Integer Programming and IP Proof Systems Part 2

Noah Fleming
University of California, San Diego

Recap of Last Time

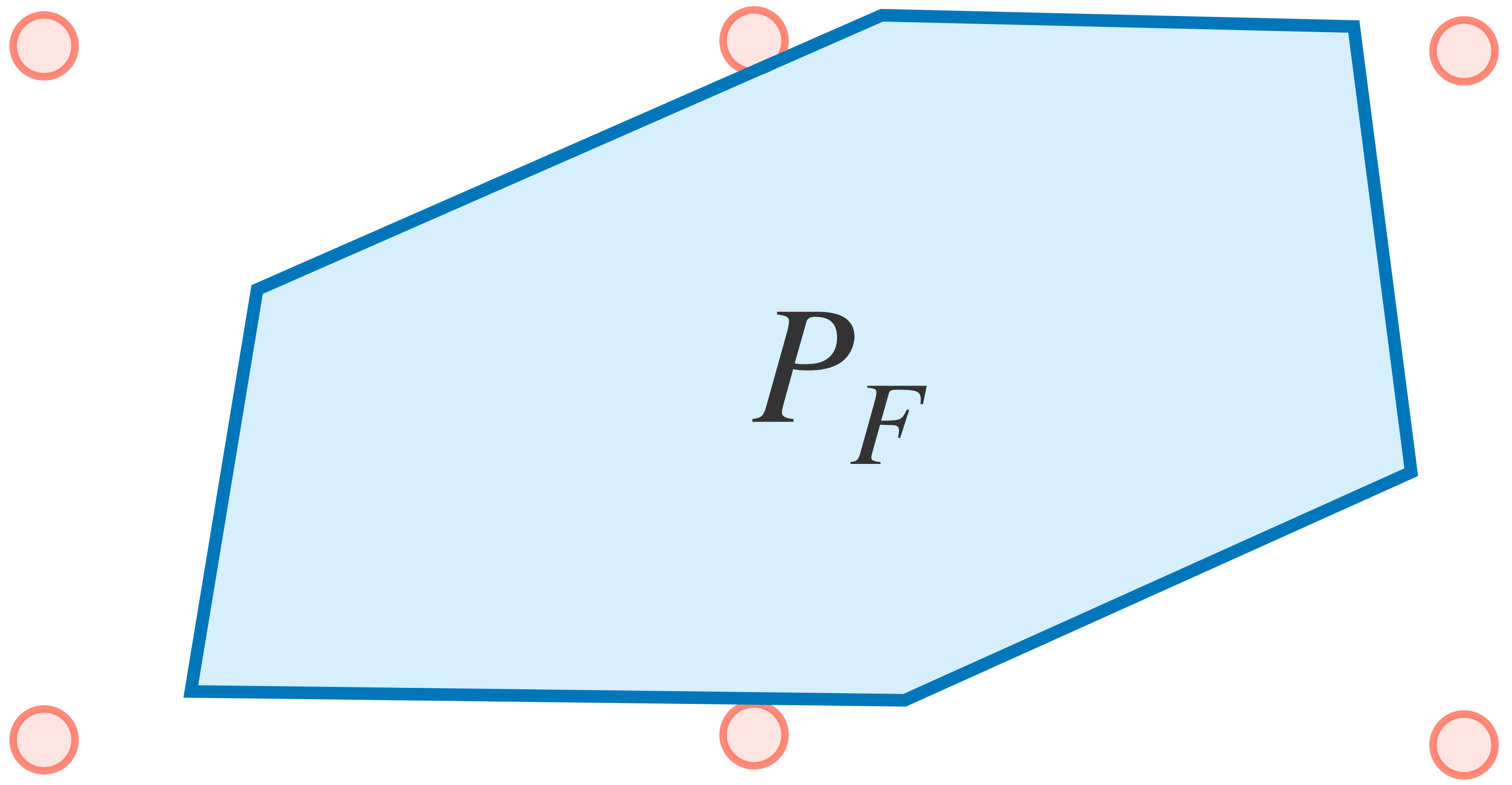
Encode unsatisfiable CNF formulas F as polytopes (systems of linear inequalities) P_F with no integer points

$$F = C_1 \wedge \ldots \wedge C_m \implies P_F = \{x : Ax \geq b\}$$

For each $C_i = \bigvee_{i \in I} x_i \vee \bigvee_{j \in J} \neg x_j$

Include in $Ax \geq b$

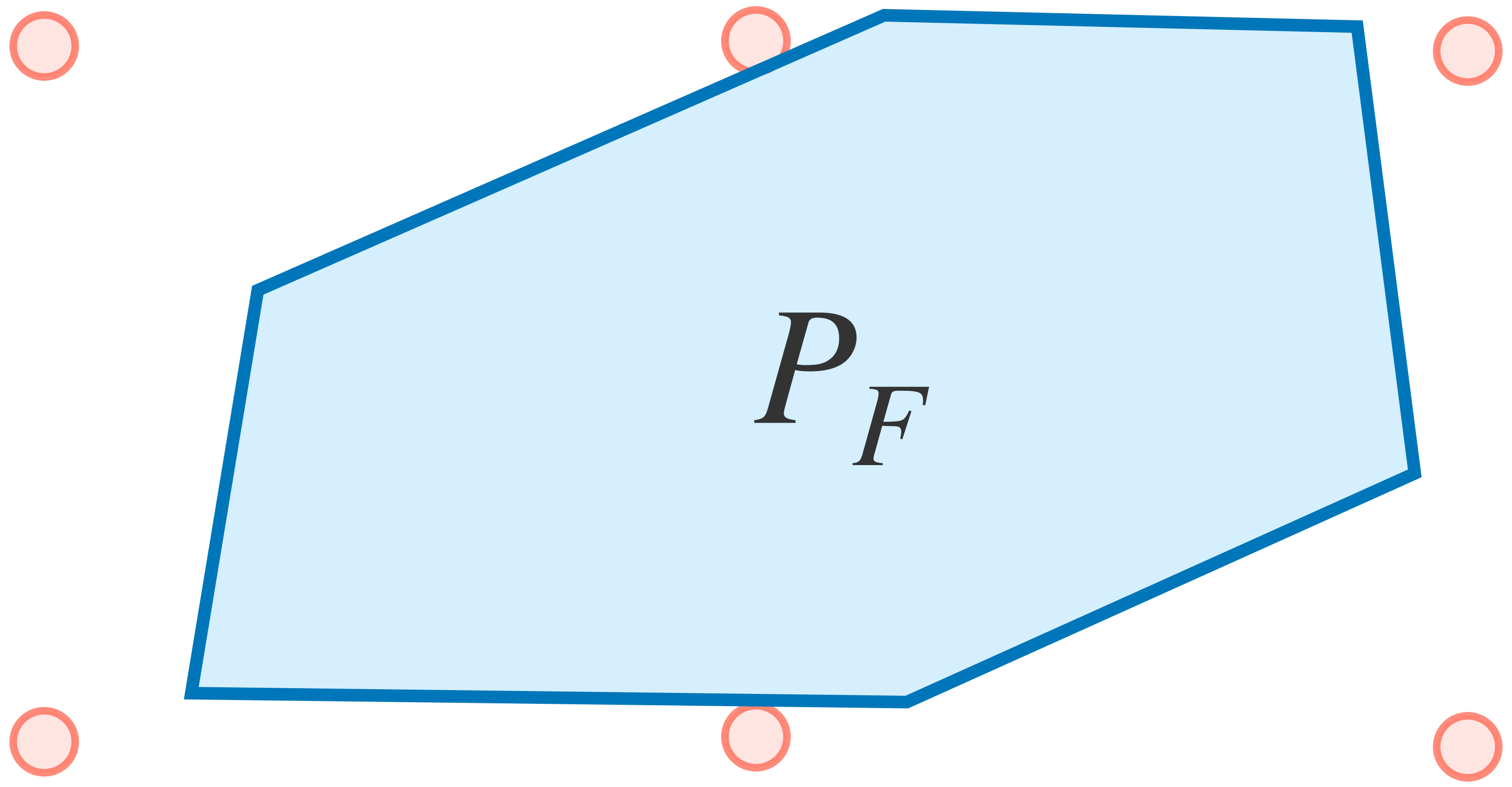
$$\sum_{i \in I} x_i + \sum_{j \in J} (1 - x_j) \geq 1 \quad \text{and} \quad x_i \geq 0, -x_i \geq -1$$



Recap of Last Time

Encode unsatisfiable CNF formulas F as polytopes (systems of linear inequalities) P_F with no integer points

$$F = C_1 \wedge \ldots \wedge C_m \implies P_F = \{x : Ax \geq b\}$$

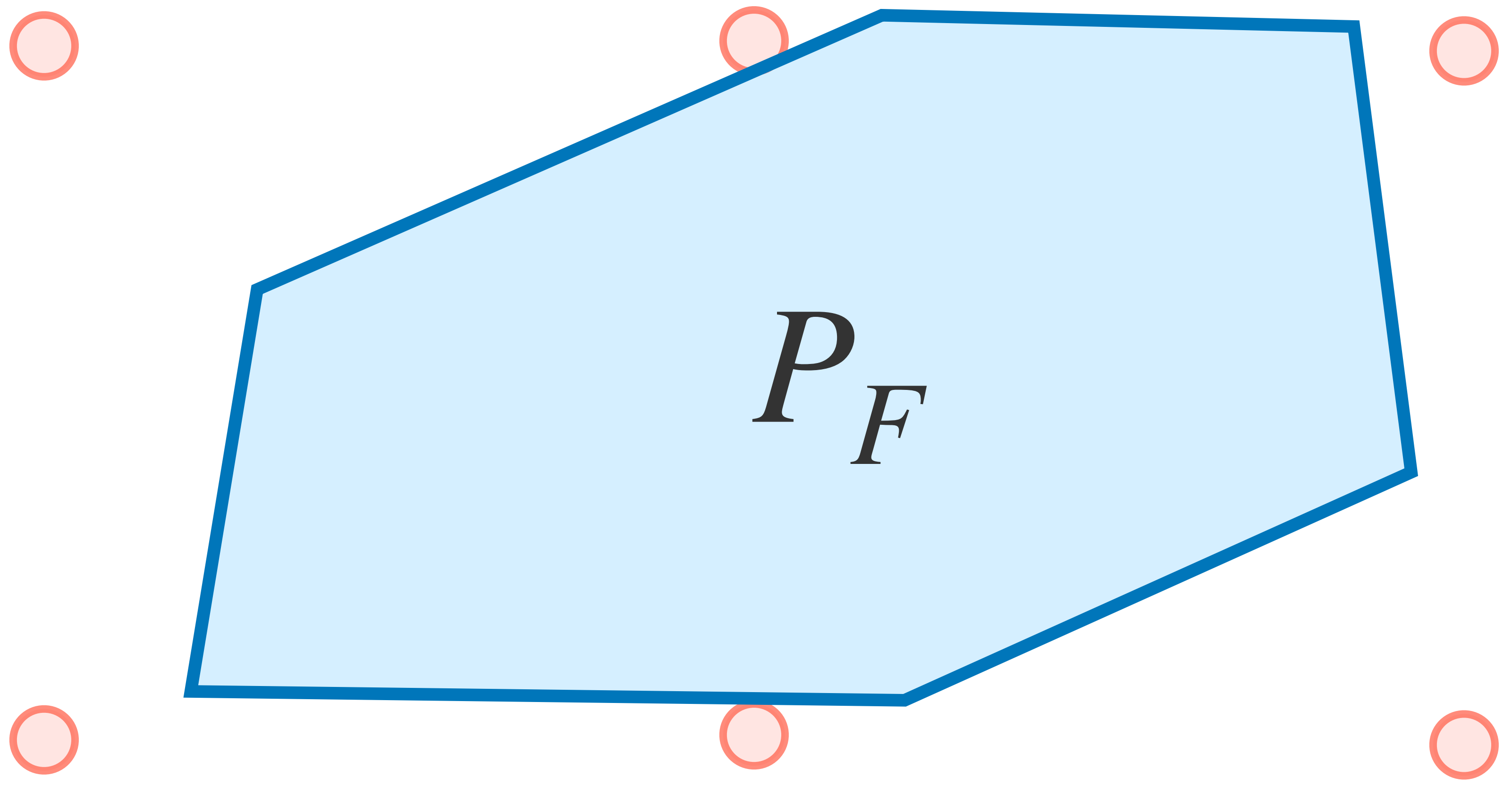


Recap of Last Time

Encode unsatisfiable CNF formulas F as polytopes (systems of linear inequalities) P_F with no integer points

$$F = C_1 \wedge \ldots \wedge C_m \implies P_F = \{x : Ax \geq b\}$$

Consider proof systems for proving that a polytope does not contain integer points



Recap of Last Time

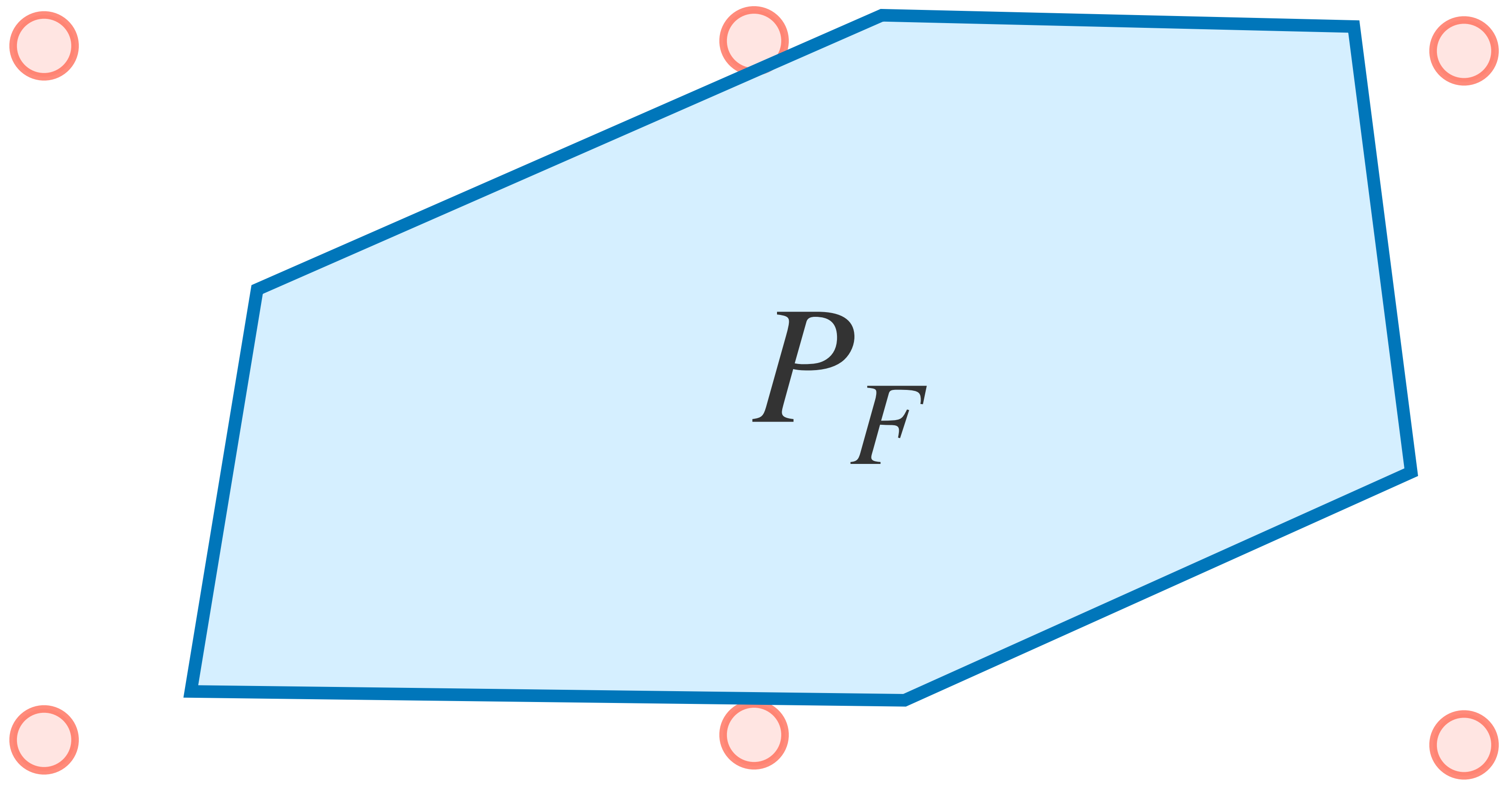
Encode unsatisfiable CNF formulas F as polytopes (systems of linear inequalities) P_F with no integer points

$$F = C_1 \land \ldots \land C_m \implies P_F = \{x : Ax \geq b\}$$

Consider proof systems for proving that a polytope does not contain integer points

$\rightarrow$ Cutting Planes — captures Cutting Planes method

$\rightarrow$ Stabbing Planes — captures branch-and-cut

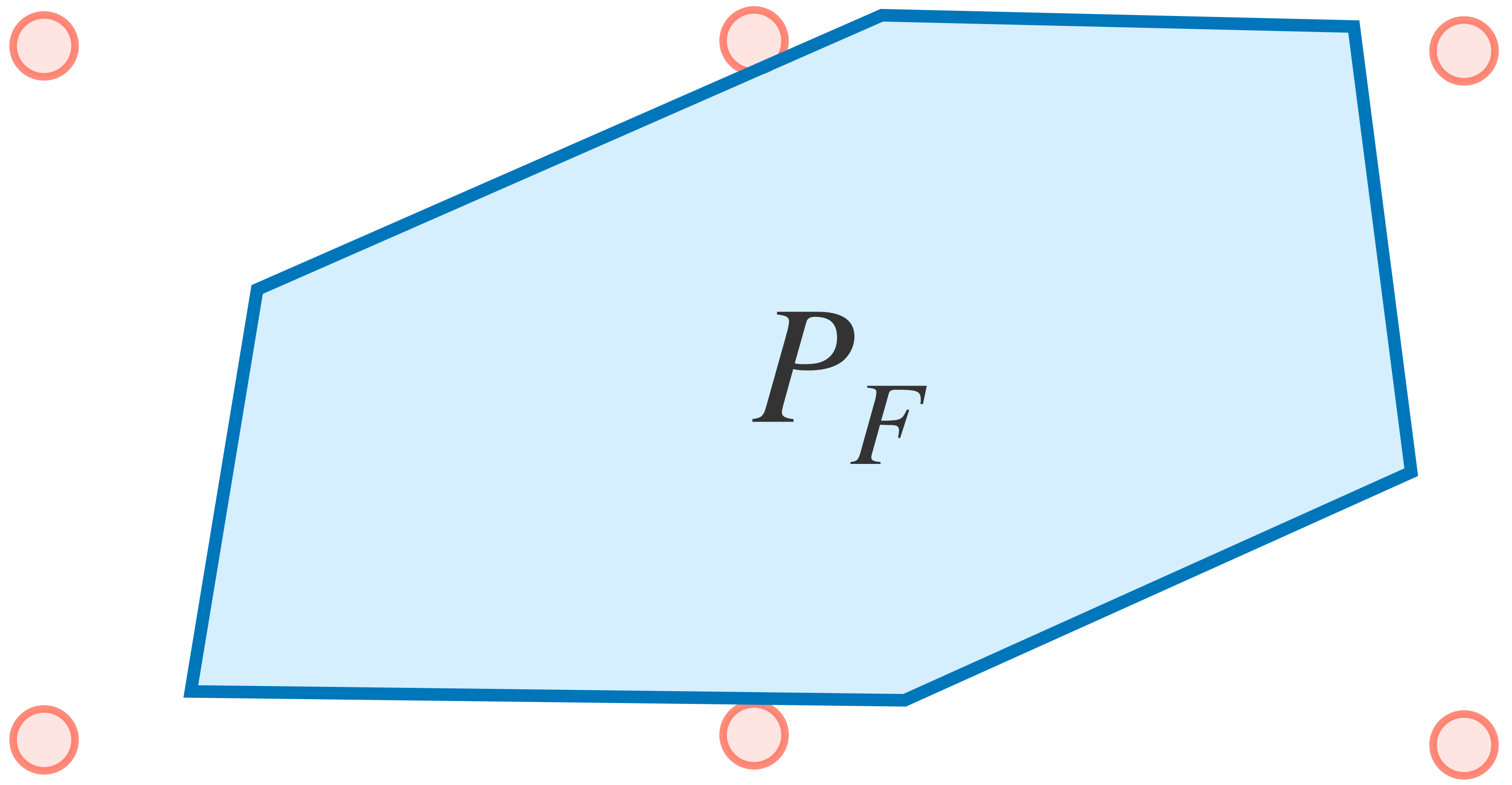


Recap of Last Time

Encode unsatisfiable CNF formulas F as polytopes (systems of linear inequalities) P_F with no integer points

$$F = C_1 \wedge \ldots \wedge C_m \implies P_F = \{x : Ax \geq b\}$$

Consider proof systems for proving that a polytope does not contain integer points



→ Cutting Planes — captures Cutting Planes method

→ Stabbing Planes — captures branch-and-cut

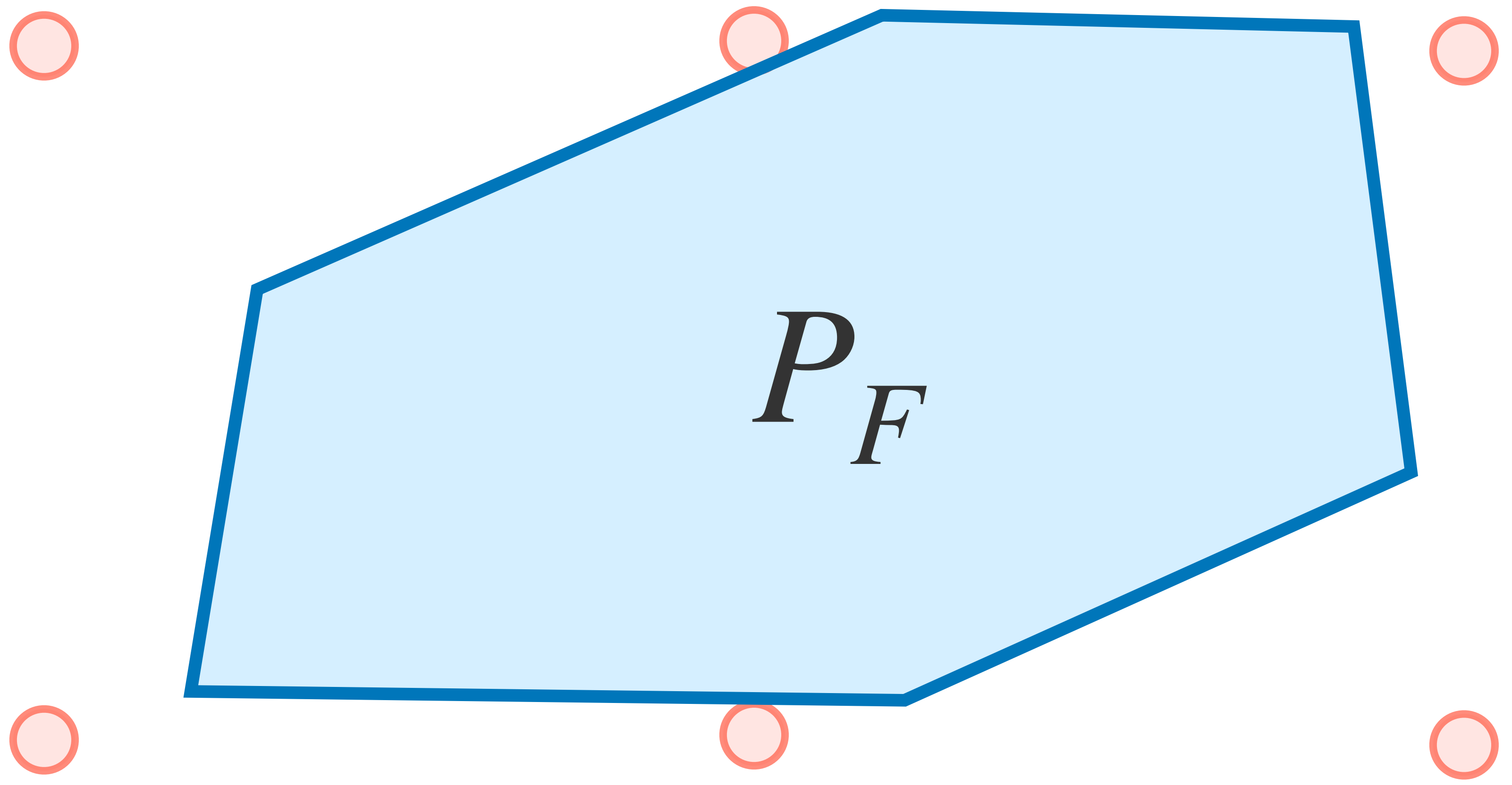
Last time: Cutting Planes $\leq$ Stabbing Planes

Recap of Last Time

Encode unsatisfiable CNF formulas F as polytopes (systems of linear inequalities) P_F with no integer points

$$F = C_1 \wedge \ldots \wedge C_m \implies P_F = \{x : Ax \geq b\}$$

Consider proof systems for proving that a polytope does not contain integer points



→ Cutting Planes — captures Cutting Planes method

→ Stabbing Planes — captures branch-and-cut

Last time: Cutting Planes $\leq$ Stabbing Planes

Thm [FGI+21]

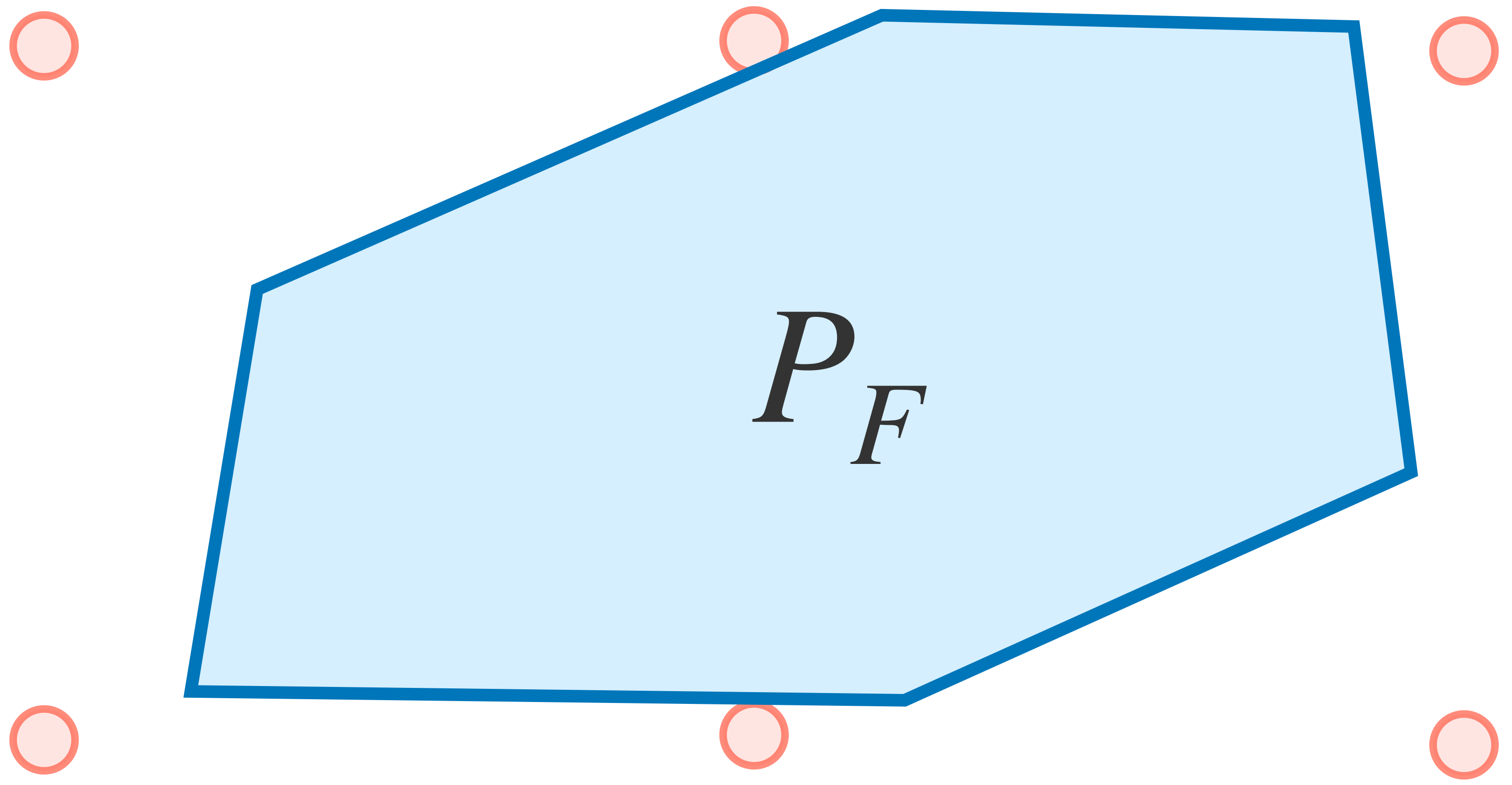
Any Stabbing Planes proof with coefficients at most $2^{\text{polylog } n}$ (SP*) can be translated into Cutting Planes with a quasi-polynomial blow-up in the size.

Recap of Last Time

Encode unsatisfiable CNF formulas F as polytopes (systems of linear inequalities) P_F with no integer points

$$F = C_1 \wedge \ldots \wedge C_m \implies P_F = \{x : Ax \geq b\}$$

Consider proof systems for proving that a polytope does not contain integer points



→ Cutting Planes — captures Cutting Planes method

→ Stabbing Planes — captures branch-and-cut

Last time: Cutting Planes $\leq$ Stabbing Planes

Thm [FGI+21]

Any Stabbing Planes proof with coefficients at most $2^{\text{polylog } n}$ (SP*) can be translated into Cutting Planes with a quasi-polynomial blow-up in the size.

$\implies$ Can prove bounds on branch-and-cut by proving bounds on Cutting Planes

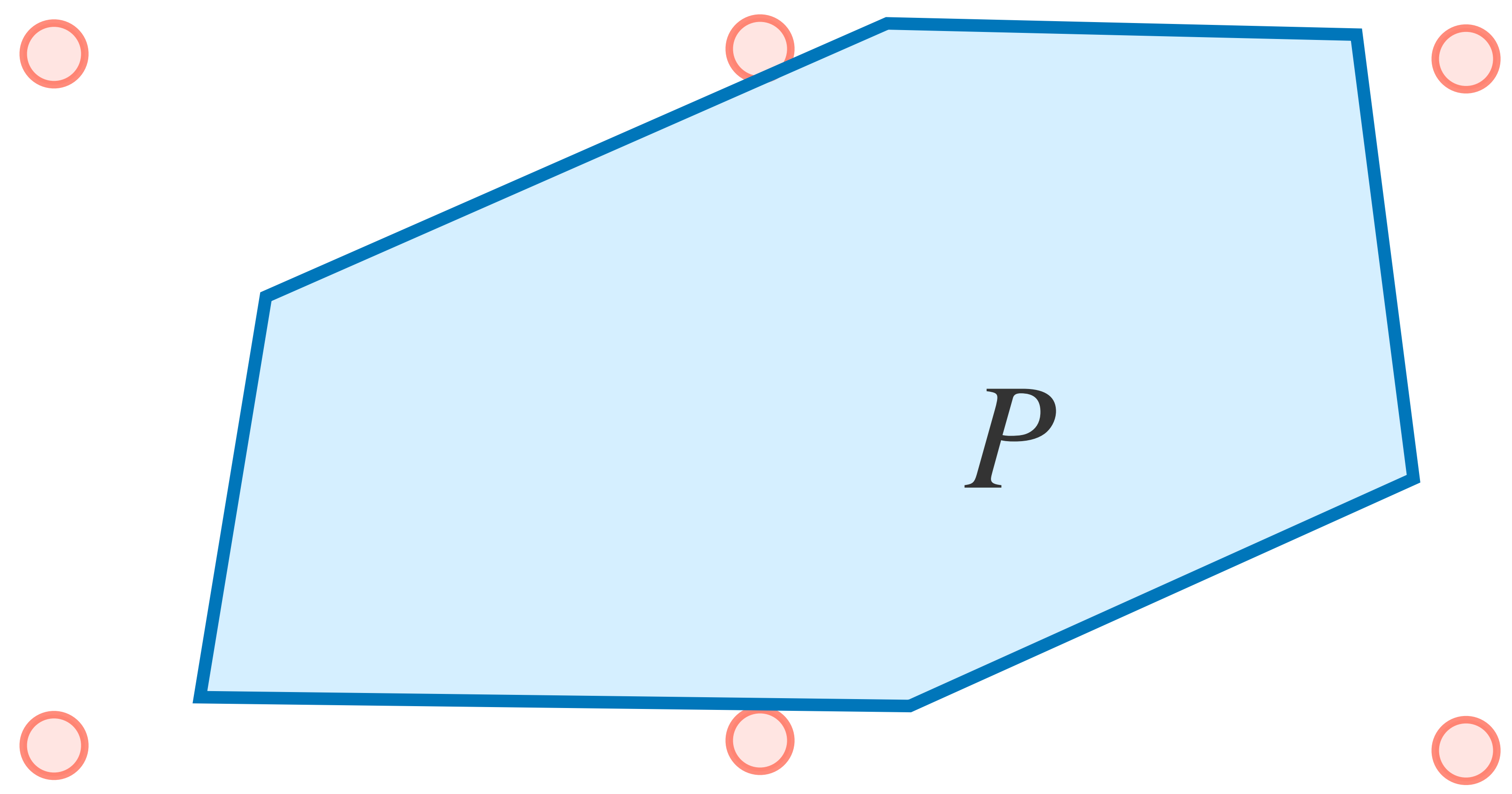
Today

Lower bounds on the size of Cutting Planes proofs!

Let’s recall Cutting Planes…

Cutting Planes Proofs

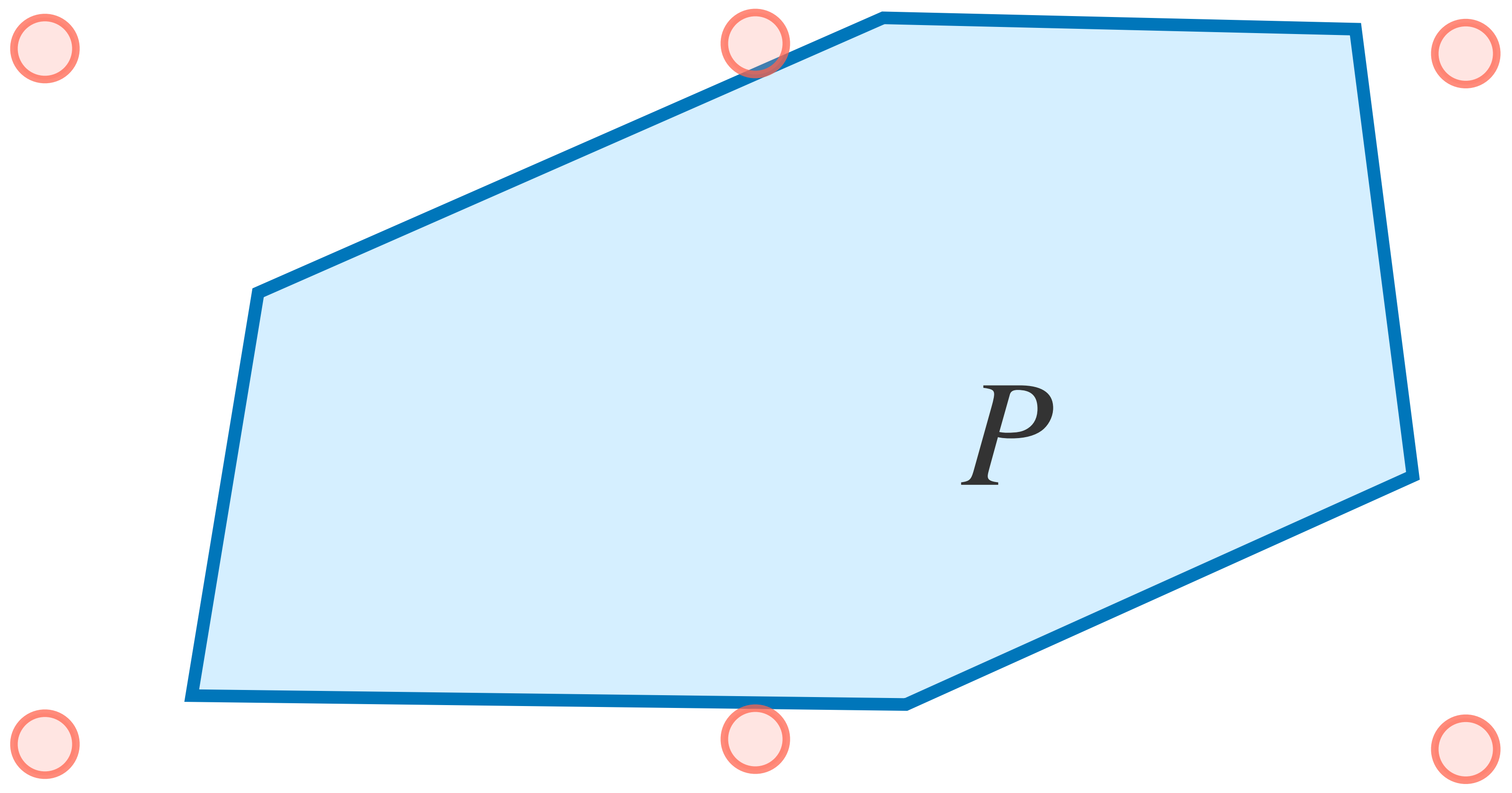
Suppose $Ax \geq b$ has **no integer solutions**



Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

$\rightarrow$ **Prove** this fact using cutting planes!



Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

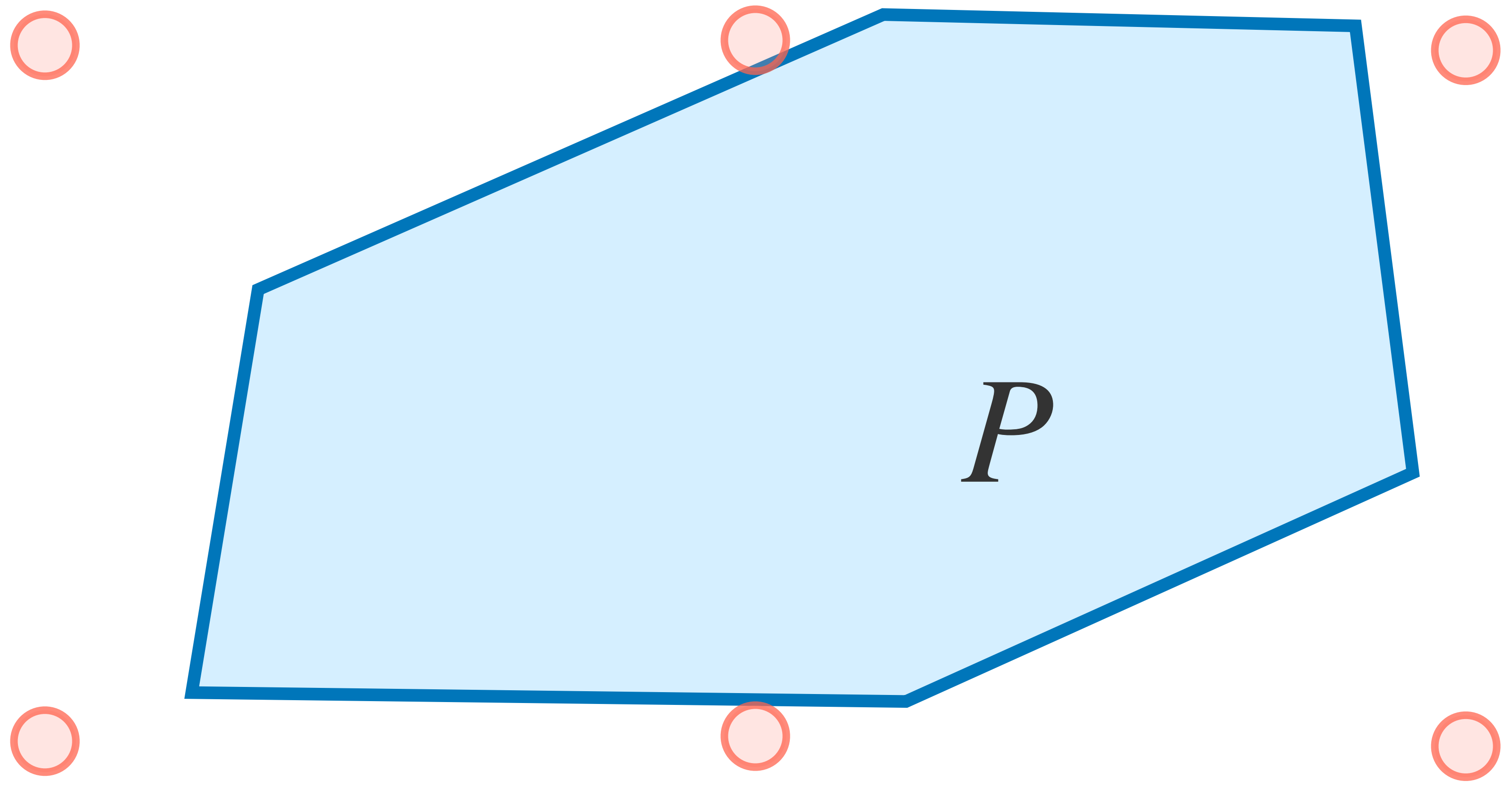
$\rightarrow$ **Prove** this fact using cutting planes!

Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

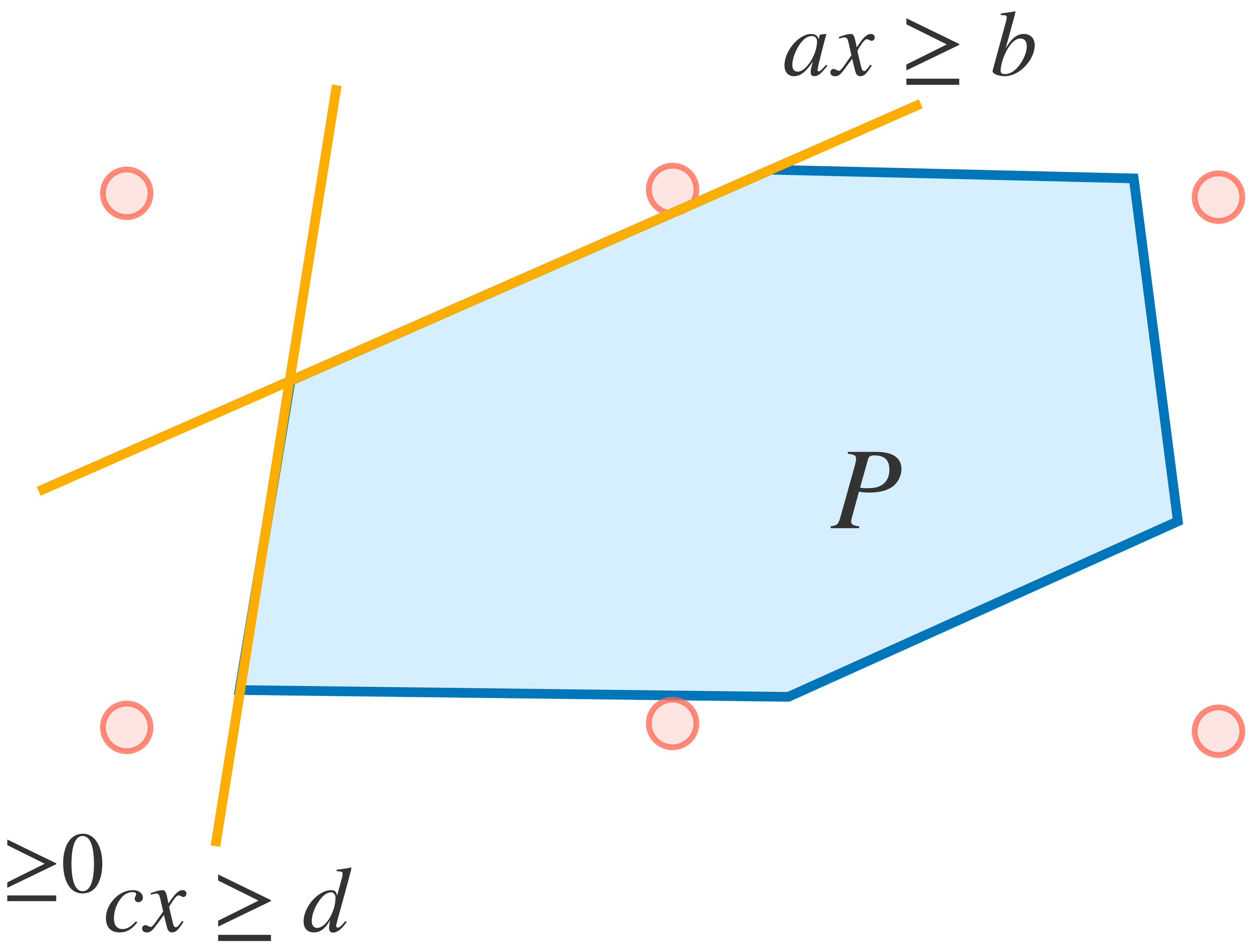
$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$



Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

$\rightarrow$ **Prove** this fact using cutting planes!



Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$

Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

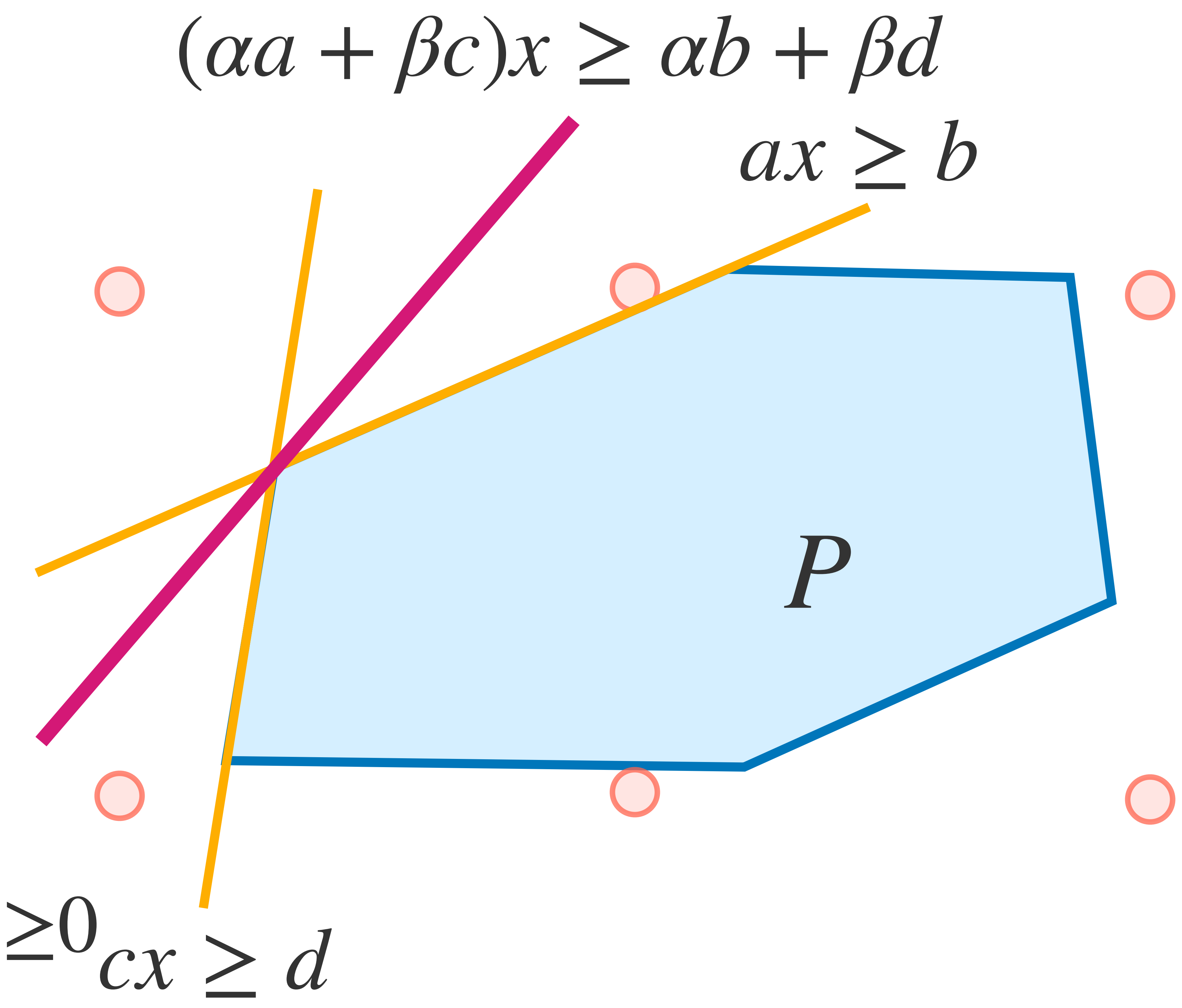
$\rightarrow$ **Prove** this fact using cutting planes!

Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$



Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

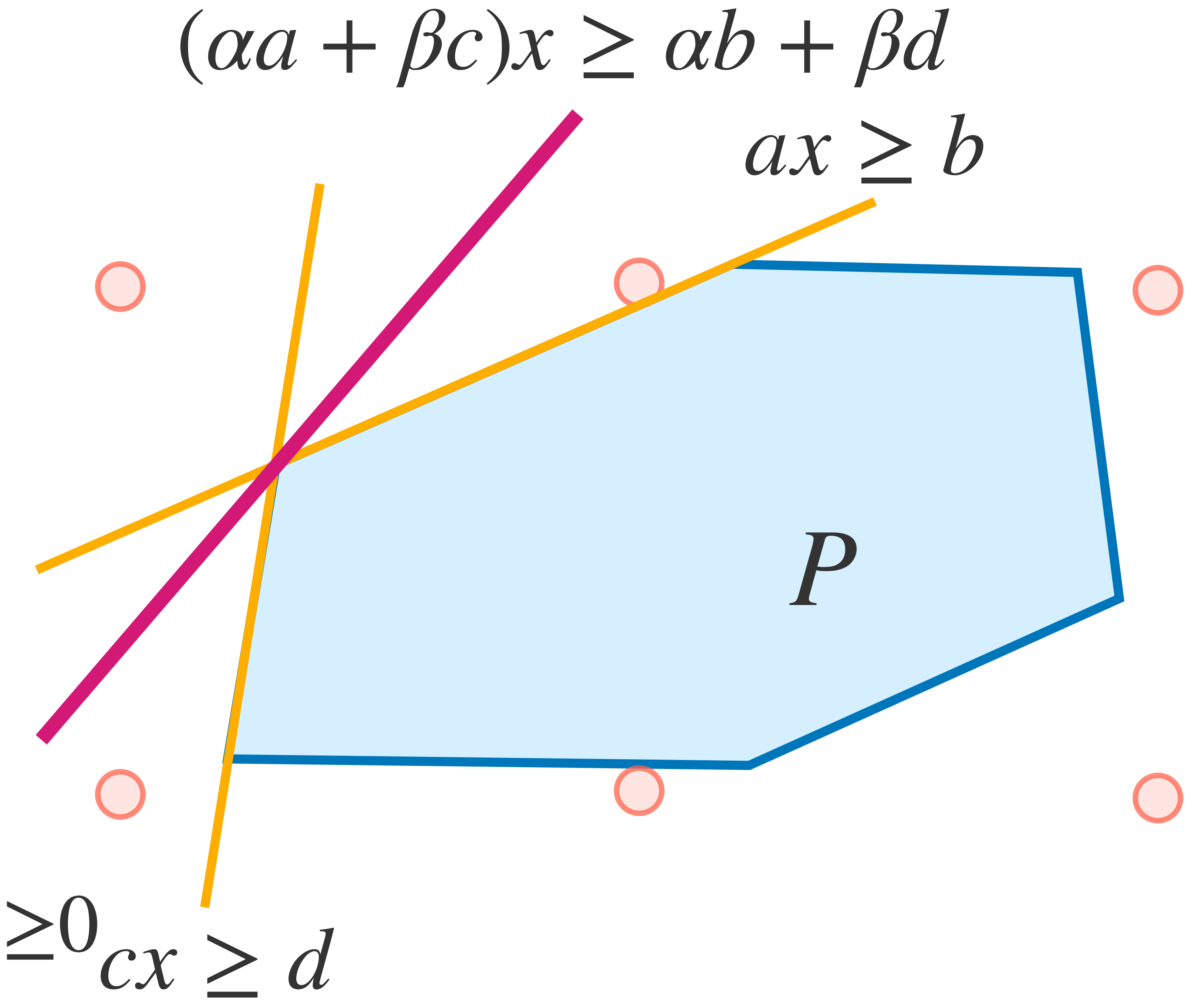
$\rightarrow$ **Prove** this fact using cutting planes!

Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$

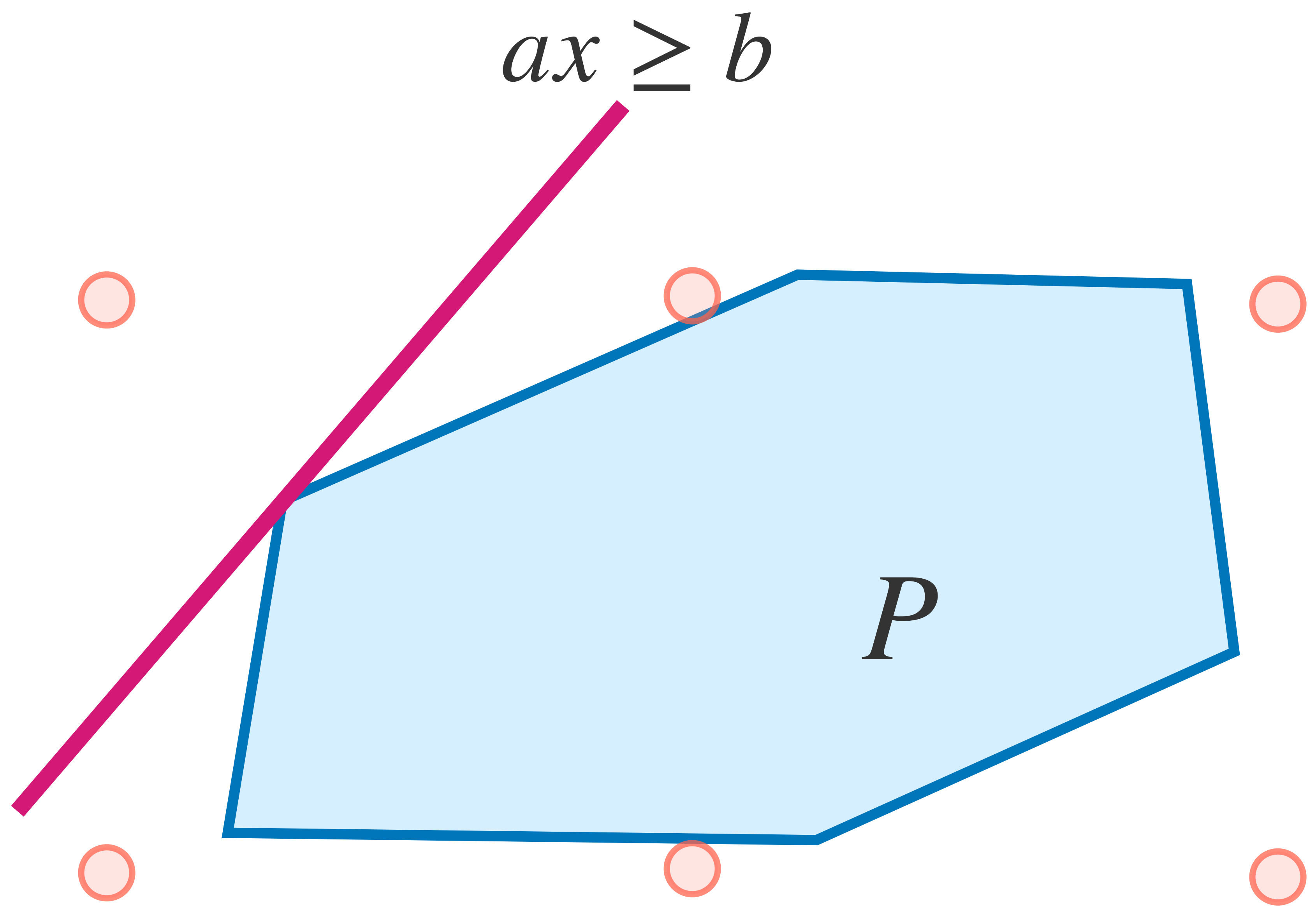


Preserves **all** points in P

Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

$\rightarrow$ **Prove** this fact using cutting planes!



Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$

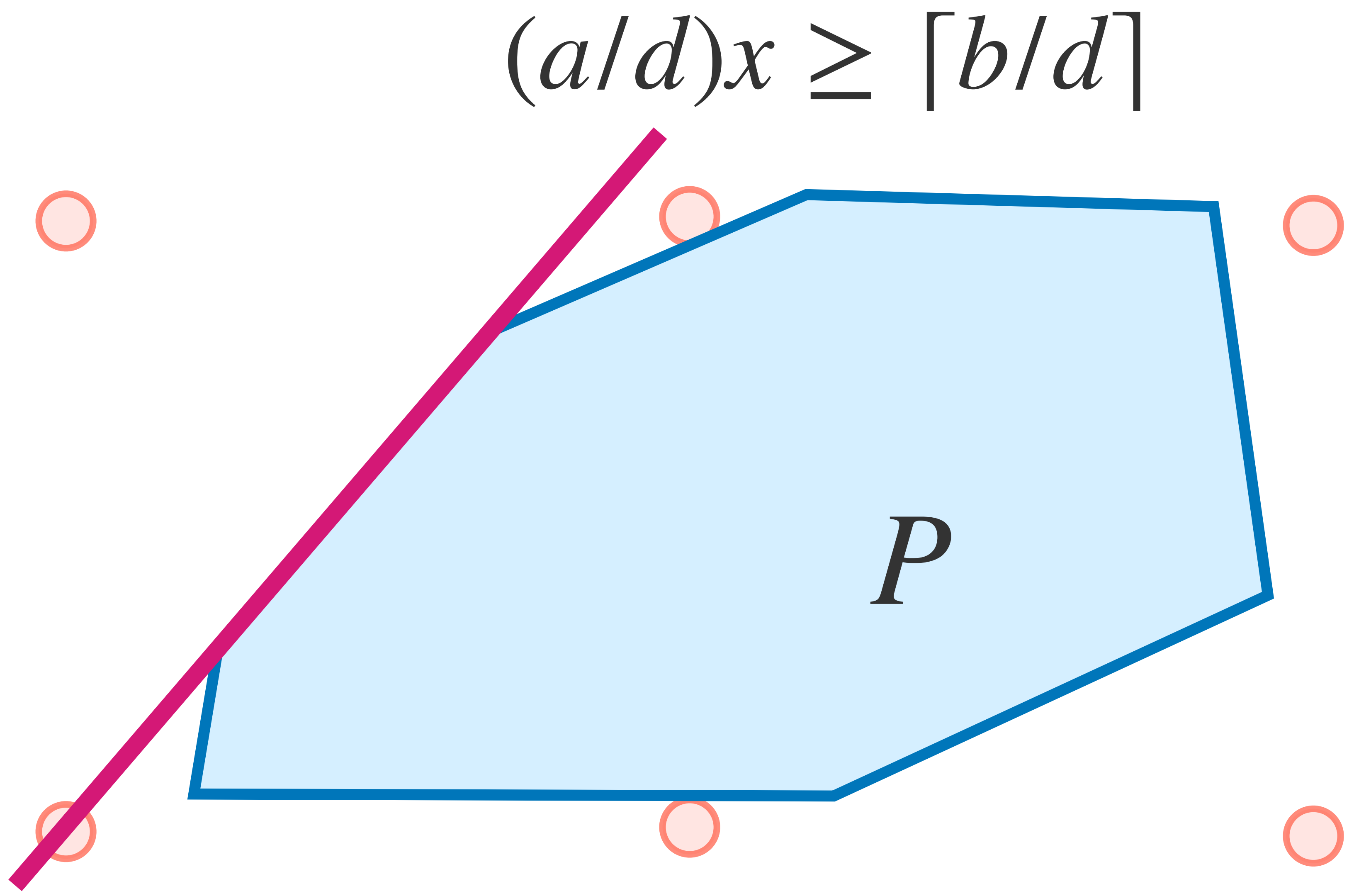
- **Cut:**

$$\frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil},$$ if $d \in \mathbb{Z}^{\geq 0}$ divides a

Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

$\rightarrow$ **Prove** this fact using cutting planes!



Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$

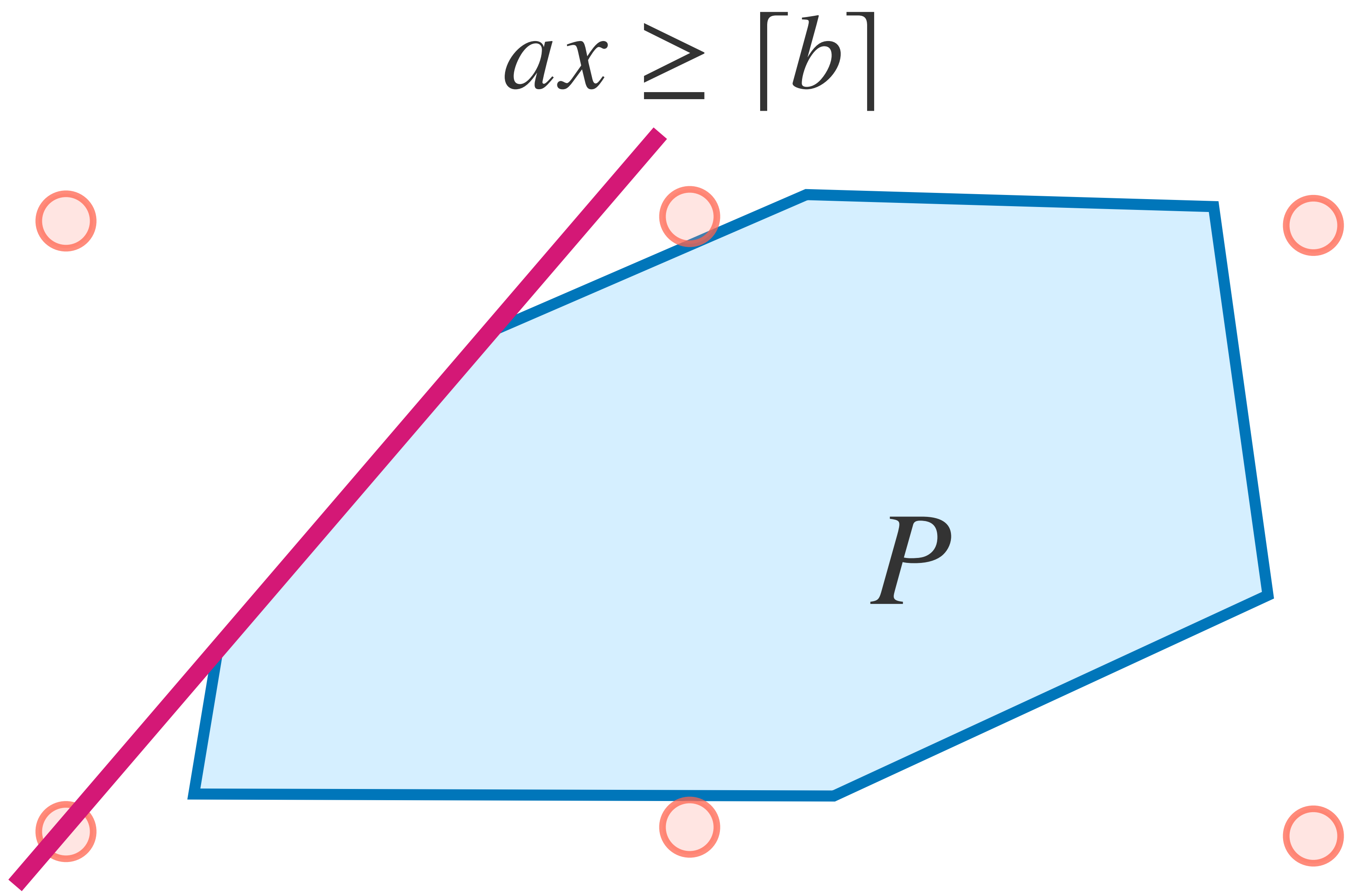
- **Cut:**

$$\frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil},$$ if $d \in \mathbb{Z}^{\geq 0}$ divides a

Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

$\rightarrow$ **Prove** this fact using cutting planes!



Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$

- **Cut:**

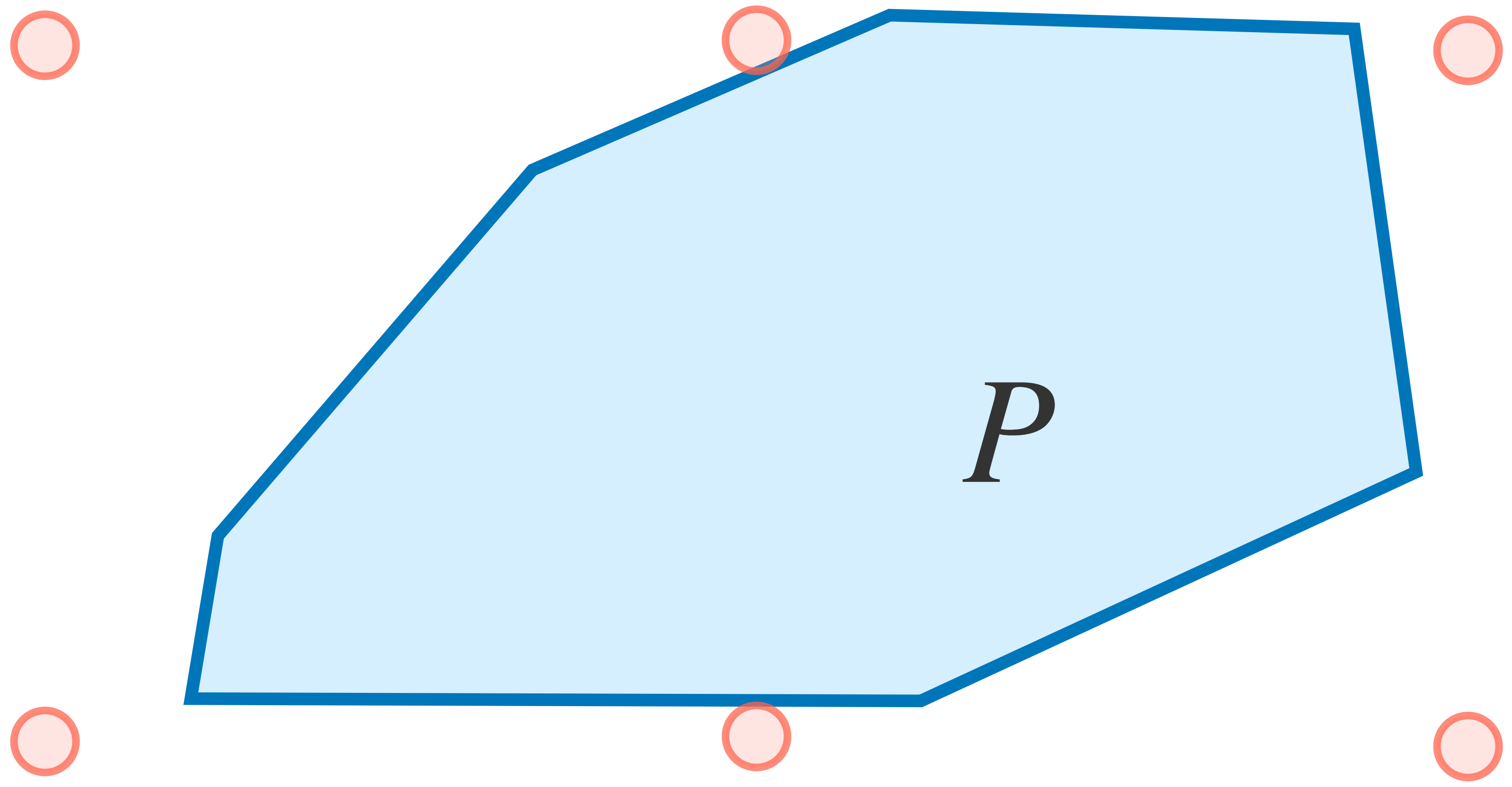
$$\frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil}, \text{ if } d \in \mathbb{Z}^{\geq 0} \text{ divides } a$$

Preserves **integer** points in P

Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

$\rightarrow$ **Prove** this fact using cutting planes!



Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$

- **Cut:**

$$\frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil}, \text{ if } d \in \mathbb{Z}^{\geq 0} \text{ divides } a$$

Cutting Planes Proof

Derivation of $0 \geq 1$ from $Ax \geq b$

- equivalently, the **empty polytope**

Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

$\rightarrow$ **Prove** this fact using cutting planes!

Rules

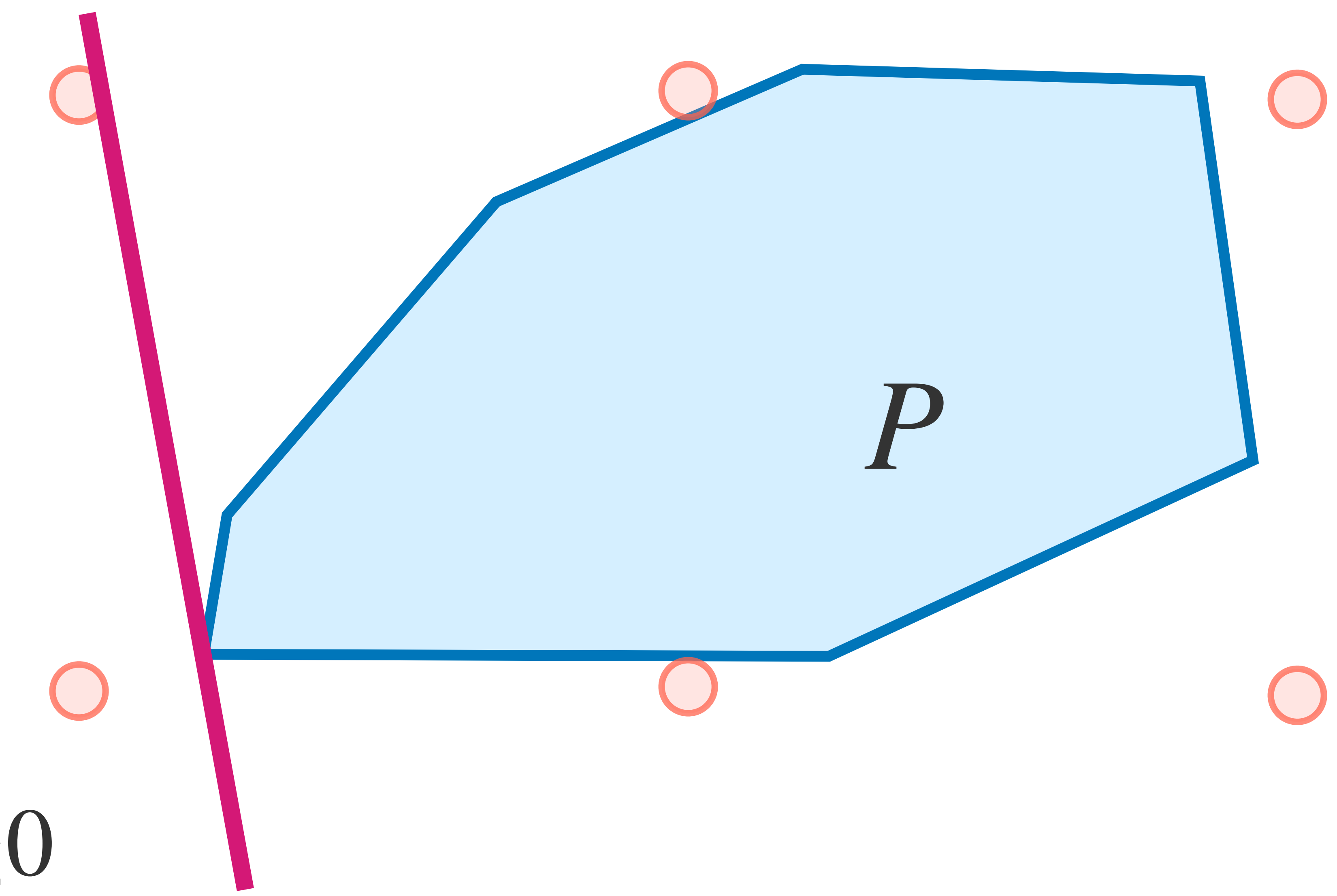
Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$

- **Cut:**

$$\frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil}, \text{ if } d \in \mathbb{Z}^{\geq 0} \text{ divides } a$$



Cutting Planes Proof

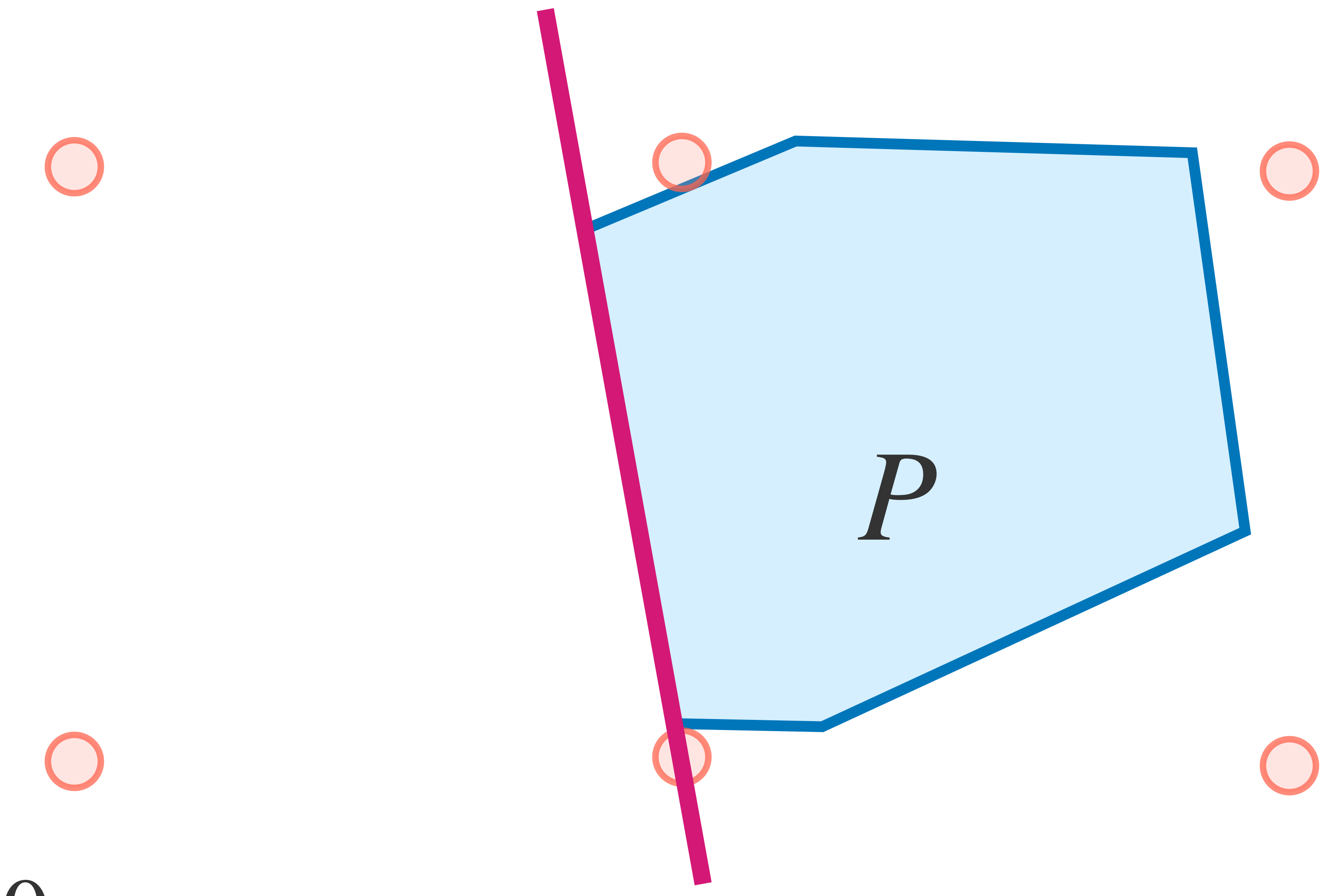
Derivation of $0 \geq 1$ from $Ax \geq b$

- equivalently, the **empty polytope**

Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

$\rightarrow$ **Prove** this fact using cutting planes!



Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$

- **Cut:**

$$\frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil}, \text{ if } d \in \mathbb{Z}^{\geq 0} \text{ divides } a$$

Cutting Planes Proof

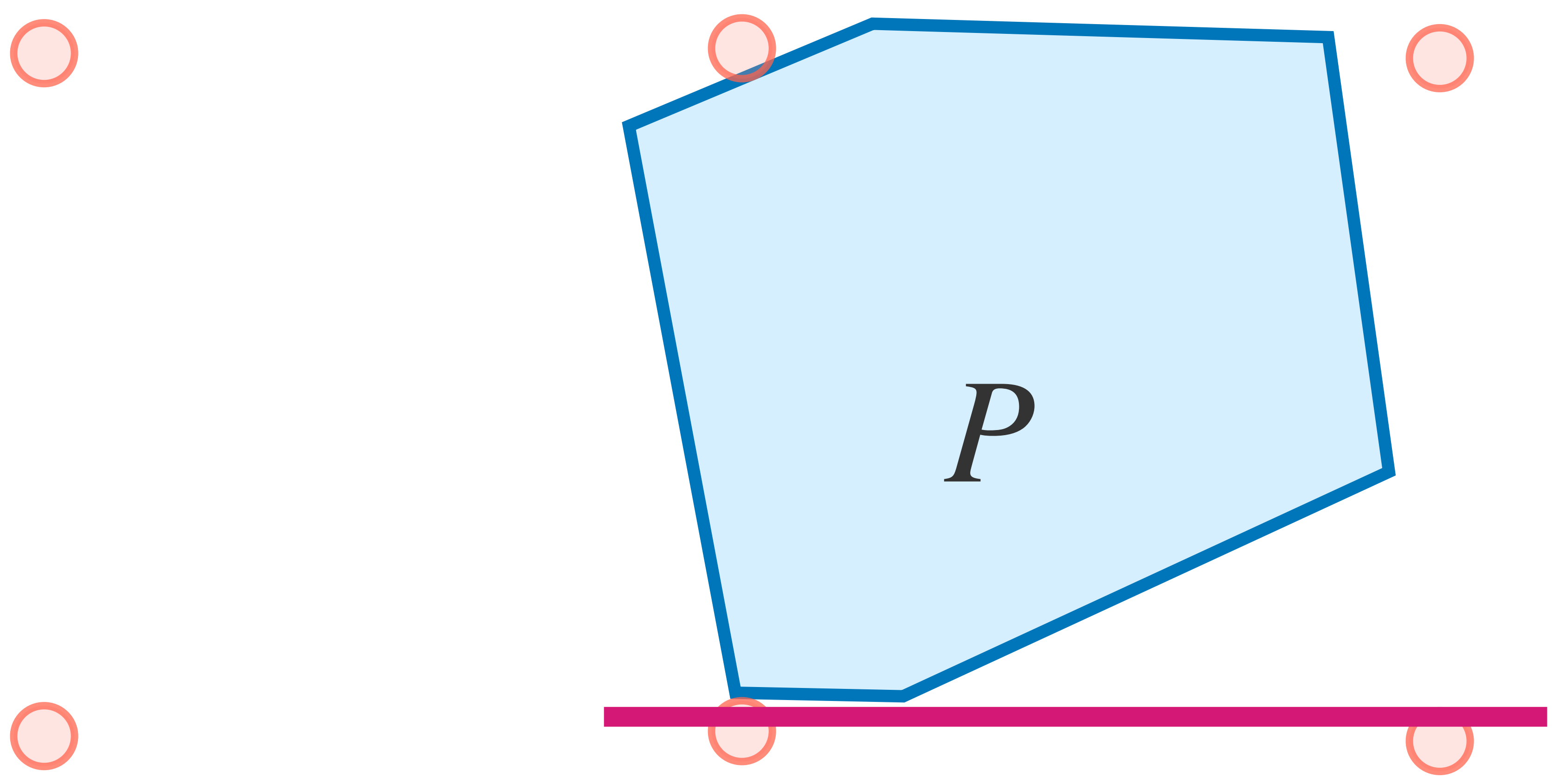
Derivation of $0 \geq 1$ from $Ax \geq b$

- equivalently, the **empty polytope**

Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

$\rightarrow$ **Prove** this fact using cutting planes!



Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$

- **Cut:**

$$\frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil}, \text{ if } d \in \mathbb{Z}^{\geq 0} \text{ divides } a$$

Cutting Planes Proof

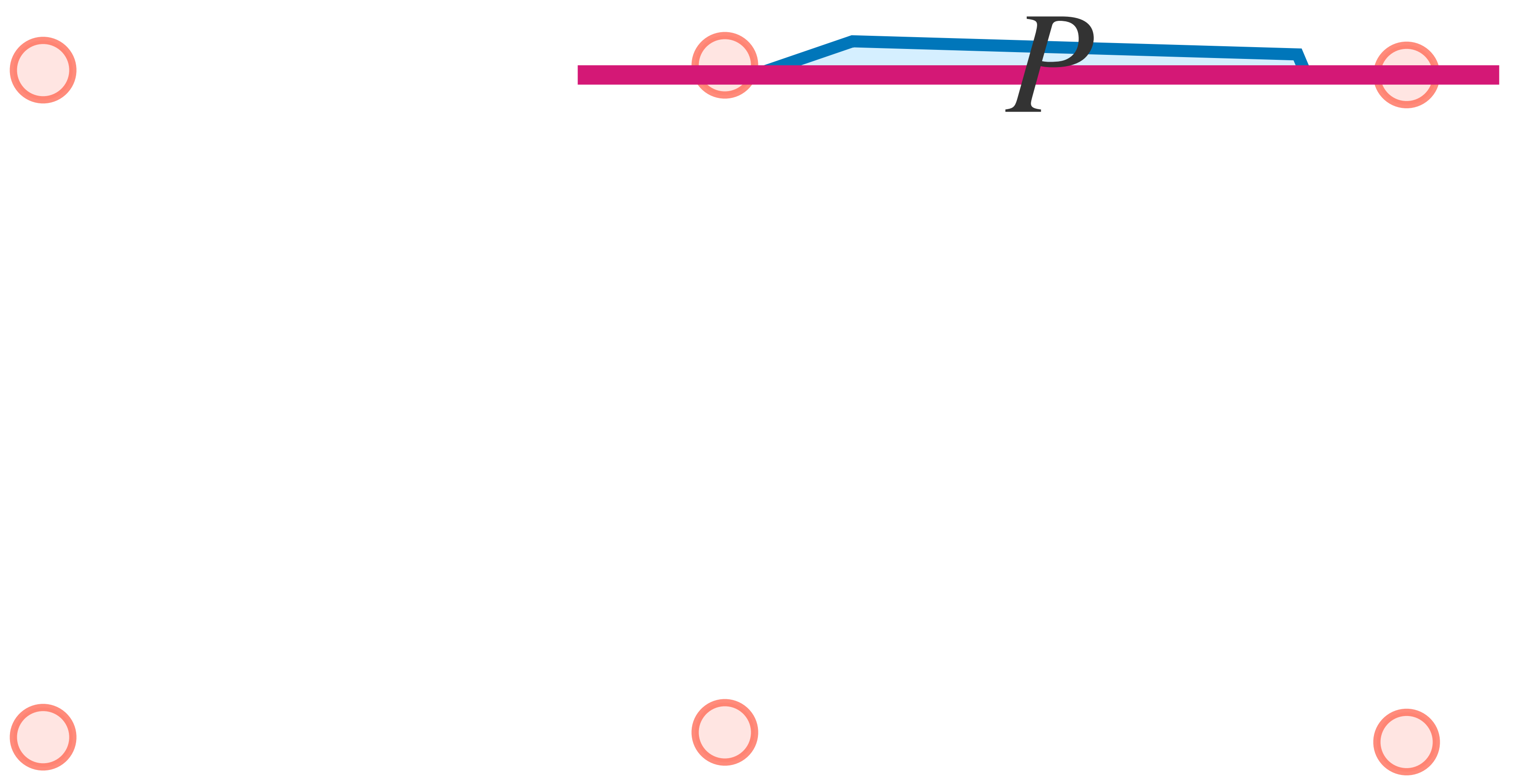
Derivation of $0 \geq 1$ from $Ax \geq b$

- equivalently, the **empty polytope**

Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

$\rightarrow$ **Prove** this fact using cutting planes!



Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$

- **Cut:**

$$\frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil}, \text{ if } d \in \mathbb{Z}^{\geq 0} \text{ divides } a$$

Cutting Planes Proof

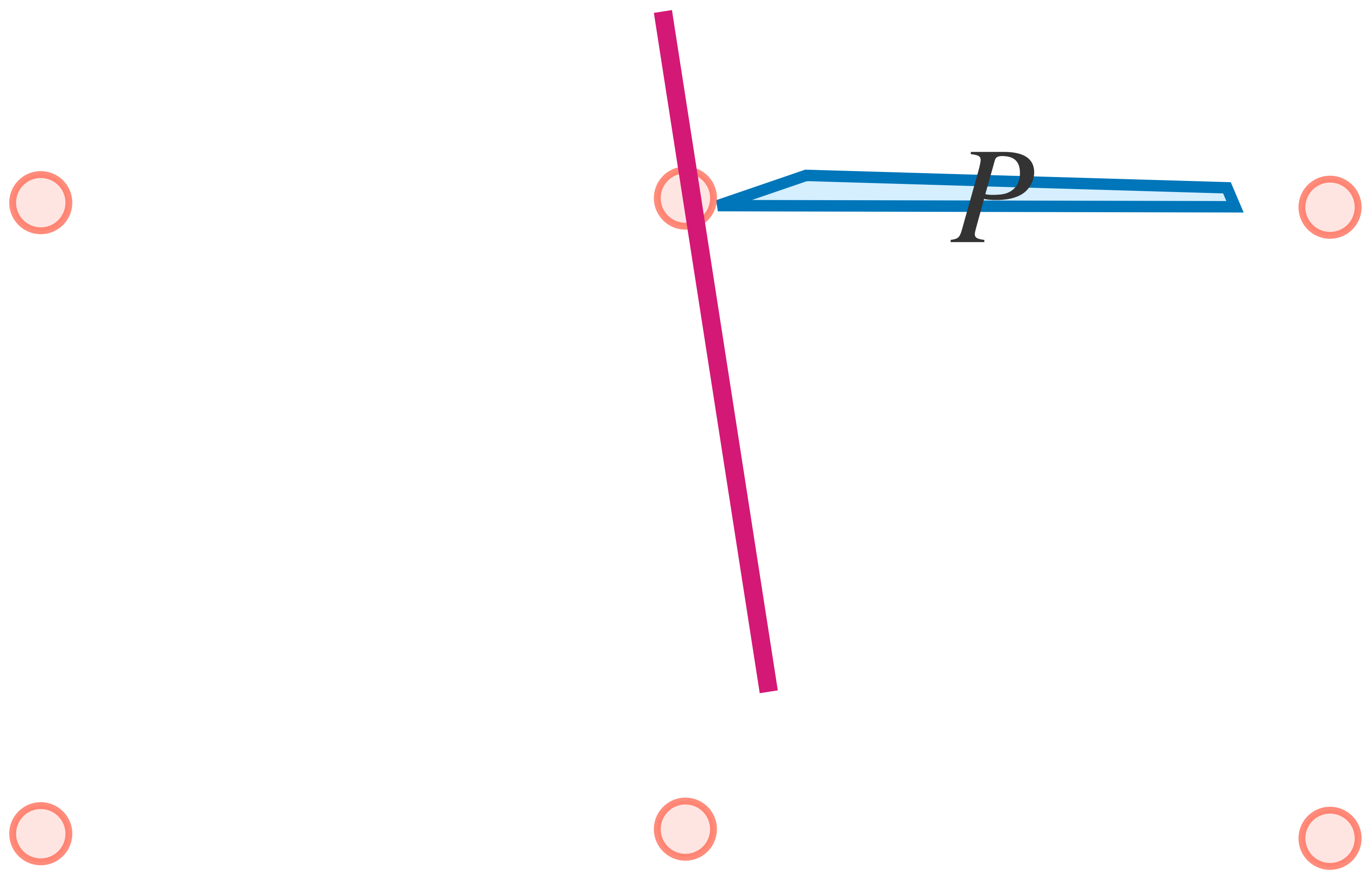
Derivation of $0 \geq 1$ from $Ax \geq b$

- equivalently, the **empty polytope**

Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

$\rightarrow$ **Prove** this fact using cutting planes!



Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$

- **Cut:**

$$\frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil}, \text{ if } d \in \mathbb{Z}^{\geq 0} \text{ divides } a$$

Cutting Planes Proof

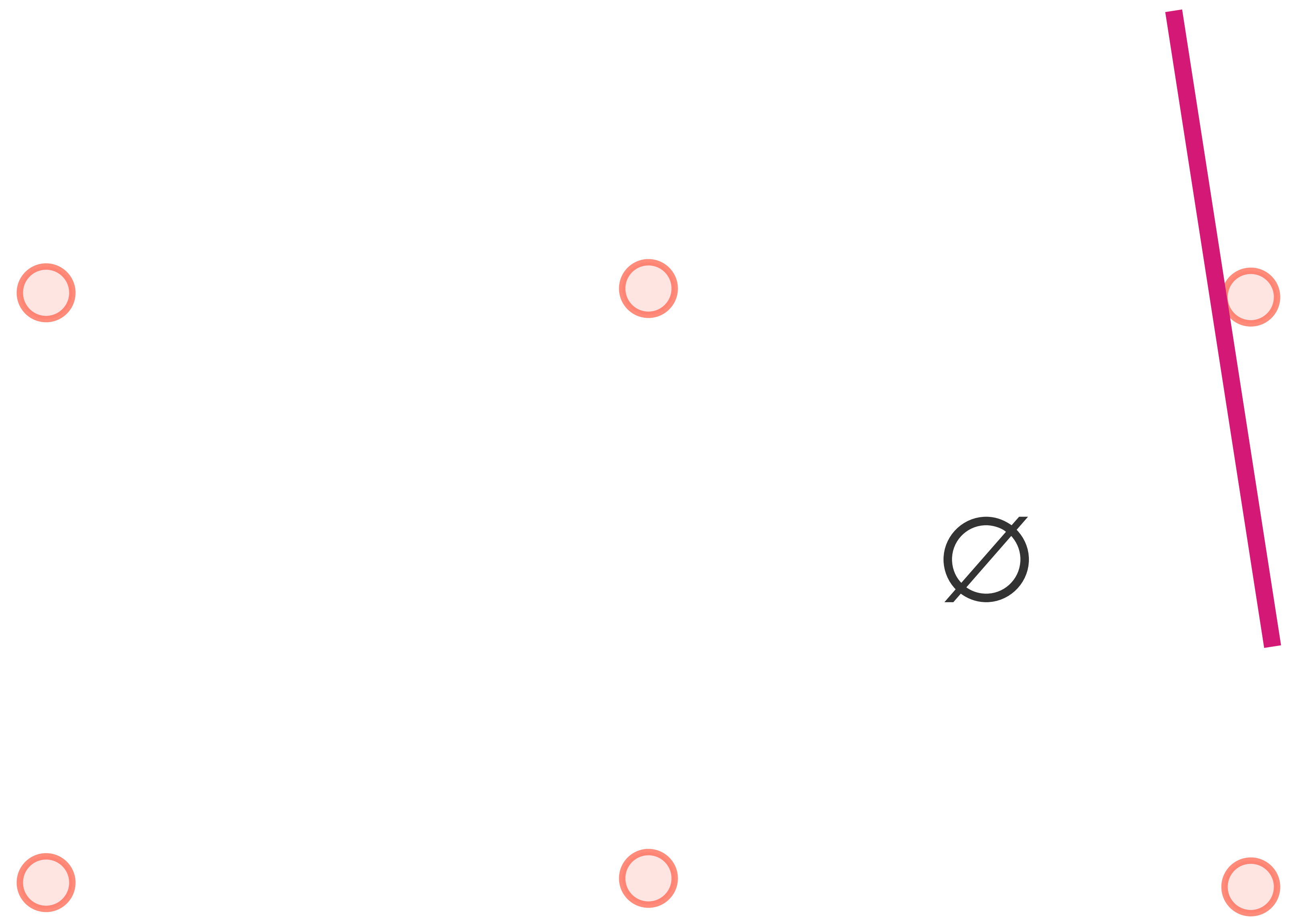
Derivation of $0 \geq 1$ from $Ax \geq b$

- equivalently, the **empty polytope**

Cutting Planes Proofs

Suppose $Ax \geq b$ has **no integer solutions**

$\rightarrow$ **Prove** this fact using cutting planes!



Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

$$\frac{ax \geq b,\ cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^{\geq 0}$$

- **Cut:**

$$\frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil}, \text{ if } d \in \mathbb{Z}^{\geq 0} \text{ divides } a$$

Cutting Planes Proof

Derivation of $0 \geq 1$ from $Ax \geq b$

- equivalently, the **empty polytope**

Today

Lower bounds on the size of Cutting Planes proofs!

Today

Lower bounds on the size of Cutting Planes proofs!

→ Unlike other proof systems, there is only one lower bound technique for Cutting Planes

Today

Lower bounds on the size of Cutting Planes proofs!

→Unlike other proof systems, there is only one lower bound technique for Cutting Planes

How? An exciting connection between proofs and circuits!

Monotone Feasible Interpolation

For many (all?) proof systems P it is possible to relate their complexity to the complexity of circuits in some associated model C_P of monotone computation

Monotone Feasible Interpolation

For many (all?) proof systems P it is possible to relate their complexity to the complexity of circuits in some associated model C_P of monotone computation

P-proof of F
of size s

$\longrightarrow$

C_P-Computation of f_F
of size $\mathrm{poly}(s)$

Where $f_F : \{0,1\}^m \to \{0,1\}$ is an associated monotone function (defined later)

Monotone Feasible Interpolation

For many (all?) proof systems P it is possible to relate their complexity to the complexity of circuits in some associated model C_P of monotone computation

P**-proof of** F of size s	$\longrightarrow$	C_P**-Computation of** f_F of size $\mathrm{poly}(s)$

Where $f_F : \{0,1\}^m \to \{0,1\}$ is an associated monotone function (defined later)

Upshot: computational lower bounds imply proof lower bounds!

Monotone Feasible Interpolation

For many (all?) proof systems P it is possible to relate their complexity to the complexity of circuits in some associated model C_P of monotone computation

P-proof of F of size s	$\longrightarrow$	**C_P-Computation of f_F** of size $\text{poly}(s)$

Where $f_F : \{0,1\}^m \to \{0,1\}$ is an associated monotone function (defined later)

Upshot: computational lower bounds imply proof lower bounds!

In many cases, a converse is possible as well!

Split Formulas

P-**proof of** F
of size s

$\rightarrow$

C_P-**Computation of** f_F
of size $\mathrm{poly}(s)$

For simplicity, we'll restrict to the case of **split** formulas:

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

Where A, B are CNF and all y variables occur positively in A

Split Formulas

P-proof of F
of size s

$\rightarrow$

C_P-Computation of f_F
of size $\text{poly}(s)$

For simplicity, we'll restrict to the case of **split** formulas:

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

Where A, B are CNF and all y variables occur positively in A

The Function Computed

Let $\alpha \in \{0,1\}^y$ be any assignment to $y \implies A(x, \alpha)$ or $B(\alpha, z)$ is unsatisfiable

Split Formulas

P-proof of F
of size s

$\longrightarrow$

C_P-Computation of f_F
of size poly(s)

For simplicity, we'll restrict to the case of **split** formulas:

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

Where A, B are CNF and all y variables occur positively in A

The Function Computed

Let $\alpha \in \{0,1\}^y$ be any assignment to $y \implies A(x, \alpha)$ or $B(\alpha, z)$ is unsatisfiable

Define monotone "interpolant" function $I_F(y) = \begin{cases} 0 & \text{if } A(x, \alpha) \text{ is unsatisfiable} \\ 1 & \text{if } B(\alpha, z) \text{ is unsatisfiable} \end{cases}$

Split Formulas

P-proof of split F
of size s

$\longrightarrow$

C_P-Computation of I_F
of size $\mathrm{poly}(s)$

For simplicity, we'll restrict to the case of **split** formulas:

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

Where A, B are CNF and all y variables occur positively in A

The Function Computed

Let $\alpha \in \{0,1\}^y$ be any assignment to $y \implies A(x, \alpha)$ or $B(\alpha, z)$ is unsatisfiable

Define monotone "interpolant" function $I_F(y) = \begin{cases} 0 & \text{if } A(x, \alpha) \text{ is unsatisfiable} \\ 1 & \text{if } B(\alpha, z) \text{ is unsatisfiable} \end{cases}$

Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

Split Formulas

$F(x, y, z) = A(x, y) \wedge B(y, z)$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$:

Split Formulas

$F(x, y, z) = A(x, y) \wedge B(y, z)$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$

Split Formulas

$F(x, y, z) = A(x, y) \wedge B(y, z)$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$:

Split Formulas

$F(x, y, z) = A(x, y) \wedge B(y, z)$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique

Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$:

Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Unsatisfiable!

Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Unsatisfiable!

e.g. suppose $n = 3, k = 3$

2

1

3

Split Formulas

$F(x, y, z) = A(x, y) \wedge B(y, z)$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

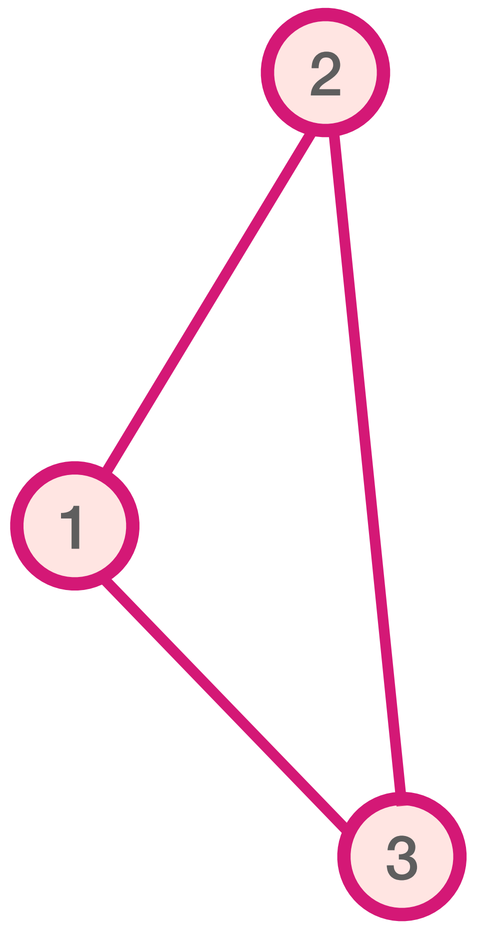
"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Unsatisfiable!

e.g. suppose $n = 3, k = 3$

If $y = [1,1,1]$



Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

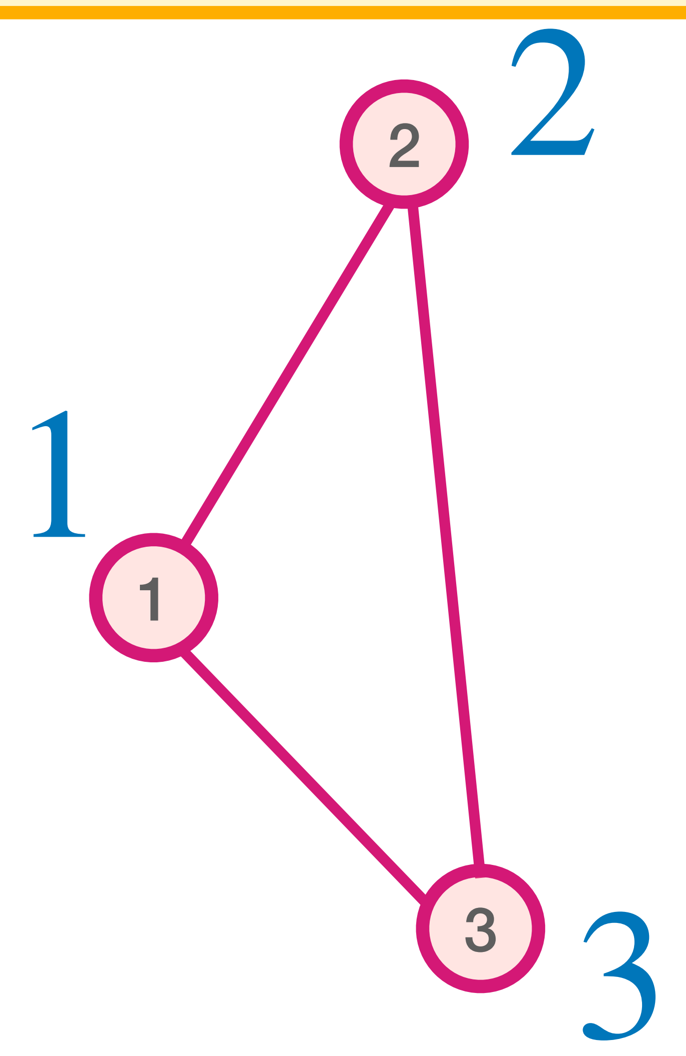
- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Unsatisfiable!

e.g. suppose $n = 3, k = 3$

If $y = [1,1,1]$

$\rightarrow$ $x = [1,0,0,0,1,0,0,0,1]$ satisfies the $Clique(x, y)$ constraints



Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

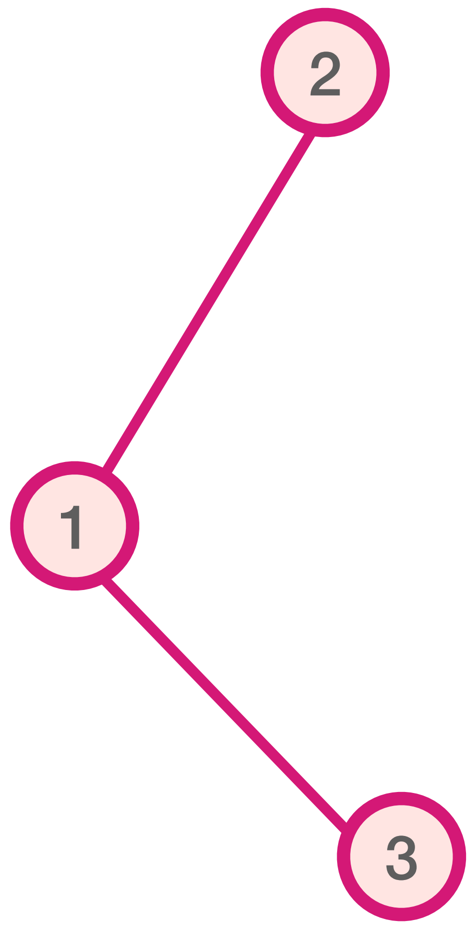
"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Unsatisfiable!

e.g. suppose $n = 3, k = 3$

If $y = [1,1,0]$



Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

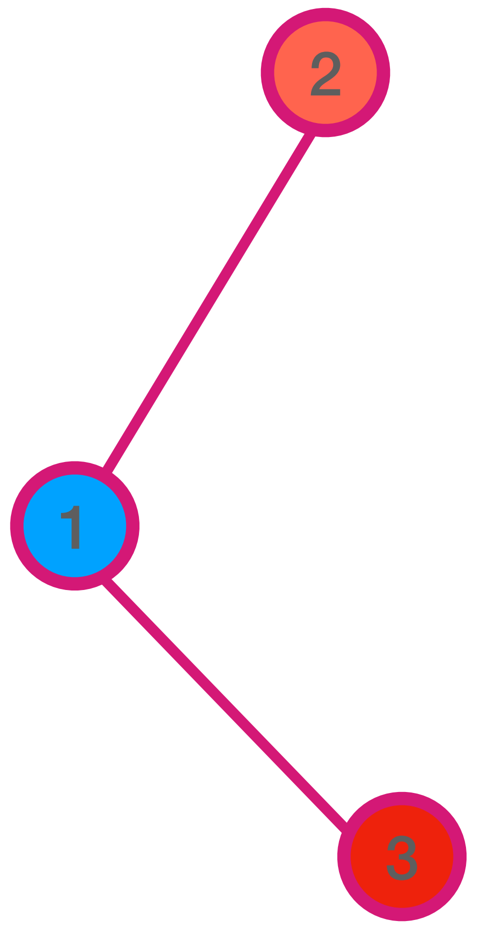
- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Unsatisfiable!

e.g. suppose $n = 3, k = 3$

If $y = [1,1,0]$

$\rightarrow z = [1,0,1,0,1,0]$ satisfies the $Color(y, z)$ constraints



Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Constraints of $Clique(x, y)$

- $\forall\, t \in [k]$: $\bigvee_{v\in[n]} x_{v,t}$ — some vertex is the t-th clique member

Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Constraints of $Clique(x, y)$

- $\forall\, t \in [k]$: $\bigvee_{v \in [n]} x_{v,t}$ — some vertex is the t-th clique member
- $\forall v,\ \forall t \neq \ell$: $\neg x_{v,t} \vee \neg x_{v,\ell}$ — no vertex is the t-th and ℓ-th clique member

Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Constraints of $Clique(x, y)$

- $\forall\, t \in [k]$: $\bigvee_{v \in [n]} x_{v,t}$ — some vertex is the t-th clique member
- $\forall v,\ \forall t \neq \ell$: $\neg x_{v,t} \vee \neg x_{v,\ell}$ — no vertex is the t-th and ℓ-th clique member
- $\forall u \neq v, \forall t \neq \ell$: $\neg x_{u,t} \vee \neg x_{v,\ell} \vee \neg y_{uv}$ — if u, v are in the clique then edge uv must be present

Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Constraints of $Color(y, z)$

- $\forall\, v \in [n]$: $\bigvee_{c\in[k-1]} z_{v,c}$ — every vertex gets a color

Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Constraints of $Color(y, z)$

- $\forall\, v \in [n]$: $\bigvee_{c \in [k-1]} z_{v,c}$ — every vertex gets a color
- $\forall v,\ \forall c \neq d$: $\neg z_{v,c} \vee \neg z_{v,d}$ — no vertex gets two different colors

Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Constraints of $Color(y, z)$

- $\forall\, v \in [n]$: $\bigvee_{c \in [k-1]} z_{v,c}$ — every vertex gets a color
- $\forall v,\ \forall c \neq d$: $\neg z_{v,c} \vee \neg z_{v,d}$ — no vertex gets two different colors
- $\forall u \neq v, \forall c$: $\neg z_{u,c} \vee \neg z_{v,c} \vee y_{uv}$ — adjacent vertices must receive different colors

Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Interpolant function: $I_F(y) = \begin{cases} 0 & \text{if } Clique(x, \alpha) \text{ is unsatisfiable} \\ 1 & \text{if } Color(\alpha, z) \text{ is unsatisfiable} \end{cases}$

Split Formulas

$$F(x, y, z) = A(x, y) \wedge B(y, z)$$

E.g. $Clique(x, y) \wedge Color(y, z)_{n,k}$

"There is a graph containing both a k-clique and a $(k-1)$-coloring"

- $y \in \{0,1\}^{\binom{n}{2}}$ defines an n-vertex graph $G(y) = (V, E)$: $e \in E$ iff $y_e = 1$
- $x \in \{0,1\}^{nk}$ defines a k-clique in $G(y)$: $x_{v,t} = 1$ iff v is t-th member of clique
- $z \in \{0,1\}^{n(k-1)}$ defines a $(k-1)$-coloring of $G(y)$: v has color c iff $z_{v,c} = 1$

Interpolant function: $I_F(y) = \begin{cases} 0 & \text{if } Clique(x, \alpha) \text{ is unsatisfiable} \\ 1 & \text{if } Color(\alpha, z) \text{ is unsatisfiable} \end{cases}$

Interpolation theorem for P implies P-proof of $Clique(x, y) \wedge Color(y, z)_{n,k} \Longrightarrow$ C_P-computation separating graphs with k-cliques from $(k-1)$-colorable graphs

Feasible Interpolation Theorems

[R95]

$S^2_1(\alpha) \Longrightarrow$ Boolean circuits

Feasible Interpolation Theorems

[R95]

[K97] Defined interpolation as a general method.

$S^2_1(\alpha) \Longrightarrow$ Boolean circuits

Resolution $\Longrightarrow$ Monotone circuits

Feasible Interpolation Theorems

[R95] $S_1^2(\alpha) \implies$ Boolean circuits

[K97] Defined interpolation as a general method.
Resolution $\implies$ Monotone circuits

[PS97] Nullstellensatz $\implies$ Monotone span programs

Feasible Interpolation Theorems

[R95]

[K97] Defined interpolation as a general method.

[PS97]

[BPR97]

$S_1^2(\alpha) \implies$ Boolean circuits

Resolution $\implies$ Monotone circuits

Nullstellensatz $\implies$ Monotone span programs

Cutting Planes* $\implies$ Monotone circuits

Feasible Interpolation Theorems

[R95]	$S_1^2(\alpha)$	$\Longrightarrow$	Boolean circuits
[K97] Defined interpolation as a general method.	Resolution	$\Longrightarrow$	Monotone circuits
[PS97]	Nullstellensatz	$\Longrightarrow$	Monotone span programs
[BPR97]	Cutting Planes*	$\Longrightarrow$	Monotone circuits
[P97]	Cutting Planes	$\Longrightarrow$	Monotone real circuits

Feasible Interpolation Theorems

[R95]	$S_1^2(\alpha) \Longrightarrow$ Boolean circuits
[K97] Defined interpolation as a general method.	Resolution $\Longrightarrow$ Monotone circuits
[PS97]	Nullstellensatz $\Longrightarrow$ Monotone span programs
[BPR97]	Cutting Planes* $\Longrightarrow$ Monotone circuits
[P97]	Cutting Planes $\Longrightarrow$ Monotone real circuits

Only worked for split formulas!

Feasible Interpolation Theorems

[R95] — $S_1^2(\alpha) \Longrightarrow$ Boolean circuits

[K97] Defined interpolation as a general method. — Resolution $\Longrightarrow$ Monotone circuits

[PS97] — Nullstellensatz $\Longrightarrow$ Monotone span programs

[BPR97] — Cutting Planes* $\Longrightarrow$ Monotone circuits

[P97] — Cutting Planes $\Longrightarrow$ Monotone real circuits

[FPPR17, HP17] Generalized Interpolation to work for **any** unsatisfiable formula — $CC_{O(\log n)} \Longleftrightarrow$ Monotone circuits; $RCC_1 \Longleftrightarrow$ Monotone real circuits

Feasible Interpolation Theorems

[R95]	$S_1^2(\alpha) \Longrightarrow$ Boolean circuits
[K97] Defined interpolation as a general method.	Resolution $\Longrightarrow$ Monotone circuits
[PS97]	Nullstellensatz $\Longrightarrow$ Monotone span programs
[BPR97]	Cutting Planes* $\Longrightarrow$ Monotone circuits
[P97]	Cutting Planes $\Longrightarrow$ Monotone real circuits
[FPPR17, HP17] Generalized Interpolation to work for **any** unsatisfiable formula	$CC_{O(\log n)} \Longleftrightarrow$ Monotone circuits $RCC_1 \Longleftrightarrow$ Monotone real circuits
[FGGR22]	Sherali-Adams $\Longrightarrow$ Extended Formulations

Feasible Interpolation Theorems

[R95]	$S_1^2(\alpha) \Longrightarrow$ Boolean circuits
[K97] Defined interpolation as a general method.	Resolution $\Longrightarrow$ Monotone circuits
[PS97]	Nullstellensatz $\Longrightarrow$ Monotone span programs
[BPR97]	Cutting Planes* $\Longrightarrow$ Monotone circuits
[P97]	Cutting Planes $\Longrightarrow$ Monotone real circuits
[FPPR17, HP17] Generalized Interpolation to work for **any** unsatisfiable formula	$CC_{O(\log n)} \Longleftrightarrow$ Monotone circuits $RCC_1 \Longleftrightarrow$ Monotone real circuits
[FGGR22]	Sherali-Adams $\Longrightarrow$ Extended Formulations
[FGR22 unpublished]	SoS $\Longrightarrow$ Semidefinite EFs

Feasible Interpolation For CP

[P97] Cutting Planes $\Longrightarrow$ Monotone real circuits

Feasible Interpolation For CP

[P97] Cutting Planes $\Longrightarrow$ Monotone real circuits

Remainder of today:

1. Prove this theorem

Feasible Interpolation For CP

[P97] Cutting Planes $\Longrightarrow$ Monotone real circuits

Remainder of today:

1. Prove this theorem
2. Use known lower bounds on monotone real circuits computing clique to obtain Cutting Planes lower bounds for $Clique - Color$

Feasible Interpolation For CP

[P97] Cutting Planes $\Longrightarrow$ Monotone real circuits

We will first prove the following simpler lemma

Feasible Interpolation For CP

[P97] Cutting Planes $\Longrightarrow$ Monotone real circuits

We will first prove the following simpler lemma

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Feasible Interpolation For CP

[P97] Cutting Planes $\Longrightarrow$ Monotone real circuits

We will first prove the following simpler lemma

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

The following claim will allow us to define our algorithm

Feasible Interpolation For CP

[P97] Cutting Planes $\Longrightarrow$ Monotone real circuits

We will first prove the following simpler lemma

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

The following claim will allow us to define our algorithm

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

Feasible Interpolation For CP

[P97] Cutting Planes $\Longrightarrow$ Monotone real circuits

We will first prove the following simpler lemma

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

The following claim will allow us to define our algorithm

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$

Feasible Interpolation For CP

[P97] Cutting Planes $\Longrightarrow$ Monotone real circuits

We will first prove the following simpler lemma

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

The following claim will allow us to define our algorithm

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α

Feasible Interpolation For CP

[P97] Cutting Planes $\Longrightarrow$ Monotone real circuits

We will first prove the following simpler lemma

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

The following claim will allow us to define our algorithm

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

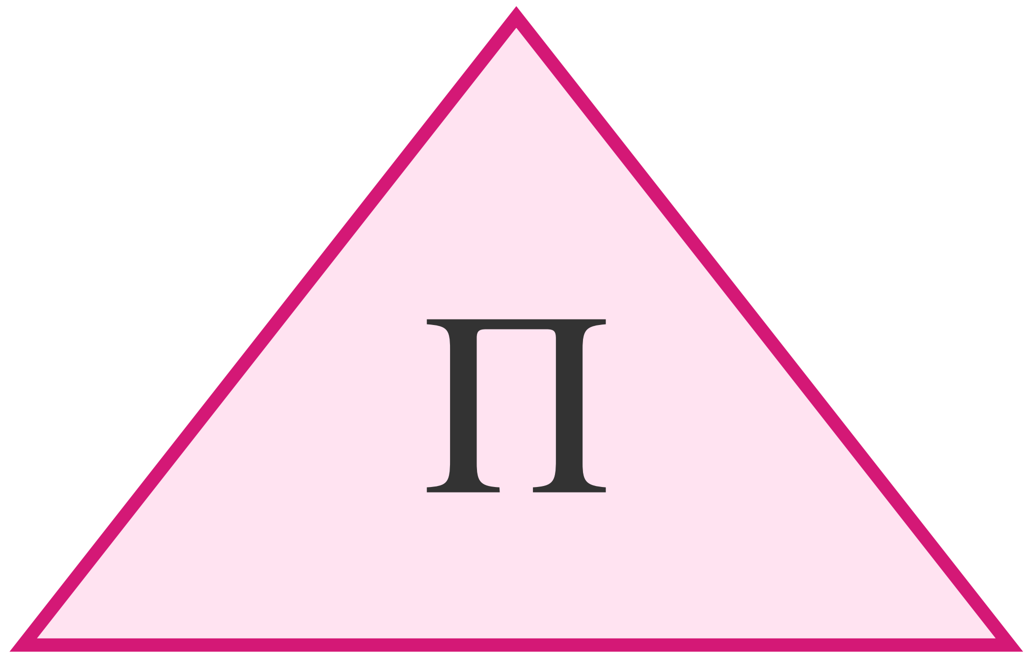
Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

$0 \geq 1$



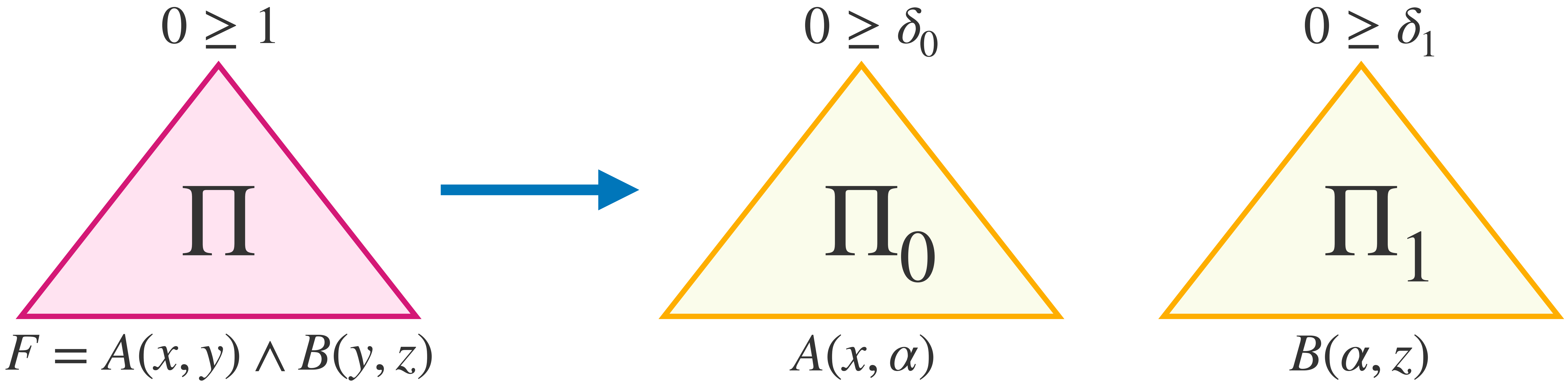
$F = A(x, y) \wedge B(y, z)$

Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x,y) \wedge B(y,z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x,\alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$



Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof of Lemma:

Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x,y) \wedge B(y,z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x,\alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof of Lemma: Claim allows us to extract from Π a proof of

$\rightarrow A(x,\alpha)$ if $A(x,\alpha)$ is unsatisfiable

$\rightarrow B(\alpha,z)$ if $B(\alpha,z)$ is unsatisfiable

Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof of Lemma: Claim allows us to extract from Π a proof of

$\rightarrow$ $A(x, \alpha)$ if $A(x, \alpha)$ is unsatisfiable

$\rightarrow$ $B(\alpha, z)$ if $B(\alpha, z)$ is unsatisfiable

Indeed, …

Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof of Lemma: Applying claim to the last line $0 \geq 1$ of Π, we get

Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x,y) \wedge B(y,z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x,\alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof of Lemma: Applying claim to the last line $0 \geq 1$ of Π, we get

- Derivation of $0 \geq \delta_0$ from $A(x,\alpha)$
- Derivation of $0 \geq \delta_1$ from $B(\alpha, z)$

Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x,y) \wedge B(y,z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x,\alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof of Lemma: Applying claim to the last line $0 \geq 1$ of Π, we get

- Derivation of $0 \geq \delta_0$ from $A(x,\alpha)$
- Derivation of $0 \geq \delta_1$ from $B(\alpha, z)$

With $\delta_0 + \delta_1 \geq 1$

Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x,y) \wedge B(y,z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x,\alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof of Lemma: Applying claim to the last line $0 \geq 1$ of Π, we get

- Derivation of $0 \geq \delta_0$ from $A(x,\alpha)$
- Derivation of $0 \geq \delta_1$ from $B(\alpha, z)$

With $\delta_0 + \delta_1 \geq 1$

Either $\delta_0 > 0$ and so $A(x,\alpha)$ is unsatisfiable

or $\delta_1 > 0$ and so $B(\alpha,z)$ is unsatisfiable

Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof of Lemma: The poly-time algorithm:

Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof of Lemma: The poly-time algorithm:

on input $\alpha \in \{0,1\}^y$

Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof of Lemma: The poly-time algorithm:

on input $\alpha \in \{0,1\}^y$

1. Constructs δ_0 and δ_1 in time poly(s)

Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof of Lemma: The poly-time algorithm:

on input $\alpha \in \{0,1\}^y$

1. Constructs δ_0 and δ_1 in time poly(s)
2. If $\delta_0 > 0$ then $A(x, \alpha)$ is unsatisfiable and we output 0

Feasible Interpolation For CP

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof of Lemma: The poly-time algorithm:

on input $\alpha \in \{0,1\}^y$

1. Constructs δ_0 and δ_1 in time poly(s)
2. If $\delta_0 > 0$ then $A(x, \alpha)$ is unsatisfiable and we output 0
3. Otherwise, $\delta_1 > 0$ and $B(\alpha, z)$ is unsatisfiable, so output 1

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: by induction. **Base case:**

- If $ax + by + cz \geq d$ belongs to $A(x, y)$ then $c = 0$
  - → Let $\delta_0 = d - b\alpha$ and the proof Π_0 be the axiom $ax \geq d - b\alpha$ of $A(x, \alpha)$
  - → Let $\delta_1 = 0$ and the proof Π_1 be the trivial axiom $0 \geq 0$
- If $ax + by + cz \geq d$ is an axiom of $B(y, z)$ then $a = 0$
  - → Let $\delta_0 = 0$ and Π_0 be $0 \geq 0$
  - → Let $\delta_1 = d - b\alpha$ and Π_1 be the axiom $cz \geq d - b\alpha$ of $B(\alpha, z)$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: by induction. **Base case:**

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: by induction. **Base case:**

- If $ax + by + cz \geq d$ belongs to $A(x, y)$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: by induction. **Base case:**

- If $ax + by + cz \geq d$ belongs to $A(x, y)$ then $c = 0$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: by induction. **Base case:**

- If $ax + by + cz \geq d$ belongs to $A(x, y)$ then $c = 0$

  $\rightarrow$ Let $\delta_0 = d - b\alpha$ and the proof Π_0 be the axiom $ax \geq d - b\alpha$ of $A(x, \alpha)$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: by induction. **Base case:**

- If $ax + by + cz \geq d$ belongs to $A(x, y)$ then $c = 0$

  $\rightarrow$ Let $\delta_0 = d - b\alpha$ and the proof Π_0 be the axiom $ax \geq d - b\alpha$ of $A(x, \alpha)$

  $\rightarrow$ Let $\delta_1 = 0$ and the proof Π_1 be the trivial axiom $0 \geq 0$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: by induction. **Base case:**

- If $ax + by + cz \geq d$ belongs to $A(x, y)$ then $c = 0$
  → Let $\delta_0 = d - b\alpha$ and the proof Π_0 be the axiom $ax \geq d - b\alpha$ of $A(x, \alpha)$
  → Let $\delta_1 = 0$ and the proof Π_1 be the trivial axiom $0 \geq 0$
- If $ax + by + cz \geq d$ is an axiom of $B(y, z)$ then $a = 0$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: **Cut:** Suppose that $ax + by + cz \geq d$ is deduced by cut in Π

$$\frac{a'x + b'y + c'z \geq d'}{(a'/t)x + (b'/t)y + (c'/t)z \geq \lceil d'/t \rceil}$$

For t dividing a', b', c'

$ax + by + cz \geq d$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Cut: Suppose that $ax + by + cz \geq d$ is deduced by cut in Π

$$\frac{a'x + b'y + c'z \geq d'}{(a'/t)x + (b'/t)y + (c'/t)z \geq \lceil d'/t \rceil}$$

For t dividing a', b', c'

And by induction we have derived

From $A(x, \alpha)$: $\boxed{a'x \geq \delta_0'}$

From $B(\alpha, z)$: $\boxed{c'z \geq \delta_1'}$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Cut: Suppose that $ax + by + cz \geq d$ is deduced by cut in Π

$$\frac{a'x + b'y + c'z \geq d'}{(a'/t)x + (b'/t)y + (c'/t)z \geq \lceil d'/t \rceil}$$

For t dividing a', b', c'

And by induction we have derived

From $A(x, \alpha)$: $\boxed{a'x \geq \delta_0'}$

From $B(\alpha, z)$: $\boxed{c'z \geq \delta_1'}$

With $\delta_0' + \delta_1' \geq d' - b'\alpha$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Cut: Suppose that $ax + by + cz \geq d$ is deduced by cut in Π

$$\frac{a'x + b'y + c'z \geq d'}{(a'/t)x + (b'/t)y + (c'/t)z \geq \lceil d'/t \rceil}$$ For t dividing a', b', c'

And by induction we have derived

From $A(x, \alpha)$: $a'x \geq \delta_0'$ → Cut → $(a'/t)x \geq \lceil \delta_0'/t \rceil$

From $B(\alpha, z)$: $c'z \geq \delta_1'$ → Cut → $(c'/t)z \geq \lceil \delta_1'/t \rceil$

With $\delta_0' + \delta_1' \geq d' - b'\alpha$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: **Cut:** Suppose that $ax + by + cz \geq d$ is deduced by cut in Π

$$\frac{a'x + b'y + c'z \geq d'}{(a'/t)x + (b'/t)y + (c'/t)z \geq \lceil d'/t \rceil}$$

For t dividing a', b', c'

And by induction we have derived

From $A(x, \alpha)$: $a'x \geq \delta_0'$ $\rightarrow$ Cut $\rightarrow$ $(a'/t)x \geq \lceil \delta_0'/t \rceil = \delta_0$

From $B(\alpha, z)$: $c'z \geq \delta_1'$ $\rightarrow$ Cut $\rightarrow$ $(c'/t)z \geq \lceil \delta_1'/t \rceil = \delta_1$

With $\delta_0' + \delta_1' \geq d' - b'\alpha$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Cut: Suppose that $ax + by + cz \geq d$ is deduced by cut in Π

$$\frac{a'x + b'y + c'z \geq d'}{(a'/t)x + (b'/t)y + (c'/t)z \geq \lceil d'/t \rceil}$$

For t dividing a', b', c'

And by induction we have derived

$a'x \geq \delta_0'$ → Cut → $(a'/t)x \geq \lceil \delta_0'/t \rceil = \delta_0$

$c'z \geq \delta_1'$ → Cut → $(c'/t)z \geq \lceil \delta_1'/t \rceil = \delta_1$

Invariant:

$\delta_0 + \delta_1$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Cut: Suppose that $ax + by + cz \geq d$ is deduced by cut in Π

$$\frac{a'x + b'y + c'z \geq d'}{(a'/t)x + (b'/t)y + (c'/t)z \geq \lceil d'/t \rceil}$$

For t dividing a', b', c'

And by induction we have derived

$a'x \geq \delta_0'$ $\rightarrow$ Cut $\rightarrow$ $(a'/t)x \geq \lceil \delta_0'/t \rceil = \delta_0$

$c'z \geq \delta_1'$ $\rightarrow$ Cut $\rightarrow$ $(c'/t)z \geq \lceil \delta_1'/t \rceil = \delta_1$

Invariant:

$$\delta_0 + \delta_1 = \lceil \delta_0'/t \rceil + \lceil \delta_1'/t \rceil$$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Cut: Suppose that $ax + by + cz \geq d$ is deduced by cut in Π

$$\frac{a'x + b'y + c'z \geq d'}{(a'/t)x + (b'/t)y + (c'/t)z \geq \lceil d'/t \rceil}$$

For t dividing a', b', c'

And by induction we have derived

$a'x \geq \delta_0'$ → Cut → $(a'/t)x \geq \lceil \delta_0'/t \rceil = \delta_0$

$c'z \geq \delta_1'$ → Cut → $(c'/t)z \geq \lceil \delta_1'/t \rceil = \delta_1$

Invariant:

$$\delta_0 + \delta_1 = \lceil \delta_0'/t \rceil + \lceil \delta_1'/t \rceil \geq \lceil (\delta_0' + \delta_1')/t \rceil$$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: **Cut:** Suppose that $ax + by + cz \geq d$ is deduced by cut in Π

$$\frac{a'x + b'y + c'z \geq d'}{(a'/t)x + (b'/t)y + (c'/t)z \geq \lceil d'/t \rceil}$$ For t dividing a', b', c'

And by induction we have derived

$a'x \geq \delta_0'$ $\rightarrow$ Cut $\rightarrow$ $(a'/t)x \geq \lceil \delta_0'/t \rceil = \delta_0$

$c'z \geq \delta_1'$ $\rightarrow$ Cut $\rightarrow$ $(c'/t)z \geq \lceil \delta_1'/t \rceil = \delta_1$

Invariant:

$$\delta_0 + \delta_1 = \lceil \delta_0'/t \rceil + \lceil \delta_1'/t \rceil \geq \lceil (\delta_0' + \delta_1')/t \rceil \geq \lceil (d - b\alpha)/t \rceil$$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: **Cut:** Suppose that $ax + by + cz \geq d$ is deduced by cut in Π

$$\frac{a'x + b'y + c'z \geq d'}{(a'/t)x + (b'/t)y + (c'/t)z \geq \lceil d'/t \rceil}$$

For t dividing a', b', c'

And by induction we have derived

$a'x \geq \delta_0'$ $\rightarrow$ Cut $\rightarrow$ $(a'/t)x \geq \lceil \delta_0'/t \rceil = \delta_0$

$c'z \geq \delta_1'$ $\rightarrow$ Cut $\rightarrow$ $(c'/t)z \geq \lceil \delta_1'/t \rceil = \delta_1$

Invariant:

$$\delta_0 + \delta_1 = \lceil \delta_0'/t \rceil + \lceil \delta_1'/t \rceil \geq \lceil (\delta_0' + \delta_1')/t \rceil \geq \lceil (d - b\alpha)/t \rceil = \lceil d/t \rceil - b\alpha/t$$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Non-negative Linear Combination:

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Non-negative Linear Combination:

$$\frac{a'x + b'y + c'z \geq d', \qquad a''x + b''y + c''z \geq d''}{(\gamma'a' + \gamma''a'')x + (\gamma'b' + \gamma''b'')y + (\gamma'c' + \gamma''c'')z \geq \gamma'd' + \gamma''d''} \quad \text{For } \gamma', \gamma'' \geq 0$$

$$ax + by + cz \geq d$$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: **Non-negative Linear Combination:**

$$\frac{a'x + b'y + c'z \geq d', \qquad a''x + b''y + c''z \geq d''}{(\gamma'a' + \gamma''a'')x + (\gamma'b' + \gamma''b'')y + (\gamma'c' + \gamma''c'')z \geq \gamma'd' + \gamma''d''}$$ For $\gamma', \gamma'' \geq 0$

And by induction we have derived

$\boxed{a'x \geq \delta_0', \quad a''x \geq \delta_0''}$ From $A(x, \alpha)$

$\boxed{c'z \geq \delta_1', \quad c''z \geq \delta_1''}$ From $B(\alpha, z)$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Non-negative Linear Combination:

$$\frac{a'x + b'y + c'z \geq d', \qquad a''x + b''y + c''z \geq d''}{(\gamma'a' + \gamma''a'')x + (\gamma'b' + \gamma''b'')y + (\gamma'c' + \gamma''c'')z \geq \gamma'd' + \gamma''d''} \quad \text{For } \gamma', \gamma'' \geq 0$$

And by induction we have derived

$a'x \geq \delta_0', \quad a''x \geq \delta_0''$ → non-neg combo → $(\gamma'a' + \gamma''a'')x \geq \gamma'\delta_0' + \gamma''\delta_0''$

$c'z \geq \delta_1', \quad c''z \geq \delta_1''$ → non-neg combo → $(\gamma'c' + \gamma''c'')z \geq \gamma'\delta_1' + \gamma''\delta_1''$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Non-negative Linear Combination:

$$\frac{a'x + b'y + c'z \geq d', \qquad a''x + b''y + c''z \geq d''}{(\gamma'a' + \gamma''a'')x + (\gamma'b' + \gamma''b'')y + (\gamma'c' + \gamma''c'')z \geq \gamma'd' + \gamma''d''} \quad \text{For } \gamma', \gamma'' \geq 0$$

And by induction we have derived

$a'x \geq \delta_0', \quad a''x \geq \delta_0''$ → non-neg combo → $(\gamma'a' + \gamma''a'')x \geq \gamma'\delta_0' + \gamma''\delta_0'' = \delta_0$

$c'z \geq \delta_1', \quad c''z \geq \delta_1''$ → non-neg combo → $(\gamma'c' + \gamma''c'')z \geq \gamma'\delta_1' + \gamma''\delta_1'' = \delta_1$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Non-negative Linear Combination:

$$\frac{a'x + b'y + c'z \geq d', \qquad a''x + b''y + c''z \geq d''}{(\gamma'a' + \gamma''a'')x + (\gamma'b' + \gamma''b'')y + (\gamma'c' + \gamma''c'')z \geq \gamma'd' + \gamma''d''} \quad \text{For } \gamma', \gamma'' \geq 0$$

And by induction we have derived

$a'x \geq \delta_0', \quad a''x \geq \delta_0''$ → non-neg combo → $(\gamma'a' + \gamma''a'')x \geq \gamma'\delta_0' + \gamma''\delta_0'' = \delta_0$

$c'z \geq \delta_1', \quad c''z \geq \delta_1''$ → non-neg combo → $(\gamma'c' + \gamma''c'')z \geq \gamma'\delta_1' + \gamma''\delta_1'' = \delta_1$

Invariant: $\delta_0 + \delta_1$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Non-negative Linear Combination:

$$\frac{a'x + b'y + c'z \geq d', \qquad a''x + b''y + c''z \geq d''}{(\gamma'a' + \gamma''a'')x + (\gamma'b' + \gamma''b'')y + (\gamma'c' + \gamma''c'')z \geq \gamma'd' + \gamma''d''} \quad \text{For } \gamma', \gamma'' \geq 0$$

And by induction we have derived

$a'x \geq \delta_0', \quad a''x \geq \delta_0''$ → non-neg combo → $(\gamma'a' + \gamma''a'')x \geq \gamma'\delta_0' + \gamma''\delta_0'' = \delta_0$

$c'z \geq \delta_1', \quad c''z \geq \delta_1''$ → non-neg combo → $(\gamma'c' + \gamma''c'')z \geq \gamma'\delta_1' + \gamma''\delta_1'' = \delta_1$

Invariant: $\delta_0 + \delta_1 = \gamma'(\delta_0' + \delta_1') + \gamma''(\delta_0'' + \delta_1'')$

Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in Π there are constants δ_0, δ_1 s.t.

1. There are CP derivations of $ax \geq \delta_0$ from $A(x, \alpha)$ and $cz \geq \delta_1$ from $B(\alpha, z)$
2. δ_0, δ_1 are constructible in poly(s) time from Π and α
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Non-negative Linear Combination:

$$\frac{a'x + b'y + c'z \geq d', \qquad a''x + b''y + c''z \geq d''}{(\gamma'a' + \gamma''a'')x + (\gamma'b' + \gamma''b'')y + (\gamma'c' + \gamma''c'')z \geq \gamma'd' + \gamma''d''}$$

For $\gamma', \gamma'' \geq 0$

And by induction we have derived

$a'x \geq \delta_0', \quad a''x \geq \delta_0''$ → non-neg combo → $(\gamma'a' + \gamma''a'')x \geq \gamma'\delta_0' + \gamma''\delta_0'' = \delta_0$

$c'z \geq \delta_1', \quad c''z \geq \delta_1''$ → non-neg combo → $(\gamma'c' + \gamma''c'')z \geq \gamma'\delta_1' + \gamma''\delta_1'' = \delta_1$

Invariant: $\delta_0 + \delta_1 = \gamma'(\delta_0' + \delta_1') + \gamma''(\delta_0'' + \delta_1'') \geq \gamma'(d' - b'\alpha) + \gamma''(d'' - b''\alpha)$

Feasible Interpolation by Real Circuits

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

This lemma is overkill!

Feasible Interpolation by Real Circuits

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \land B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

This lemma is overkill!

$\rightarrow$ Don't need the full power of poly-time algorithms to construct δ_0, δ_1.

Feasible Interpolation by Real Circuits

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

This lemma is overkill!

$\rightarrow$ Don't need the full power of poly-time algorithms to construct δ_0, δ_1.

$\rightarrow$ In order to calculate δ_0, δ_1, only need a computational model which supports addition, multiplication, division, ceiling

Feasible Interpolation by Real Circuits

Lemma: There is a time poly(s) algorithm which given a split formula $F = A(x, y) \wedge B(y, z)$, a size s CP proof of Π of F, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$

This lemma is overkill!

$\rightarrow$ Don't need the full power of poly-time algorithms to construct δ_0, δ_1.

$\rightarrow$ In order to calculate δ_0, δ_1, only need a computational model which supports addition, multiplication, division, ceiling

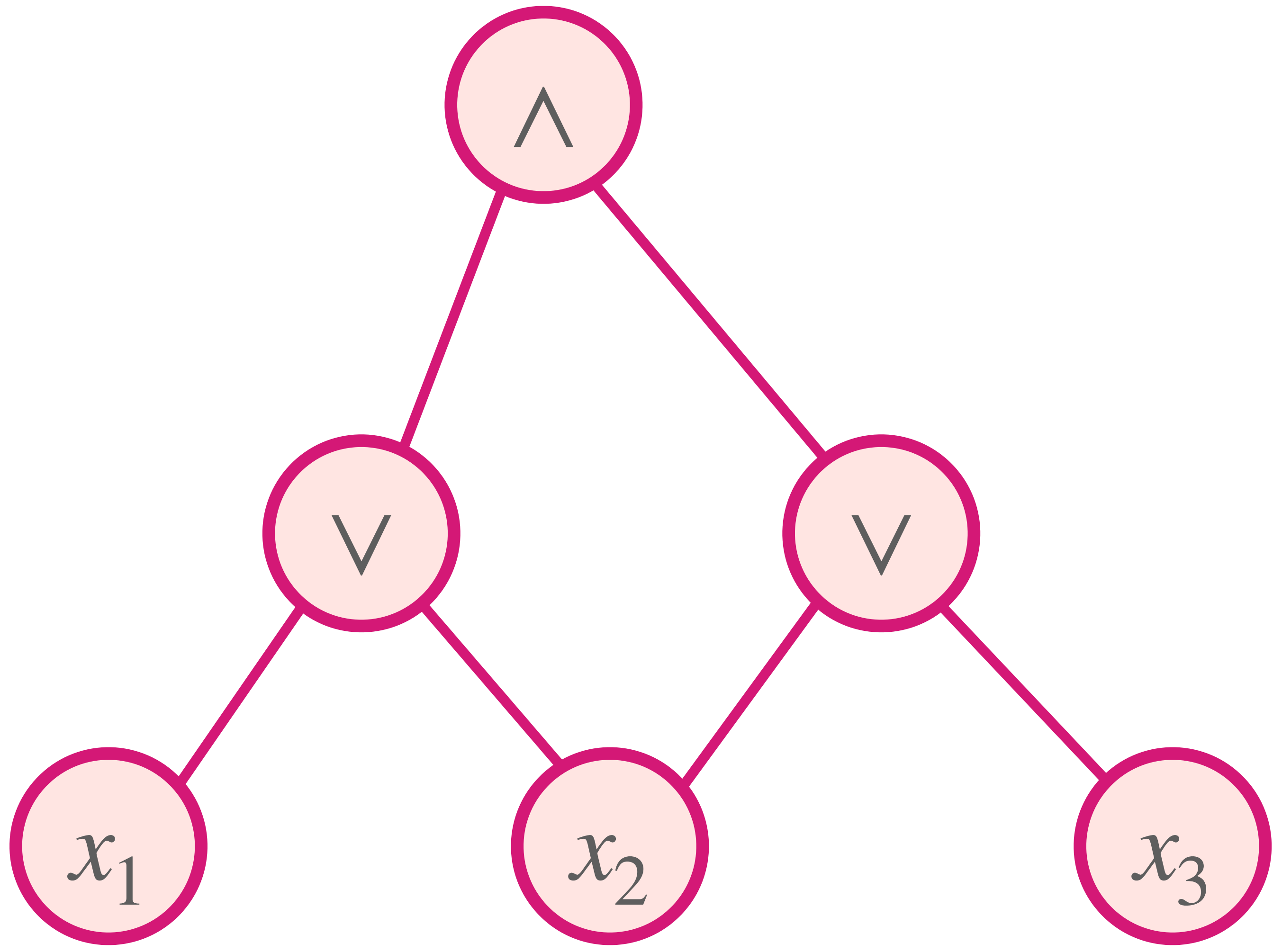
We will define a computational model can do all of this but is still weak enough to prove lower bounds on!

Feasible Interpolation by Real Circuits

Monotone Circuits: boolean circuits using only $\wedge$ and $\vee$ gates — no $\neg$

Feasible Interpolation by Real Circuits

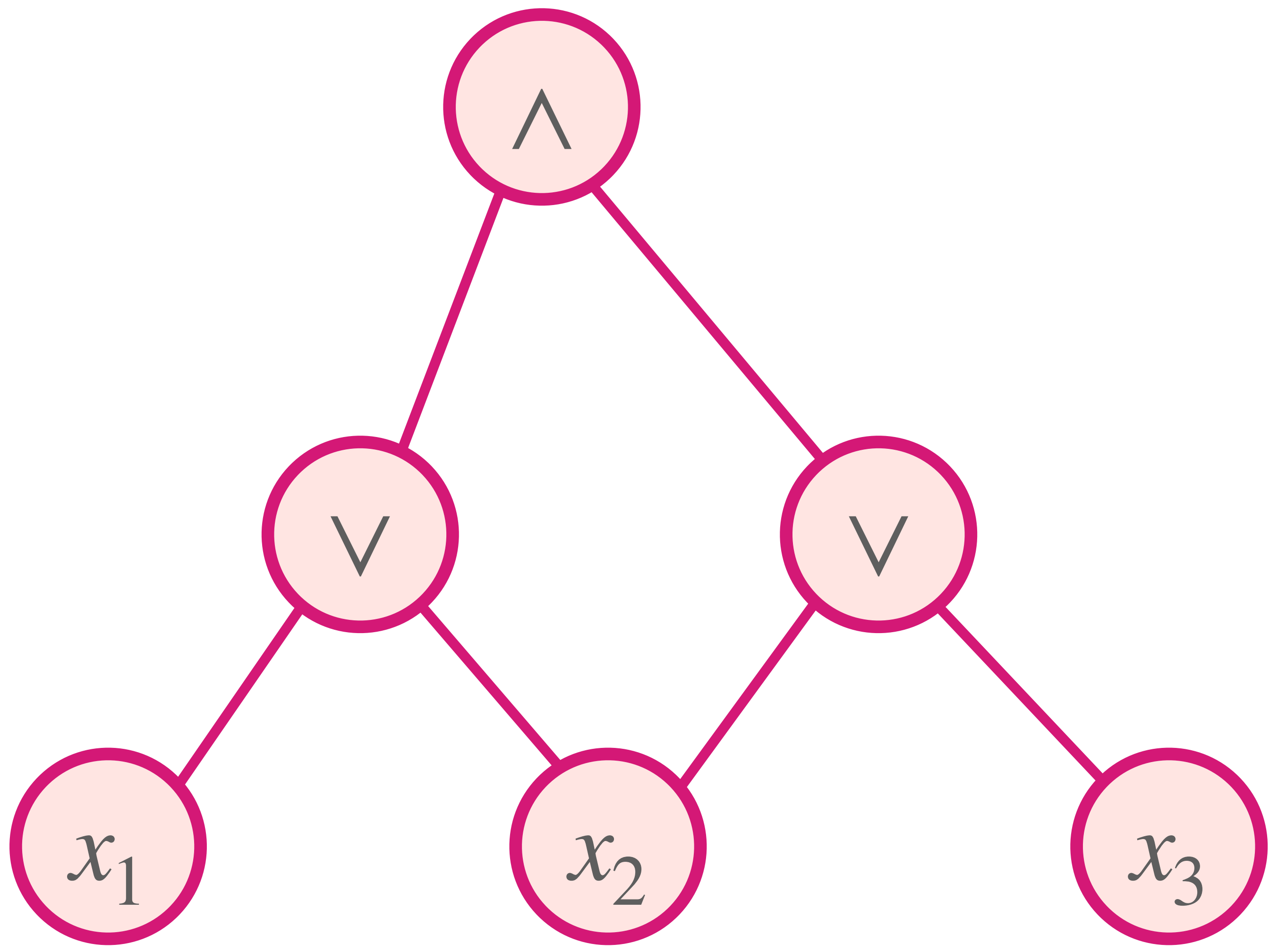
Monotone Circuits: boolean circuits using only $\wedge$ and $\vee$ gates — no $\neg$



Feasible Interpolation by Real Circuits

Monotone Circuits: boolean circuits using only $\wedge$ and $\vee$ gates — no $\neg$

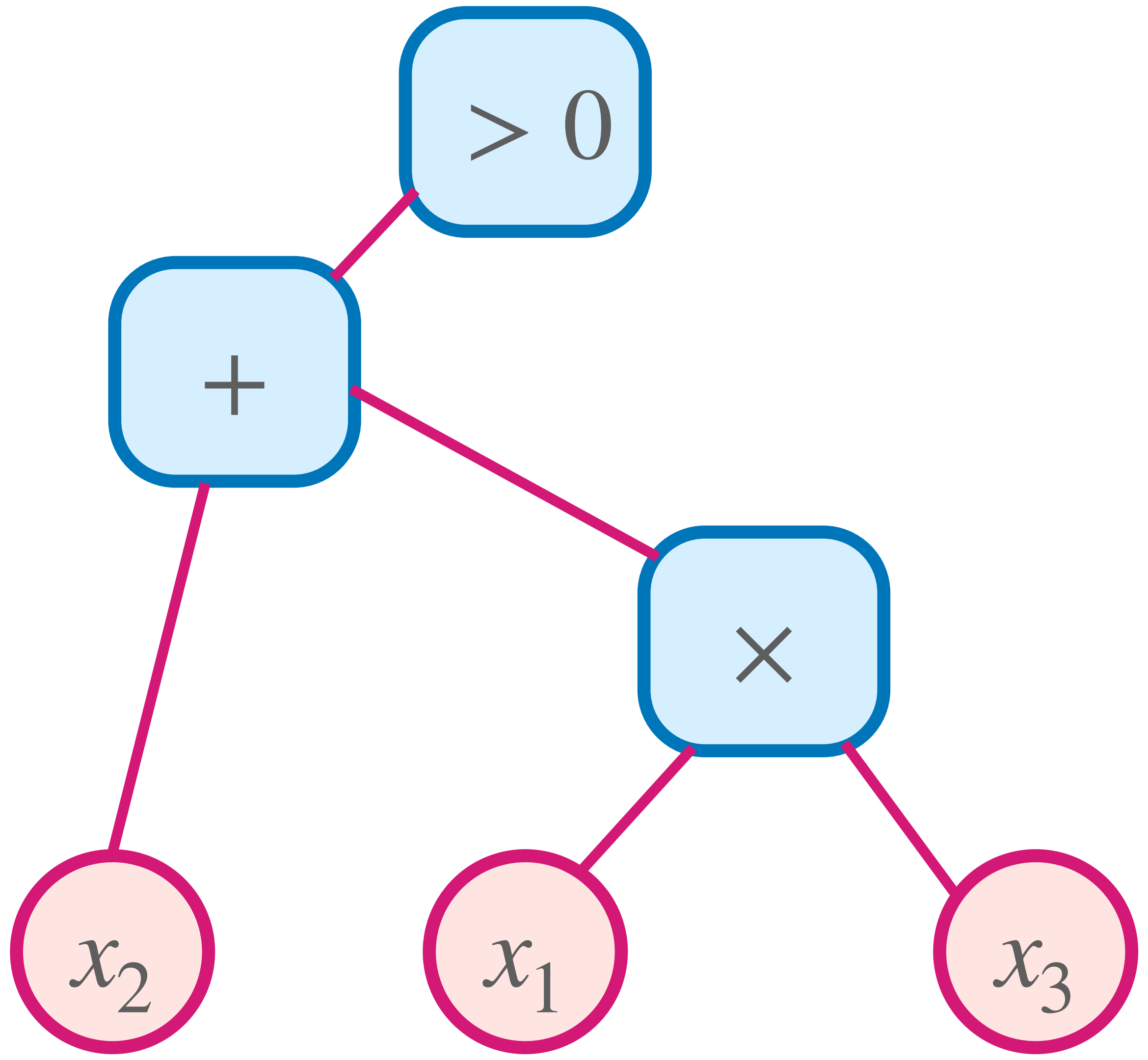
Monotone Real Circuits [P97]: A monotone real circuit computing $f : \{0,1\}^n \to \{0,1\}$ is a circuit in which gates are any monotone real-valued function $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ on (at most) two inputs!

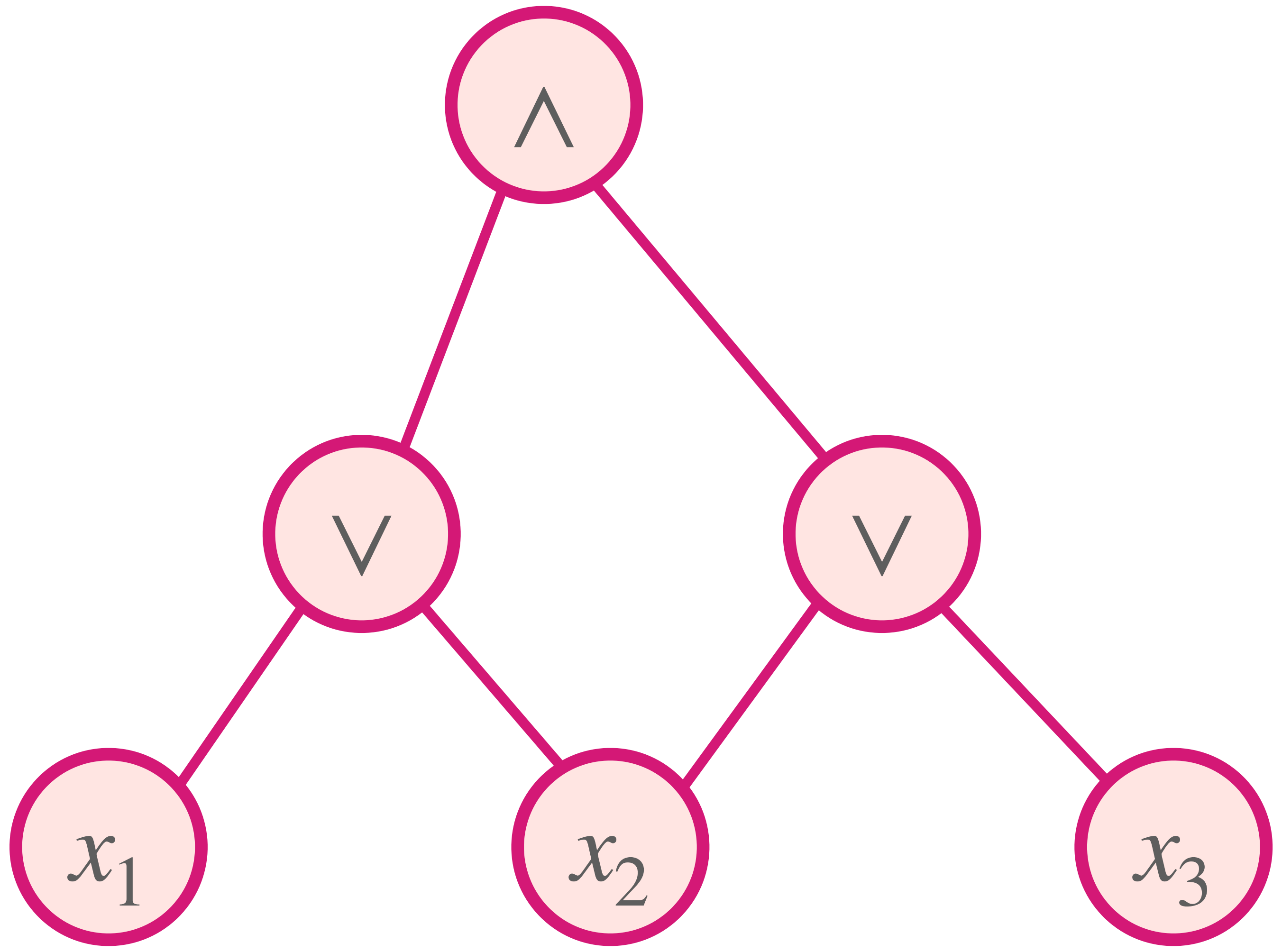


Feasible Interpolation by Real Circuits

Monotone Circuits: boolean circuits using only $\wedge$ and $\vee$ gates — no $\neg$

Monotone Real Circuits [P97]: A monotone real circuit computing $f : \{0,1\}^n \rightarrow \{0,1\}$ is a circuit in which gates are any monotone real-valued function $g : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ on (at most) two inputs!





Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Proof: Recall that y-variables occurs only positively in $A(x, y)$.

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Proof: Recall that y-variables occurs only positively in $A(x, y)$.

Calculate $-\delta_0$ using same argument as in the previous lemma, observing that each operation is monotone.

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Proof: Recall that y-variables occurs only positively in $A(x, y)$.

Calculate $-\delta_0$ using same argument as in the previous lemma, observing that each operation is monotone.

Let $ax + by + cz \geq d$ be a line in Π

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Proof: Recall that y-variables occurs only positively in $A(x, y)$.

Calculate $-\delta_0$ using same argument as in the previous lemma, observing that each operation is monotone.

Let $ax + by + cz \geq d$ be a line in Π

$\rightarrow$ Axiom of $A(x, \alpha)$: then $-\delta_0 = b\alpha - d$. Monotone in α as only positive y-vars.

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Proof: Recall that y-variables occurs only positively in $A(x, y)$.

Calculate $-\delta_0$ using same argument as in the previous lemma, observing that each operation is monotone.

Let $ax + by + cz \geq d$ be a line in Π

$\rightarrow$ Axiom of $A(x, \alpha)$: then $-\delta_0 = b\alpha - d$. Monotone in α as only positive y-vars.

$\rightarrow$ Non-neg combo: From $-\delta_0'$ and $-\delta_0''$ derive $-\delta_0 = \gamma'(-\delta_0') + \gamma''(-\delta_0'')$

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Proof: Recall that y-variables occurs only positively in $A(x, y)$.

Calculate $-\delta_0$ using same argument as in the previous lemma, observing that each operation is monotone.

Let $ax + by + cz \geq d$ be a line in Π

→ Axiom of $A(x, \alpha)$: then $-\delta_0 = b\alpha - d$. Monotone in α as only positive y-vars.

→ Non-neg combo: From $-\delta_0'$ and $-\delta_0''$ derive $-\delta_0 = \gamma'(-\delta_0') + \gamma''(-\delta_0'')$

→ Cut: From $-\delta_0'$ derive $\lceil -\delta_0'/t \rceil$

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Proof: Suppose we have calculated $-\delta_0$ for the last line in Π. What do we output?

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Proof: Suppose we have calculated $-\delta_0$ for the last line in Π. What do we output?

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Proof: Suppose we have calculated $-\delta_0$ for the last line in Π. What do we output?

$$I_F(y) = \begin{cases} 0 & \text{if } A(x, \alpha) \text{ is unsatisfiable} \\ 1 & \text{if } B(\alpha, z) \text{ is unsatisfiable} \end{cases}$$

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Proof: Suppose we have calculated $-\delta_0$ for the last line in Π. What do we output?

$$I_F(y) = \begin{cases} 0 & \text{if } A(x, \alpha) \text{ is unsatisfiable} \\ 1 & \text{if } B(\alpha, z) \text{ is unsatisfiable} \end{cases}$$

If $0 \geq \delta_0$ then $A(x, \alpha)$ is satisfiable , so we should output 1

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x,y) \wedge B(y,z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Proof: Suppose we have calculated $-\delta_0$ for the last line in Π. What do we output?

$$I_F(y) = \begin{cases} 0 & \text{if } A(x,\alpha) \text{ is unsatisfiable} \\ 1 & \text{if } B(\alpha,z) \text{ is unsatisfiable} \end{cases}$$

If $0 \geq \delta_0$ then $A(x,\alpha)$ is satisfiable , so we should output 1

$\Longrightarrow$ Let the output gate of the circuit be $-\delta_0 \geq 0$.

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Lower bounds on the size of monotone real circuits computing $I_F \Longrightarrow$ Cutting Planes lower bounds on split formula F!

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Lower bounds on the size of monotone real circuits computing $I_F \implies$ Cutting Planes lower bounds on split formula F!

Recall $Clique - Color$ formula

Interpolant function: $$I_F(y) = \begin{cases} 0 & \text{if } Clique(x, \alpha) \text{ is unsatisfiable} \\ 1 & \text{if } Color(\alpha, z) \text{ is unsatisfiable} \end{cases}$$

Upshot: Lower bounds on **Clique** imply lower bounds on I_F

Feasible Interpolation by Real Circuits

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$

Lower bounds on the size of monotone real circuits computing $I_F \Longrightarrow$ Cutting Planes lower bounds on split formula F!

Recall $Clique - Color$ formula

Interpolant function: $$I_F(y) = \begin{cases} 0 & \text{if } Clique(x, \alpha) \text{ is unsatisfiable} \\ 1 & \text{if } Color(\alpha, z) \text{ is unsatisfiable} \end{cases}$$

Upshot: Lower bounds on **Clique** imply lower bounds on I_F

Thm[P97]**:** Any monotone real circuit computing Clique requires exponential size

Interpolation for any Formula

Thm: If there is a size s CP proof Π of $F = A(x, y) \wedge B(y, z)$ then there is a size poly(s) monotone real circuit computing $I_F(y)$