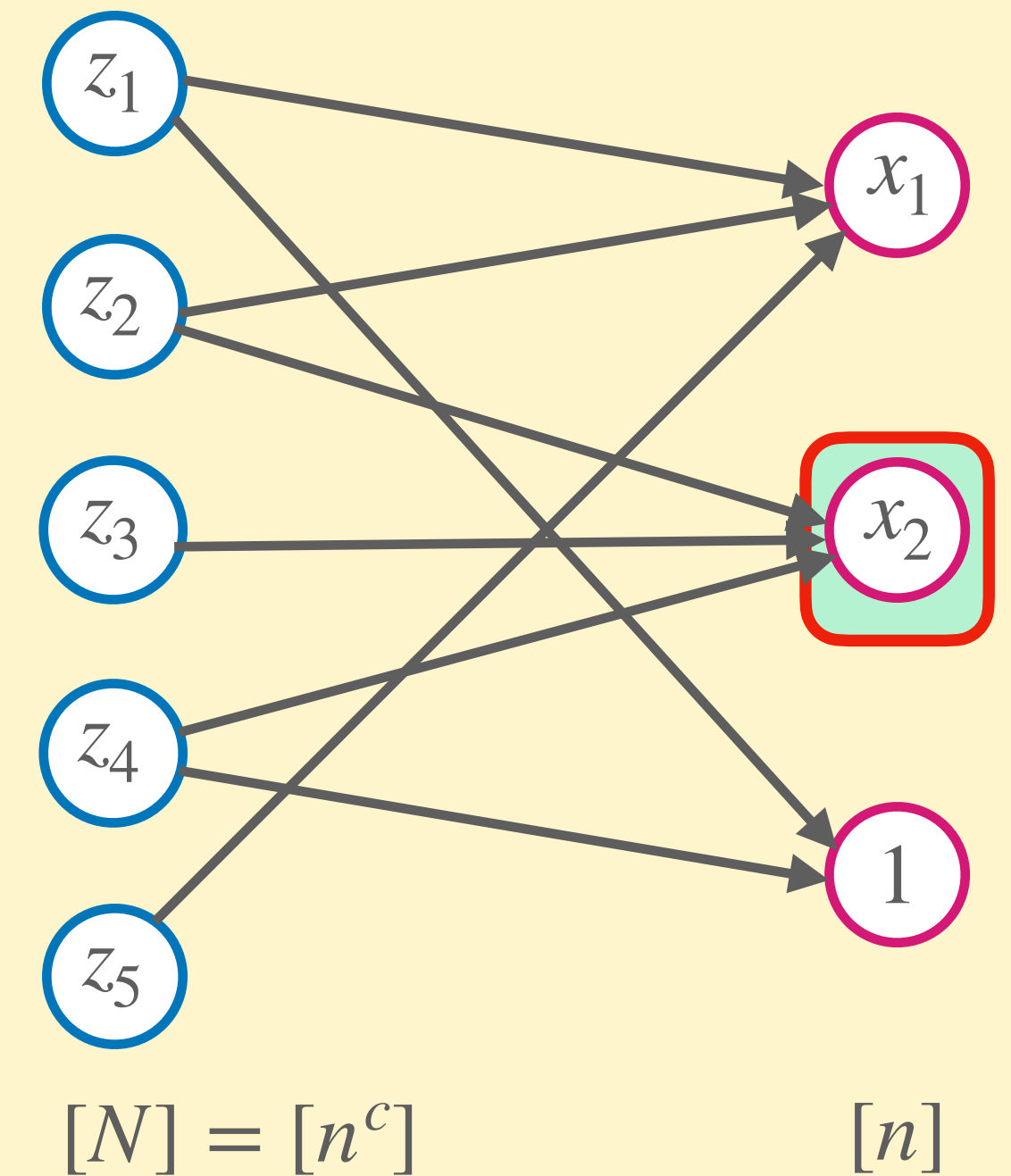
## High Level of Proof:

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  $\geq 2$  variables in  $N(z_j)$  for every  $z_j \in N(x_i)$ :  
set  $x_i$  arbitrarily
- If  $x_i$  is the last variable in  $N(z_j)$  (for some  $z_j$ ) not set in  $\rho$ :



# (New) Proof of Depth Condensation

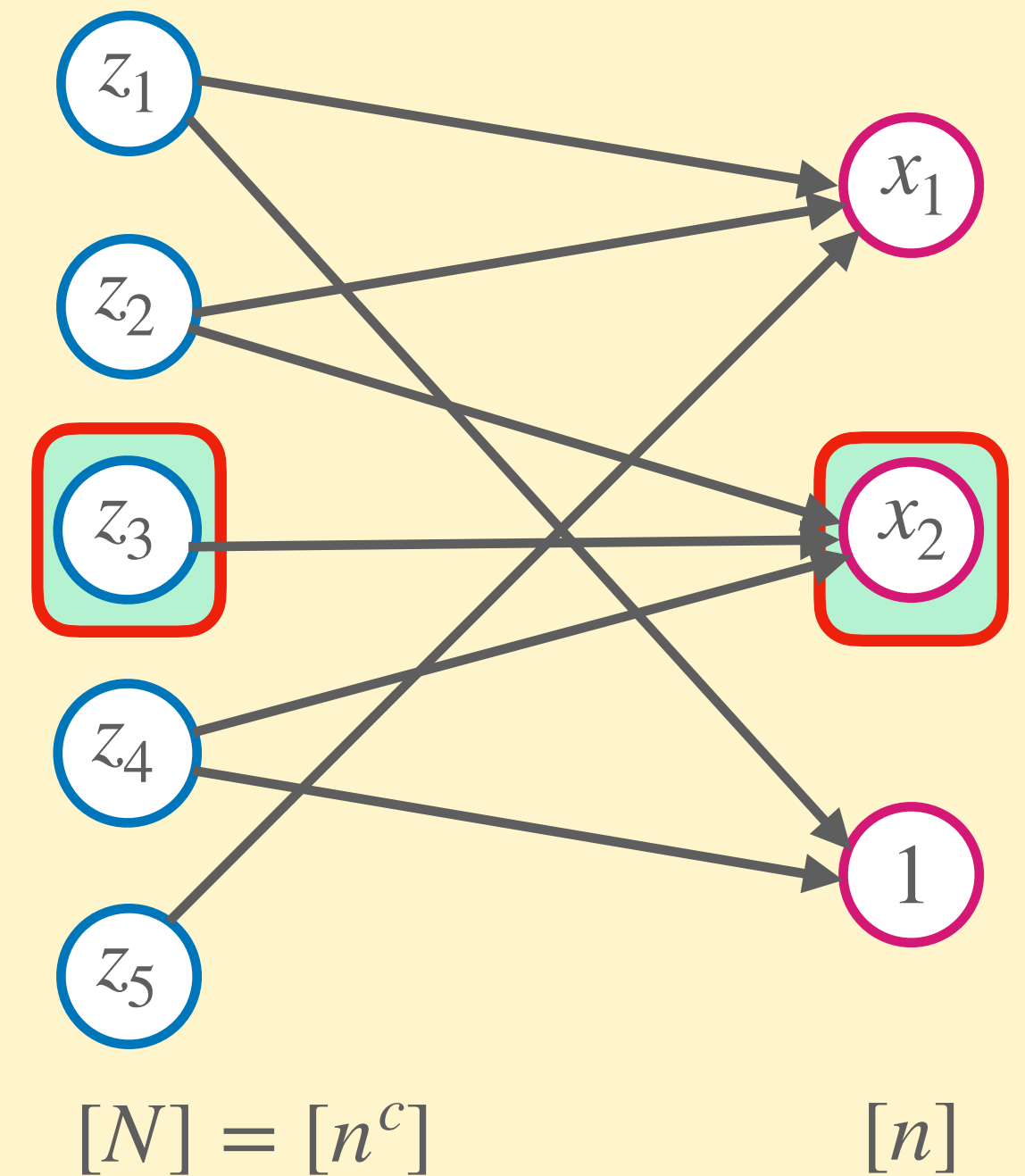
## High Level of Proof:

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  $\geq 2$  variables in  $N(z_j)$  for every  $z_j \in N(x_i)$ :  
set  $x_i$  arbitrarily
- If  $x_i$  is the last variable in  $N(z_j)$  (for some  $z_j$ ) not set in  $\rho$ :



# (New) Proof of Depth Condensation

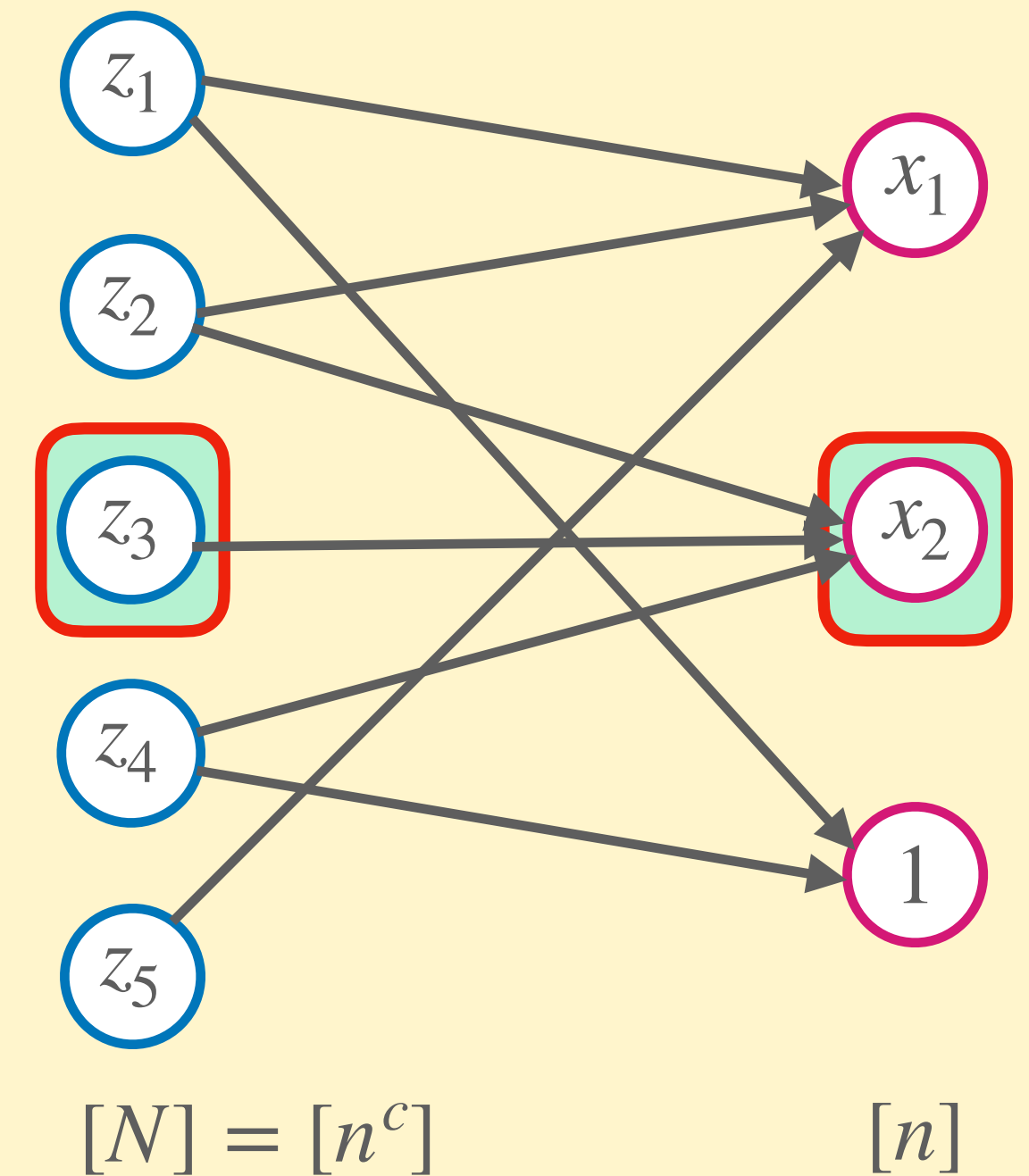
## High Level of Proof:

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





# (New) Proof of Depth Condensation

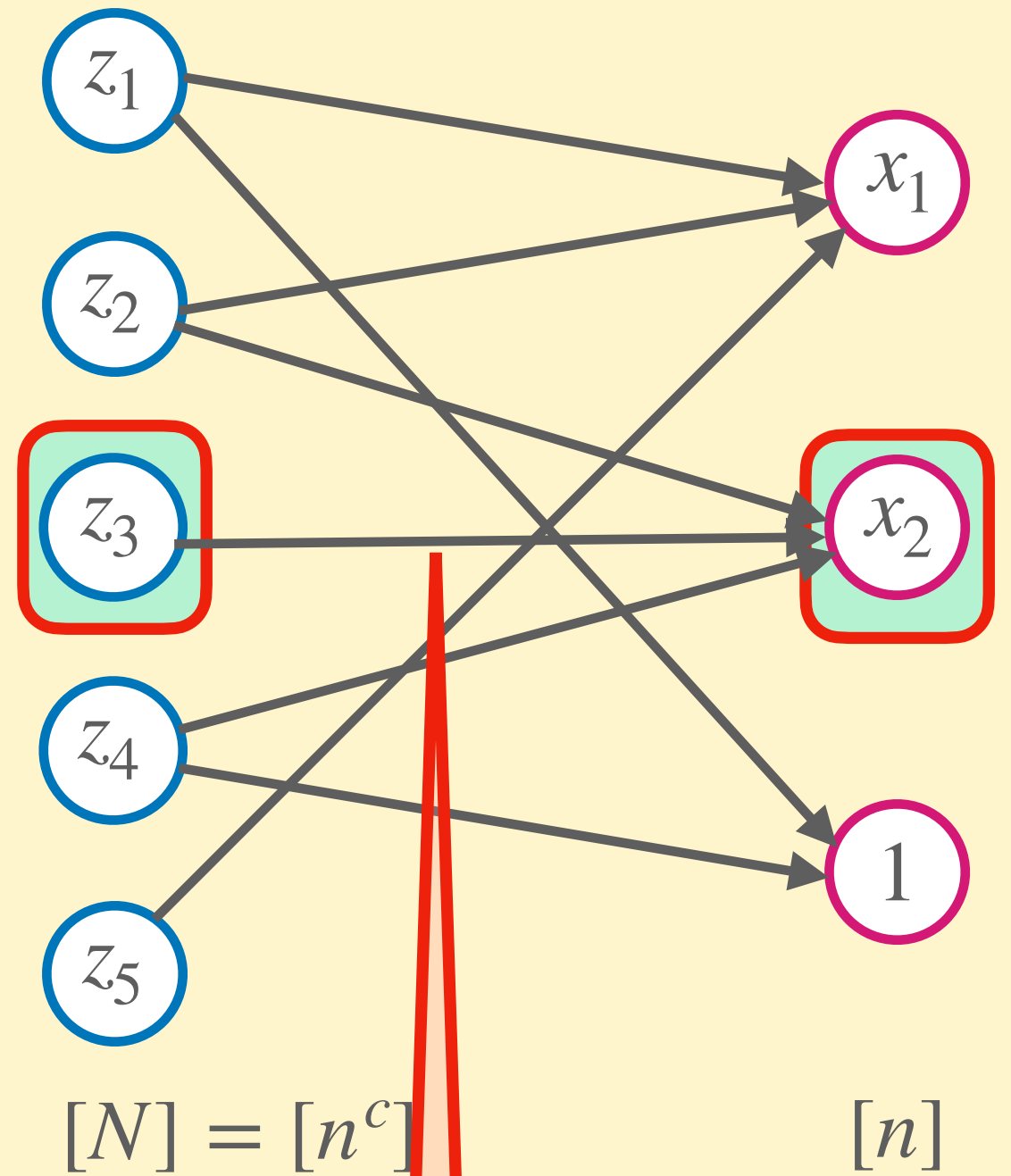
## High Level of Proof:

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



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

# (New) Proof of Depth Condensation

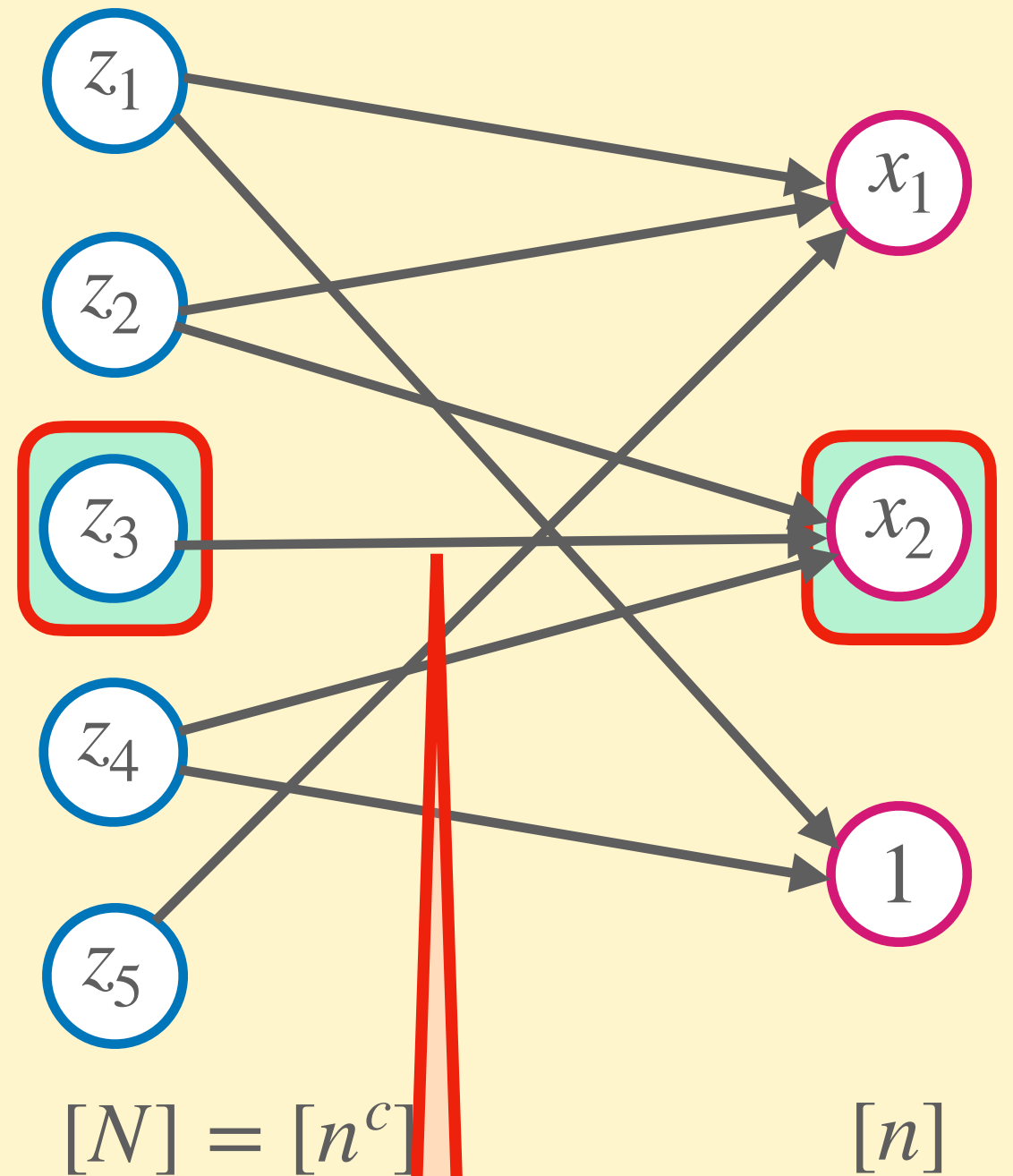
## High Level of Proof:

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  $\geq 2$  variables in  $N(z_j)$  for every  $z_j \in N(x_i)$ :  
set  $x_i$  arbitrarily
- If  $x_i$  is the last variable in  $N(z_j)$  (for some  $z_j$ ) not set in  $\rho$ :  
– 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

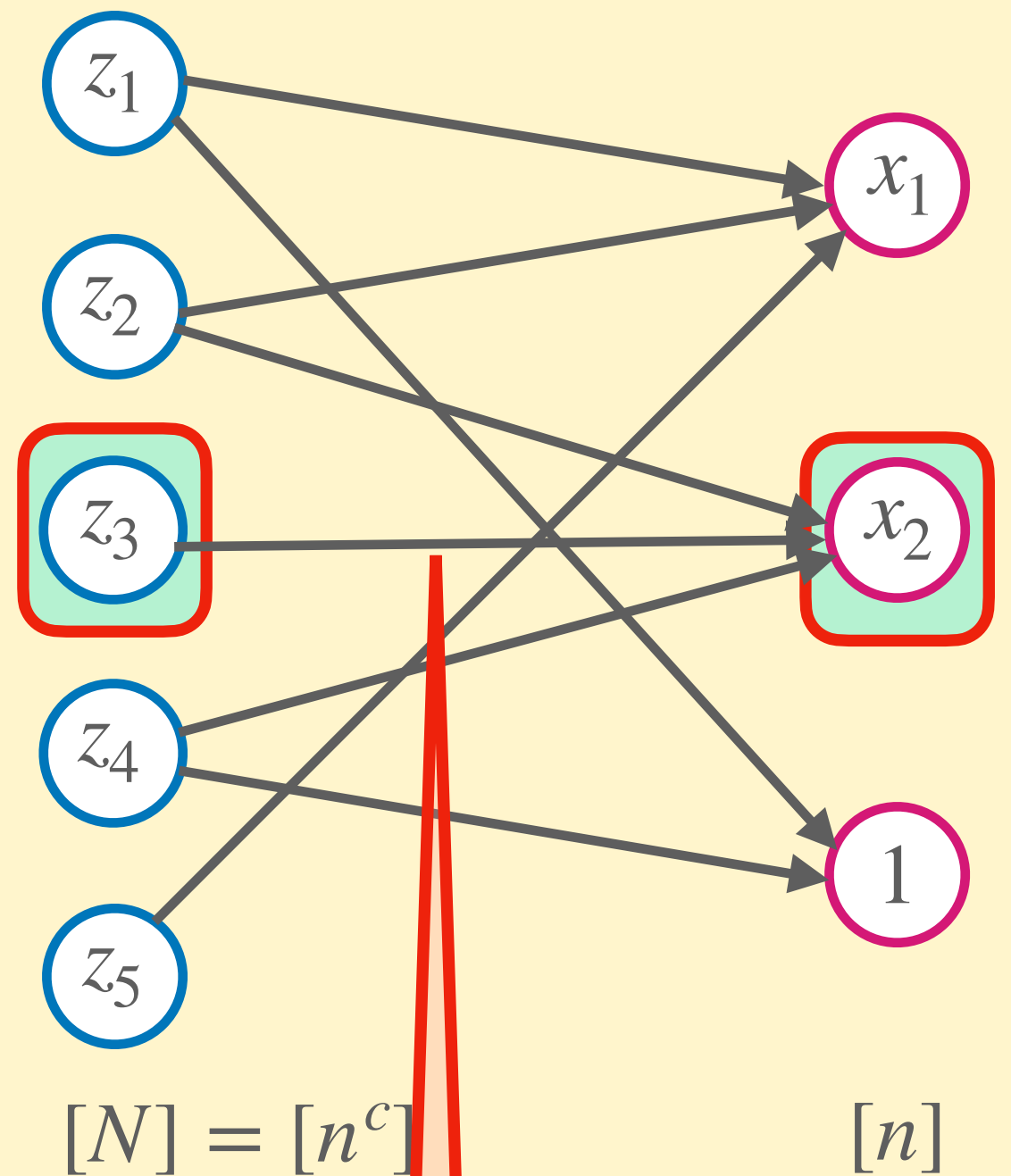
## High Level of Proof:

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  $\geq 2$  variables in  $N(z_j)$  for every  $z_j \in N(x_i)$ :  
set  $x_i$  arbitrarily
- If  $x_i$  is the **last** variable in  $N(z_j)$  (for some  $z_j$ ) not set in  $\rho$ :
  - 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

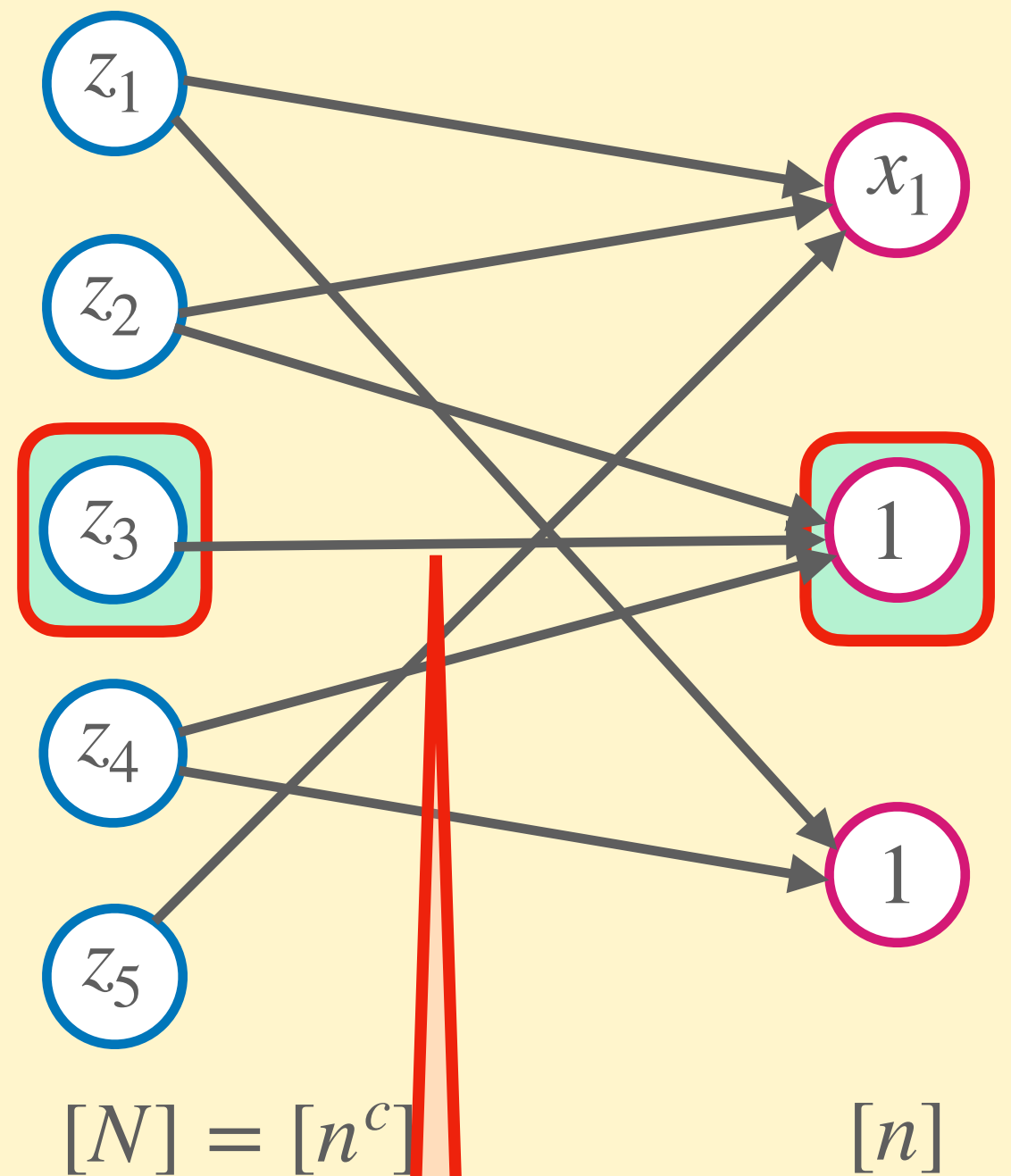
## High Level of Proof:

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  $\geq 2$  variables in  $N(z_j)$  for every  $z_j \in N(x_i)$ :  
set  $x_i$  arbitrarily
- If  $x_i$  is the **last** variable in  $N(z_j)$  (for some  $z_j$ ) not set in  $\rho$ :
  - 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

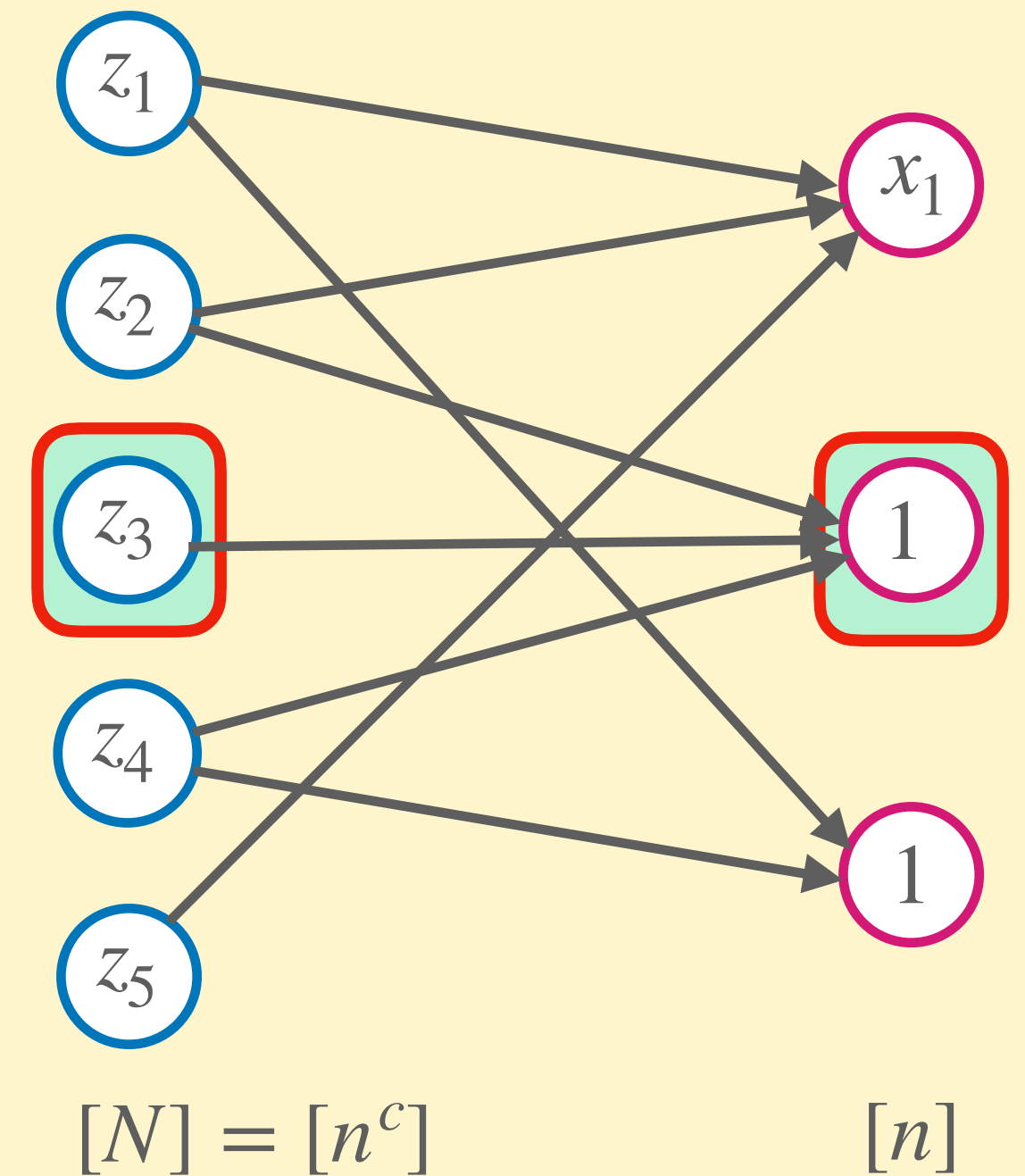
## High Level of Proof:

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  $\geq 2$  variables in  $N(z_j)$  for every  $z_j \in N(x_i)$ :  
set  $x_i$  arbitrarily
- If  $x_i$  is the last variable in  $N(z_j)$  (for some  $z_j$ ) not set in  $\rho$ :
  - 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  $z_4 = 0$

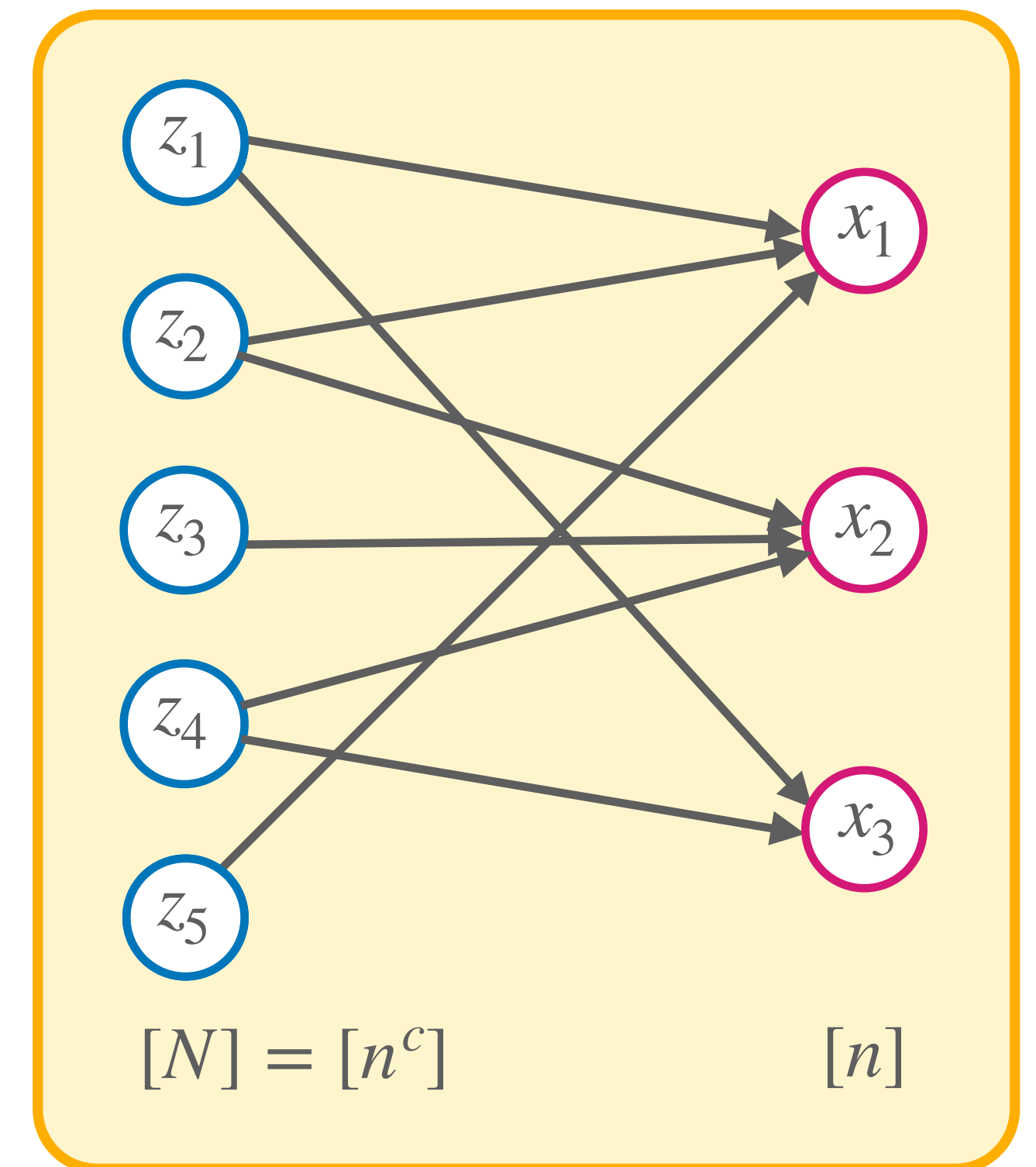
What if  $A$  sets  $z_4 = 1$ ?

# Proof Overview

**Problem:**  $z$ -variables are correlated

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

→ Cannot follow  $A$  in this case



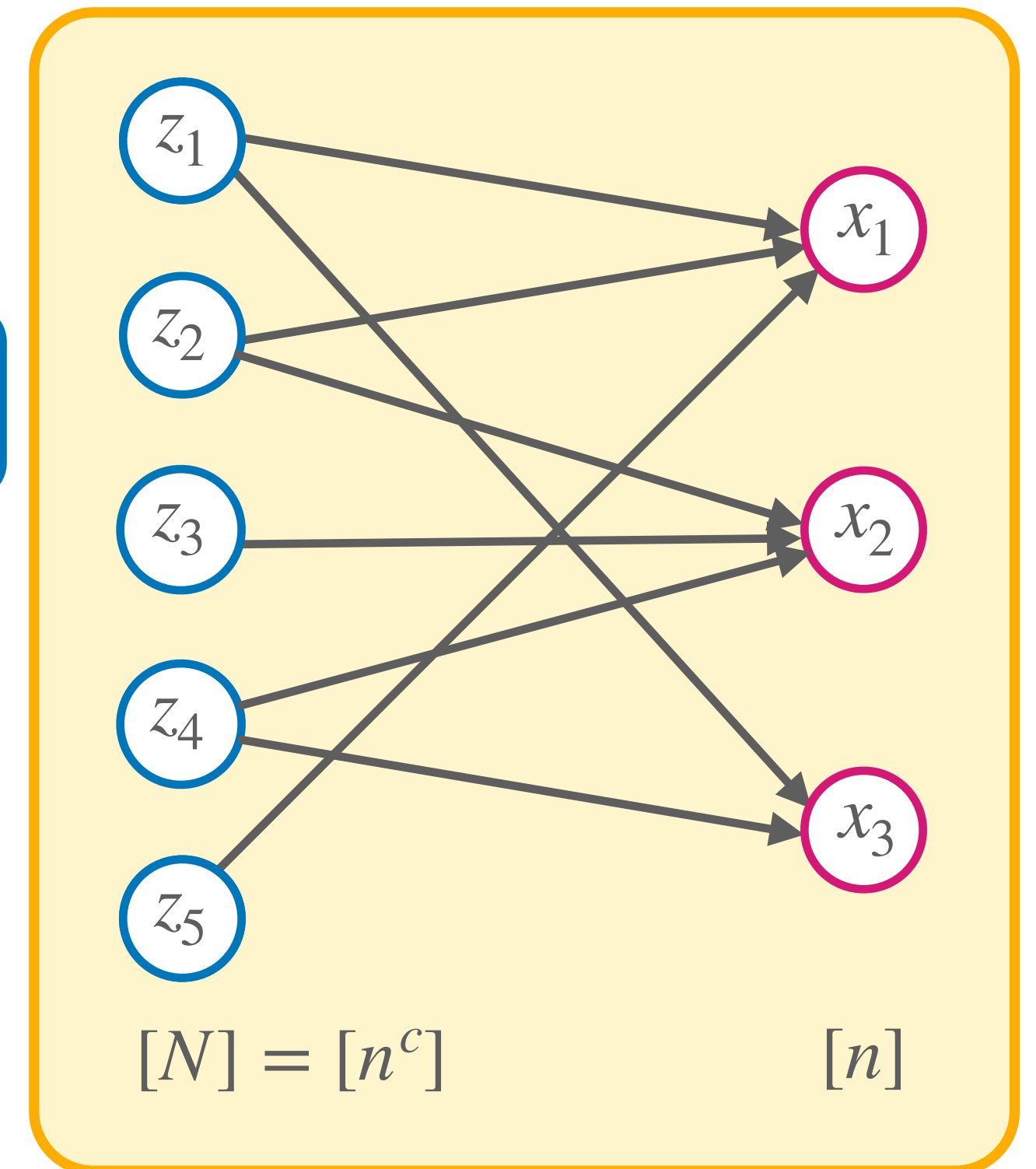
# Proof Overview

**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

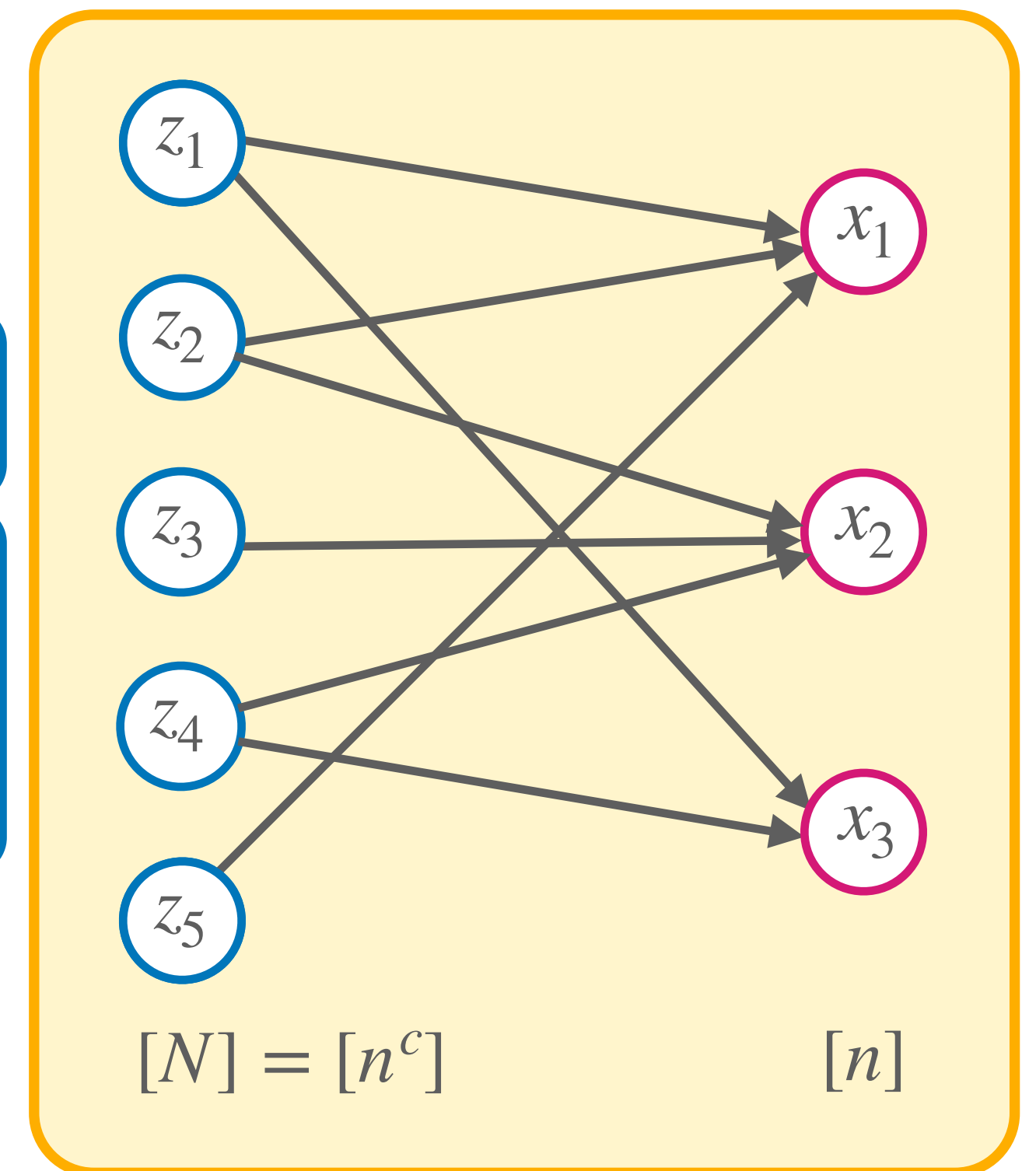
**Problem:**  $z$ -variables are correlated

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

→ Cannot follow  $A$  in this case

Use **expansion** to avoid this scenario!

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





# Proof Overview

**Problem:**  $z$ -variables are correlated

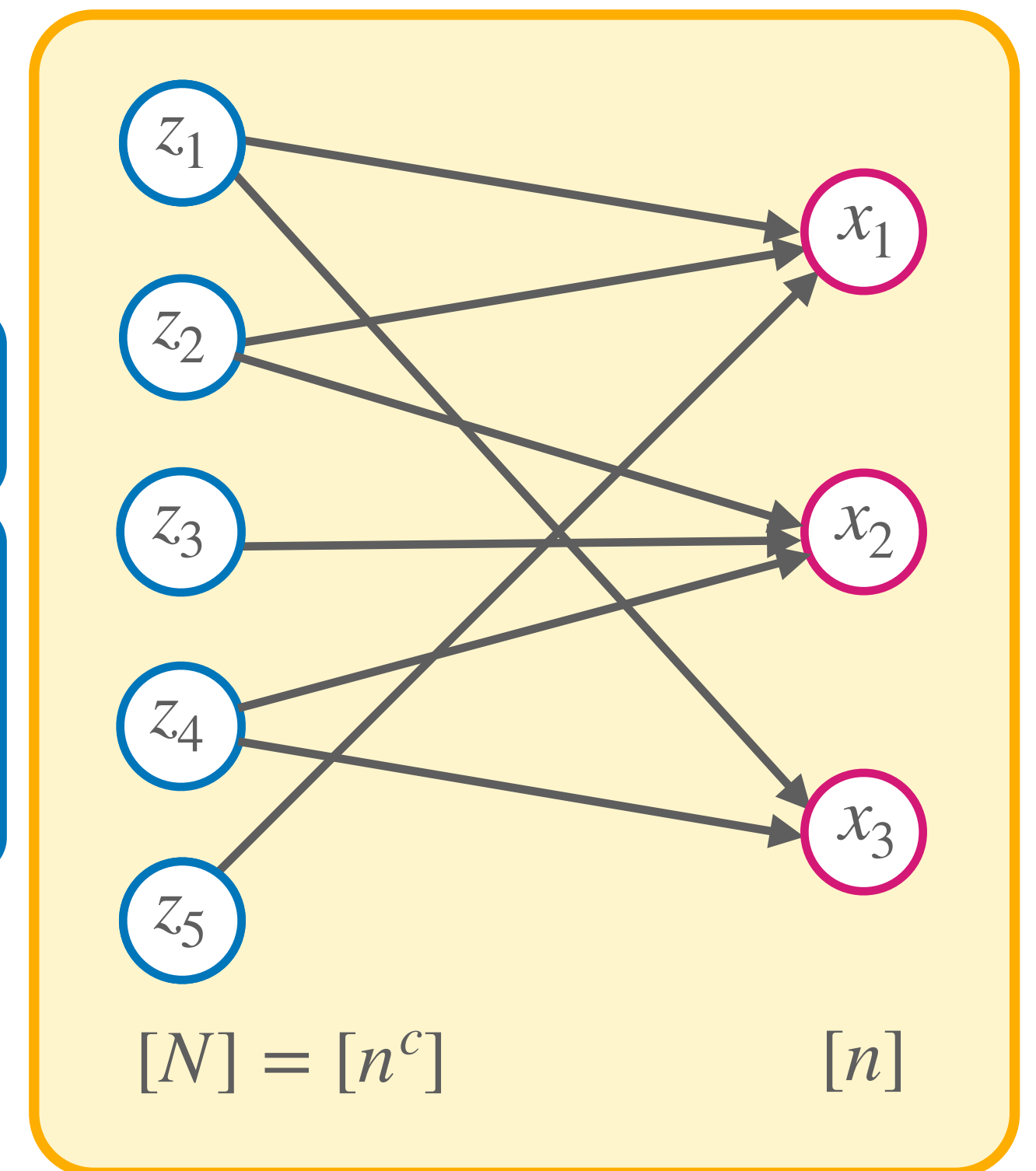
→ Setting one  $x$ -variable can force several  $z$ -variables

→ Cannot follow  $A$  in this case

Use **expansion** to avoid this scenario!

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

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



# Proof Overview

**Problem:**  $z$ -variables are correlated

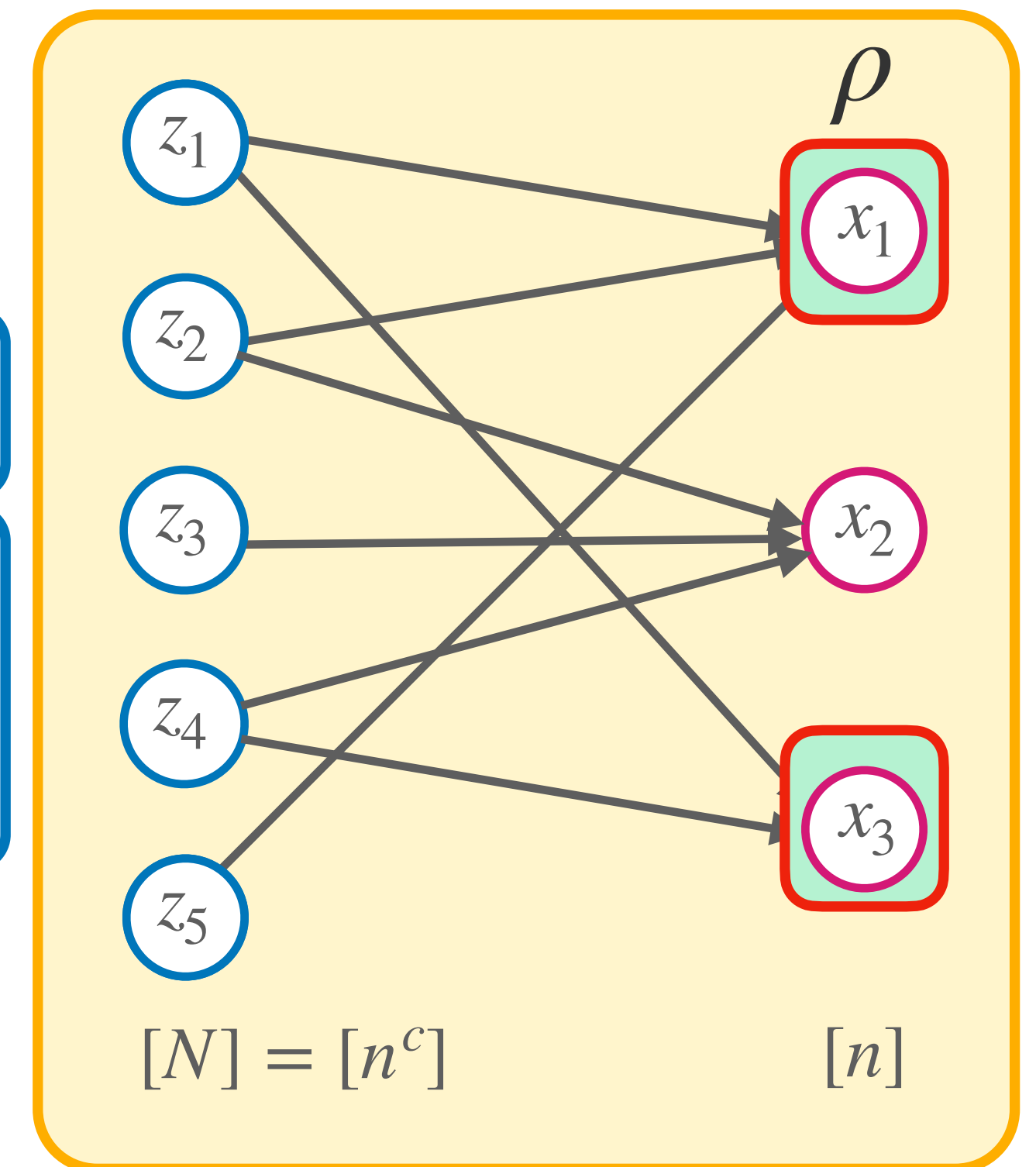
→ Setting one  $x$ -variable can force several  $z$ -variables

→ Cannot follow  $A$  in this case

Use **expansion** to avoid this scenario!

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

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





# Proof Overview

**Problem:**  $z$ -variables are correlated

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

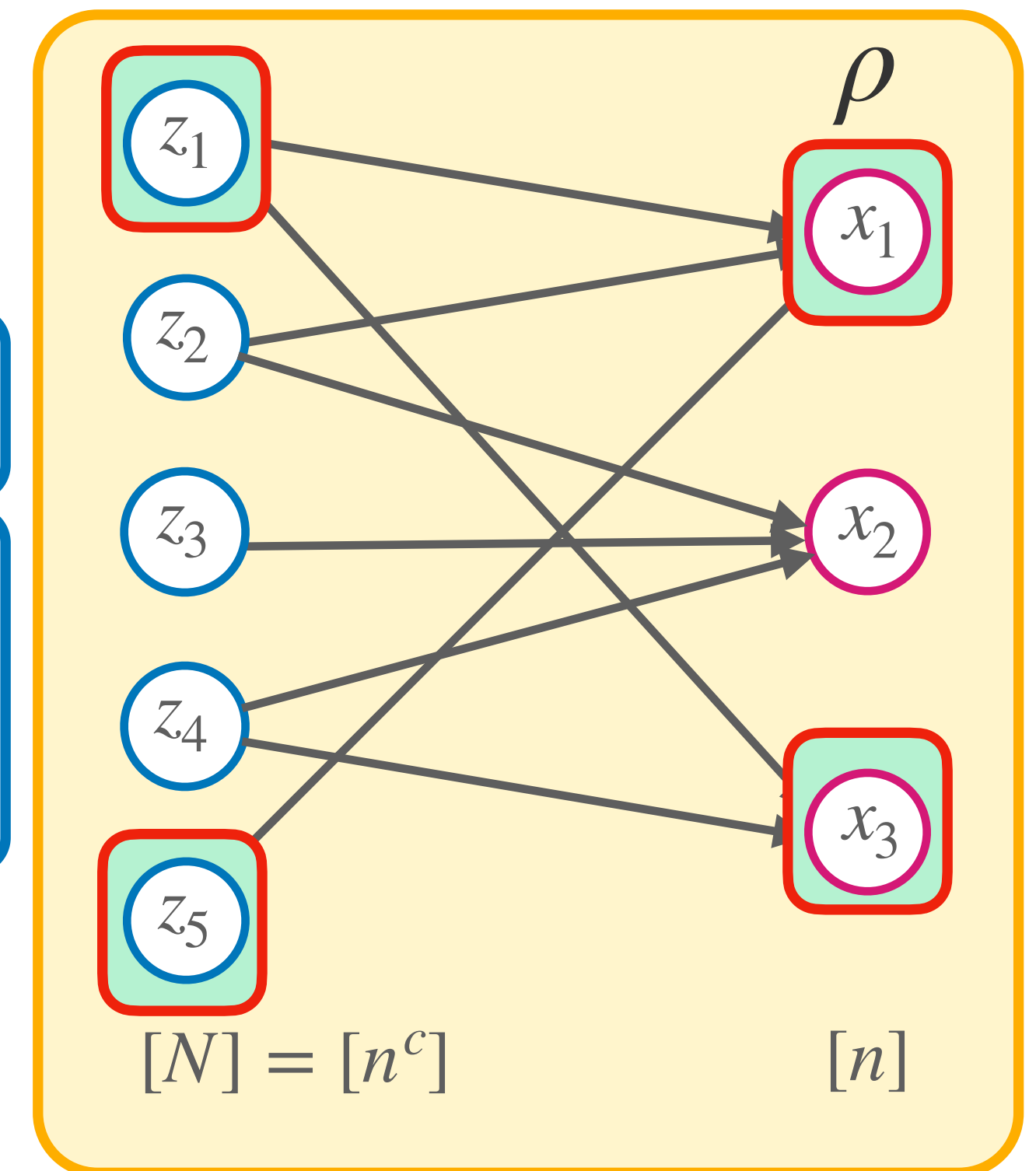
→ Cannot follow  $A$  in this case

Use **expansion** to avoid this scenario!

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

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

Determined



# Proof Overview

**Problem:**  $z$ -variables are correlated

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

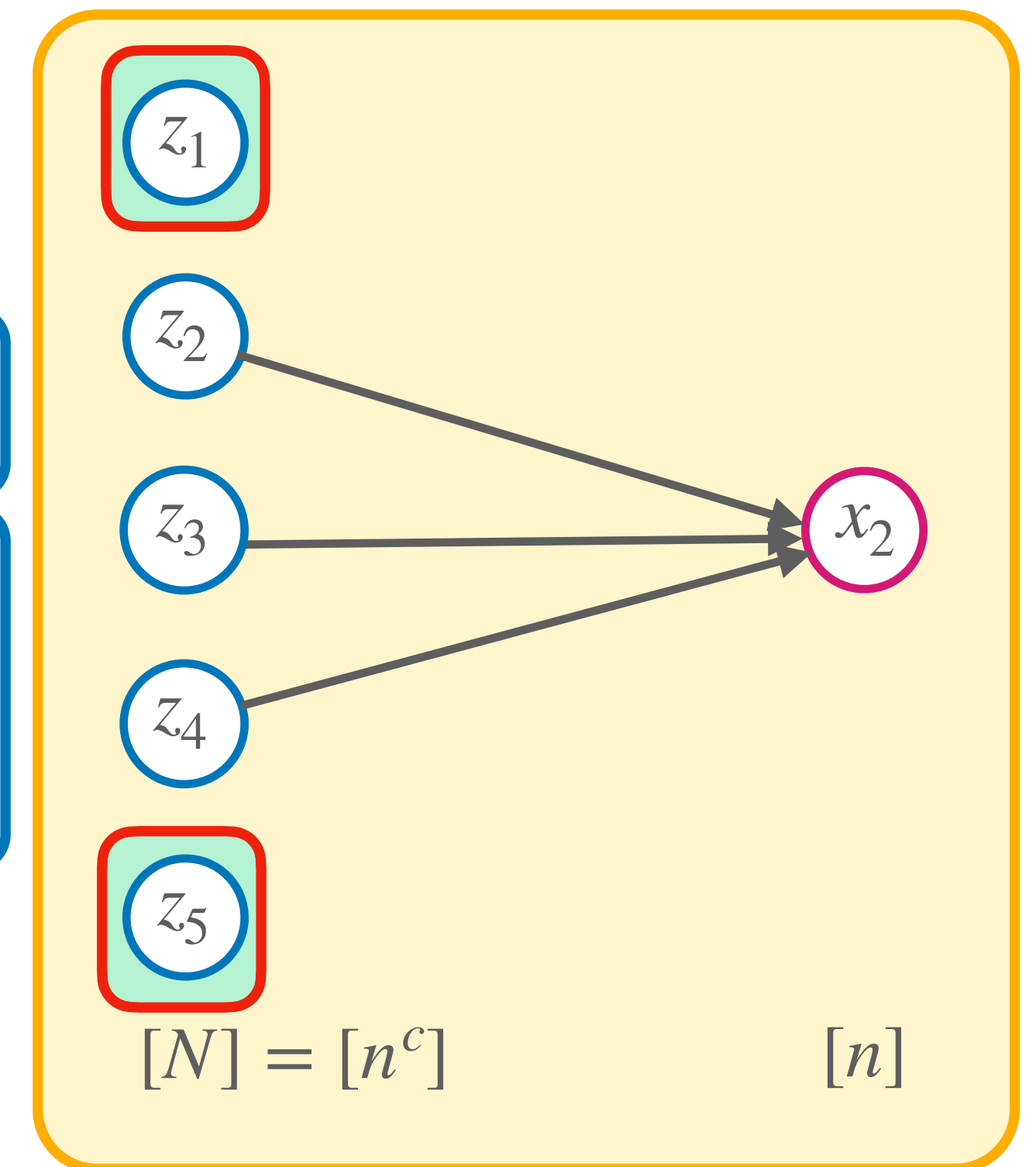
→ Cannot follow  $A$  in this case

Use **expansion** to avoid this scenario!

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

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

Determined



# Proof Overview

**Problem:**  $z$ -variables are correlated

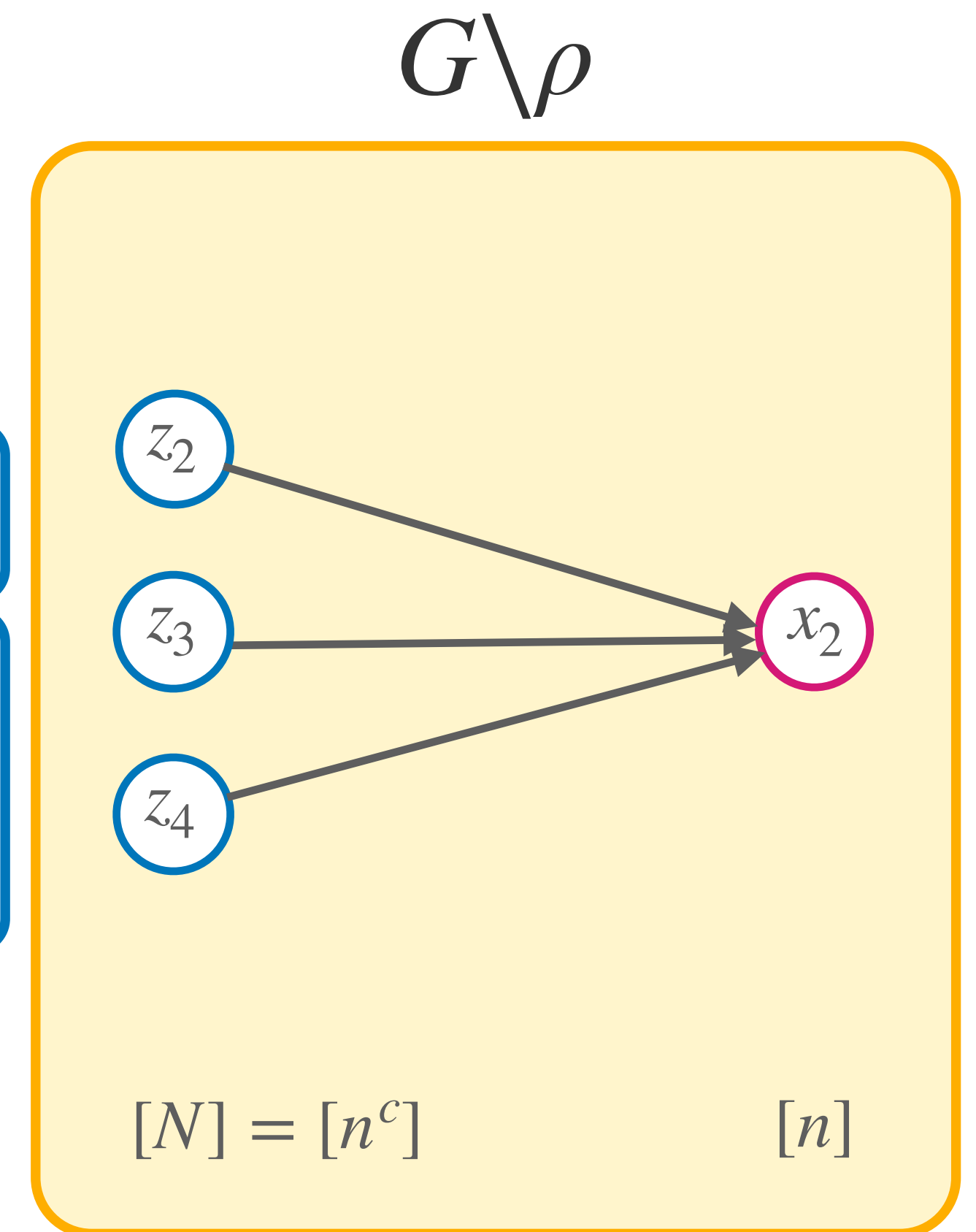
→ Setting one  $x$ -variable can force several  $z$ -variables

→ Cannot follow  $A$  in this case

Use **expansion** to avoid this scenario!

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

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



# Proof Overview

**Problem:**  $z$ -variables are correlated

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

→ Cannot follow  $A$  in this case

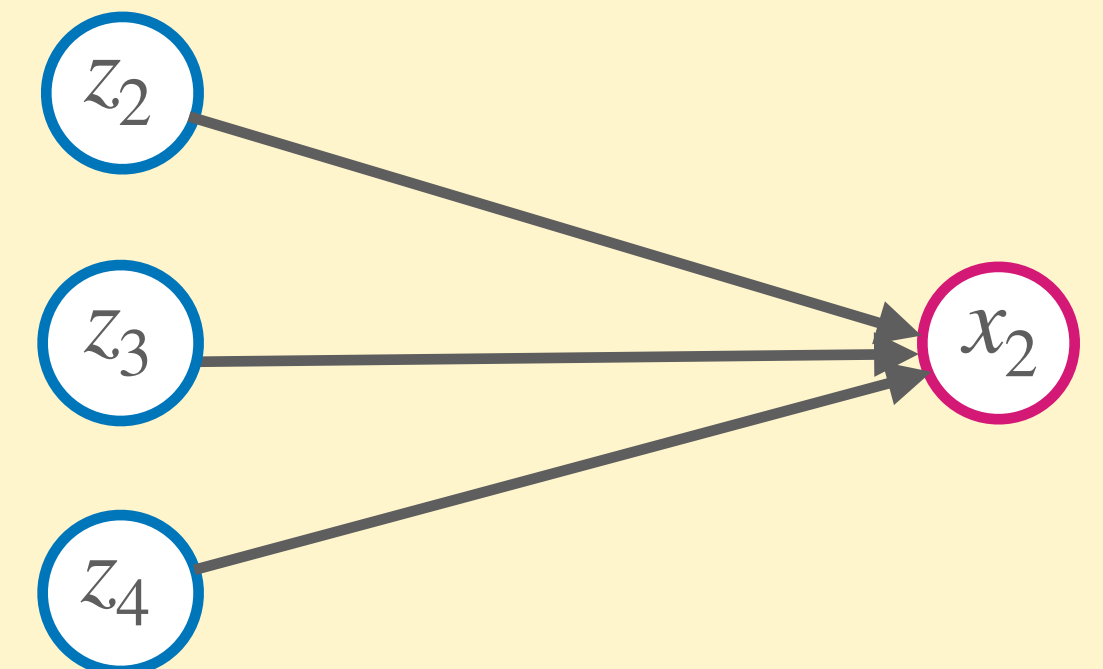
Use **expansion** to avoid this scenario!

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

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

**Invariant:**  $G \setminus \rho$  is expanding

$G \setminus \rho$



$[N] = [n^c]$

$[n]$

# Proof Overview

**Problem:**  $z$ -variables are correlated

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

→ Cannot follow  $A$  in this case

Use **expansion** to avoid this scenario!

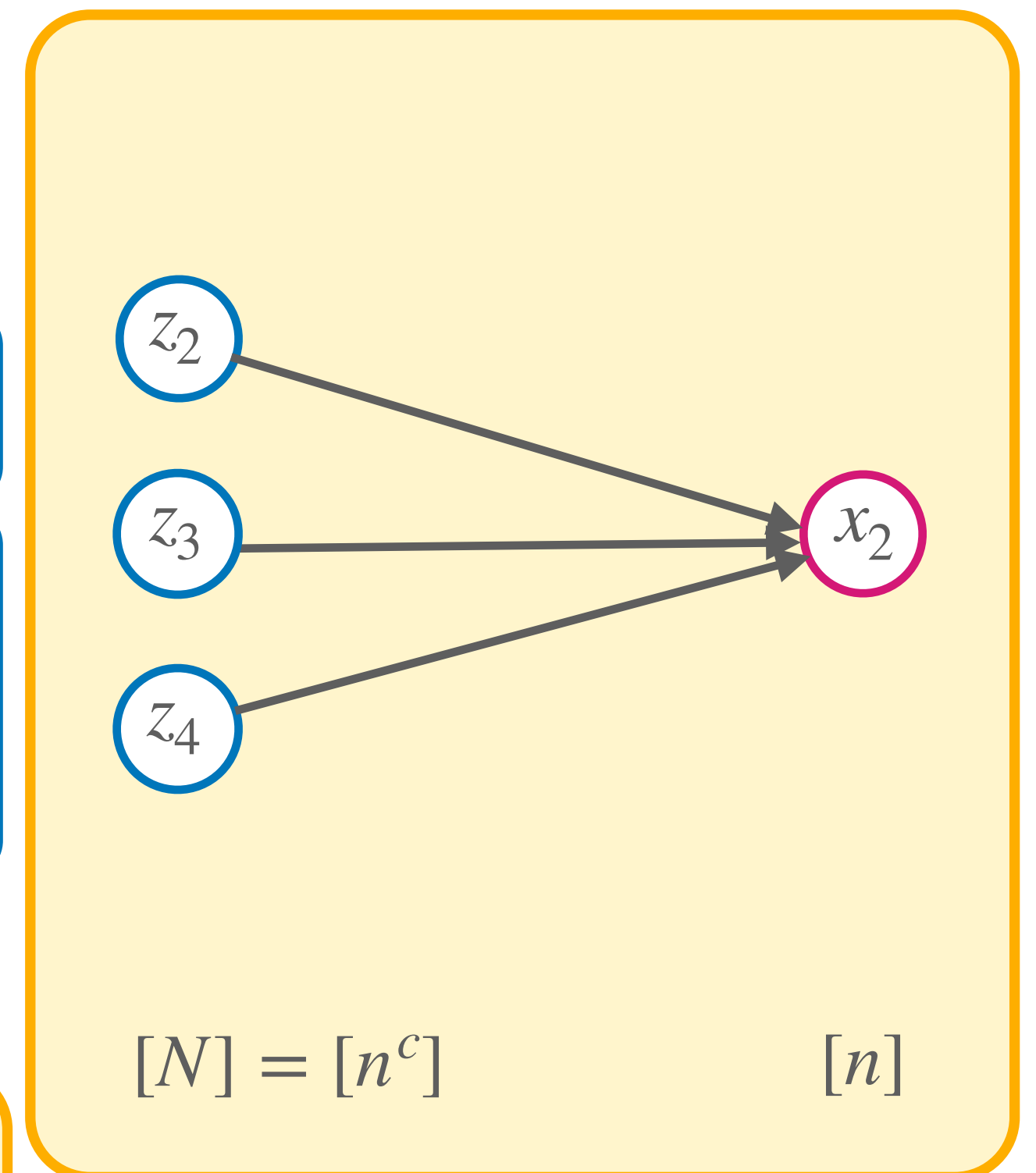
Let  $G \setminus \rho$  be induced by removing the  $x$ -variables set by  $\rho$  and  $z$ -variables determined by  $\rho$

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

**Invariant:**  $G \setminus \rho$  is expanding

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

$G \setminus \rho$



# Expansion Restoration

**However...** after setting an  $x_i$ ,  $G \setminus \rho$  may no longer be expanding

# Expansion Restoration

**However...** after setting an  $x_i$ ,  $G \setminus \rho$  may no longer be expanding  
→ Query **additional**  $x$ -variables to **restore expansion!**

# Expansion Restoration

**However...** after setting an  $x_i$ ,  $G \setminus \rho$  may no longer be 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



# Expansion Restoration

**However...** after setting an  $x_i$ ,  $G \setminus \rho$  may no longer be 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  $G \setminus \rho$  is expanding and  $\rho'$  is obtained by querying some  $x_i$ , then there exists  $\text{Cl}(\rho') \supseteq \text{vars}(\rho')$  such that

# Expansion Restoration

**However...** after setting an  $x_i$ ,  $G \setminus \rho$  may no longer be 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  $G \setminus \rho$  is expanding and  $\rho'$  is obtained by querying some  $x_i$ , then there exists  $Cl(\rho') \supseteq vars(\rho')$  such that

1.  $Cl(\rho')$  fixes at most  $2w$   $z$ -variables

# Expansion Restoration

**However...** after setting an  $x_i$ ,  $G \setminus \rho$  may no longer be 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  $G \setminus \rho$  is expanding and  $\rho'$  is obtained by querying some  $x_i$ , then there exists  $Cl(\rho') \supseteq vars(\rho')$  such that

1.  $Cl(\rho')$  fixes at most  $2w$   $z$ -variables
2.  $G \setminus Cl(\rho')$  is **expanding**

# Expansion Restoration

**However...** after setting an  $x_i$ ,  $G \setminus \rho$  may no longer be 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  $G \setminus \rho$  is expanding and  $\rho'$  is obtained by querying some  $x_i$ , then there exists  $Cl(\rho') \supseteq \text{vars}(\rho')$  such that

1.  $Cl(\rho')$  fixes at most  $2w$   $z$ -variables
2.  $G \setminus Cl(\rho')$  is **expanding**
3. The variables of  $Cl(\rho') \setminus \text{vars}(\rho')$  can be set **consistently** with  $A$

# Expansion Restoration

**However...** after setting an  $x_i$ ,  $G \setminus \rho$  may no longer be 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  $G \setminus \rho$  is expanding and  $\rho'$  is obtained by querying some  $x_i$ , then there exists  $Cl(\rho') \supseteq vars(\rho')$  such that

1.  $Cl(\rho')$  fixes at most  $2w$   $z$ -variables
2.  $G \setminus Cl(\rho')$  is **expanding**
3. The variables of  $Cl(\rho') \setminus vars(\rho')$  can be set **consistently** with  $A$

→ To restore expansion, set the variables of  $Cl(\rho') \setminus vars(\rho')$ !

# 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

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:



# 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:**  $G \setminus \rho$  is expanding



# 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:**  $G \setminus \rho$  is expanding

**Query:** If Prover asks for the value of  $x_i$

→ Set  $x_i$  arbitrarily

# 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:**  $G \setminus \rho$  is expanding

**Query:** If Prover asks for the value of  $x_i$

→ 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:**  $G \setminus \rho$  is expanding

**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  $z_j$

**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:**  $G \setminus \rho$  is expanding

**Query:** If Prover asks for the value of  $x_i$

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

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

→ 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:**  $G \setminus \rho$  is expanding

**Query:** If Prover asks for the value of  $x_i$

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

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

→ 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!

# 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:**  $G \setminus \rho$  is expanding

**Query:** If Prover asks for the value of  $x_i$

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

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

→ 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!

**Upshot:** any width  $w$  proof of  $F \circ \text{XOR}_G$  requires depth  $\Omega(d/w)$

# 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?

→ There are functions which have poly-size **monotone circuits** but require depth  $\Omega(n/\log^{O(1)} n)$  [deRMN+20]



# Open Problems

What about supercritical size/depth tradeoffs for other models of computation?  
→ There are functions which have poly-size **monotone circuits** but require 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?  
→ There are functions which have poly-size **monotone circuits** but require depth  $\Omega(n/\log^{O(1)} n)$  [deRMN+20]

*Q.* Can this be extended to **supercritical**?

**Interpolation:** Our tradeoff actually holds for a proof system which is **equivalent** to monotone circuits

# Open Problems

What about supercritical size/depth tradeoffs for other models of computation?  
→ There are functions which have poly-size **monotone circuits** but require depth  $\Omega(n/\log^{O(1)} n)$  [deRMN+20]

*Q.* Can this be extended to **supercritical**?

**Interpolation:** Our tradeoff actually holds for a proof system which is **equivalent** to monotone circuits — Any proof of  $F$  is equivalent to a monotone circuit with the **same topology** computing an associated function  $f_F$

# Open Problems

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

→ There are functions which have poly-size **monotone circuits** but require depth  $\Omega(n/\log^{O(1)} n)$  [deRMN+20]

*Q.* Can this be extended to **supercritical**?

**Interpolation:** Our tradeoff actually holds for a proof system which is **equivalent** to monotone circuits — Any proof of  $F$  is equivalent to a monotone circuit with the **same topology** computing an associated function  $f_F$

→ 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?  
→ There are functions which have poly-size **monotone circuits** but require depth  $\Omega(n/\log^{O(1)} n)$  [deRMN+20]

*Q.* Can this be extended to **supercritical**?

**Interpolation:** Our tradeoff actually holds for a proof system which is **equivalent** to monotone circuits — Any proof of  $F$  is equivalent to a monotone circuit with the **same topology** computing an associated function  $f_F$

→ However, the number of variables of  $f_F$  is **equal** to the number of clauses of  $F$

⇒ Our tradeoffs **do not** imply supercritical tradeoffs for monotone circuits

# Open Problems

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

→ There are functions which have poly-size **monotone circuits** but require depth  $\Omega(n/\log^{O(1)} n)$  [deRMN+20]

*Q.* Can this be extended to **supercritical**?

**Interpolation:** Our tradeoff actually holds for a proof system which is **equivalent** to monotone circuits — Any proof of  $F$  is equivalent to a monotone circuit with the **same topology** computing an associated function  $f_F$

→ However, the number of variables of  $f_F$  is **equal** to the number of clauses of  $F$

⇒ Our tradeoffs **do not** imply supercritical tradeoffs for monotone circuits

**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 which have poly-size **monotone circuits** but require depth  $\Omega(n/\log^{O(1)} n)$  [deRMN+20]

*Q.* Can this be extended to **supercritical**?

**Interpolation:** Our tradeoff actually holds for a proof system which is **equivalent** to monotone circuits — Any proof of  $F$  is equivalent to a monotone circuit with the **same topology** computing an associated function  $f_F$

→ However, the number of variables of  $f_F$  is **equal** to the number of clauses of  $F$

⇒ Our tradeoffs **do not** imply supercritical tradeoffs for monotone circuits

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

# 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  $\implies$  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  $\implies$  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  $\implies$  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  $\implies$  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?