Extremely Deep Proofs

Noah Fleming
University of California, San Diego
Joint work with Toniann Pitassi and Robert Robere

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

  → Our work based on [Razborov16]

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

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

Resolution

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

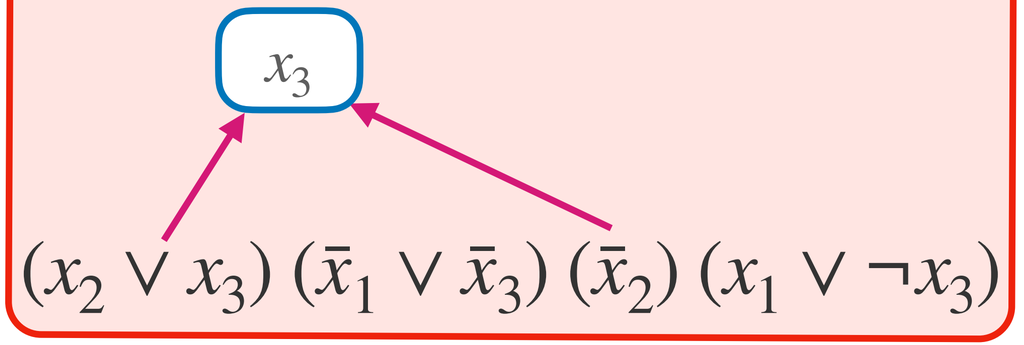
Given an unsatisfiable CNF formula F as a set of clauses

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



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:

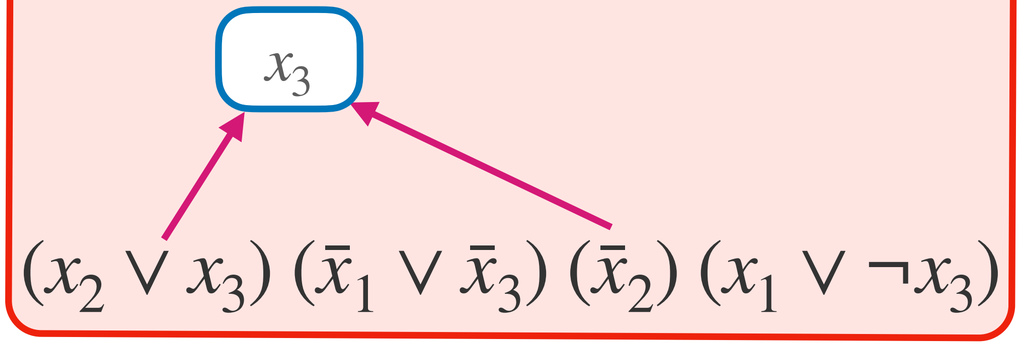
Resolution rule:

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

Goal: Derive empty clause Λ

Resolution is sound $\Longrightarrow$ F is unsatisfiable

Π



Resolution

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

Given an unsatisfiable CNF formula F as a set of clauses

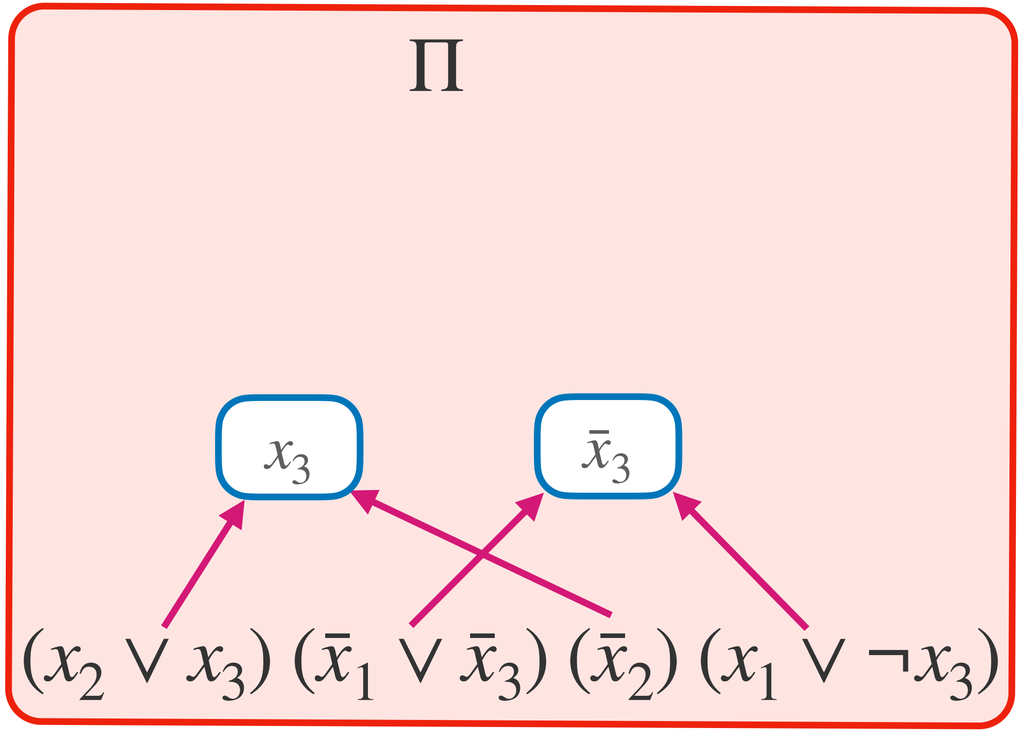
Derive new clauses from old ones using:

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



Resolution

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

Given an unsatisfiable CNF formula F as a set of clauses

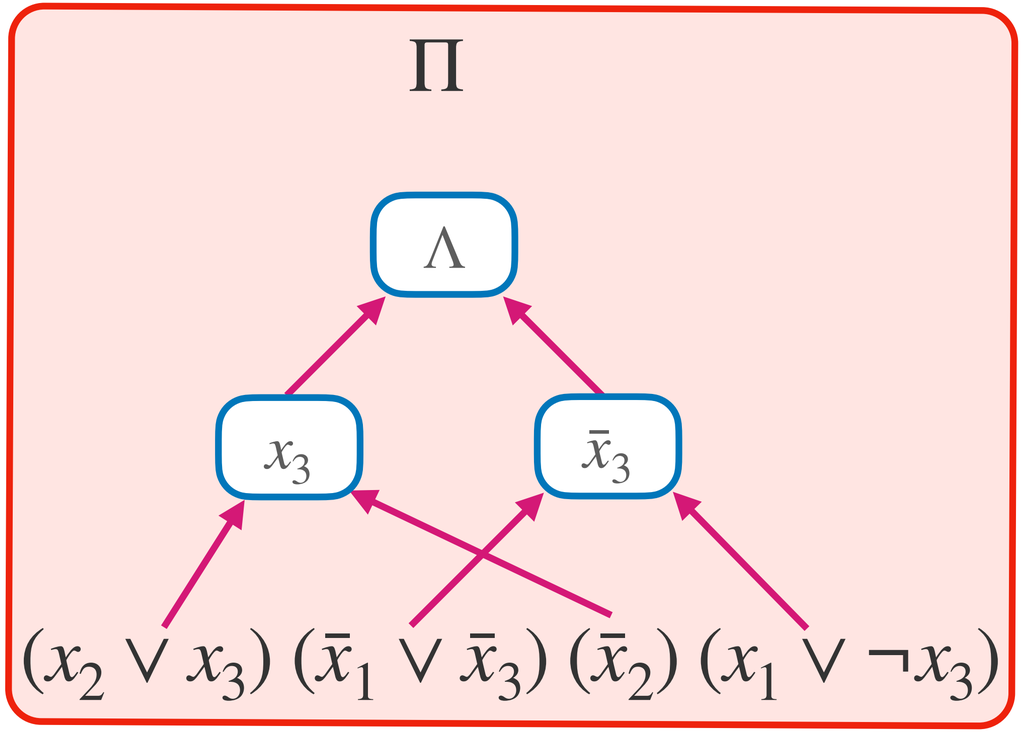
Derive new clauses from old ones using:

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



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:

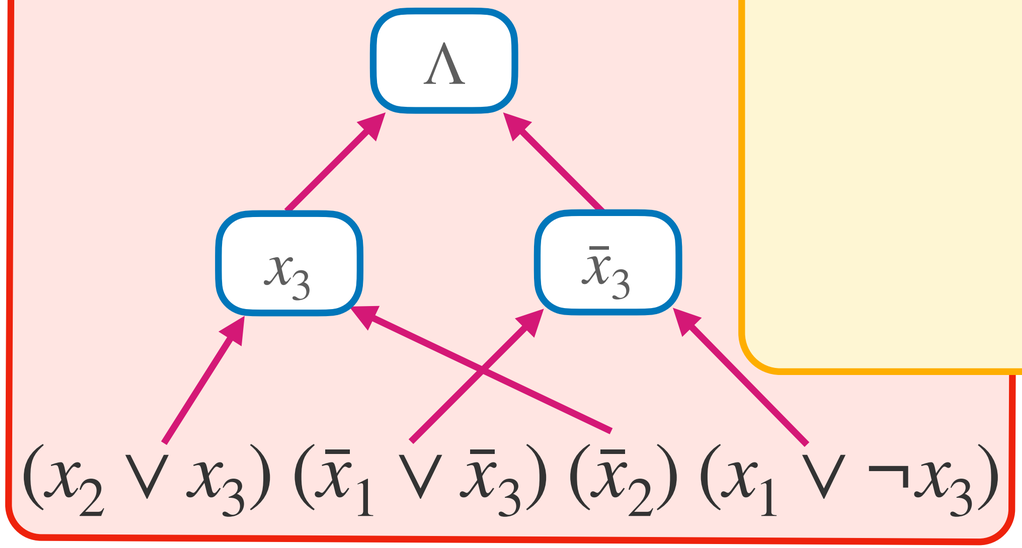
Resolution rule:

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

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

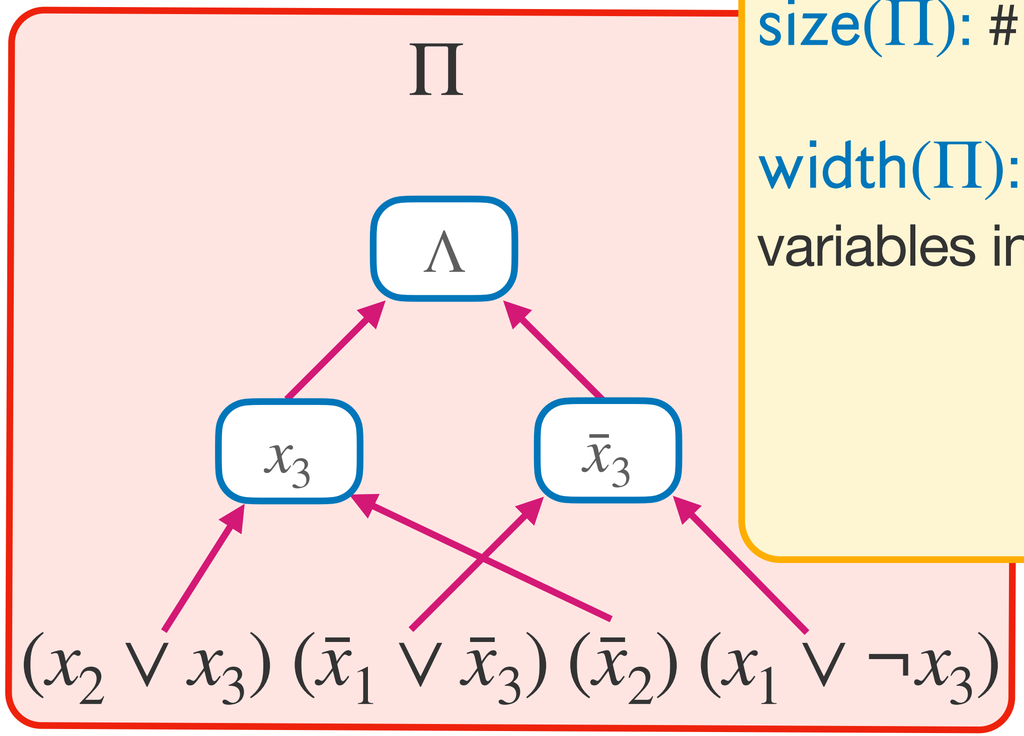
Derive new clauses from old ones using:

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

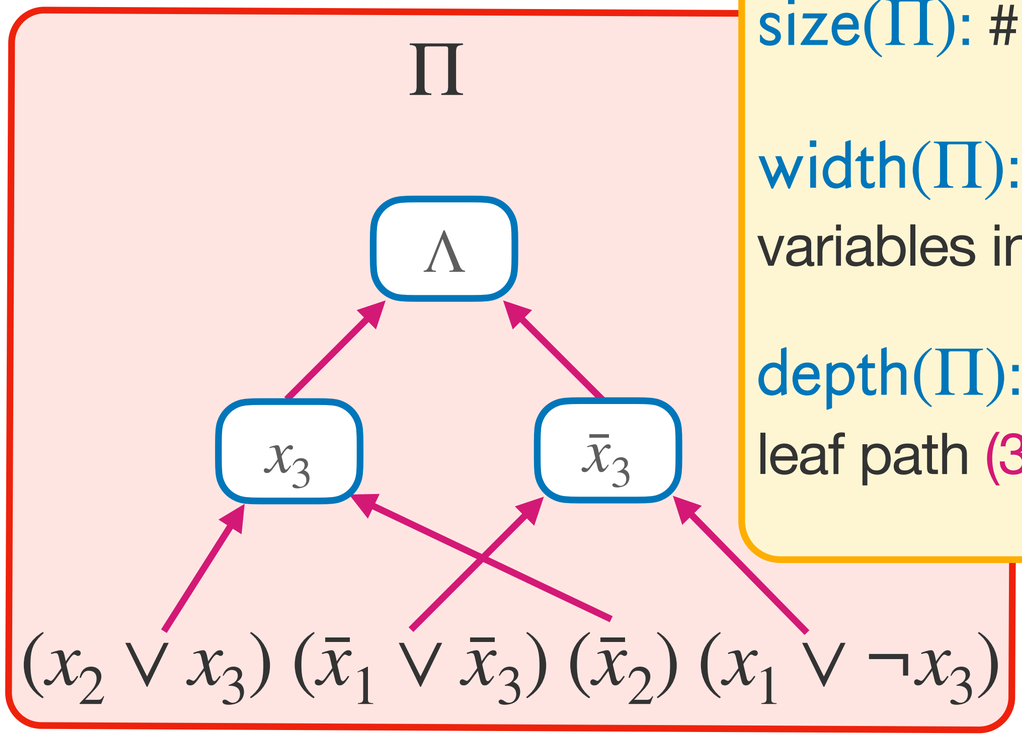
Derive new clauses from old ones using:

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 $\Longrightarrow$ 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

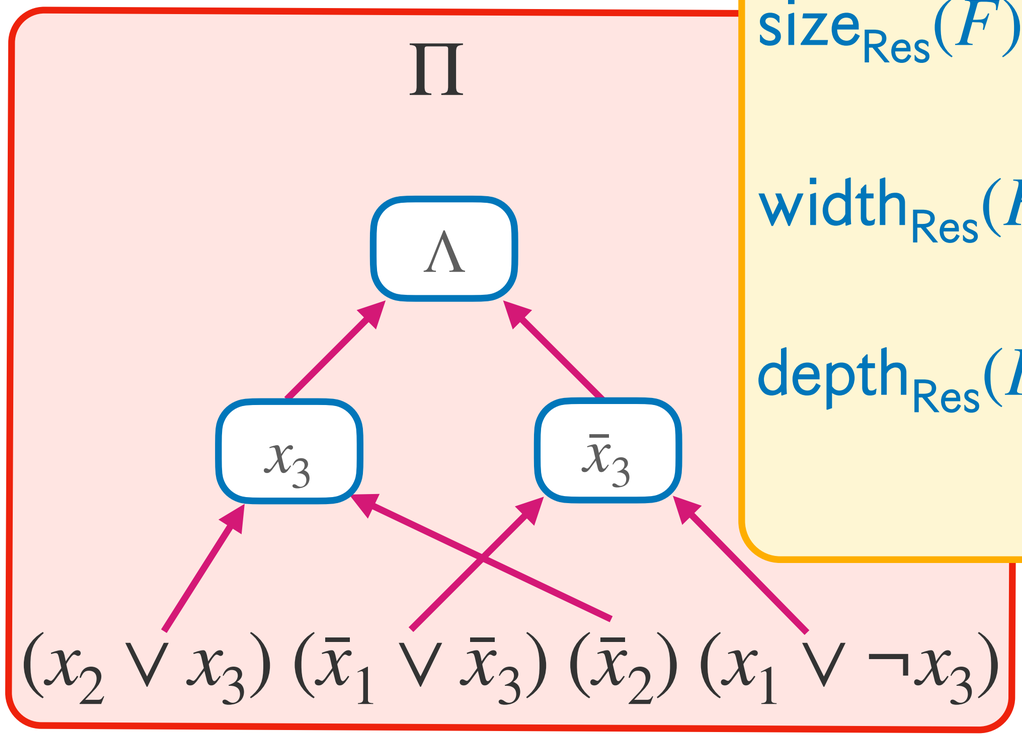
Derive new clauses from old ones using:

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

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

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

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

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

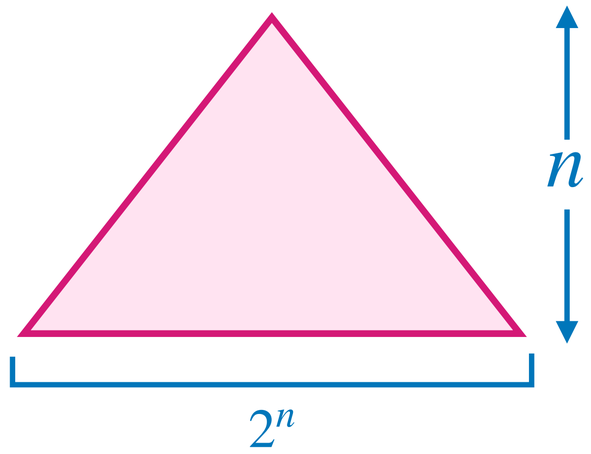
Like circuit depth, proof depth captures a notion of “parallelism” of a proof

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

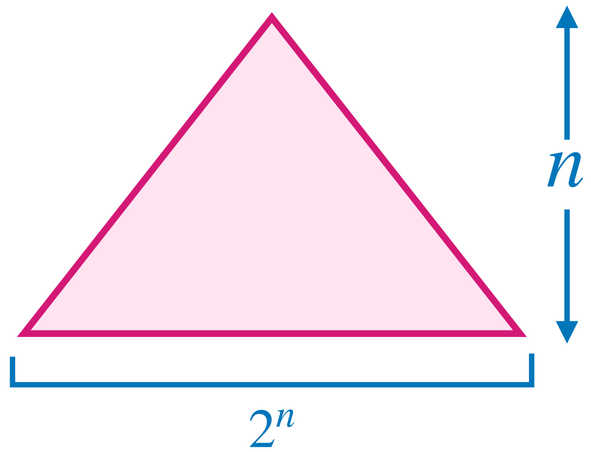
Like circuit depth, proof depth captures a notion of "parallelism" of a proof

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

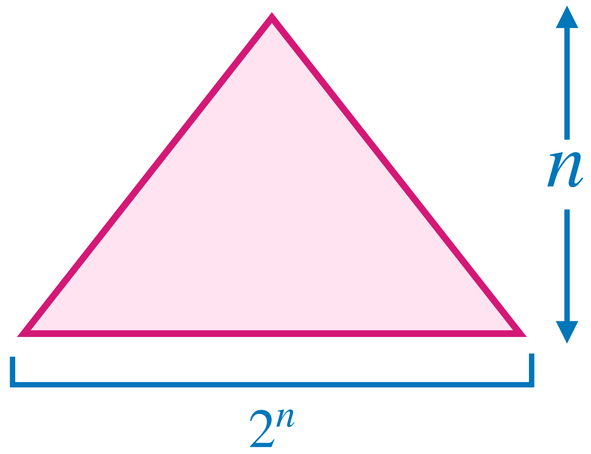
Like circuit depth, proof depth captures a notion of "parallelism" of a proof

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

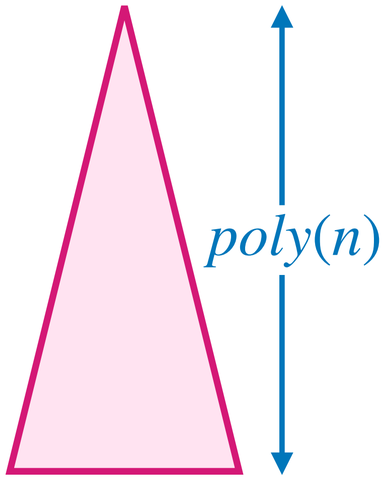
→ Resolution (Res(k), Cutting Planes) cannot always be balanced

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 polynomial depth

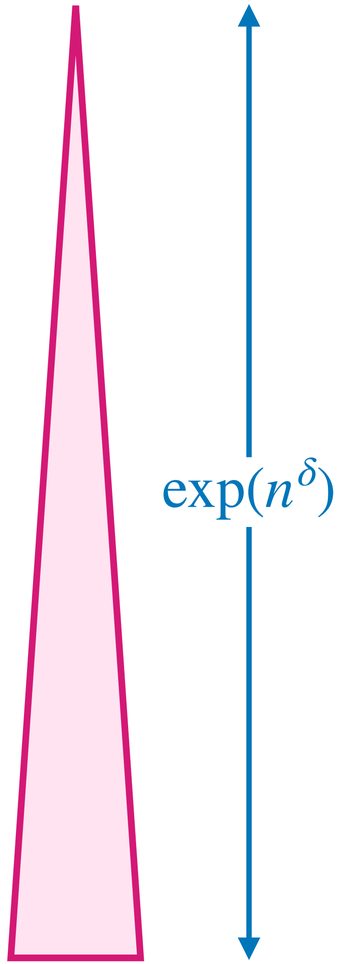


This Work

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 $\text{size}(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ then

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

This Work

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 $\text{size}(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ then

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

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

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 this 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 this 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 $F(z_1, \ldots, z_N) = C_1 \wedge \ldots \wedge C_m$ 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 $F \circ g := F(g(x_1), \ldots, g(x_N))$

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 $F \circ g := F(g(x_1), \ldots, g(x_N))$

Typically $x_1, \ldots, x_N$ are disjoint sets

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 $F \circ g := F(g(x_1), \ldots, g(x_N))$

Typically $x_1, \ldots, x_N$ are disjoint sets

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: $g = \mathrm{XOR}_2$ then $F \circ \mathrm{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Lifting (Composition)

Simple Example: $g = \mathrm{XOR}_2$ then $F \circ \mathrm{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Width-to-Size Lifting Theorem: Let F be any unsatisfiable formula. Then

$$\mathrm{size}_{\mathrm{Res}}(F \circ \mathrm{XOR}_2) \geq 2^{\Omega(\mathrm{width}_{\mathrm{Res}}(F))}$$

Lifting (Composition)

Simple Example: $g = \mathrm{XOR}_2$ then $F \circ \mathrm{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Width-to-Size Lifting Theorem: Let F be any unsatisfiable formula. Then

$$\mathrm{size}_{\mathrm{Res}}(F \circ \mathrm{XOR}_2) \geq 2^{\Omega(\mathrm{width}_{\mathrm{Res}}(F))}$$

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

Lifting (Composition)

Simple Example: $g = \mathrm{XOR}_2$ then $F \circ \mathrm{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Width-to-Size Lifting Theorem: Let F be any unsatisfiable formula. Then

$$\mathrm{size}_{\mathrm{Res}}(F \circ \mathrm{XOR}_2) \geq 2^{\Omega(\mathrm{width}_{\mathrm{Res}}(F))}$$

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

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

Lifting (Composition)

Simple Example: $g = \mathrm{XOR}_2$ then $F \circ \mathrm{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Width-to-Size Lifting Theorem: Let F be any unsatisfiable formula. Then

$$\mathrm{size}_{\mathrm{Res}}(F \circ \mathrm{XOR}_2) \geq 2^{\Omega(\mathrm{width}_{\mathrm{Res}}(F))}$$

- $P =$ 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: $g = \mathsf{XOR}_2$ then $F \circ \mathsf{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Width-to-Size Lifting Theorem: Let F be any unsatisfiable formula. Then

$$\mathsf{size}_{\mathsf{Res}}(F \circ \mathsf{XOR}_2) \geq 2^{\Omega(\mathsf{width}_{\mathsf{Res}}(F))}$$

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

If F has a proof of size s and width $w \implies F \circ \mathsf{XOR}_2$ 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!

Lifting (Composition)

Simple Example: $g = \mathrm{XOR}_2$ then $F \circ \mathrm{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Width-to-Size Lifting Theorem: Let F be any unsatisfiable formula. Then

$$\mathrm{size}_{\mathrm{Res}}(F \circ \mathrm{XOR}_2) \geq 2^{\Omega(\mathrm{width}_{\mathrm{Res}}(F))}$$

- $P =$ 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

$\Longrightarrow$ Naively simulation is essentially the best!

$\rightarrow$ Theme of lifting theorems

Lifting (Composition)

Width-to-Size Lifting Theorem: $\mathsf{size}_{\mathsf{Res}}(F \circ \mathsf{XOR}_2) \geq 2^{\Omega(\mathsf{width}_{\mathsf{Res}}(F))}$

Proof: Let Π be a proof of $F \circ \mathsf{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Lifting (Composition)

Width-to-Size Lifting Theorem: $\text{size}_{\text{Res}}(F \circ \text{XOR}_2) \geq 2^{\Omega(\text{width}_{\text{Res}}(F))}$

Proof: Let Π be a proof of $F \circ \text{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Let $\rho \in \{0,1,*\}^{2N}$ be generated as follows

Lifting (Composition)

Width-to-Size Lifting Theorem: $\mathsf{size}_{\mathsf{Res}}(F \circ \mathsf{XOR}_2) \geq 2^{\Omega(\mathsf{width}_{\mathsf{Res}}(F))}$

Proof: Let Π be a proof of $F \circ \mathsf{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Let $\rho \in \{0,1,*\}^{2N}$ be generated as follows — Flip a coin for each $i \in [N]$:

- Heads: set x_i to a random bit, set $y_i = *$
- Tails: set y_i to a random bit, set $x_i = *$

Lifting (Composition)

Width-to-Size Lifting Theorem: $\mathsf{size}_{\mathsf{Res}}(F \circ \mathsf{XOR}_2) \geq 2^{\Omega(\mathsf{width}_{\mathsf{Res}}(F))}$

Proof: Let Π be a proof of $F \circ \mathsf{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Let $\rho \in \{0,1,*\}^{2N}$ be generated as follows — Flip a coin for each $i \in [N]$:

- Heads: set x_i to a random bit, set $y_i = *$
- Tails: set y_i to a random bit, set $x_i = *$

Observe: $F \circ \mathsf{XOR}_2 \restriction \rho = F$ (some variables negated)

Lifting (Composition)

Width-to-Size Lifting Theorem: $\mathrm{size}_{\mathrm{Res}}(F \circ \mathrm{XOR}_2) \geq 2^{\Omega(\mathrm{width}_{\mathrm{Res}}(F))}$

Proof: Let Π be a proof of $F \circ \mathrm{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Let $\rho \in \{0,1,*\}^{2N}$ be generated as follows — Flip a coin for each $i \in [N]$:

- Heads: set x_i to a random bit, set $y_i = *$
- Tails: set y_i to a random bit, set $x_i = *$

Observe: $F \circ \mathrm{XOR}_2 \restriction \rho = F$ (some variables negated) $\implies \Pi \restriction \rho$ is a proof of F

Lifting (Composition)

Width-to-Size Lifting Theorem: $\mathrm{size}_{\mathrm{Res}}(F \circ \mathrm{XOR}_2) \geq 2^{\Omega(\mathrm{width}_{\mathrm{Res}}(F))}$

Proof: Let Π be a proof of $F \circ \mathrm{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Let $\rho \in \{0,1,*\}^{2N}$ be generated as follows — Flip a coin for each $i \in [N]$:

- Heads: set x_i to a random bit, set $y_i = *$
- Tails: set y_i to a random bit, set $x_i = *$

Observe: $F \circ \mathrm{XOR}_2 \restriction \rho = F$ (some variables negated) $\implies \Pi \restriction \rho$ is a proof of F

If C is a clause of width $\geq w := \mathrm{width}_{\mathrm{Res}}(F)$ then $\Pr[C \restriction \rho \neq 1] \leq (3/4)^w$

Lifting (Composition)

Width-to-Size Lifting Theorem: $\text{size}_{\text{Res}}(F \circ \text{XOR}_2) \geq 2^{\Omega(\text{width}_{\text{Res}}(F))}$

Proof: Let Π be a proof of $F \circ \text{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Let $\rho \in \{0,1,*\}^{2N}$ be generated as follows — Flip a coin for each $i \in [N]$:

- Heads: set x_i to a random bit, set $y_i = *$
- Tails: set y_i to a random bit, set $x_i = *$

Observe: $F \circ \text{XOR}_2 \restriction \rho = F$ (some variables negated) $\implies \Pi \restriction \rho$ is a proof of F

If C is a clause of width $\geq w := \text{width}_{\text{Res}}(F)$ then $\Pr[C \restriction \rho \neq 1] \leq (3/4)^w$

Union bound $\implies \Pr[\exists C \in \Pi : \text{width}(C) \geq w, C \restriction \rho \neq 1] \leq |\Pi|(3/4)^w$

Lifting (Composition)

Width-to-Size Lifting Theorem: $\mathrm{size}_{\mathrm{Res}}(F \circ \mathrm{XOR}_2) \geq 2^{\Omega(\mathrm{width}_{\mathrm{Res}}(F))}$

Proof: Let Π be a proof of $F \circ \mathrm{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Let $\rho \in \{0,1,*\}^{2N}$ be generated as follows — Flip a coin for each $i \in [N]$:

- Heads: set x_i to a random bit, set $y_i = *$
- Tails: set y_i to a random bit, set $x_i = *$

Observe: $F \circ \mathrm{XOR}_2 \restriction \rho = F$ (some variables negated) $\Longrightarrow \Pi \restriction \rho$ is a proof of F

If C is a clause of width $\geq w := \mathrm{width}_{\mathrm{Res}}(F)$ then $\Pr[C \restriction \rho \neq 1] \leq (3/4)^w$

Union bound $\Longrightarrow \Pr[\exists C \in \Pi : \mathrm{width}(C) \geq w, C \restriction \rho \neq 1] \leq |\Pi|(3/4)^w$

If $|\Pi| < (4/3)^w$ then

Lifting (Composition)

Width-to-Size Lifting Theorem: $\mathrm{size}_{\mathrm{Res}}(F \circ \mathrm{XOR}_2) \geq 2^{\Omega(\mathrm{width}_{\mathrm{Res}}(F))}$

Proof: Let Π be a proof of $F \circ \mathrm{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Let $\rho \in \{0,1,*\}^{2N}$ be generated as follows — Flip a coin for each $i \in [N]$:

- Heads: set x_i to a random bit, set $y_i = *$
- Tails: set y_i to a random bit, set $x_i = *$

Observe: $F \circ \mathrm{XOR}_2 \restriction \rho = F$ (some variables negated) $\implies \Pi \restriction \rho$ is a proof of F

If C is a clause of width $\geq w := \mathrm{width}_{\mathrm{Res}}(F)$ then $\Pr[C \restriction \rho \neq 1] \leq (3/4)^w$

Union bound $\implies \Pr[\exists C \in \Pi : \mathrm{width}(C) \geq w, C \restriction \rho \neq 1] \leq |\Pi|(3/4)^w < 1$

If $|\Pi| < (4/3)^w$ then

Lifting (Composition)

Width-to-Size Lifting Theorem: $\text{size}_{\text{Res}}(F \circ \text{XOR}_2) \geq 2^{\Omega(\text{width}_{\text{Res}}(F))}$

Proof: Let Π be a proof of $F \circ \text{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Let $\rho \in \{0,1,*\}^{2N}$ be generated as follows — Flip a coin for each $i \in [N]$:

- Heads: set x_i to a random bit, set $y_i = *$
- Tails: set y_i to a random bit, set $x_i = *$

Observe: $F \circ \text{XOR}_2 \restriction \rho = F$ (some variables negated) $\implies \Pi \restriction \rho$ is a proof of F

If C is a clause of width $\geq w := \text{width}_{\text{Res}}(F)$ then $\Pr[C \restriction \rho \neq 1] \leq (3/4)^w$

Union bound $\implies \Pr[\exists C \in \Pi : \text{width}(C) \geq w, C \restriction \rho \neq 1] \leq |\Pi|(3/4)^w < 1$

If $|\Pi| < (4/3)^w$ then $\exists \rho$ such that $\text{width}(\Pi \restriction \rho) < \text{width}_{\text{Res}}(F)$

Lifting (Composition)

Width-to-Size Lifting Theorem: $\text{size}_{\text{Res}}(F \circ \text{XOR}_2) \geq 2^{\Omega(\text{width}_{\text{Res}}(F))}$

Proof: Let Π be a proof of $F \circ \text{XOR}_2 := F(x_1 \oplus y_1, \ldots, x_N \oplus y_N)$

Let $\rho \in \{0,1,*\}^{2N}$ be generated as follows — Flip a coin for each $i \in [N]$:

- Heads: set x_i to a random bit, set $y_i = *$
- Tails: set y_i to a random bit, set $x_i = *$

Observe: $F \circ \text{XOR}_2 \restriction \rho = F$ (some variables negated) $\implies \Pi \restriction \rho$ is a proof of F

If C is a clause of width $\geq w := \text{width}_{\text{Res}}(F)$ then $\Pr[C \restriction \rho \neq 1] \leq (3/4)^w$

Union bound $\implies \Pr[\exists C \in \Pi : \text{width}(C) \geq w, C \restriction \rho \neq 1] \leq |\Pi|(3/4)^w < 1$

If $|\Pi| < (4/3)^w$ then $\exists \rho$ such that $\text{width}(\Pi \restriction \rho) < \text{width}_{\text{Res}}(F)$

Contradiction!

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

$F(\mathrm{XOR}(x_1), \ldots, \mathrm{XOR}(x_N))$

The Gadget

Our gadget will be the XOR function

$F(\text{XOR}(x_1), \ldots, \text{XOR}(x_N))$ … With a **twist**!

The variable sets $x_1, \ldots, x_N$ will no longer be disjoint!

The Gadget

Our gadget will be the XOR function

$F(\text{XOR}(x_1), \ldots, \text{XOR}(x_N))$ … With a **twist**!

The variable sets $x_1, \ldots, x_N$ will no longer be disjoint!

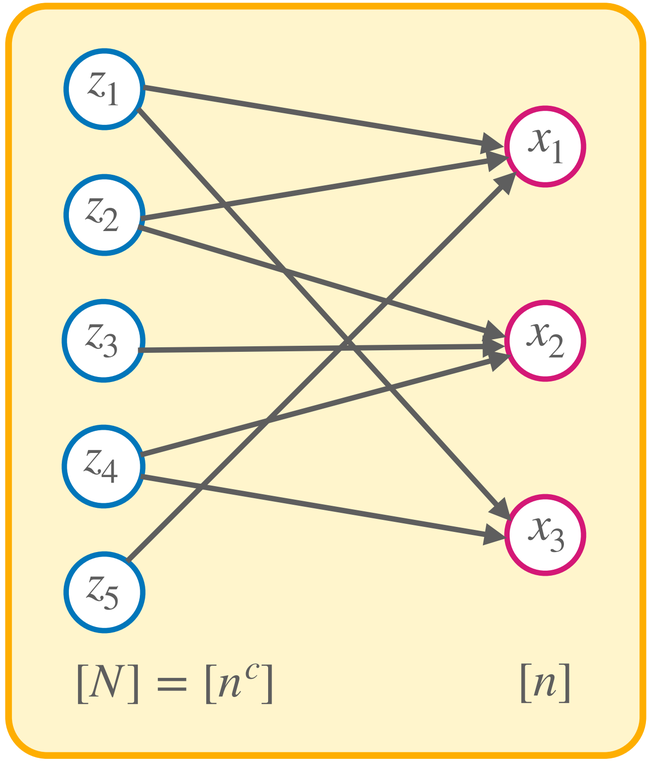
→ Composing will reduce the total number of variables to $n \ll N$

The Gadget

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

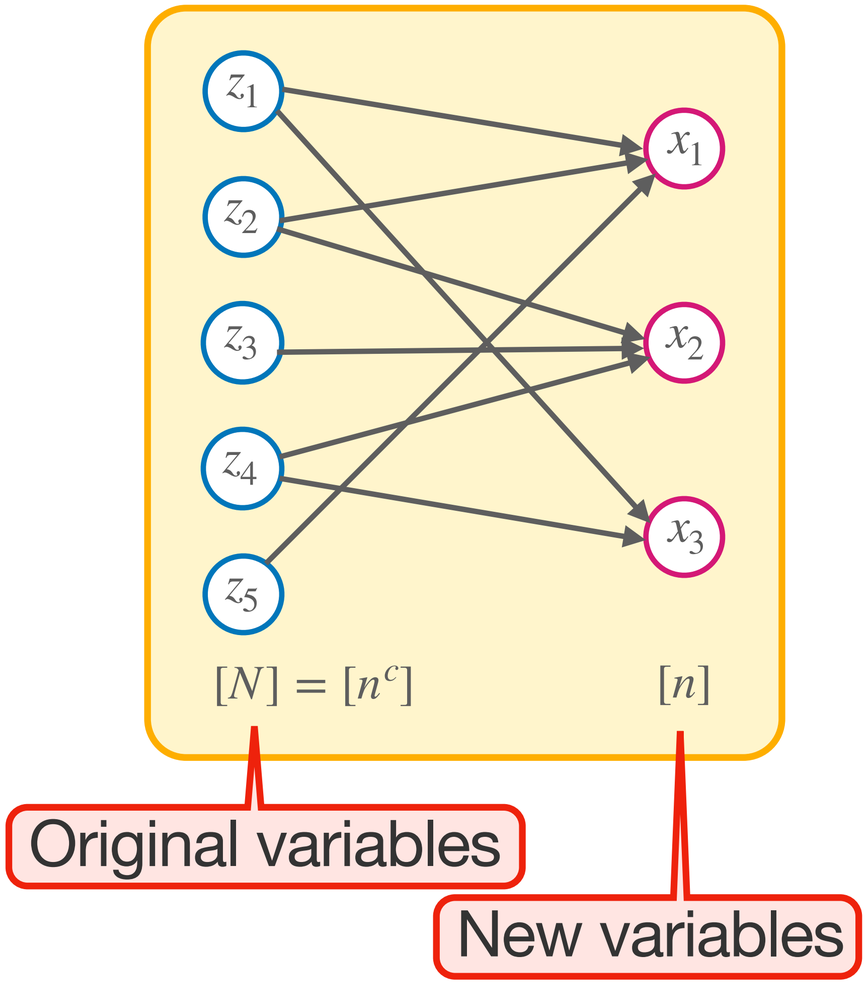
The Gadget

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



The Gadget

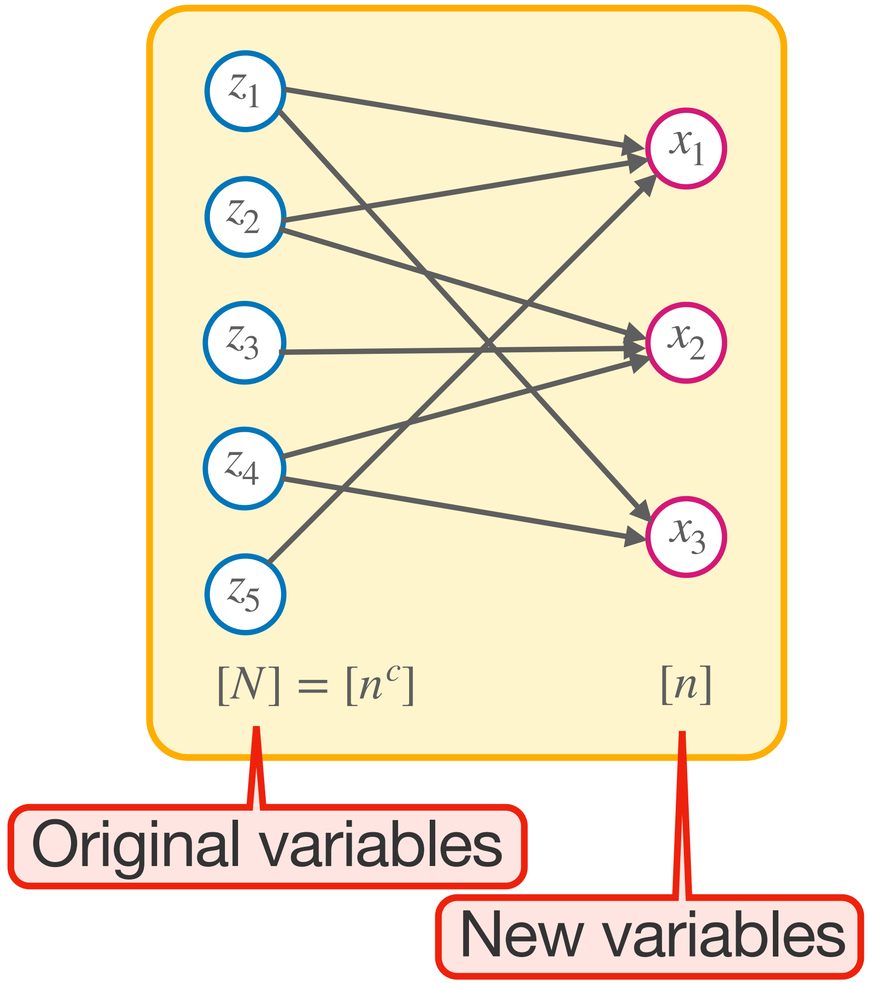
Let G be an $N \times n$ bipartite graph



The Gadget

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

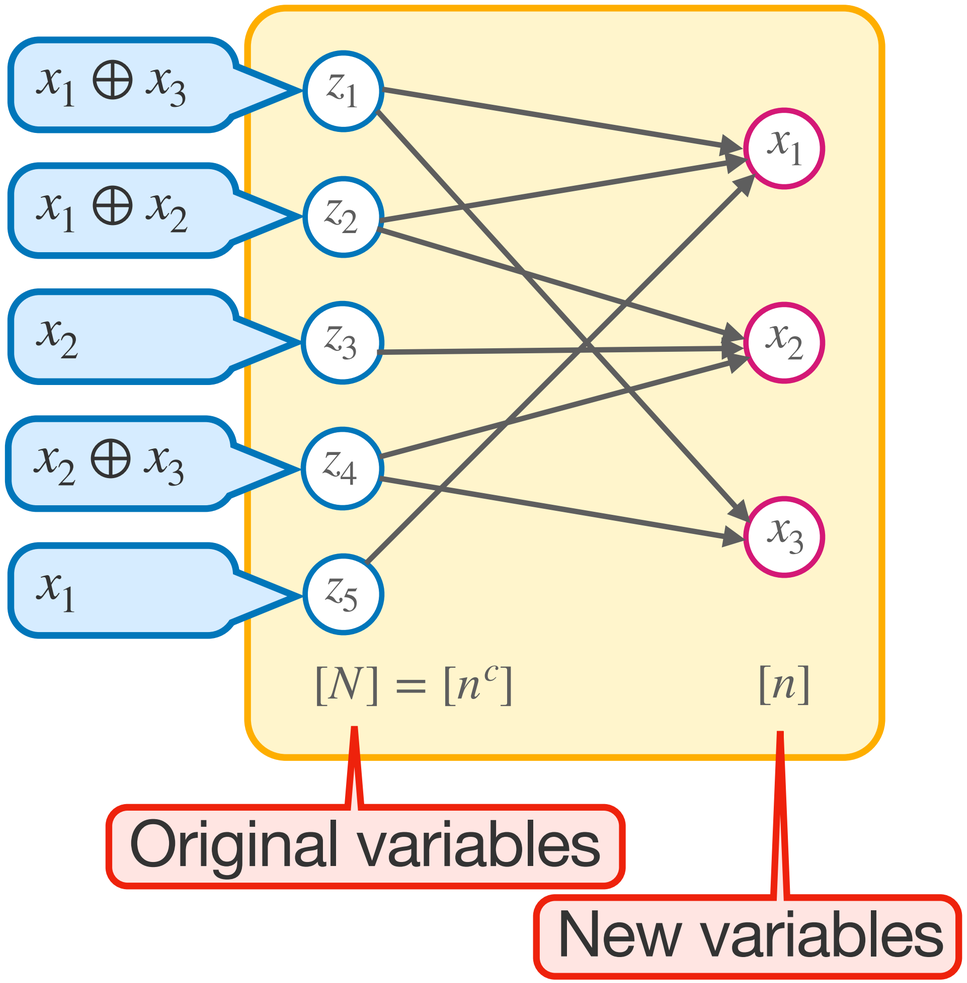
$F \circ \mathrm{XOR}_G$ replaces $z_i \mapsto \bigoplus_{x_j \in \mathrm{N}(z_i)} x_j$



The Gadget

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

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

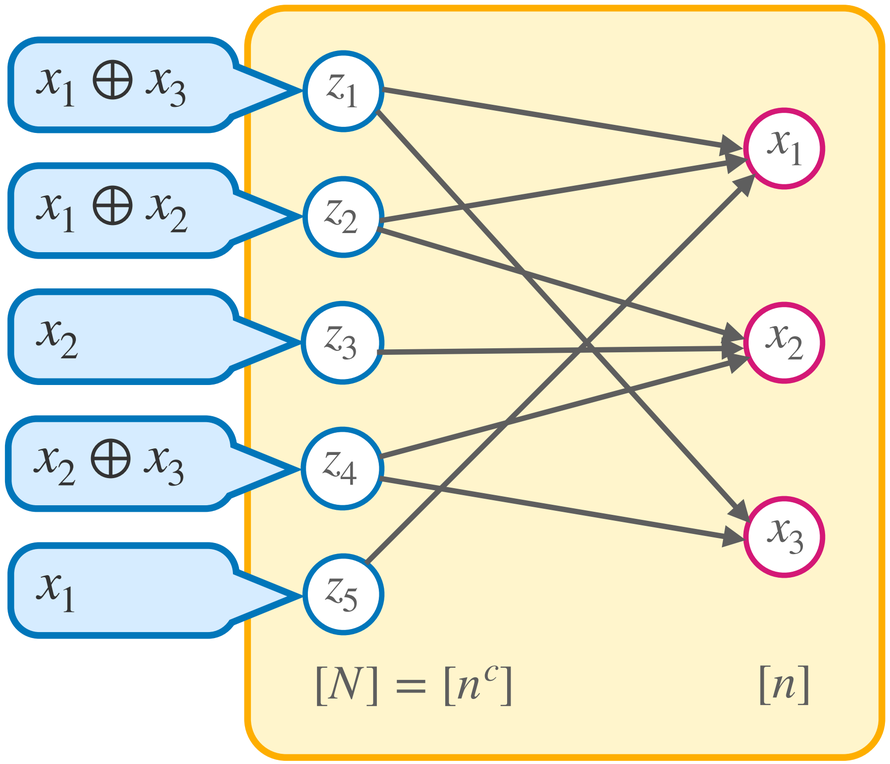


The Gadget

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

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

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



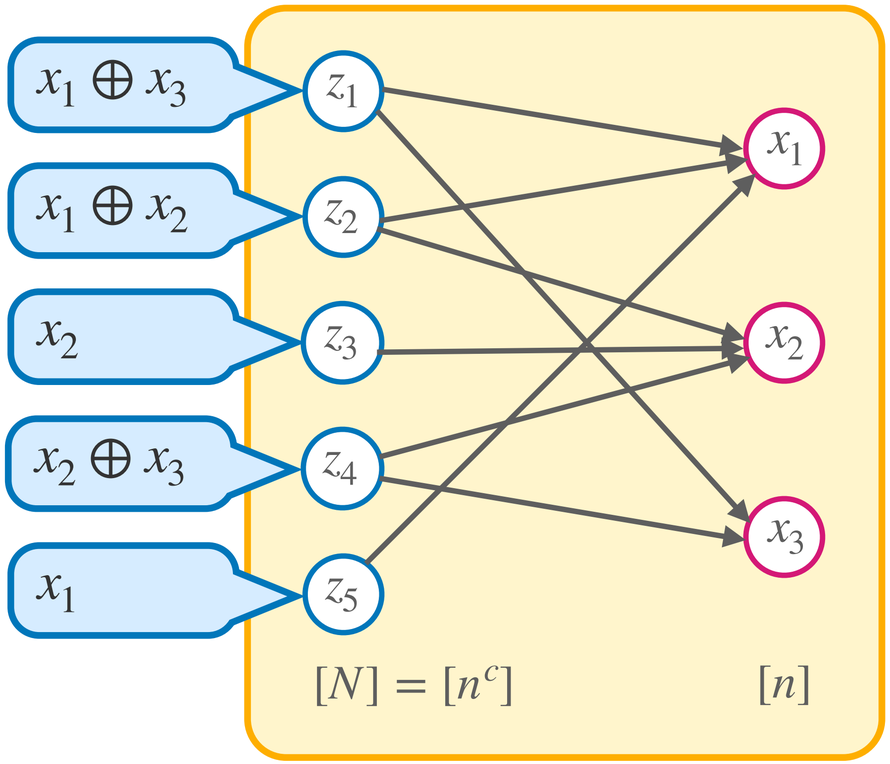
The Gadget

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

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

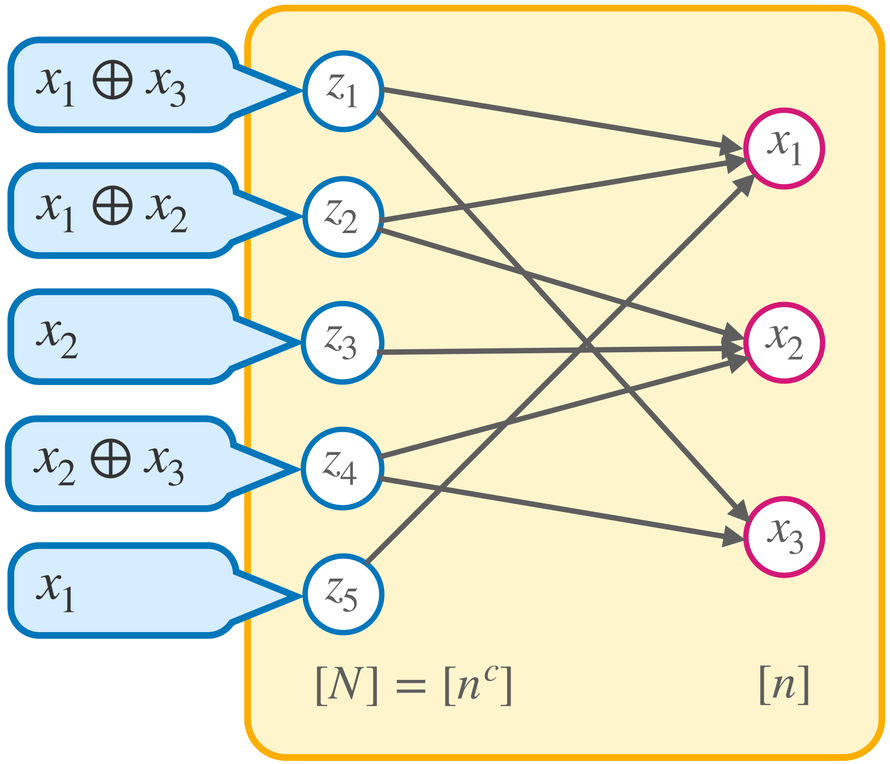


The Gadget

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

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

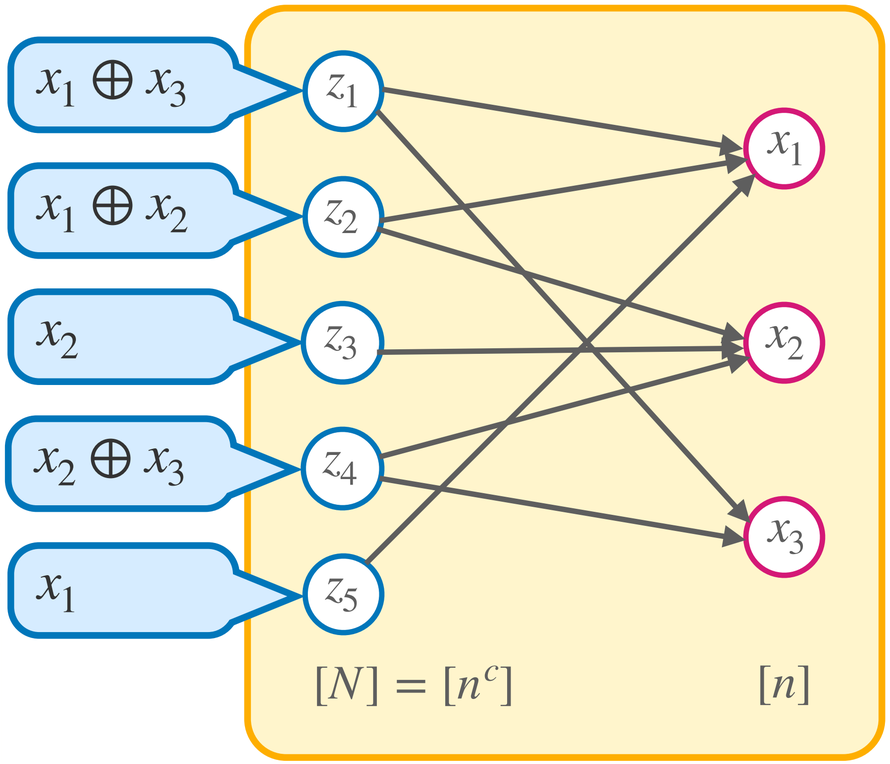
Idea: If the edges of G are sufficiently "spread out"



The Gadget

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

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



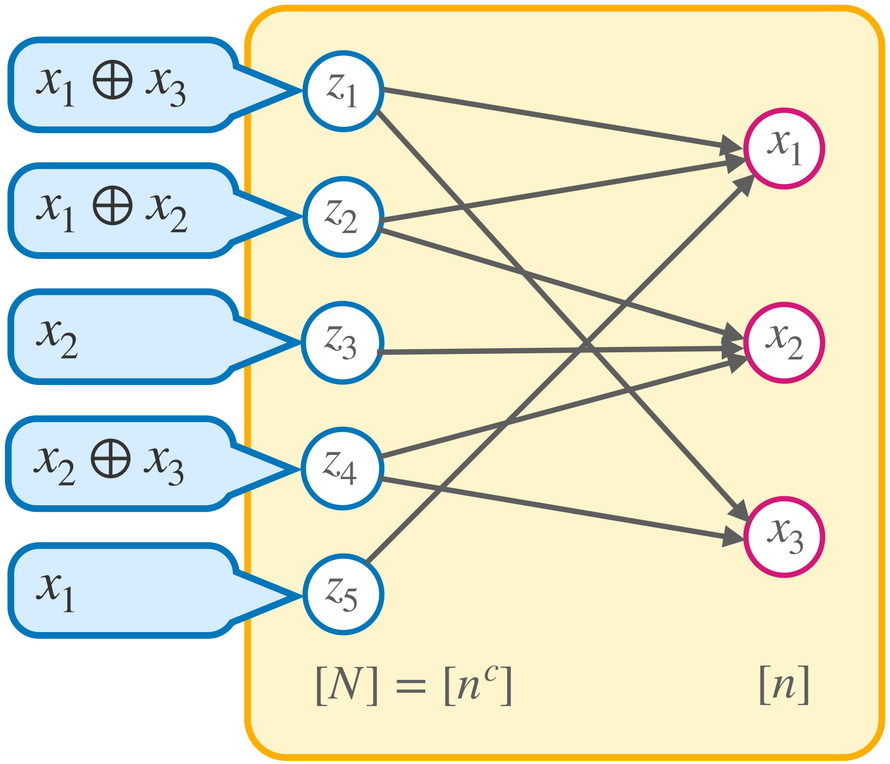
Idea: If the edges of G are sufficiently "spread out"

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

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 the edges of G are sufficiently "spread out"

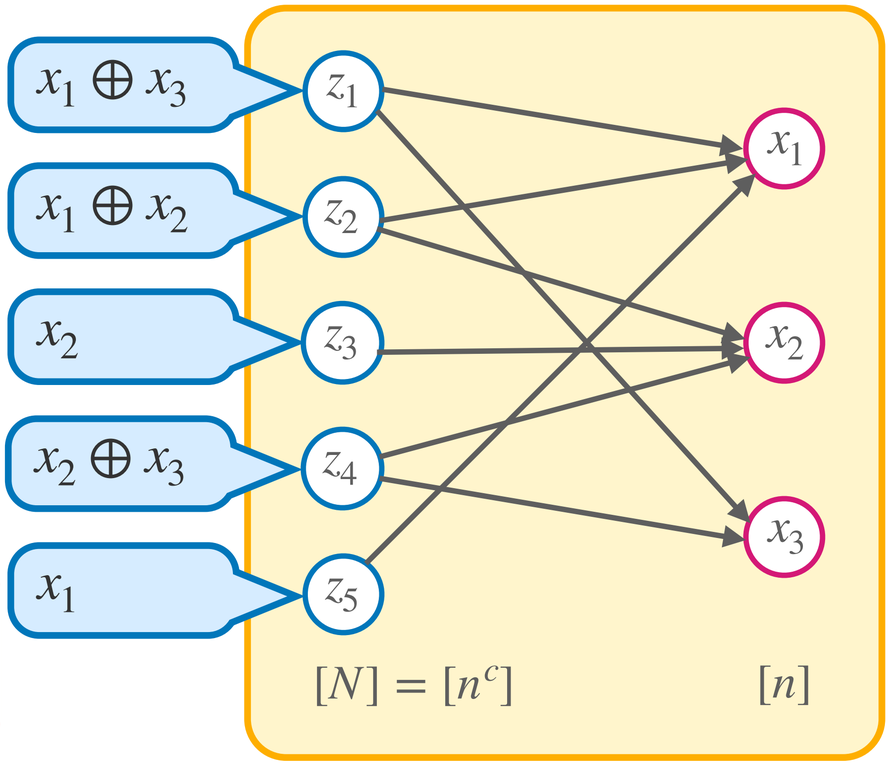
$\rightarrow$ learning the value of one XOR won't reveal much information about any other XOR

$\rightarrow$ 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 $z_i \mapsto \bigoplus_{x_j \in \mathrm{N}(z_i)} x_j$



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

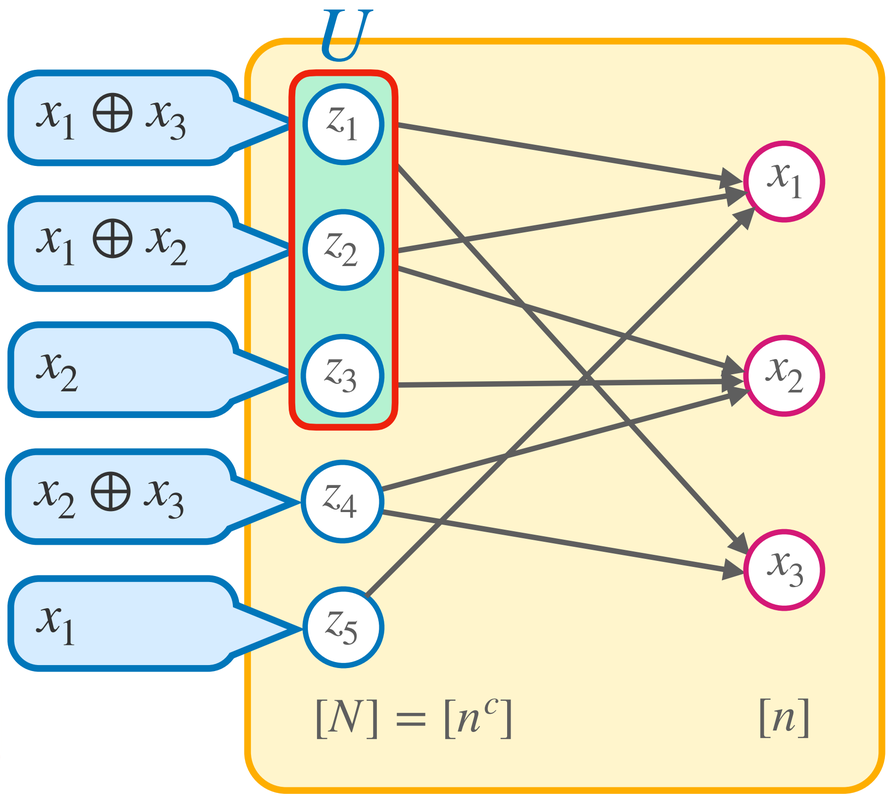
Boundary: $\delta(U)$ of $U \subseteq [N]$ is the number of "unique neighbours" of U

$\rightarrow$ Number of variables that occur in exactly one XOR in 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 the edges of G are sufficiently "spread out"

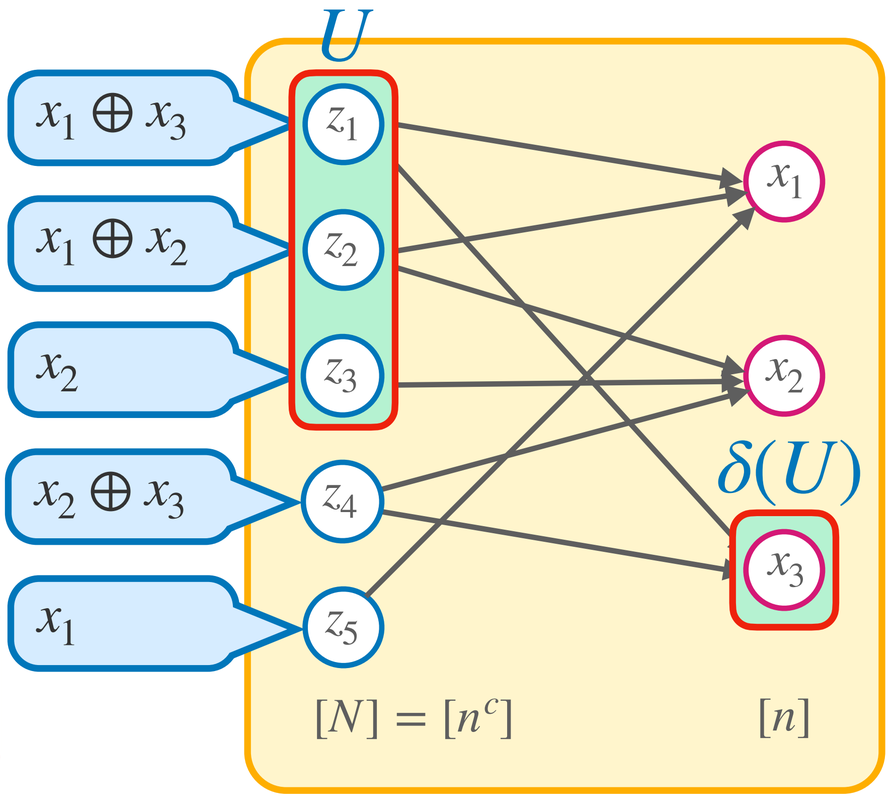
Boundary: $\delta(U)$ of $U \subseteq [N]$ is the number of "unique neighbours" of U

$\rightarrow$ Number of variables that occur in exactly one XOR in U

The Gadget

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

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



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

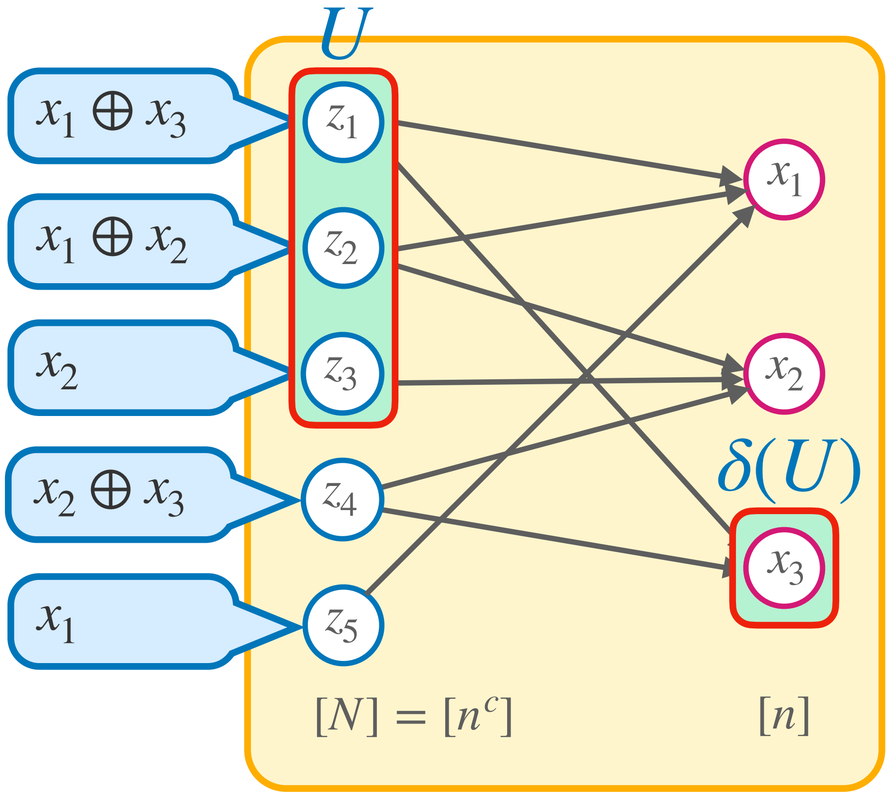
Boundary: $\delta(U)$ of $U \subseteq [N]$ is the number of "unique neighbours" of U

$\rightarrow$ Number of variables that occur in exactly one XOR in U

The Gadget

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

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



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

Boundary: $\delta(U)$ of $U \subseteq [N]$ is the number of "unique neighbours" of U

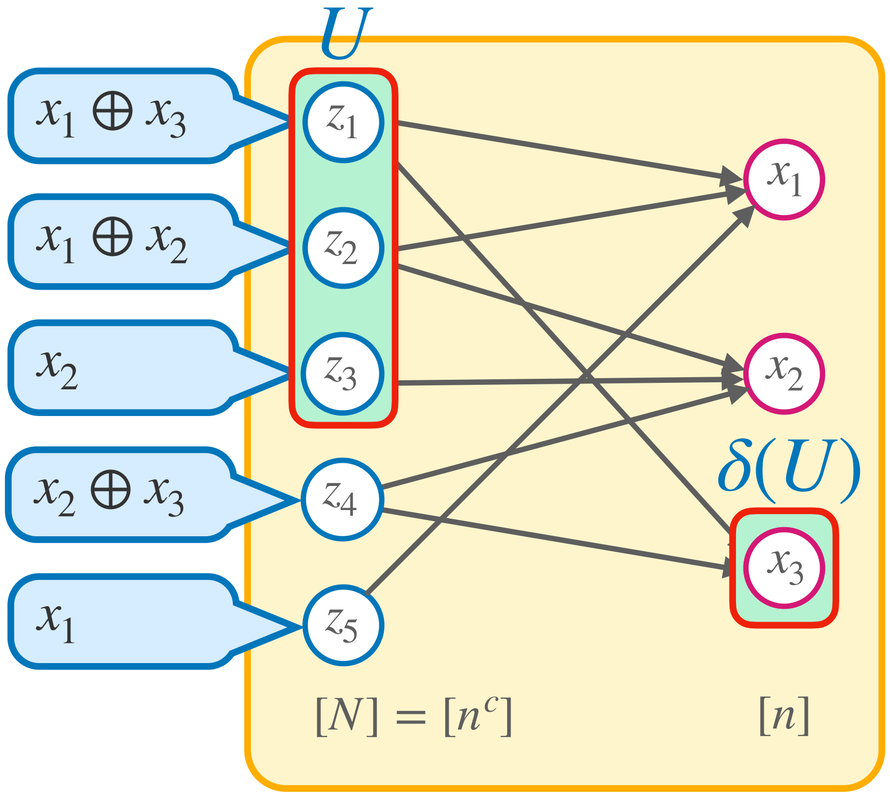
$\rightarrow$ Number of variables that occur in exactly one XOR in U

(r, c)-(Boundary) Expander: If every $U \subseteq [N]$ with $|U| \leq r$ has $|\delta(U)| \geq c\,|U|$

The Gadget

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

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



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

Boundary: $\delta(U)$ of $U \subseteq [N]$ is the number of "unique neighbours" of U

$\rightarrow$ Number of variables that occur in exactly one XOR in U

(r, c)-(Boundary) Expander: If every $U \subseteq [N]$ with $|U| \leq r$ has $|\delta(U)| \geq c\,|U|$

$\rightarrow$ Our gadget g will be XOR_G for expanding G

Depth Condensation

Main workhorse behind our tradeoff:

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

Let G be an $(r,2)$-boundary expander, F any unsatisfiable formula.

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

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

Depth Condensation

Main workhorse behind our tradeoff:

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

Let G be an $(r,2)$-boundary expander, F any unsatisfiable formula.

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

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

$\rightarrow$ We give a simple proof

Depth Condensation

Main workhorse behind our tradeoff:

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

Let G be an $(r,2)$-boundary expander, F any unsatisfiable formula.

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

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

→ We give a simple proof

→ Combine this with the **width-to-size lifting theorem** to prove our main tradeoff!

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

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

Main Tradeoff (For Resolution)

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

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

Idea: Set $N = n^c$.

Main Tradeoff (For Resolution)

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

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

Idea: Set $N = n^c$. Take a formula on N variables which requires large depth but small size

Main Tradeoff (For Resolution)

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

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

Idea: Set $N = n^c$. Take a formula on N variables which requires large depth but small size — Pebbling requires $\Omega(N/\log N)$ depth but has size N proofs

Main Tradeoff (For Resolution)

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

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

Idea: Set $N = n^c$. Take a formula on N variables which requires large depth but small size — Pebbling requires $\Omega(N/\log N)$ depth but has size N proofs

$\rightarrow$ Compose with XOR_G and the **depth condensation theorem** to compress

Main Tradeoff (For Resolution)

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

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

Idea: Set $N = n^c$. Take a formula on N variables which requires large depth but small size — Pebbling requires $\Omega(N/\log N)$ depth but has size N proofs

$\rightarrow$ Compose with XOR_G and the **depth condensation theorem** to compress

$\rightarrow$ Compose with XOR_2 to translate the **width** bound to **size**

Main Tradeoff (For Resolution)

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

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

Pf: Set $N = n^c$. Peb_N requires $\Omega(N/\log N)$ depth but has size N proofs

Main Tradeoff (For Resolution)

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

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

Pf: Set $N = n^c$. Peb_N requires $\Omega(N/\log N)$ depth but has size N proofs

[R16]: Exist $[N] \times [n]$ bipartite $(n^{1-\varepsilon}/c,\ 2)$-expander G

Main Tradeoff (For Resolution)

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

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

Pf: Set $N = n^c$. Peb_N requires $\Omega(N/\log N)$ depth but has size N proofs

[R16]: Exist $[N] \times [n]$ bipartite $(n^{1-\varepsilon}/c,\ 2)$-expander G

Depth Condensation: If Π is a proof of $\text{Peb}_N \circ \text{XOR}_G$ with $\text{width}(\Pi) \leq (n^{1-\varepsilon}/c)$ then

$$\text{width}(\Pi)\text{depth}(\Pi) = \Omega(\text{depth}_{\text{Res}}(\text{Peb}_N)) = \Omega(n^c/c \log n)$$

Main Tradeoff (For Resolution)

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

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

Pf: Set $N = n^c$. Peb_N requires $\Omega(N/\log N)$ depth but has size N proofs

[R16]: Exist $[N] \times [n]$ bipartite $(n^{1-\varepsilon}/c,\ 2)$-expander G

Depth Condensation: If Π is a proof of $\text{Peb}_N \circ \text{XOR}_G$ with $\text{width}(\Pi) \leq (n^{1-\varepsilon}/c)$ then

$$\text{width}(\Pi)\text{depth}(\Pi) = \Omega(\text{depth}_{\text{Res}}(\text{Peb}_N)) = \Omega(n^c/c \log n)$$

Width-to-Size Lifting: If Π is a Resolution proof of $\text{Peb}_N \circ \text{XOR}_G \circ \text{XOR}_2$ then

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

Main Tradeoff (For Resolution)

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

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

Pf: Set $N = n^c$. Peb_N requires $\Omega(N/\log N)$ depth but has size N proofs

[R16]: Exist $[N] \times [n]$ bipartite $(n^{1-\varepsilon}/c,\ 2)$-expander G (left-degree $O(c)$)

Main Tradeoff (For Resolution)

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

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

Pf: Set $N = n^c$. Peb_N requires $\Omega(N/\log N)$ depth but has size N proofs

[R16]: Exist $[N] \times [n]$ bipartite $(n^{1-\varepsilon}/c,\ 2)$-expander G (left-degree $O(c)$)

$\rightarrow$ Each XOR in $\text{Peb}_N \circ \text{XOR}_G \circ \text{XOR}_2$ contains $O(c)$ variables

Main Tradeoff (For Resolution)

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

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

Pf: Set $N = n^c$. Peb_N requires $\Omega(N/\log N)$ depth but has size N proofs

[R16]: Exist $[N] \times [n]$ bipartite $(n^{1-\varepsilon}/c,\ 2)$-expander G (left-degree $O(c)$)

$\rightarrow$ Each XOR in $\text{Peb}_N \circ \text{XOR}_G \circ \text{XOR}_2$ contains $O(c)$ variables

Locally simulate the XOR in every step of the size N proof of Peb_N

Main Tradeoff (For Resolution)

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

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

Pf: Set $N = n^c$. Peb_N requires $\Omega(N/\log N)$ depth but has size N proofs

[R16]: Exist $[N] \times [n]$ bipartite $(n^{1-\varepsilon}/c,\ 2)$-expander G (left-degree $O(c)$)

$\rightarrow$ Each XOR in $\text{Peb}_N \circ \text{XOR}_G \circ \text{XOR}_2$ contains $O(c)$ variables

Locally simulate the XOR in every step of the size N proof of Peb_N

$\rightarrow$ size $n^c \cdot 2^{O(c)}$ Resolution proof

Main Tradeoffs

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

Main Tradeoffs

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

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

Main Tradeoffs

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 lifting theorem XOR_2 from Resolution width to Res(k) size using the switching lemma of **[SBI04]**

Proof of Depth Condensation

Prover Adversary Games: Characterizes Resolution depth of proving F

Proof of Depth Condensation

Prover Adversary Games: Characterizes Resolution depth of proving F

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

Proof of Depth Condensation

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 ρ such that there is $C \in F$ such that $C(\rho) = 0$

Proof of Depth Condensation

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 ρ such that there is $C \in F$ such that $C(\rho) = 0$
- Adversary wants to prolong the game

Proof of Depth Condensation

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 ρ such that there is $C \in F$ such that $C(\rho) = 0$
- Adversary wants to prolong the game

Each round:

Proof of Depth Condensation

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 ρ such that there is $C \in F$ such that $C(\rho) = 0$
- Adversary wants to prolong the game

Each round:

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

Proof of Depth Condensation

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 ρ such that there is $C \in F$ such that $C(\rho) = 0$
- 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 $\rho_i = b$

Proof of Depth Condensation

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 ρ such that there is $C \in F$ such that $C(\rho) = 0$
- 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)

Proof of Depth Condensation

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 ρ such that there is $C \in F$ such that $C(\rho) = 0$
- 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: If $|\rho| \leq w$ always

Proof of Depth Condensation

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 ρ such that there is $C \in F$ such that $C(\rho) = 0$
- 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: If $|\rho| \leq w$ always

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

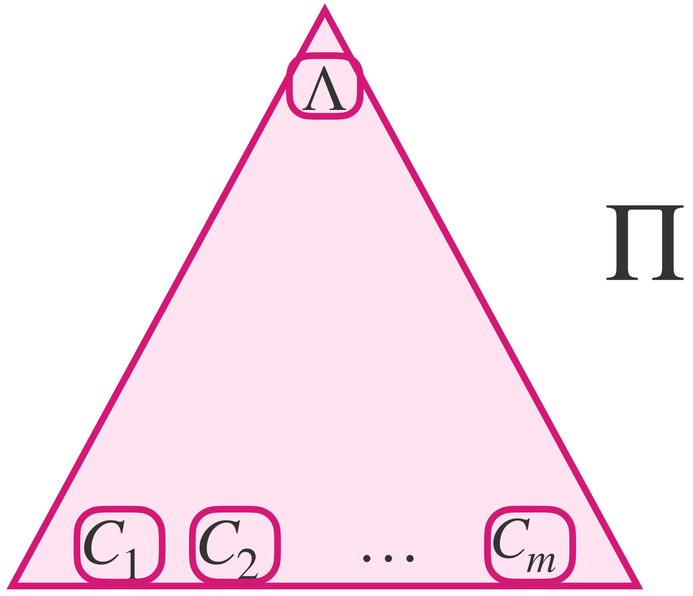
Proof of Depth Condensation

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.

Proof of Depth Condensation

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:

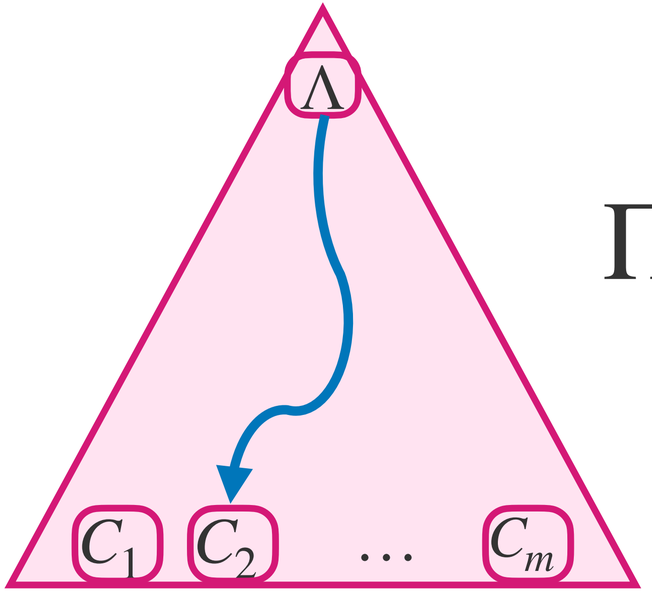


Π

Proof of Depth Condensation

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



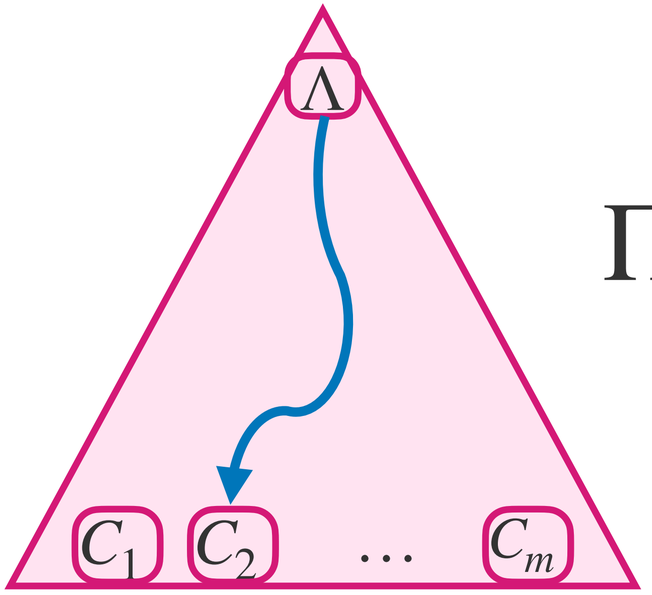
Π

Proof of Depth Condensation

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 $C(\rho) = 0$, $|\rho| \leq w$



Π

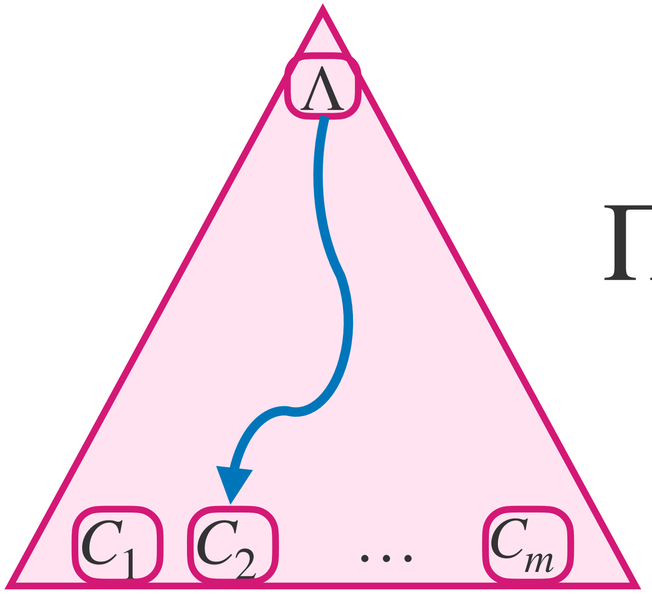
Proof of Depth Condensation

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



Π

Proof of Depth Condensation

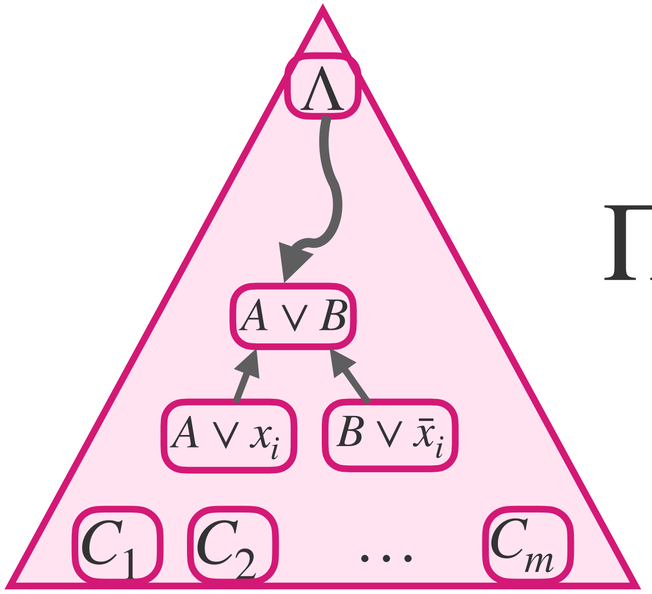
Claim: For any F, if there is a Resolution proof Π of F of width $\leq w$ and depth $\leq d$ then there is a strategy for the Prover to win the $(w+1)$-bounded game in d rounds.

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 $A \vee B$



Π

Proof of Depth Condensation

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.

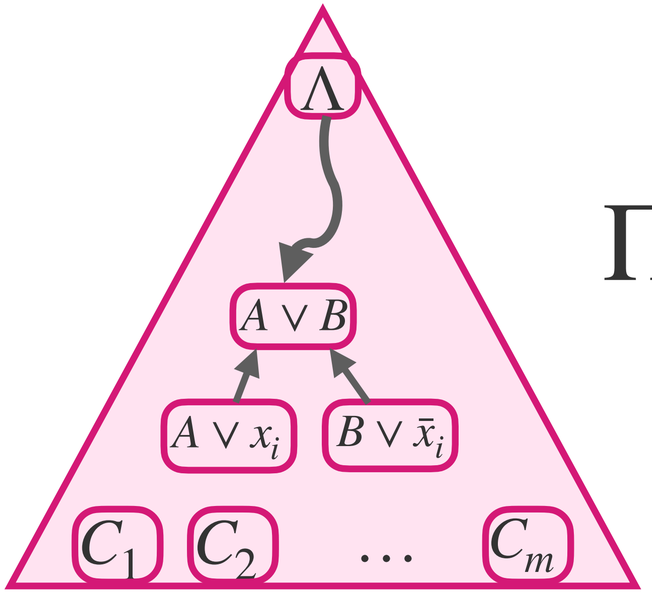
Pf: Prover will walk from the root of Π to a leaf

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



Π

Proof of Depth Condensation

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.

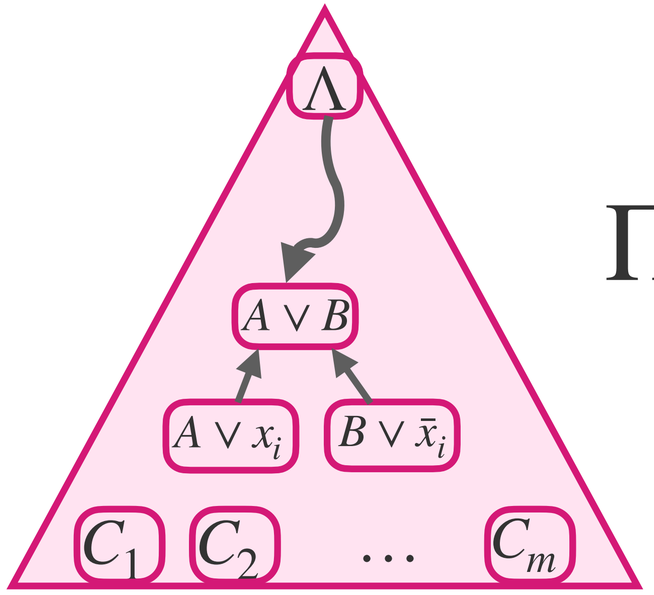
Pf: Prover will walk from the root of Π to a leaf

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 Delayer says $x_i = 0$ then move to $A \vee x_i$. Forget $B \backslash A$



Π

Proof of Depth Condensation

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.

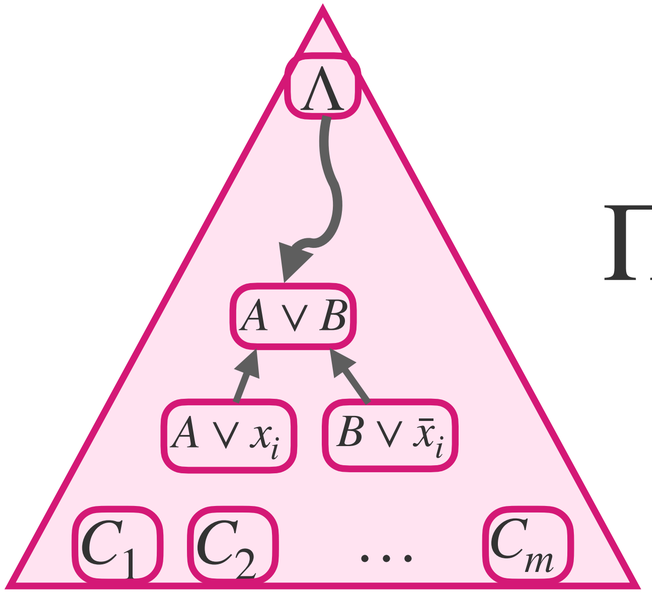
Pf: Prover will walk from the root of Π to a leaf

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 Delayer says $x_i = 0$ then move to $A \vee x_i$. Forget $B \backslash A$
- Otherwise, move to $B \vee \bar{x}_i$. Forget $A \backslash B$



Π

Proof of Depth Condensation

Depth Condensation Theorem:

Let G be an $(r,2)$-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

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

Proof of Depth Condensation

Depth Condensation Theorem:

Let G be an $(r,2)$-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

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

Simplifying Assumption: Delayer can query variables as well

Proof of Depth Condensation

Depth Condensation Theorem:

Let G be an $(r,2)$-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

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

Simplifying Assumption: Delayer can query variables as well

High Level: If $\text{depth}_{\text{Res}}(F) \geq d \Longrightarrow$ strategy A for the Adversary to survive d rounds in the unbounded game on F

Proof of Depth Condensation

Depth Condensation Theorem:

Let G be an $(r,2)$-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

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

Simplifying Assumption: Delayer can query variables as well

High Level: If $\text{depth}_{\text{Res}}(F) \geq d \Longrightarrow$ strategy A for the Adversary to survive d rounds in the unbounded game on F

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

Proof Overview

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

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

Proof Overview

High Level: If $\mathsf{depth}_{\mathsf{Res}}(F) \geq d \implies$ 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 there are two cases

Proof Overview

High Level: If $\mathsf{depth}_{\mathsf{Res}}(F) \geq d \implies$ 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 there are two cases

- If x_i is the last variable in $\mathsf{N}(z_j)$ (for some z_j) not set in ρ:

Proof Overview

High Level: If $\mathsf{depth}_{\mathsf{Res}}(F) \geq d \implies$ 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 there are two cases

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

Proof Overview

High Level: If $\mathrm{depth}_{\mathrm{Res}}(F) \geq d \implies$ 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 there are two cases

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

Proof Overview

High Level: If $\mathsf{depth}_{\mathsf{Res}}(F) \geq d \implies$ 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 there are two cases

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

Proof Overview

High Level: If $\mathsf{depth}_{\mathsf{Res}}(F) \geq d \implies$ 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 there are two cases

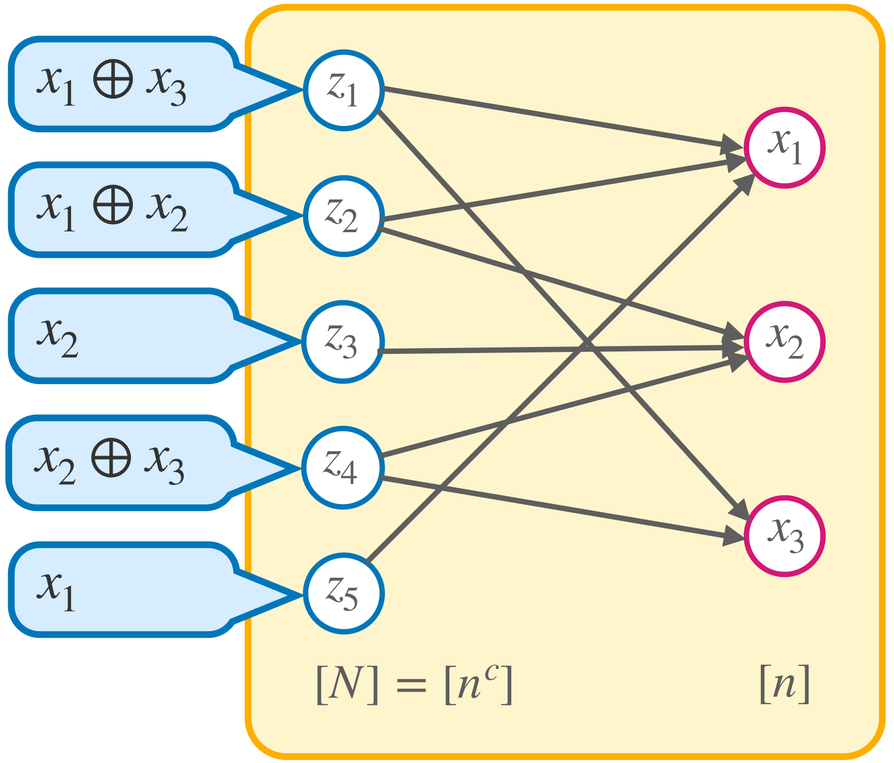
- If x_i is the last variable in $\mathsf{N}(z_j)$ (for some z_j) not set in ρ:
  - Query A for the value b of z_j on state $\mathsf{XOR}_G(\rho)$.
  - Set x_i so that $\bigoplus_{t:x_t \in N(z_j)} x_t = b$
- If there are at least two variables in $\mathsf{N}(z_j)$ for every z_j: set x_i arbitrarily

Unfortunately there is a problem — constraints are correlated!

Proof Overview

Unfortunately there is a problem — constraints are correlated!

e.g. Suppose $n = 3$ and currently $\rho = [1, *, *]$

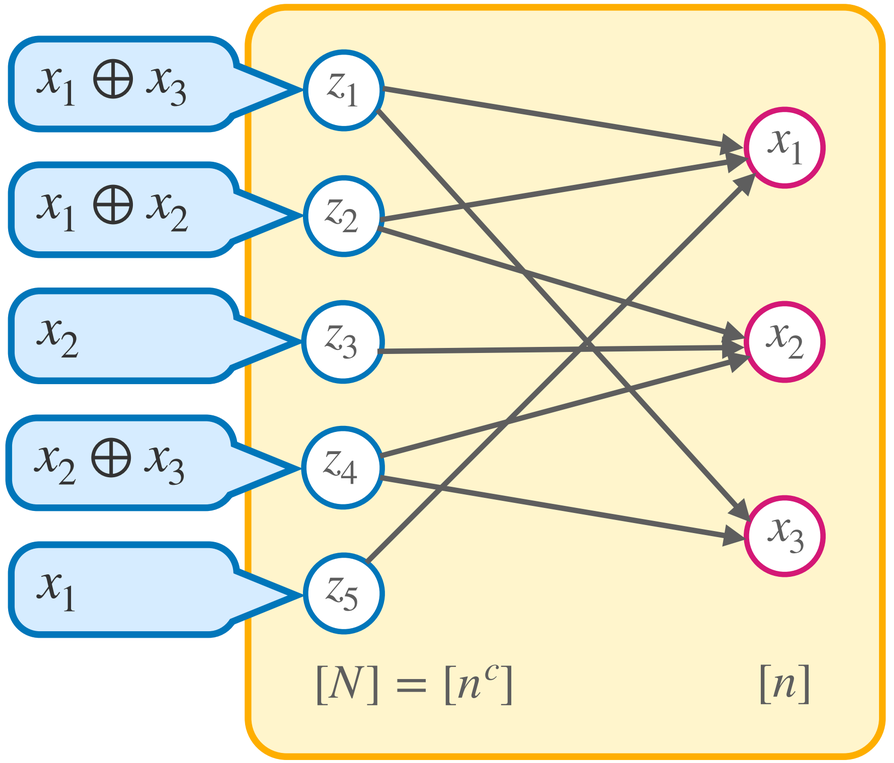


Proof Overview

Unfortunately there is a problem — constraints are correlated!

e.g. Suppose $n = 3$ and currently $\rho = [1, *, *]$

Suppose Prover asks about x_2



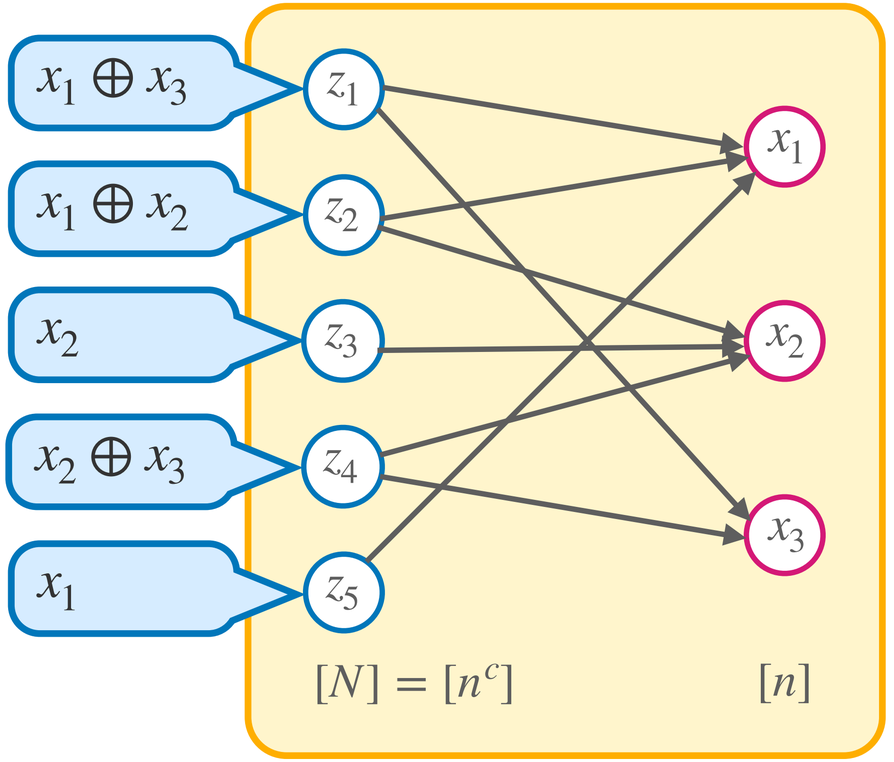
Proof Overview

Unfortunately there is a problem — constraints are correlated!

e.g. Suppose $n = 3$ and currently $\rho = [1, *, *]$

Suppose Prover asks about x_2

$\rightarrow$ x_2 is the last unset variable in $\mathsf{N}(z_2)$



Proof Overview

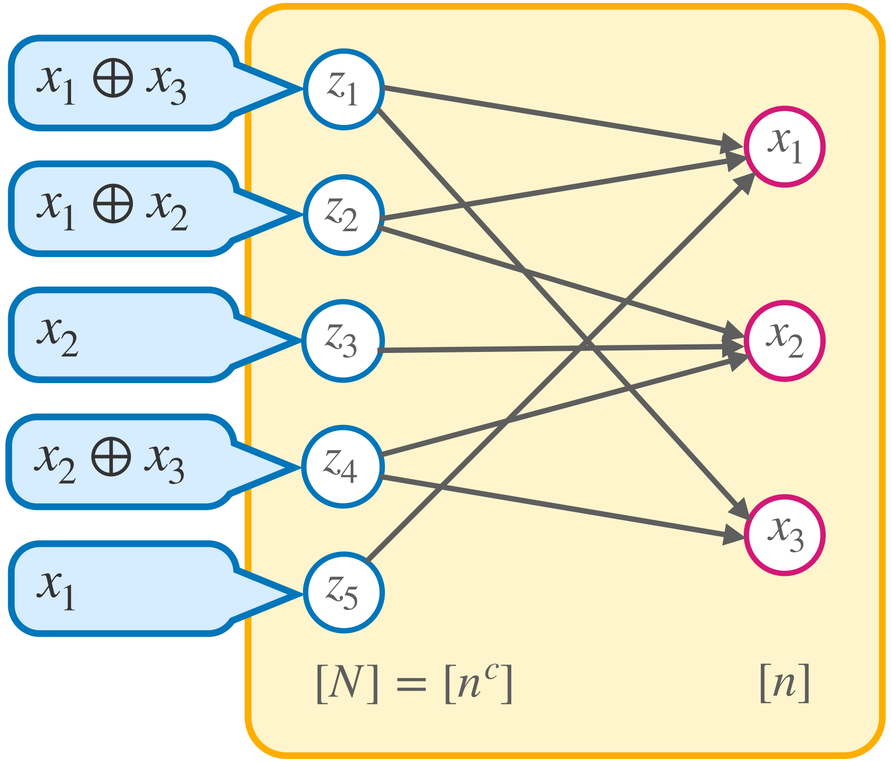
Unfortunately there is a problem — constraints are correlated!

e.g. Suppose $n = 3$ and currently $\rho = [1, *, *]$

Suppose Prover asks about x_2

$\rightarrow$ x_2 is the last unset variable in $\mathsf{N}(z_2)$

Query A for z_2 on state $\mathsf{XOR}_G(\rho) = [*, *, *, *, 1]$



Proof Overview

Unfortunately there is a problem — constraints are correlated!

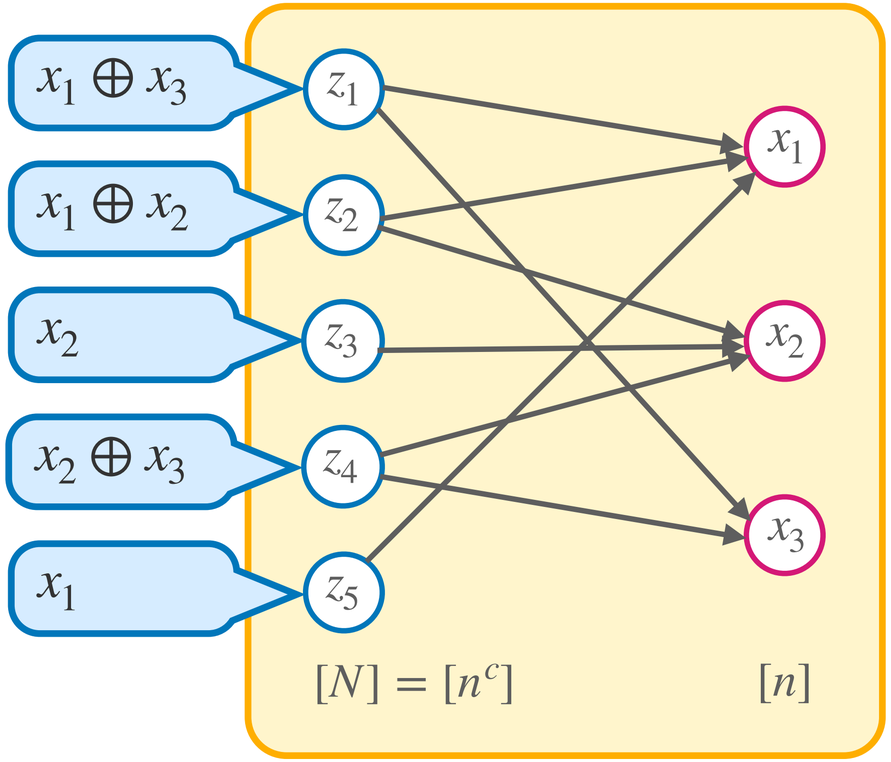
e.g. Suppose $n = 3$ and currently $\rho = [1, *, *]$

Suppose Prover asks about x_2

$\rightarrow$ x_2 is the last unset variable in $\mathsf{N}(z_2)$

Query A for z_2 on state $\mathsf{XOR}_G(\rho) = [*, *, *, *, 1]$

$\rightarrow$ Suppose A responds with $z_2 = 0$



Proof Overview

Unfortunately there is a problem — constraints are correlated!

e.g. Suppose $n = 3$ and currently $\rho = [1, *, *]$

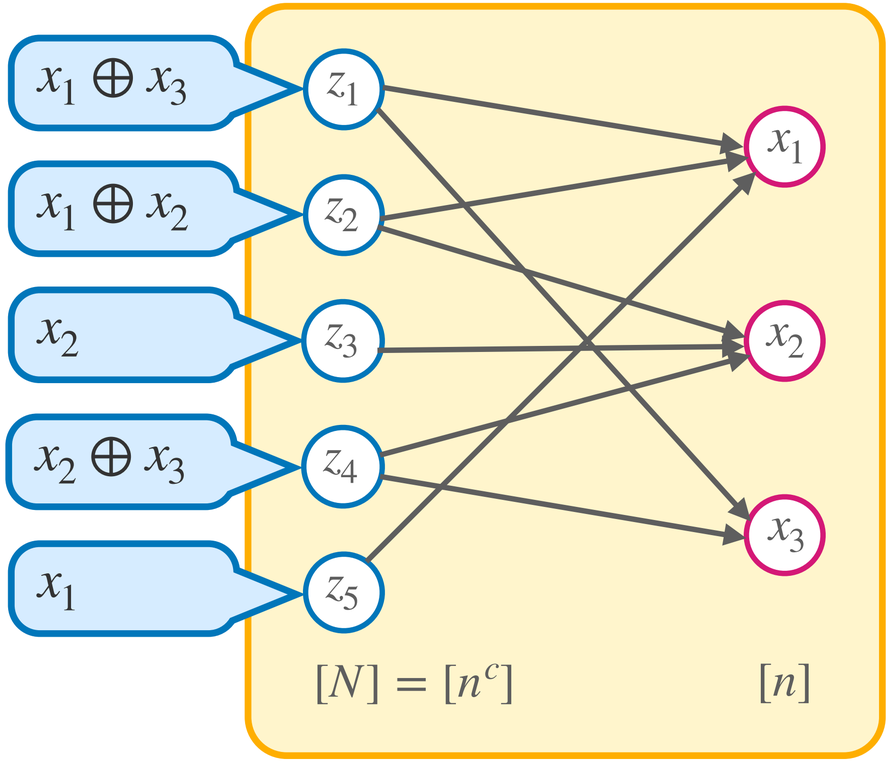
Suppose Prover asks about x_2

→ x_2 is the last unset variable in $\mathsf{N}(z_2)$

Query A for z_2 on state $\mathsf{XOR}_G(\rho) = [*, *, *, *, 1]$

→ Suppose A responds with $z_2 = 0$

→ Update $\rho = [1,1,*]$ so that $z_2 = x_1 \oplus x_2 = 0$



Proof Overview

Unfortunately there is a problem — constraints are correlated!

e.g. Suppose $n = 3$ and currently $\rho = [1, *, *]$

Suppose Prover asks about x_2

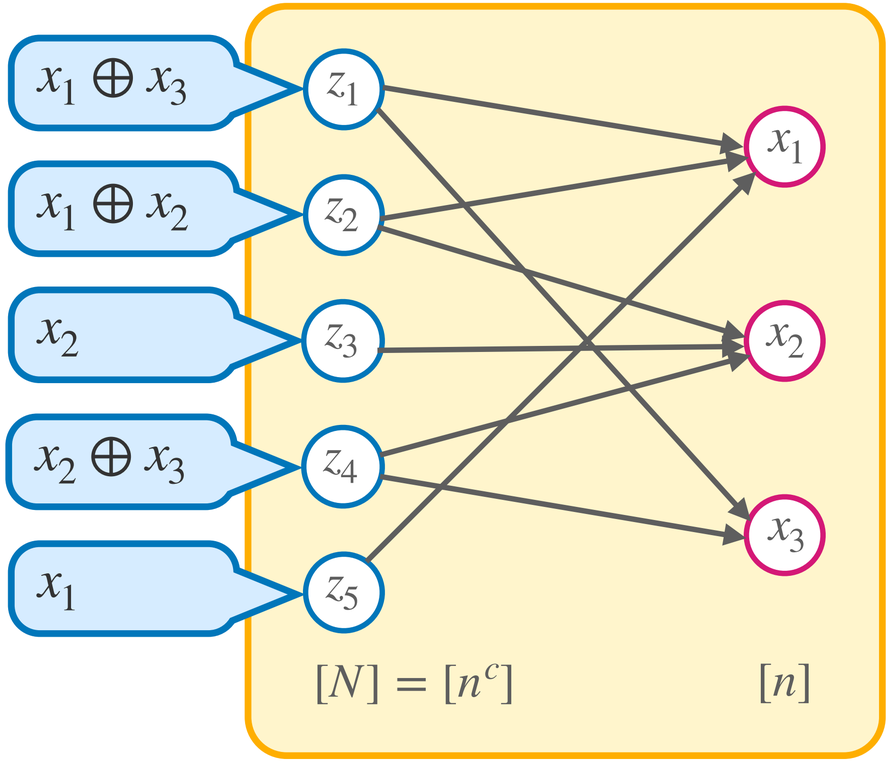
$\rightarrow$ x_2 is the last unset variable in $\mathsf{N}(z_2)$

Query A for z_2 on state $\mathsf{XOR}_G(\rho) = [*, *, *, *, 1]$

$\rightarrow$ Suppose A responds with $z_2 = 0$

$\rightarrow$ Update $\rho = [1,1,*]$ so that $z_2 = x_1 \oplus x_2 = 0$

This forces $z_3 = 1$!



Proof Overview

Unfortunately there is a problem — constraints are correlated!

e.g. Suppose $n = 3$ and currently $\rho = [1, *, *]$

Suppose Prover asks about x_2

→ x_2 is the last unset variable in $\mathsf{N}(z_2)$

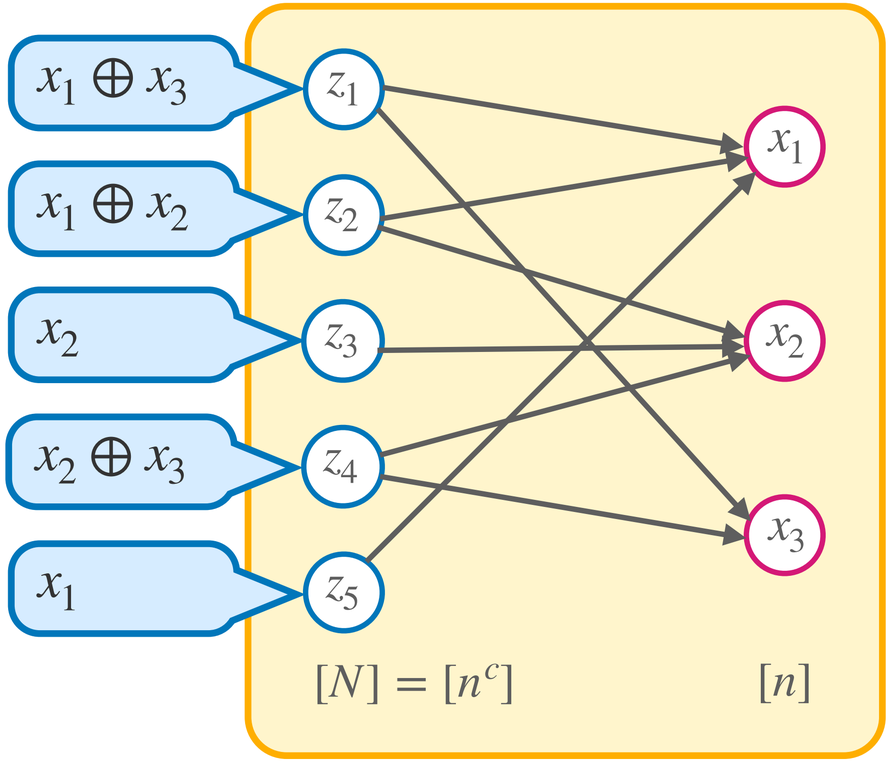
Query A for z_2 on state $\mathsf{XOR}_G(\rho) = [*, *, *, *, 1]$

→ Suppose A responds with $z_2 = 0$

→ Update $\rho = [1,1,*]$ so that $z_2 = x_1 \oplus x_2 = 0$

This forces $z_3 = 1$!

— If on state $[*,0,*,*,1]$ A sets $z_3 = 0$



Proof Overview

Unfortunately there is a problem — constraints are correlated!

e.g. Suppose $n = 3$ and currently $\rho = [1, *, *]$

Suppose Prover asks about x_2

→ x_2 is the last unset variable in $\mathsf{N}(z_2)$

Query A for z_2 on state $\mathsf{XOR}_G(\rho) = [*, *, *, *, 1]$

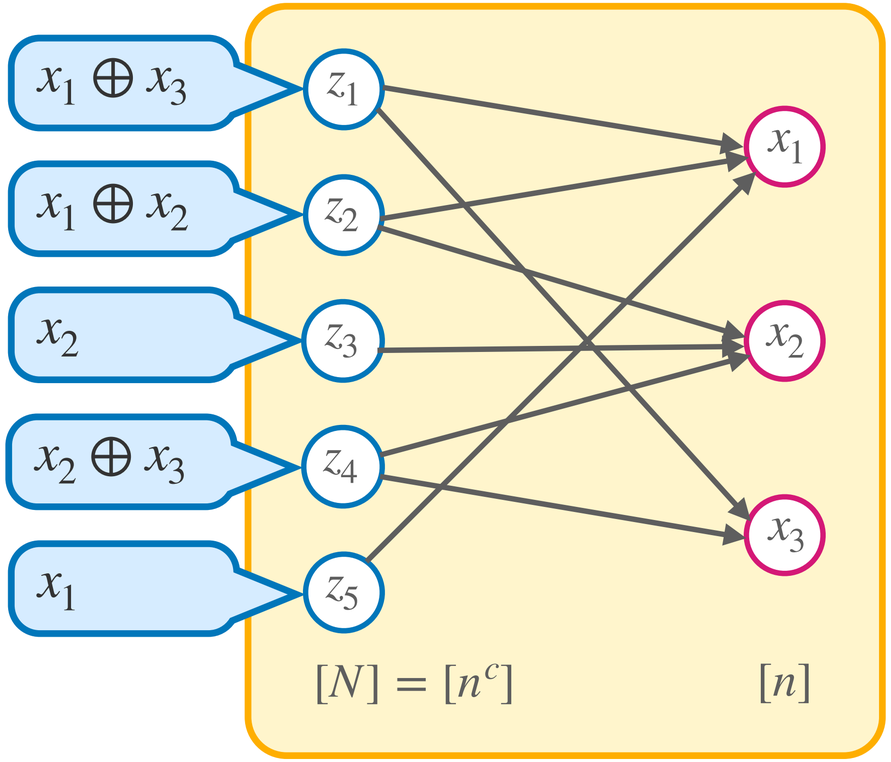
→ Suppose A responds with $z_2 = 0$

→ Update $\rho = [1,1,*]$ so that $z_2 = x_1 \oplus x_2 = 0$

This forces $z_3 = 1$!

— If on state $[*,0,*,*,1]$ A sets $z_3 = 0$

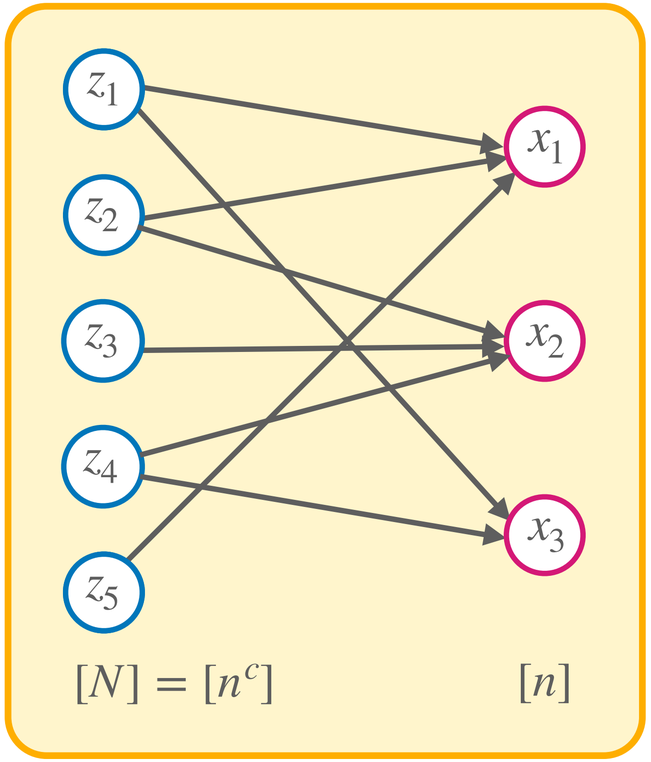
→ We cannot follow A!



Proof Overview

Unfortunately there is a problem — constraints are correlated!

Use expansion to avoid bad situations where setting the value of x_i determines more than one z-variable!



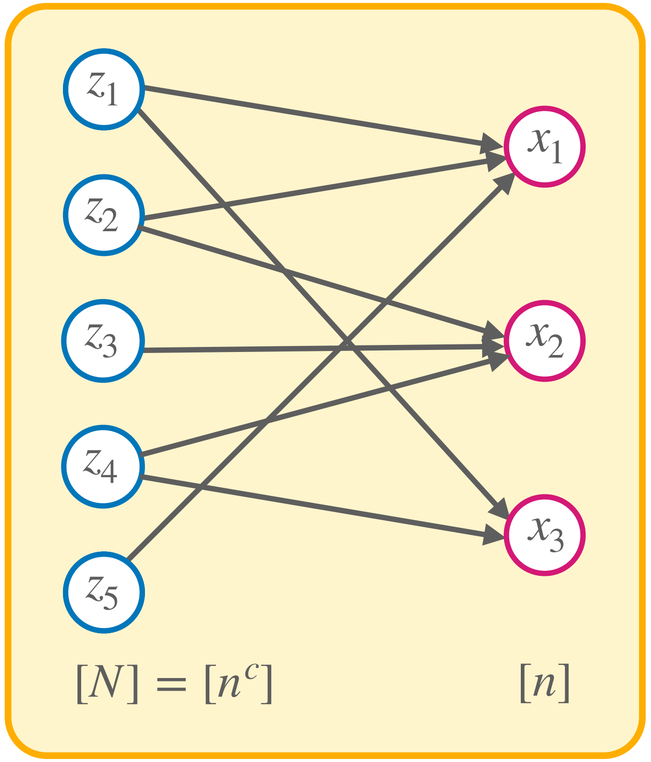
Proof Overview

Unfortunately there is a problem — constraints are correlated!

Use expansion to avoid bad situations where setting the value of x_i determines more than one z-variable!

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

$$\text{Fixed}(\rho) := \{z_j \in [N] : \text{N}(z_j) \text{ is in } \rho\}$$



Proof Overview

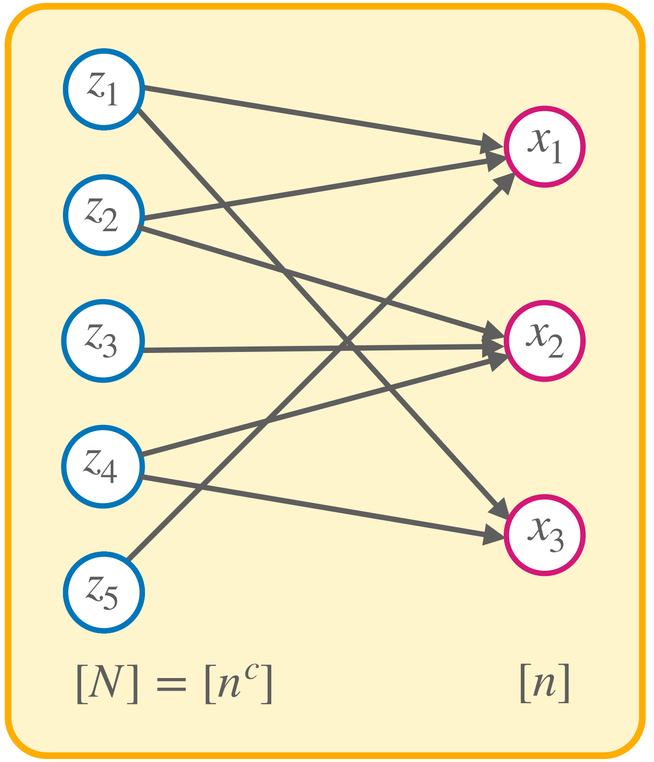
Unfortunately there is a problem — constraints are correlated!

Use expansion to avoid bad situations where setting the value of x_i determines more than one z-variable!

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

$$\text{Fixed}(\rho) := \{z_j \in [N] : \text{N}(z_j) \text{ is in } \rho\}$$

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



Proof Overview

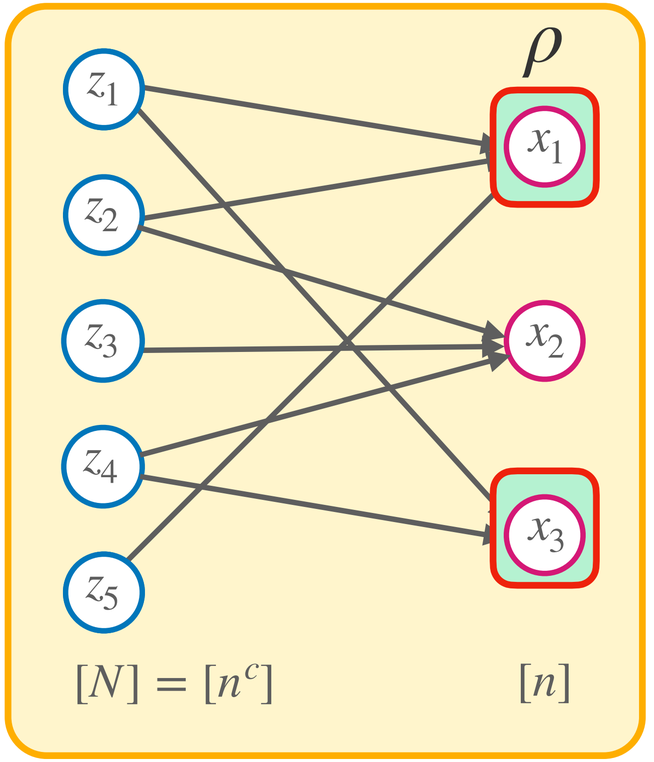
Unfortunately there is a problem — constraints are correlated!

Use expansion to avoid bad situations where setting the value of x_i determines more than one z-variable!

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

$$\text{Fixed}(\rho) := \{z_j \in [N] : \text{N}(z_j) \text{ is in } \rho\}$$

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



Proof Overview

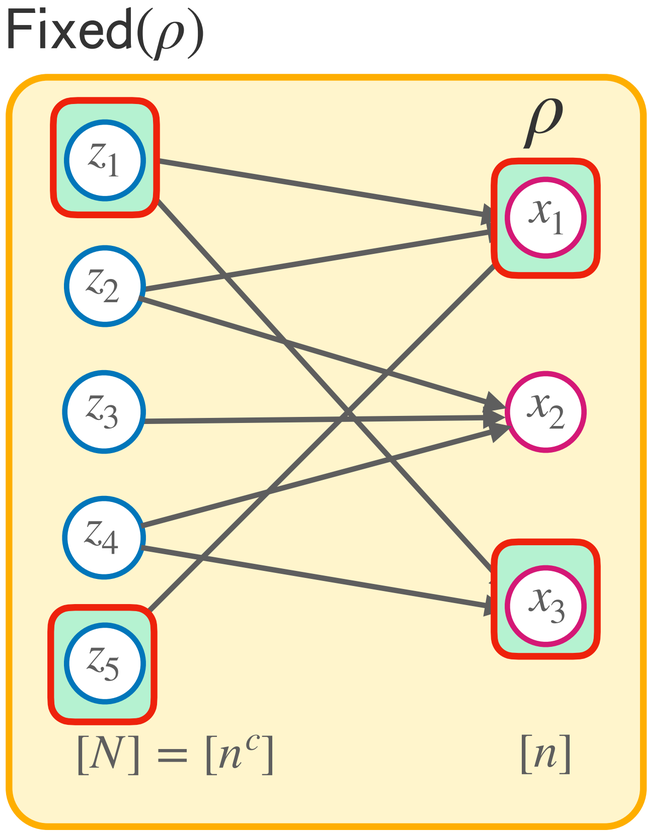
Unfortunately there is a problem — constraints are correlated!

Use expansion to avoid bad situations where setting the value of x_i determines more than one z-variable!

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

$$\text{Fixed}(\rho) := \{z_j \in [N] : \text{N}(z_j) \text{ is in } \rho\}$$

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



Proof Overview

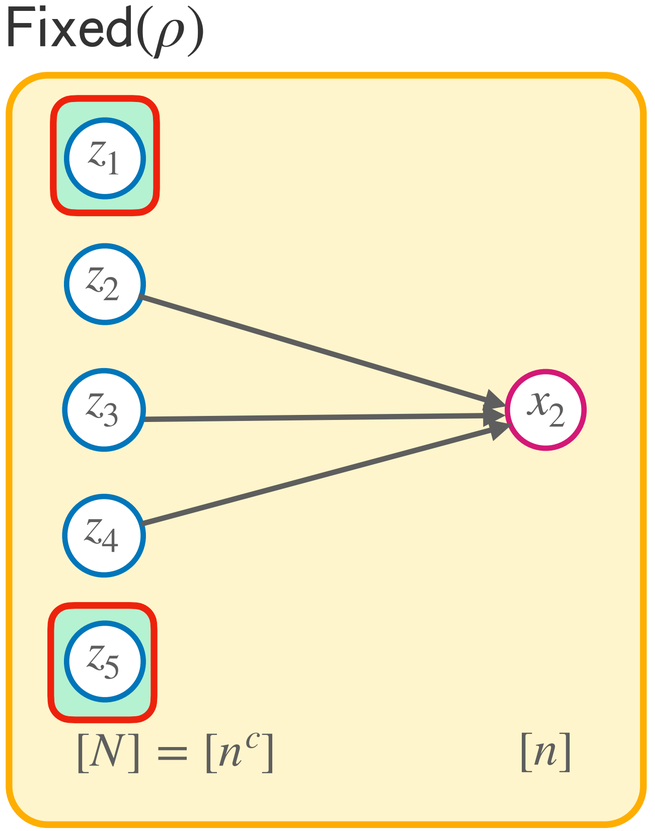
Unfortunately there is a problem — constraints are correlated!

Use expansion to avoid bad situations where setting the value of x_i determines more than one z-variable!

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

$$\text{Fixed}(\rho) := \{z_j \in [N] : \text{N}(z_j) \text{ is in } \rho\}$$

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



Proof Overview

Unfortunately there is a problem — constraints are correlated!

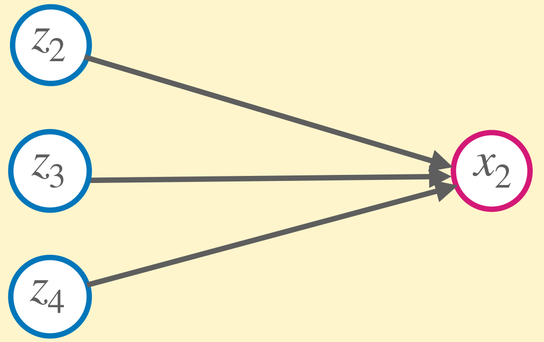
Use expansion to avoid bad situations where setting the value of x_i determines more than one z-variable!

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

$$\text{Fixed}(\rho) := \{z_j \in [N] : \text{N}(z_j) \text{ is in } \rho\}$$

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

$G\backslash\rho$



$[N] = [n^c]$ $[n]$

Proof Overview

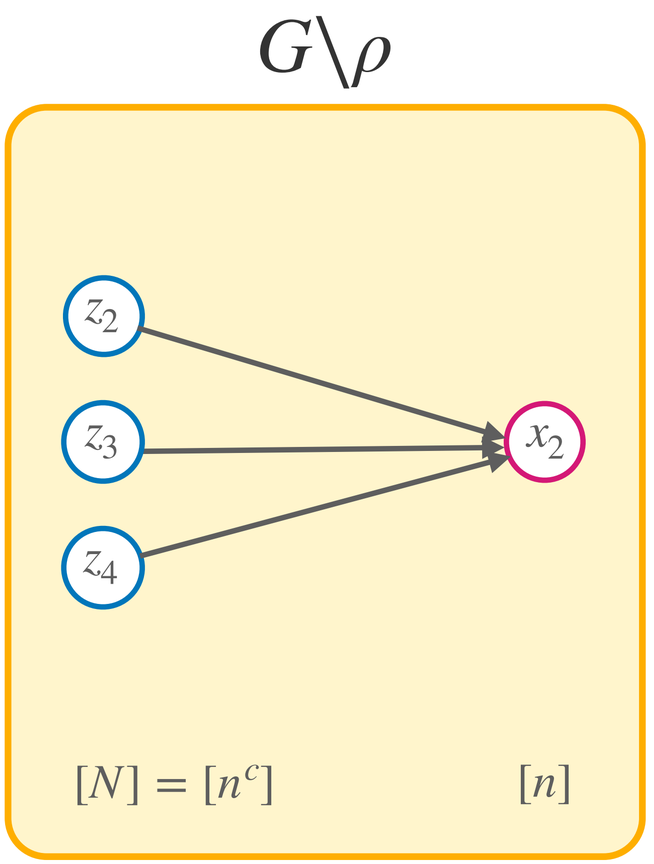
Unfortunately there is a problem — constraints are correlated!

Use expansion to avoid bad situations where setting the value of x_i determines more than one z-variable!

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

$$\text{Fixed}(\rho) := \{z_j \in [N] : \text{N}(z_j) \text{ is in } \rho\}$$

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



We will maintain the following invariant

Invariant: $G\backslash\rho$ is $(r/2, 3/2)$-expanding

Proof Overview

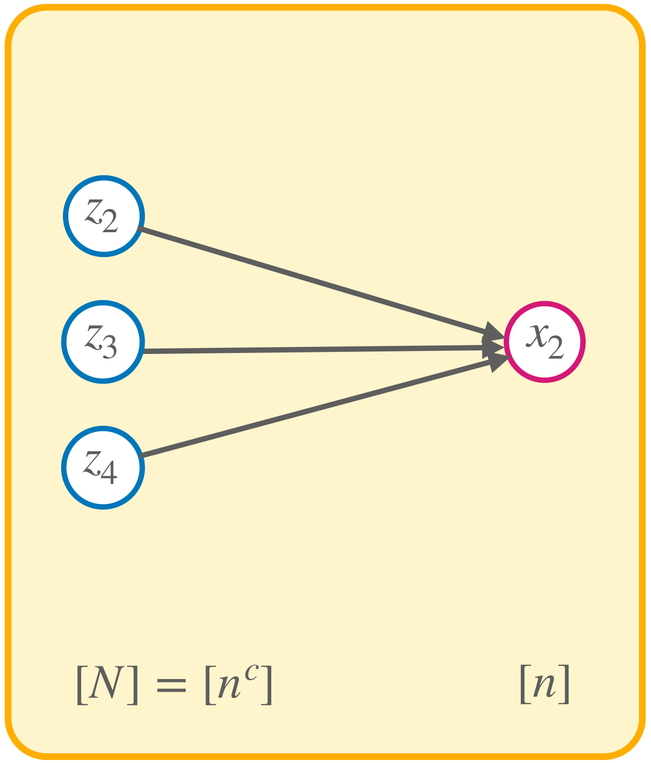
Unfortunately there is a problem — constraints are correlated!

Use expansion to avoid bad situations where setting the value of x_i determines more than one z-variable!

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

$$\text{Fixed}(\rho) := \{z_j \in [N] : \text{N}(z_j) \text{ is in } \rho\}$$

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



We will maintain the following invariant

Invariant: $G\backslash\rho$ is $(r/2, 3/2)$-expanding

→ Setting x_i doesn't determine any z-variable

Expansion Restoration

However, after setting x_i, $G \backslash \rho$ may no longer be $(r/2, 3/2)$-expanding…

Expansion Restoration

However, after setting x_i, $G \backslash \rho$ may no longer be $(r/2, 3/2)$-expanding…

$\rightarrow$ Query additional variables to restore expansion!

Expansion Restoration

However, after setting x_i, $G \backslash \rho$ may no longer be $(r/2, 3/2)$-expanding…

$\rightarrow$ Query additional variables to restore expansion!

Want to assign few z-variables while doing this

Expansion Restoration

However, after setting x_i, $G \backslash \rho$ may no longer be $(r/2, 3/2)$-expanding…

→ Query additional variables to restore expansion!

Want to assign few z-variables while doing this

→ each time we fix a z-variable we query the Adversary strategy A for its value

– can only do at most d times in total

Expansion Restoration

However, after setting x_i, $G \backslash \rho$ may no longer be $(r/2, 3/2)$-expanding…

→ Query additional variables to restore expansion!

Want to assign few z-variables while doing this

→ each time we fix a z-variable we query the Adversary strategy A for its value

– can only do at most d times in total

Closure Lemma [Alek05]: If G is an $(r,2)$-boundary expander, then for any ρ, $|\rho| \leq r/4$ there exists $\mathsf{Cl}(\rho) \subseteq [n]$, $\mathsf{Cl}(\rho) \supseteq \rho$ such that

Expansion Restoration

However, after setting x_i, $G \setminus \rho$ may no longer be $(r/2, 3/2)$-expanding…

→ Query additional variables to restore expansion!

Want to assign few z-variables while doing this

→ each time we fix a z-variable we query the Adversary strategy A for its value

– can only do at most d times in total

Closure Lemma [Alek05]: If G is an $(r,2)$-boundary expander, then for any ρ, $|\rho| \leq r/4$ there exists $\mathsf{Cl}(\rho) \subseteq [n]$, $\mathsf{Cl}(\rho) \supseteq \rho$ such that

1. The sets few z-variables: $|\mathsf{Fixed}(\mathsf{Cl}(\rho))| \leq 2|\rho|$

Expansion Restoration

However, after setting x_i, $G \backslash \rho$ may no longer be $(r/2, 3/2)$-expanding…

→ Query additional variables to restore expansion!

Want to assign few z-variables while doing this

→ each time we fix a z-variable we query the Adversary strategy A for its value

– can only do at most d times in total

Closure Lemma [Alek05]: If G is an $(r, 2)$-boundary expander, then for any ρ, $|\rho| \leq r/4$ there exists $\mathrm{Cl}(\rho) \subseteq [n]$, $\mathrm{Cl}(\rho) \supseteq \rho$ such that

1. The sets few z-variables: $|\mathrm{Fixed}(\mathrm{Cl}(\rho))| \leq 2|\rho|$
2. $G \backslash \mathrm{Cl}(\rho)$ is an $(r/2, 3/2)$-boundary expander

Expansion Restoration

However, after setting x_i, $G \backslash \rho$ may no longer be $(r/2, 3/2)$-expanding…

→ Query additional variables to restore expansion!

Want to assign few z-variables while doing this

→ each time we fix a z-variable we query the Adversary strategy A for its value

– can only do at most d times in total

Closure Lemma [Alek05]: If G is an $(r,2)$-boundary expander, then for any ρ, $|\rho| \leq r/4$ there exists $\mathrm{Cl}(\rho) \subseteq [n]$, $\mathrm{Cl}(\rho) \supseteq \rho$ such that

1. The sets few z-variables: $|\mathrm{Fixed}(\mathrm{Cl}(\rho))| \leq 2|\rho|$
2. $G \backslash \mathrm{Cl}(\rho)$ is an $(r/2, 3/2)$-boundary expander

→ To restore expansion, set the variables in $\mathrm{Cl}(\rho)$

Expansion Restoration

However, after setting x_i, $G \backslash \rho$ may no longer be $(r/2, 3/2)$-expanding…

→ Query additional variables to restore expansion!

Want to assign few z-variables while doing this

→ each time we fix a z-variable we query the Adversary strategy A for its value

– can only do at most d times in total

Closure Lemma [Alek05]: If G is an $(r,2)$-boundary expander, then for any ρ, $|\rho| \leq r/4$ there exists $\mathsf{Cl}(\rho) \subseteq [n]$, $\mathsf{Cl}(\rho) \supseteq \rho$ such that

1. The sets few z-variables: $|\mathsf{Fixed}(\mathsf{Cl}(\rho))| \leq 2|\rho|$
2. $G \backslash \mathsf{Cl}(\rho)$ is an $(r/2, 3/2)$-boundary expander

→ To restore expansion, set the variables in $\mathsf{Cl}(\rho)$

– Must be able to set z-variables in $\mathsf{Fixed}(\mathsf{Cl}(\rho))$ consistent with A while doing this

Expansion Restoration

However, after setting x_i, $G\backslash\rho$ may no longer be $(r/2,3/2)$-expanding…

→ Query additional variables to restore expan

Want to assign few z-variables while doing this

→ each time we fix a z-variable we query the

– can only do at most d times in total

$|\mathsf{Cl}(\rho)|$ may be larger than $w = r/4$, but don't worry about that for now

Closure Lemma [Alek05]: If G is an $(r,2)$-boundary expander, then for any ρ, $|\rho| \leq r/4$ there exists $\mathsf{Cl}(\rho) \subseteq [n]$, $\mathsf{Cl}(\ ) \supseteq \rho$ such that

1. The sets few z-variables: $|\mathsf{Fixed}(\mathsf{Cl}(\rho))| \leq 2|\rho|$
2. $G\backslash\mathsf{Cl}(\rho)$ is an $(r/2,3/2)$-boundary expander

→ To restore expansion, set the variables in $\mathsf{Cl}(\rho)$

– Must be able to set z-variables in $\mathsf{Fixed}(\mathsf{Cl}(\rho))$ consistent with A while doing this

Expansion Restoration

However, after setting x_i, $G \setminus \rho$ may no longer be $(r/2, 3/2)$-expanding…

Expansion Restoration

However, after setting x_i, $G \backslash \rho$ may no longer be $(r/2, 3/2)$-expanding…

But it is $(r/2, 1/2)$-expanding!

Expansion Restoration

However, after setting x_i, $G \backslash \rho$ may no longer be $(r/2, 3/2)$-expanding…

But it is $(r/2, 1/2)$-expanding!

$\rightarrow$ Setting a single x-variable can only decrease the boundary by at most 1

Expansion Restoration

However, after setting x_i, $G \setminus \rho$ may no longer be $(r/2, 3/2)$-expanding…

But it is $(r/2, 1/2)$-expanding!

$\rightarrow$ Setting a single x-variable can only decrease the boundary by at most 1

Use this remaining expansion to set the variables in $\mathrm{Cl}(\rho)$ consistently with A!

Expansion Restoration

However, after setting x_i, $G \setminus \rho$ may no longer be $(r/2, 3/2)$-expanding…

But it is $(r/2, 1/2)$-expanding!

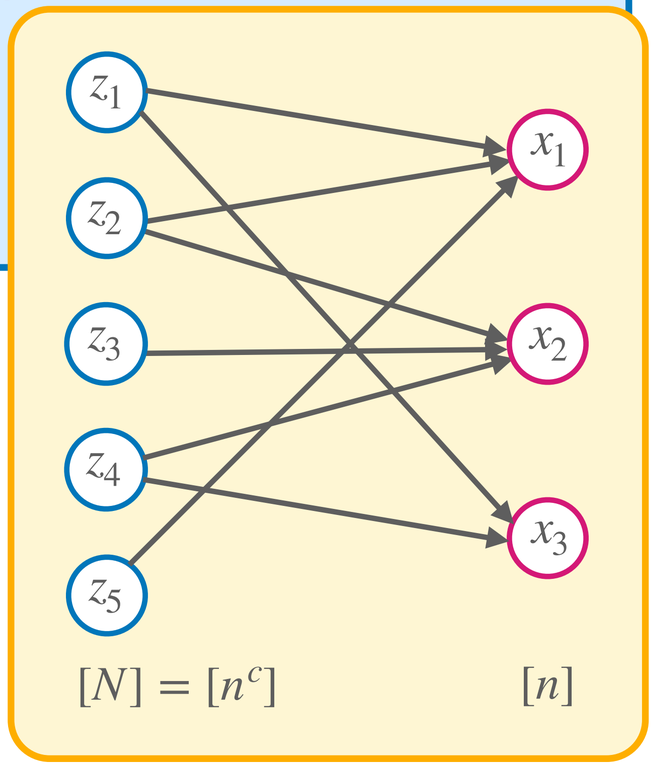
$\rightarrow$ Setting a single x-variable can only decrease the boundary by at most 1

Use this remaining expansion to set the variables in $\mathrm{Cl}(\rho)$ consistently with A!

$\rightarrow$ If $G \setminus \rho$ is $(r/2, 1/2)$-expanding then we can find a strong system of distinct representatives (SDR)

Expansion Restoration

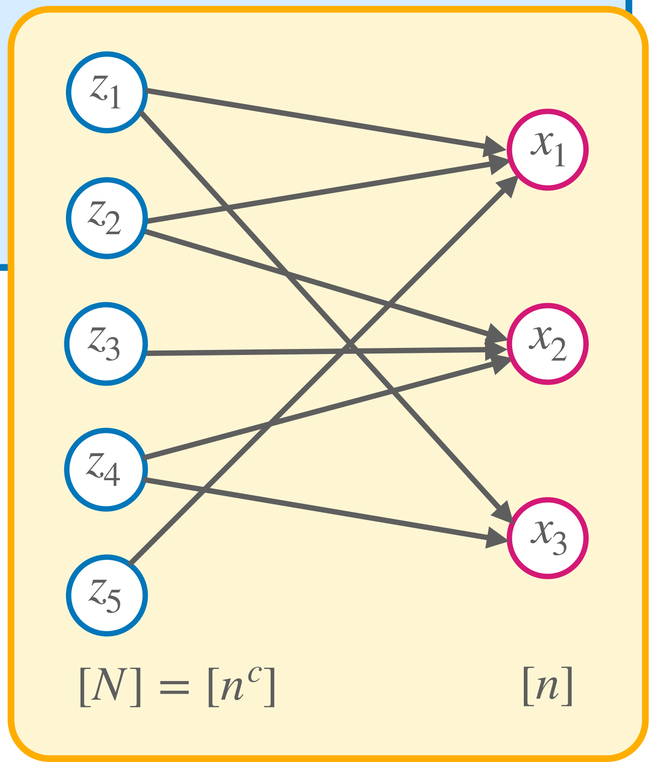
A **strong SDR** of $I = \{I_1, \ldots, I_t\} \subseteq [N]$ is a set $J = \{J_1, \ldots J_t\} \subseteq [n]$ such that



Expansion Restoration

A **strong SDR** of $I = \{I_1, \ldots, I_t\} \subseteq [N]$ is a set $J = \{J_1, \ldots J_t\} \subseteq [n]$ such that

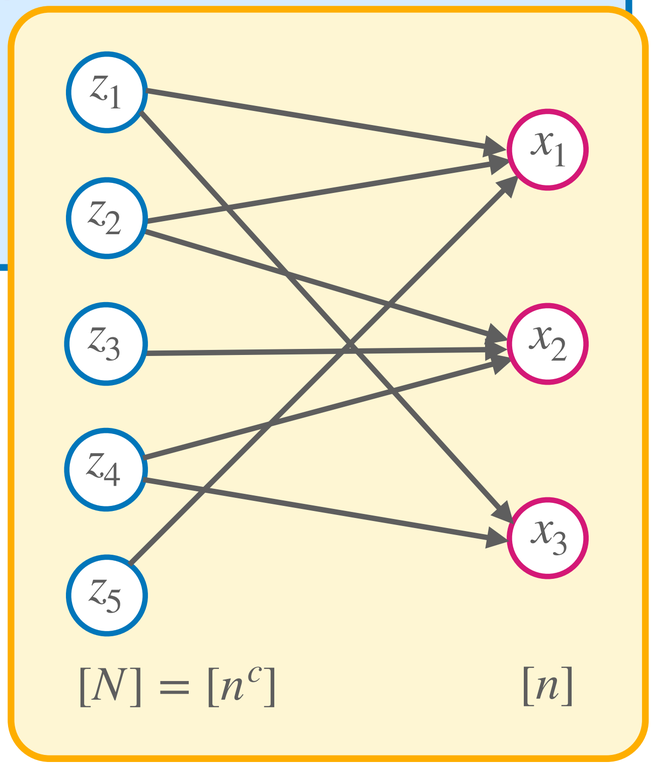
1. There is a matching between I and J in G



Expansion Restoration

A **strong SDR** of $I = \{I_1, \ldots, I_t\} \subseteq [N]$ is a set $J = \{J_1, \ldots J_t\} \subseteq [n]$ such that

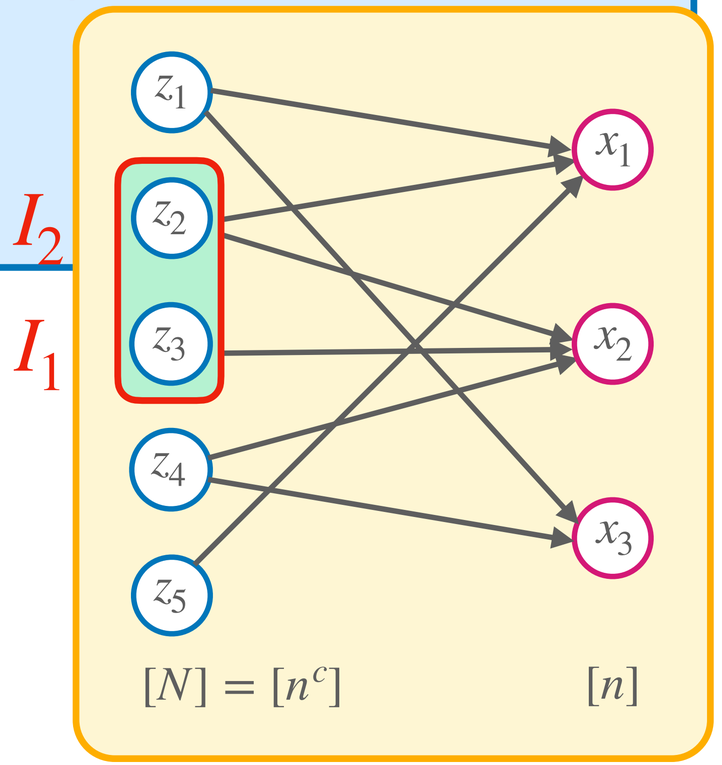
1. There is a matching between I and J in G
2. z_{I_i} is not adjacent to x_{J_j} for $j > i$



Expansion Restoration

A **strong SDR** of $I = \{I_1, \ldots, I_t\} \subseteq [N]$ is a set $J = \{J_1, \ldots J_t\} \subseteq [n]$ such that

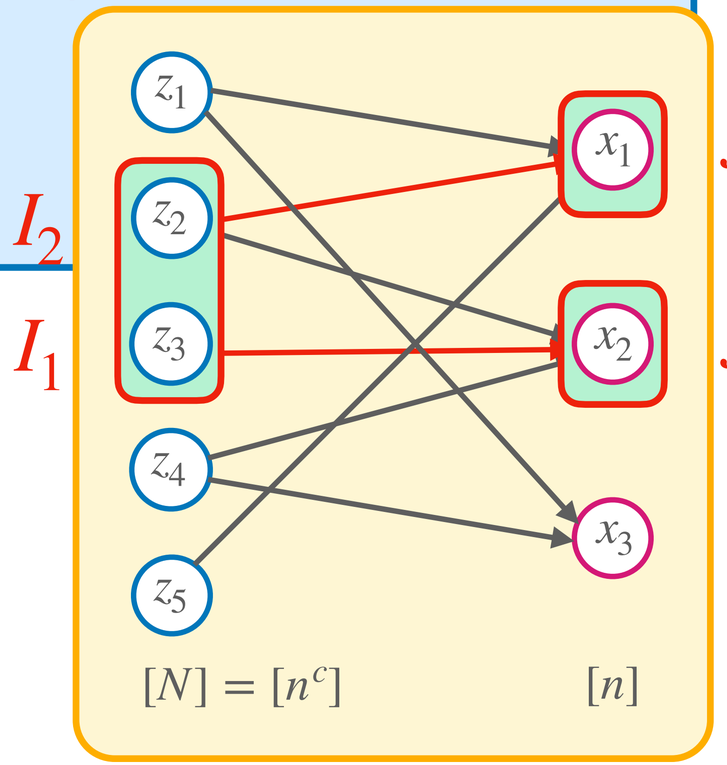
1. There is a matching between I and J in G
2. z_{I_i} is not adjacent to x_{J_j} for $j > i$



Expansion Restoration

A **strong SDR** of $I = \{I_1, \ldots, I_t\} \subseteq [N]$ is a set $J = \{J_1, \ldots J_t\} \subseteq [n]$ such that

1. There is a matching between I and J in G
2. z_{I_i} is not adjacent to x_{J_j} for $j > i$

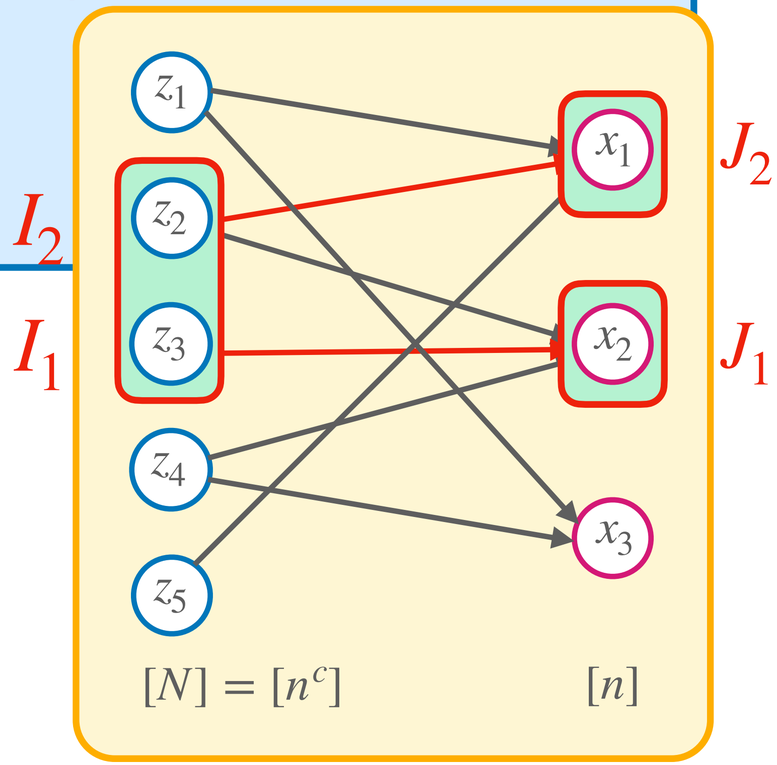


Expansion Restoration

A **strong SDR** of $I = \{I_1, \ldots, I_t\} \subseteq [N]$ is a set $J = \{J_1, \ldots J_t\} \subseteq [n]$ such that

1. There is a matching between I and J in G
2. z_{I_i} is not adjacent to x_{J_j} for $j > i$

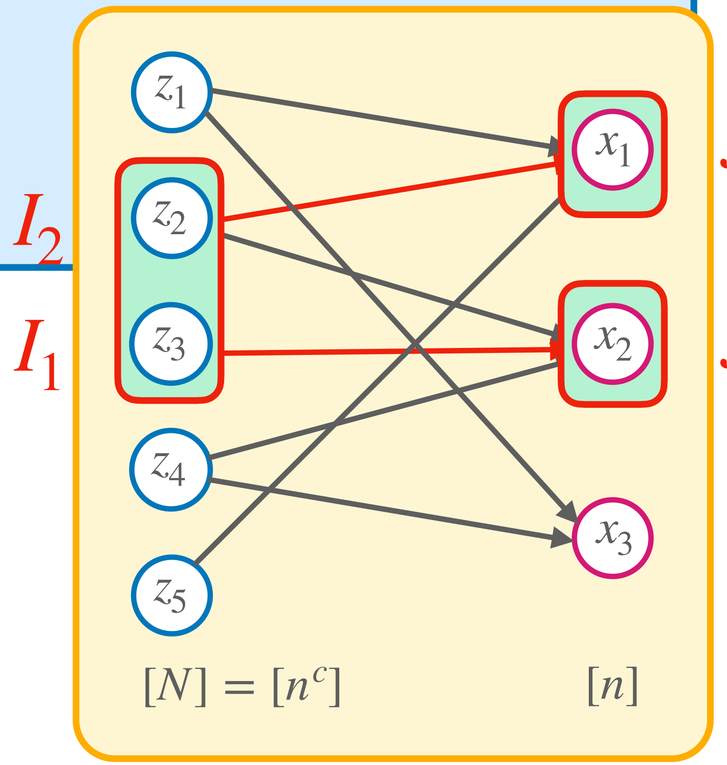
Allows us to set the constraints in I however we like



Expansion Restoration

A **strong SDR** of $I = \{I_1, \ldots, I_t\} \subseteq [N]$ is a set $J = \{J_1, \ldots J_t\} \subseteq [n]$ such that

1. There is a matching between I and J in G
2. z_{I_i} is not adjacent to x_{J_j} for $j > i$



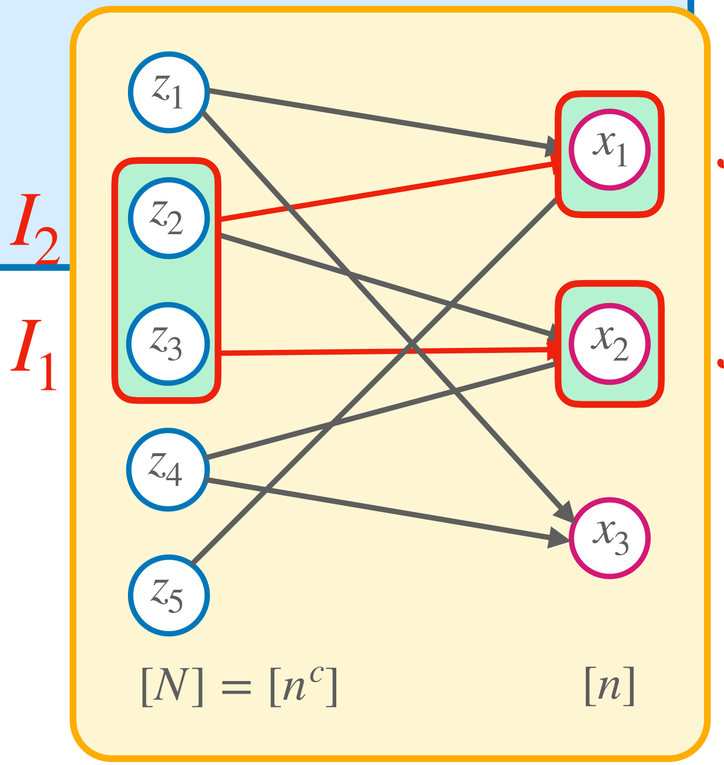
Allows us to set the constraints in I however we like

$\rightarrow$ Fix the variables in $\mathsf{N}(z_{I_1})$

Expansion Restoration

A **strong SDR** of $I = \{I_1, \ldots, I_t\} \subseteq [N]$ is a set $J = \{J_1, \ldots J_t\} \subseteq [n]$ such that

1. There is a matching between I and J in G
2. z_{I_i} is not adjacent to x_{J_j} for $j > i$



Allows us to set the constraints in I however we like

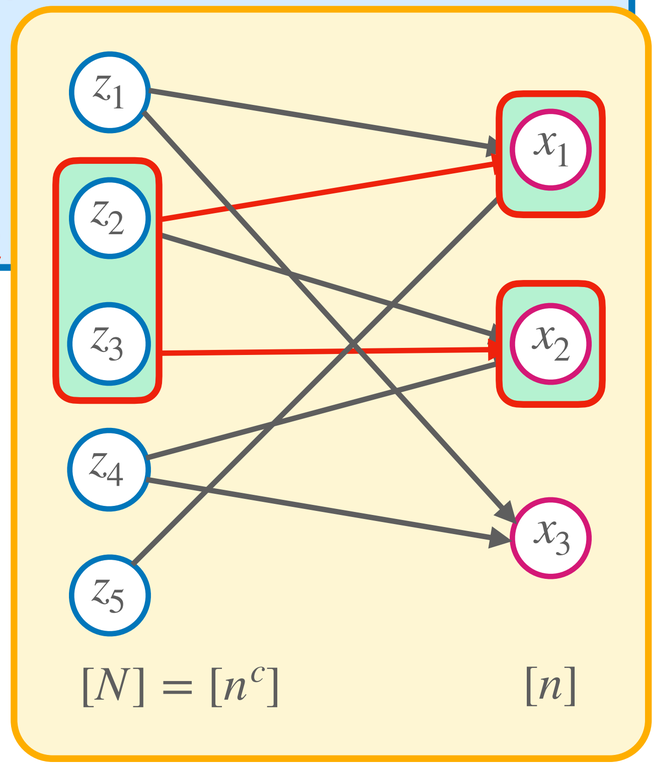
→ Fix the variables in $\mathsf{N}(z_{I_1})$

→ by (2) there is at least one free variable for $z_{I_2}, \ldots, z_{I_t}$

Expansion Restoration

A **strong SDR** of $I = \{I_1, \ldots, I_t\} \subseteq [N]$ is a set $J = \{J_1, \ldots J_t\} \subseteq [n]$ such that

1. There is a matching between I and J in G
2. z_{I_i} is not adjacent to x_{J_j} for $j > i$



Allows us to set the constraints in I however we like

→ Fix the variables in $\mathsf{N}(z_{I_1})$

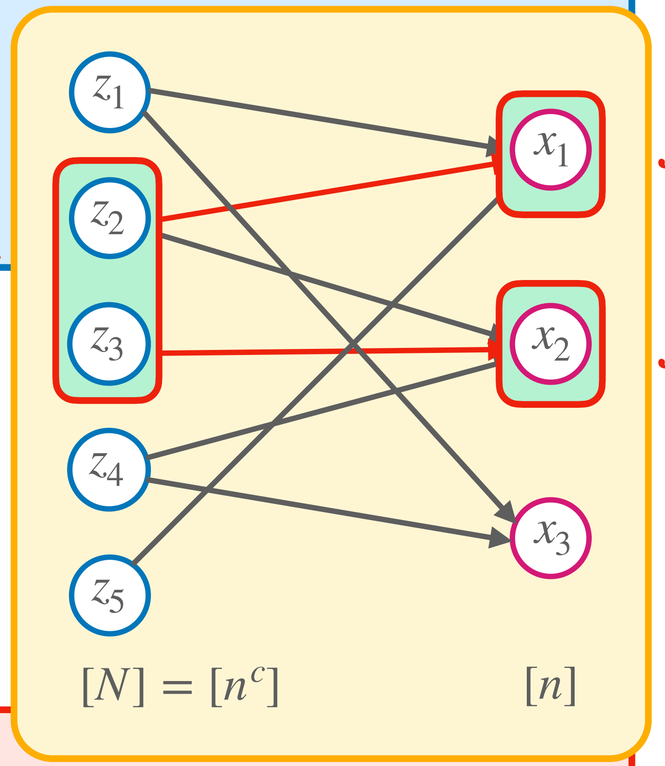
→ by (2) there is at least one free variable for $z_{I_2}, \ldots, z_{I_t}$

→ etc.

Expansion Restoration

A **strong SDR** of $I = \{I_1, \ldots, I_t\} \subseteq [N]$ is a set $J = \{J_1, \ldots J_t\} \subseteq [n]$ such that

1. There is a matching between I and J in G
2. z_{I_i} is not adjacent to x_{J_j} for $j > i$



Allows us to set the constraints in I however we like

→ Fix the variables in $\mathsf{N}(z_{I_1})$

→ by (2) there is at least one free variable for $z_{I_2}, \ldots, z_{I_t}$

→ etc.

SDR Lemma:

If $G \backslash \rho$ is a $(r/2, 1/2)$-boundary expander $\Longrightarrow$ any $|I| \leq r/2$ has a strong SDR

Expansion Restoration

To restore expansion, set the variables in $\mathrm{Cl}(\rho)$ as follows: let A be the adversary for F

Expansion Restoration

To restore expansion, set the variables in $\mathsf{Cl}(\rho)$ as follows: let A be the adversary for F

RestoreExpansion(ρ, $\mathsf{Cl}(\rho)$): Such that $G \setminus \rho$ is a $(r/2, 1/2)$-expander

Expansion Restoration

To restore expansion, set the variables in $\mathrm{Cl}(\rho)$ as follows: let A be the adversary for F

RestoreExpansion($\rho, \mathrm{Cl}(\rho)$): Such that $G \setminus \rho$ is a $(r/2, 1/2)$-expander

SDR Lemma: $\mathrm{Fixed}(\mathrm{Cl}(\rho)) \setminus \mathrm{Fixed}(\rho) = \{I_1, \ldots, I_t\}$ has a strong SDR $J = J_1, \ldots, J_t$

Expansion Restoration

To restore expansion, set the variables in $\mathrm{Cl}(\rho)$ as follows: let A be the adversary for F

RestoreExpansion($\rho, \mathrm{Cl}(\rho)$): Such that $G \backslash \rho$ is a $(r/2, 1/2)$-expander

SDR Lemma: $\mathrm{Fixed}(\mathrm{Cl}(\rho)) \backslash \mathrm{Fixed}(\rho) = \{I_1, \ldots, I_t\}$ has a strong SDR $J = J_1, \ldots, J_t$

$\rightarrow$ Set the variables in $\mathrm{Cl}(\rho) \backslash J$ arbitrarily — they do not fix any z-variables

Expansion Restoration

To restore expansion, set the variables in $\mathsf{Cl}(\rho)$ as follows: let A be the adversary for F

RestoreExpansion($\rho, \mathsf{Cl}(\rho)$): Such that $G \backslash \rho$ is a $(r/2, 1/2)$-expander

SDR Lemma: $\mathsf{Fixed}(\mathsf{Cl}(\rho)) \backslash \mathsf{Fixed}(\rho) = \{I_1, \ldots, I_t\}$ has a strong SDR $J = J_1, \ldots, J_t$

$\rightarrow$ Set the variables in $\mathsf{Cl}(\rho) \backslash J$ arbitrarily — they do not fix any z-variables

$\rightarrow$ For $\ell = 1, \ldots, t$:

Expansion Restoration

To restore expansion, set the variables in $\mathsf{Cl}(\rho)$ as follows: let A be the adversary for F

RestoreExpansion($\rho, \mathsf{Cl}(\rho)$): Such that $G \backslash \rho$ is a $(r/2, 1/2)$-expander

SDR Lemma: $\mathsf{Fixed}(\mathsf{Cl}(\rho)) \backslash \mathsf{Fixed}(\rho) = \{I_1, \ldots, I_t\}$ has a strong SDR $J = J_1, \ldots, J_t$

→ Set the variables in $\mathsf{Cl}(\rho) \backslash J$ arbitrarily — they do not fix any z-variables

→ For $\ell = 1, \ldots, t$:

- z_{I_ℓ} has exactly one unset variable, x_{J_ℓ}

Expansion Restoration

To restore expansion, set the variables in $\mathsf{Cl}(\rho)$ as follows: let A be the adversary for F

RestoreExpansion(ρ, $\mathsf{Cl}(\rho)$): Such that $G \setminus \rho$ is a $(r/2, 1/2)$-expander

SDR Lemma: $\mathsf{Fixed}(\mathsf{Cl}(\rho)) \setminus \mathsf{Fixed}(\rho) = \{I_1, \ldots, I_t\}$ has a strong SDR $J = J_1, \ldots, J_t$

$\rightarrow$ Set the variables in $\mathsf{Cl}(\rho) \setminus J$ arbitrarily — they do not fix any z-variables

$\rightarrow$ For $\ell = 1, \ldots, t$:

- z_{I_ℓ} has exactly one unset variable, x_{J_ℓ}
- Query A on state $\mathsf{XOR}_G(\rho)$ for the value $b \in \{0,1\}$ to set z_{I_ℓ}

Expansion Restoration

To restore expansion, set the variables in $\mathsf{Cl}(\rho)$ as follows: let A be the adversary for F

RestoreExpansion($\rho, \mathsf{Cl}(\rho)$): Such that $G \backslash \rho$ is a $(r/2, 1/2)$-expander

SDR Lemma: $\mathsf{Fixed}(\mathsf{Cl}(\rho)) \backslash \mathsf{Fixed}(\rho) = \{I_1, \ldots, I_t\}$ has a strong SDR $J = J_1, \ldots, J_t$

$\rightarrow$ Set the variables in $\mathsf{Cl}(\rho) \backslash J$ arbitrarily — they do not fix any z-variables

$\rightarrow$ For $\ell = 1, \ldots, t$:

- z_{I_ℓ} has exactly one unset variable, x_{J_ℓ}
- Query A on state $\mathsf{XOR}_G(\rho)$ for the value $b \in \{0,1\}$ to set z_{I_ℓ}
- Set x_{I_j} so that $\bigoplus_{x_i \in \mathsf{N}(z_{I_\ell})} x_i = b$

Expansion Restoration

To restore expansion, set the variables in $\mathsf{Cl}(\rho)$ as follows: let A be the adversary for F

RestoreExpansion($\rho, \mathsf{Cl}(\rho)$): Such that $G \setminus \rho$ is a $(r/2, 1/2)$-expander

SDR Lemma: $\mathsf{Fixed}(\mathsf{Cl}(\rho)) \setminus \mathsf{Fixed}(\rho) = \{I_1, \ldots, I_t\}$ has a strong SDR $J = J_1, \ldots, J_t$

→ Set the variables in $\mathsf{Cl}(\rho) \setminus J$ arbitrarily — they do not fix any z-variables

→ For $\ell = 1, \ldots, t$:

- z_{I_ℓ} has exactly one unset variable, x_{J_ℓ}
- Query A on state $\mathsf{XOR}_G(\rho)$ for the value $b \in \{0,1\}$ to set z_{I_ℓ}
- Set x_{I_j} so that $\bigoplus_{x_i \in \mathsf{N}(z_{I_\ell})} x_i = b$

→ By the closure lemma, $G \setminus \rho$ is now a $(r/2, 3/2)$-expander — Invariant restored!

Expansion Restoration

To restore expansion, set the variables in $\mathsf{Cl}(\rho)$ as follows: let A be the adversary for F

RestoreExpansion($\rho, \mathsf{Cl}(\rho)$): Such that $G\backslash\rho$ is a $(r/2, 1/2)$-expander

SDR Lemma: $\mathsf{Fixed}(\mathsf{Cl}(\rho))\backslash\mathsf{Fixed}(\rho) = \{I_1, \ldots, I_t\}$ has a strong SDR $J = J_1, \ldots, J_t$

$\rightarrow$ Set the variables in $\mathsf{Cl}(\rho)\backslash J$ arbitrarily — they do not fix any z-variables

$\rightarrow$ For $\ell = 1, \ldots, t$:

$\vdots$

Cost:

$|\rho| \leq r/4 \Longrightarrow t \leq |\mathsf{Fixed}(\mathsf{Cl}(\rho))| \leq r/2$ (Closure Lemma)

Expansion Restoration

To restore expansion, set the variables in $\mathsf{Cl}(\rho)$ as follows: let A be the adversary for F

RestoreExpansion($\rho, \mathsf{Cl}(\rho)$): Such that $G \setminus \rho$ is a $(r/2, 1/2)$-expander

SDR Lemma: $\mathsf{Fixed}(\mathsf{Cl}(\rho)) \setminus \mathsf{Fixed}(\rho) = \{I_1, \ldots, I_t\}$ has a strong SDR $J = J_1, \ldots, J_t$

$\rightarrow$ Set the variables in $\mathsf{Cl}(\rho) \setminus J$ arbitrarily — they do not fix any z-variables

$\rightarrow$ For $\ell = 1, \ldots, t$:

$\vdots$

Cost:

$|\rho| \leq r/4 \implies t \leq |\mathsf{Fixed}(\mathsf{Cl}(\rho))| \leq r/2$ (Closure Lemma)

$\implies$ We query A at most $r/2 = O(w)$ times

Adversary Strategy

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

Adversary Strategy

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

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

Adversary Strategy

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

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

Invariant: $G \backslash \rho$ is an $(r/2, 3/2)$-boundary expander

Adversary Strategy

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

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

Invariant: $G \backslash \rho$ is an $(r/2, 3/2)$-boundary expander

Query: If Prover asks for the value of x_i

$\rightarrow$ Set x_i arbitrarily

Adversary Strategy

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

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

Invariant: $G \backslash \rho$ is an $(r/2, 3/2)$-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

Adversary Strategy

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

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

Invariant: $G \backslash \rho$ is an $(r/2, 3/2)$-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: Run RestoreExpansion$(\rho, \mathsf{Cl}(\rho))$

Adversary Strategy

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

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

Invariant: $G \backslash \rho$ is an $(r/2, 3/2)$-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: Run RestoreExpansion$(\rho, \mathsf{Cl}(\rho))$

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

Adversary Strategy

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

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

Invariant: $G \backslash \rho$ is an $(r/2, 3/2)$-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 y_j

Restore Expansion: Run RestoreExpansion$(\rho, \mathsf{Cl}(\rho))$

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

Problem! Only the Prover can query variables $\rightarrow$ Cannot carry out RestoreExpansion

Adversary Strategy

Problem! Only the Prover can query variables → Cannot carry out RestoreExpansion

Adversary Strategy

Problem! Only the Prover can query variables → Cannot carry out RestoreExpansion

Simulate querying by having the Adversary track an additional state $\rho * \supseteq \rho$

Adversary Strategy

Problem! Only the Prover can query variables $\rightarrow$ Cannot carry out RestoreExpansion

Simulate querying by having the Adversary track an additional state $\rho^* \supseteq \rho$

$\rightarrow \rho^*$ will record the assignment to $\mathsf{Cl}(\rho)$

Adversary Strategy

Problem! Only the Prover can query variables → Cannot carry out RestoreExpansion

Simulate querying by having the Adversary track an additional state $\rho^* \supseteq \rho$

→ ρ^* will record the assignment to $\mathsf{Cl}(\rho)$

→ We will maintain that $G \backslash \rho^*$ is expanding, rather than $G \backslash \rho$

Adversary Strategy

Problem! Only the Prover can query variables → Cannot carry out RestoreExpansion

Simulate querying by having the Adversary track an additional state $\rho^* \supseteq \rho$

→ ρ^* will record the assignment to $\mathsf{Cl}(\rho)$

→ We will maintain that $G|\rho^*$ is expanding, rather than $G|\rho$

→ If the Prover asks about a variable x_i such that $\rho_i^* \neq *$ → set $x_i = \rho_i^*$

Adversary Strategy

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

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

Adversary Strategy

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

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

Invariant: $G \backslash \rho^*$ is an $(r/2, 3/2)$-boundary expander

Adversary Strategy

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

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

Invariant: $G \backslash \rho^*$ is an $(r/2, 3/2)$-boundary expander

Query: If Prover asks for the value of x_i, set $x_i = b$ where

Adversary Strategy

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

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

Invariant: $G \backslash \rho^*$ is an $(r/2, 3/2)$-boundary expander

Query: If Prover asks for the value of x_i, set $x_i = b$ where

$\rightarrow$ If $\rho_i^* \neq *$ then $b = \rho_i^*$

Adversary Strategy

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

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

Invariant: $G \backslash \rho^*$ is an $(r/2, 3/2)$-boundary expander

Query: If Prover asks for the value of x_i, set $x_i = b$ where

$\rightarrow$ If $\rho_i^* \neq *$ then $b = \rho_i^*$

$\rightarrow$ If $\rho_i^* = *$ then b is an arbitrary value in $\{0,1\}$ (we know $G \backslash \rho^*$ is expanding)

Adversary Strategy

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

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

Invariant: $G \backslash \rho^*$ is an $(r/2, 3/2)$-boundary expander

Query: If Prover asks for the value of x_i, set $x_i = b$ where

$\rightarrow$ If $\rho_i^* \neq *$ then $b = \rho_i^*$

$\rightarrow$ If $\rho_i^* = *$ then b is an arbitrary value in $\{0,1\}$ (we know $G \backslash \rho^*$ is expanding)

Let μ be the state that results after querying x_i and forgetting some other variables

Adversary Strategy

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

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

Invariant: $G \backslash \rho^*$ is an $(r/2, 3/2)$-boundary expander

Query: If Prover asks for the value of x_i, set $x_i = b$ where

$\rightarrow$ If $\rho_i^* \neq *$ then $b = \rho_i^*$

$\rightarrow$ If $\rho_i^* = *$ then b is an arbitrary value in $\{0,1\}$ (we know $G \backslash \rho^*$ is expanding)

Let μ be the state that results after querying x_i and forgetting some other variables

Restore Expansion:

Adversary Strategy

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

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

Invariant: $G \backslash \rho^*$ is an $(r/2, 3/2)$-boundary expander

Query: If Prover asks for the value of x_i, set $x_i = b$ where

$\rightarrow$ If $\rho_i^* \neq *$ then $b = \rho_i^*$

$\rightarrow$ If $\rho_i^* = *$ then b is an arbitrary value in $\{0,1\}$ (we know $G \backslash \rho^*$ is expanding)

Let μ be the state that results after querying x_i and forgetting some other variables

Restore Expansion: Run RestoreExpansion$(\rho^* \cup \{x_i = b\}, \mathsf{Cl}(\mu))$ to get μ^*

Adversary Strategy

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

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

Invariant: $G \backslash \rho^*$ is an $(r/2, 3/2)$-boundary expander

Query: If Prover asks for the value of x_i, set $x_i = b$ where

$\rightarrow$ If $\rho_i^* \neq *$ then $b = \rho_i^*$

$\rightarrow$ If $\rho_i^* = *$ then b is an arbitrary value in $\{0,1\}$ (we know $G \backslash \rho^*$ is expanding)

Let μ be the state that results after querying x_i and forgetting some other variables

Restore Expansion: Run RestoreExpansion$(\rho^* \cup \{x_i = b\}, \text{Cl}(\mu))$ to get μ^*

— μ^* extends ρ^* to set the variables in $\text{Cl}(\mu)$ consistently with A

Adversary Strategy

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

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

Invariant: $G \backslash \rho^*$ is an $(r/2, 3/2)$-boundary expander

Query: If Prover asks for the value of x_i, set $x_i = b$ where

$\rightarrow$ If $\rho_i^* \neq *$ then $b = \rho_i^*$

$\rightarrow$ If $\rho_i^* = *$ then b is an arbitrary value in $\{0,1\}$ (we know $G \backslash \rho^*$ is expanding)

Let μ be the state that results after querying x_i and forgetting some other variables

Restore Expansion: Run RestoreExpansion$(\rho^* \cup \{x_i = b\}, \text{Cl}(\mu))$ to get μ^*

– μ^* extends ρ^* to set the variables in $\text{Cl}(\mu)$ consistently with A

Forget from μ^* the variables not in $\text{Cl}(\mu)$

Adversary Strategy

Uses $O(w)$ queries to A

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

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

Invariant: $G \backslash \rho^*$ is an $(r/2, 3/2)$-boundary expander

Query: If Prover asks for the value of x_i, set $x_i = b$ where

$\rightarrow$ If $\rho_i^* \neq *$ then $b = \rho_i^*$

$\rightarrow$ If $\rho_i^* = *$ then b is an arbitrary value in $\{0,1\}$ (we know $G \backslash \rho^*$ is expanding)

Let μ be the state that results after querying x_i and forgetting some other variables

Restore Expansion: Run RestoreExpansion$(\rho^* \cup \{x_i = b\}, \text{Cl}(\mu))$ to get μ^*

– μ^* extends ρ^* to set the variables in $\text{Cl}(\mu)$ consistently with A

Forget from μ^* the variables not in $\text{Cl}(\mu)$

Adversary Strategy

Uses $O(w)$ queries to $A \Longrightarrow$ Adversary can continue the game for $\Omega(d/w)$ rounds

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

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

Invariant: $G \backslash \rho^*$ is an $(r/2, 3/2)$-boundary expander

Query: If Prover asks for the value of x_i, set $x_i = b$ where

$\rightarrow$ If $\rho_i^* \neq *$ then $b = \rho_i^*$

$\rightarrow$ If $\rho_i^* = *$ then b is an arbitrary value in $\{0,1\}$ (we know $G \backslash \rho^*$ is expanding)

Let μ be the state that results after querying x_i and forgetting some other variables

Restore Expansion: Run RestoreExpansion$(\rho^* \cup \{x_i = b\}, \mathsf{Cl}(\mu))$ to get μ^*

— μ^* extends ρ^* to set the variables in $\mathsf{Cl}(\mu)$ consistently with A

Forget from μ^* the variables not in $\mathsf{Cl}(\mu)$

Depth Condensation Theorem

Depth Condensation Theorem:

Let G be an $(r,2)$-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

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

Open Questions

Q . Supercritical size/depth tradeoffs for monotone circuits?

Open Questions

Q . Supercritical size/depth tradeoffs for monotone circuits?

$\rightarrow$ For any F, a Cutting Planes proof of F implies a monotone circuits computing an associated function f_F with the same topology [P96, HP17, FPPR17].

Open Questions

Q . Supercritical size/depth tradeoffs for monotone circuits?

$\rightarrow$ For any F, a Cutting Planes proof of 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 Questions

Q . Supercritical size/depth tradeoffs for monotone circuits?

$\rightarrow$ For any F, a Cutting Planes proof of 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 Questions

Q. Supercritical size/depth tradeoffs for monotone circuits?

$\rightarrow$ For any F, a Cutting Planes proof of 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

Q. Does every formula F on m clauses have a Resolution proof of depth $O(m)$?

Open Questions

Q. Supercritical size/depth tradeoffs for monotone circuits?

$\rightarrow$ For any F, a Cutting Planes proof of 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

Q. Does every formula F on m clauses have a Resolution proof of depth $O(m)$?

$\rightarrow$ If no, then supercritical size/depth tradeoffs for monotone circuits follow from the lifting theorem of [GGKS18].