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

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

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

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

\[ F = C_1 \land \ldots \land C_m \quad \Rightarrow \quad P_F = \{x : Ax \geq b\} \]
Recap of Last Time

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

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

Consider proof systems for proving that a polytope does not contain integer points
Recap of Last Time

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

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

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
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

→ 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

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

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

→ 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 $2^{\mathrm{polylog } n}$ (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

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

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

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

→ **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**
→ **Prove** this fact using cutting planes!

**Rules**

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

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

  $$cx \geq d$$
Cutting Planes Proofs

Suppose $Ax \geq b$ has no integer solutions
→ Prove this fact using cutting planes!

Rules
Deduce new inequalities from old ones by:

○ Non-negative linear Combination:

\[
\begin{align*}
    ax & \geq b, \ cx \geq d \\
    (\alpha a + \beta c)x & \geq \alpha b + \beta d, \quad \alpha, \beta \in \mathbb{Z}^\geq0
\end{align*}
\]
Cutting Planes Proofs

Suppose $Ax \geq b$ has no integer solutions → Prove this fact using cutting planes!

Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

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

  Preserves all points in $P$
Cutting Planes Proofs

Suppose $Ax \geq b$ has no integer solutions
→ Prove this fact using cutting planes!

Rules
Deduce new inequalities from old ones by:

○ Non-negative linear Combination:
  \[
  ax \geq b, \quad cx \geq d \\
  \frac{(\alpha a + \beta c)x}{\alpha a + \beta c} \geq \frac{\alpha b + \beta d}{\alpha a + \beta c}, \quad \alpha, \beta \in \mathbb{Z} \geq 0
  \]

○ Cut:
  \[
  \frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil}, \text{ if } d \in \mathbb{Z} \geq 0 \text{ divides } a
  \]
Suppose $Ax \geq b$ has no integer solutions
→ Prove this fact using cutting planes!

Rules
Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**
  \[
  ax \geq b, \ cx \geq d \\
  (\alpha a + \beta c)x \geq \alpha b + \beta d,
  \alpha, \beta \in \mathbb{Z}_{\geq 0}
  \]

- **Cut:**
  \[
  ax \geq b \\
  (a/d)x \geq \lceil b/d \rceil,
  \text{if } d \in \mathbb{Z}_{\geq 0} \text{ divides } a
  \]
Cutting Planes Proofs

Suppose $Ax \geq b$ has no integer solutions
→ 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:
\[
\frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil}, \quad \text{if } d \in \mathbb{Z}_{\geq 0} \text{ divides } a
\]

Preserves integer points in $P$
Cutting Planes Proofs

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

→ **Prove** this fact using cutting planes!

**Rules**

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**
  
  \[
  ax \geq b, \quad cx \geq d \\
  (\alpha a + \beta c)x \geq \alpha b + \beta d, \quad \alpha, \beta \in \mathbb{Z}^\geq 0
  \]

- **Cut:**
  
  \[
  ax \geq b \\
  (a/d)x \geq \lceil b/d \rceil, \quad \text{if } d \in \mathbb{Z}^\geq 0 \text{ divides } a
  \]

**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
→ Prove this fact using cutting planes!

Rules
Deduce new inequalities from old ones by:

○ Non-negative linear Combination:
$$ax \geq b, \ c x \geq d \Rightarrow (\alpha a + \beta c)x \geq \alpha b + \beta d, \ \alpha, \beta \in \mathbb{Z}^\geq$$

○ Cut:
$$ax \geq b \Rightarrow (a/d)x \geq \lceil b/d \rceil, \text{ if } d \in \mathbb{Z}^\geq \text{ divides } a$$

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
→ Prove this fact using cutting planes!

Rules

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**
  \[
  \frac{ax \geq b, cx \geq d}{(\alpha a + \beta c)x \geq \alpha b + \beta d}, \quad \alpha, \beta \in \mathbb{Z}^\geq
  \]

- **Cut:**
  \[
  \frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil}, \text{ if } d \in \mathbb{Z}^\geq \text{ divides } a
  \]

- Equivalently, the empty polytope
Suppose $Ax \geq b$ has no integer solutions
→ 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:**
  
  \[
  \frac{ax \geq b}{(a/d)x \geq \lceil b/d \rceil}, \text{ if } d \in \mathbb{Z}_{\geq 0} \text{ divides } a
  \]

- equivalently, the **empty polytope**
Cutting Planes Proofs

Suppose \( Ax \geq b \) has no integer solutions
→ **Prove** this fact using cutting planes!

**Rules**

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**

\[
(ax \geq b, cx \geq d) \implies (\alpha a + \beta c)x \geq \alpha b + \beta d, \quad \alpha, \beta \in \mathbb{Z}^\geq
\]

- **Cut:**

\[
ax \geq b \implies (a/d)x \geq \lceil b/d \rceil, \quad \text{if } d \in \mathbb{Z}^\geq \text{ divides } a
\]

---

**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**

→ **Prove** this fact using cutting planes!

**Rules**

Deduce new inequalities from old ones by:

- **Non-negative linear Combination:**
  
  \[
  (\alpha a + \beta c)x \geq \alpha b + \beta d, \quad \alpha, \beta \in \mathbb{Z}_\geq 0
  \]

- **Cut:**
  
  \[
  (a/d)x \geq \left\lfloor b/d \right\rfloor, \quad \text{if } d \in \mathbb{Z}_\geq 0 \text{ divides } a
  \]

**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

→ **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:**

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

**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.

Where $f_F : \{0,1\}^m \rightarrow \{0,1\}$ 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.

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

Where $f_F : \{0,1\}^m \rightarrow \{0,1\}$ 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

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

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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:

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

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

For simplicity, we’ll restrict to the case of \textit{split} formulas:

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

Where \( A, B \) are CNF and all \( y \) variables occur \textit{positively} in \( A \).

The Function Computed

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

Define monotone \textit{“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

\[ P \text{-proof of split } F \]
\[ \text{of size } s \]
\[ \rightarrow \]
\[ C_P \text{-Computation of } I_F \]
\[ \text{of size } \text{poly}(s) \]

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

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

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

“There is a graph containing both a $k$-clique and a $(k - 1)$-coloring”
Split Formulas

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“There is a graph containing both a \( k \)-clique and a \( (k - 1) \)-coloring”
Split Formulas

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

“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

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“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 \( G(y) = (V, E) \):
  \[ e \in E \text{ iff } y_e = 1 \]
**Split Formulas**

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“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 \( G(y) = (V, E) \):
  \[ e \in E \text{ iff } y_e = 1 \]
- \( x \in \{0,1\}^{nk} \) defines a \( k \)-clique in \( G(y) \):
Split Formulas

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“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

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“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

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“There is a graph containing both a \( k \)-clique and a \((k - 1)\)-coloring”

- \( y \in \{0,1\}^{n \choose 2} \) 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

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“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 \( n = 3, k = 3 \)
Split Formulas

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“There is a graph containing both a \textit{k-clique} and a \textit{(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

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“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 \( \text{Clique}(x, y) \) constraints
Split Formulas

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“There is a graph containing both a \textit{k-clique} and a \textit{(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

E.g. Clique\((x, y)\) \& 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 \(G(y)\):
  \[x_{v,t} = 1 \iff v \text{ is } t\text{-th member of clique}\]
- \(z \in \{0, 1\}^{n(k-1)}\) defines a \((k – 1)\)-coloring of \(G(y)\):
  \[v \text{ has color } c \iff z_{v,c} = 1\]

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

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“There is a graph containing both a \( k \)-clique and a \( (k - 1) \)-coloring”

- \( y \in \{0,1\}^{n\choose 2} \) 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 \( G(y) \):
  \[ x_{v,t} = 1 \iff v \text{ is } t\text{-th member of clique} \]
- \( z \in \{0,1\}^{n(k-1)} \) defines a \( (k - 1) \)-coloring of \( G(y) \):
  \[ v \text{ has color } c \iff z_{v,c} = 1 \]

Constraints of \( \text{Clique}(x, y) \)

- \( \forall t \in [k]: \quad \forall v \in [n] \quad x_{v,t} \quad \text{— some vertex is the } t\text{-th clique member} \)
Split Formulas

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“There is a graph containing both a \( k \)-clique and a \((k - 1)\)-coloring”

- \( y \in \{0,1\}^{n \choose 2} \) 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 \( \text{Clique}(x, y) \)

- \( \forall t \in [k]: \forall v \in [n] \) \( x_{v,t} \) — some vertex is the \( t \)-th clique member
- \( \forall v, \forall t \neq \ell: \neg x_{v,t} \lor \neg x_{v,\ell} \) — no vertex is the \( t \)-th and \( \ell \)-th clique member
Split Formulas

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“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 \( \text{Clique}(x, y) \)

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

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

e.g. \( \text{Clique}(x, y) \land \text{Color}(y, z)_{n, k} \)

“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 \( \text{Color}(y, z) \)

- \( \forall v \in [n]: \bigvee_{c \in [k-1]} z_{v,c} \) \( \quad \) every vertex gets a color
**Split Formulas**

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“There is a graph containing both a \textit{k-clique} and a \textit{(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 \textit{Color}(y, z)\**

- \( \forall v \in [n]: \forall c \in [k-1] \) \( z_{v,c} \) — every vertex gets a color
- \( \forall v, \forall c \neq d: \neg z_{v,c} \lor \neg z_{v,d} \) — no vertex gets two different colors
Split Formulas

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“There is a graph containing both a \( k \)-clique and a \( (k - 1) \)-coloring”

- \( y \in \{0,1\}^{n(k-1)} \) 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 \( \text{Color}(y, z) \)

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

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“There is a graph containing both a \( k \)-clique and a \((k - 1)\)-coloring”

• \( y \in \{0,1\}^{n/2} \) 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

\[ F(x, y, z) = A(x, y) \land B(y, z) \]

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

“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:

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

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

\[ S_1^2(\alpha) \rightarrow \text{Boolean circuits} \]
Feasible Interpolation Theorems

[R95] Defined interpolation as a general method.

[K97] Defined interpolation as a general method.

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

Resolution $\rightarrow$ Monotone circuits
Feasible Interpolation Theorems

[R95] Defined interpolation as a general method.

[K97] Defined interpolation as a general method.

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

Resolution $\rightarrow$ Monotone circuits

Nullstellensatz $\rightarrow$ Monotone span programs
<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[R95]</td>
<td>-</td>
</tr>
<tr>
<td>[K97]</td>
<td>Defined interpolation as a general method.</td>
</tr>
<tr>
<td>[PS97]</td>
<td>-</td>
</tr>
<tr>
<td>[BPR97]</td>
<td>-</td>
</tr>
</tbody>
</table>

- Feasible Interpolation Theorems
- $S^2_1(\alpha) \rightarrow$ Boolean circuits
- Resolution $\rightarrow$ Monotone circuits
- Nullstellensatz $\rightarrow$ Monotone span programs
- Cutting Planes* $\rightarrow$ Monotone circuits
Feasible Interpolation Theorems

[R95] Defined interpolation as a general method.

[K97] Designed Monotone circuits $\Rightarrow$ Boolean circuits

[PS97] Monotone span programs $\Rightarrow$ Nullstellensatz

[BPR97] Nullstellensatz $\Rightarrow$ Monotone circuits

[P97] Monotone circuits $\Rightarrow$ Monotone real circuits

Resolution $\Rightarrow$ Monotone circuits

Cutting Planes* $\Rightarrow$ Monotone circuits

Cutting Planes $\Rightarrow$ Monotone real circuits
Feasible Interpolation Theorems

[R95] Defined interpolation as a general method.

[K97] Defined interpolation as a general method.

[PS97] Defined interpolation as a general method.

[BPR97] Defined interpolation as a general method.

[P97] Defined interpolation as a general method.

\[ S_1^2(\alpha) \longrightarrow \text{Boolean circuits} \]

Resolution \( \longrightarrow \) Monotone circuits

Nullstellensatz \( \longrightarrow \) Monotone span programs

Cutting Planes* \( \longrightarrow \) Monotone circuits

Cutting Planes \( \longrightarrow \) Monotone real circuits

Only worked for split formulas!
### Feasible Interpolation Theorems

<table>
<thead>
<tr>
<th>Reference</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>[R95]</td>
<td></td>
</tr>
<tr>
<td>[K97]</td>
<td>Defined interpolation as a general method.</td>
</tr>
<tr>
<td>[PS97]</td>
<td></td>
</tr>
<tr>
<td>[BPR97]</td>
<td></td>
</tr>
<tr>
<td>[P97]</td>
<td></td>
</tr>
<tr>
<td>[FPPR17, HP17]</td>
<td>Generalized Interpolation to work for <strong>any</strong> unsatisfiable formula</td>
</tr>
</tbody>
</table>

- $S_1^2(\alpha) \rightarrow$ Boolean circuits
- Resolution $\rightarrow$ Monotone circuits
- Nullstellensatz $\rightarrow$ Monotone span programs
- Cutting Planes* $\rightarrow$ Monotone circuits
- Cutting Planes $\rightarrow$ Monotone real circuits

- $CC_{O(\log n)} \leftrightarrow$ Monotone circuits
- $RCC_1 \leftrightarrow$ Monotone real circuits
# Feasible Interpolation Theorems

<table>
<thead>
<tr>
<th>Reference</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>[R95]</td>
<td>$S_1^2(\alpha) \rightarrow$ Boolean circuits</td>
</tr>
<tr>
<td>[K97]</td>
<td>Resolution $\rightarrow$ Monotone circuits</td>
</tr>
<tr>
<td>[PS97]</td>
<td>Nullstellensatz $\rightarrow$ Monotone span programs</td>
</tr>
<tr>
<td>[BPR97]</td>
<td>Cutting Planes* $\rightarrow$ Monotone circuits</td>
</tr>
<tr>
<td>[P97]</td>
<td>Cutting Planes $\rightarrow$ Monotone real circuits</td>
</tr>
<tr>
<td>[FPPR17, HP17]</td>
<td>Generalized Interpolation to work for any unsatisfiable formula</td>
</tr>
<tr>
<td>[FGGR22]</td>
<td>$CC_{O(\log n)} \leftrightarrow$ Monotone circuits</td>
</tr>
<tr>
<td></td>
<td>$RCC_1 \leftrightarrow$ Monotone real circuits</td>
</tr>
<tr>
<td></td>
<td>Sherali-Adams $\rightarrow$ Extended Formulations</td>
</tr>
</tbody>
</table>

[K97] Defined interpolation as a general method.
Feasible Interpolation Theorems

- \([R95]\)
- \([K97]\) Defined interpolation as a general method.
- \([PS97]\)
- \([BPR97]\)
- \([P97]\)
- \([FPPR17, HP17]\) Generalized Interpolation to work for any unsatisfiable formula
- \([FGGR22]\)
- \([FGR22 \text{ unpublished}]\)

\[S_1^2(\alpha) \iff \text{Boolean circuits}\]

Resolution \(\longrightarrow\) Monotone circuits

Nullstellensatz \(\longrightarrow\) Monotone span programs

Cutting Planes* \(\longrightarrow\) Monotone circuits

Cutting Planes \(\longrightarrow\) Monotone real circuits

\(CC_{O(\log n)} \iff \text{Monotone circuits}\)

\(RCC_1 \iff \text{Monotone real circuits}\)

Sherali-Adams \(\longrightarrow\) Extended Formulations

SoS \(\longrightarrow\) Semidefinite EFs
Feasible Interpolation For CP

[P97] Cutting Planes $\Rightarrow$ Monotone real circuits
Feasible Interpolation For CP

Remainder of today:
1. Prove this theorem
Feasible Interpolation For CP

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

We will first prove the following simpler lemma
We will first prove the following simpler lemma:

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

We will first prove the following simpler lemma

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

The following claim will allow us to define our algorithm
We will first prove the following simpler lemma

**Lemma:** There is a time $\text{poly}(s)$ algorithm which given a split formula $F = A(x, y) \land B(y, z)$, a size $s$ CP proof of $\Pi$ 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 $\Pi$ there are constants $\delta_0, \delta_1$ s.t.
We will first prove the following simpler lemma

**Lemma:** There is a time $\text{poly}(s)$ algorithm which given a split formula $F = A(x, y) \land B(y, z)$, a size $s$ CP proof of $\Pi$ 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 $\Pi$ there are constants $\delta_0, \delta_1$ 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

We will first prove the following simpler lemma

Lemma: There is a time poly(s) algorithm which given a split formula
\[ F = A(x, y) \land B(y, z), \] a size s CP proof of \( \Pi \) 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 \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in poly(s) time from \( \Pi \) and \( \alpha \)
Feasible Interpolation For CP

We will first prove the following simpler lemma

**Lemma:** There is a time poly(s) algorithm which given a split formula $F = A(x, y) \land B(y, z)$, a size $s$ CP proof of $\Pi$ 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 $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly(s) time from $\Pi$ and $\alpha$
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
\(F = A(x, y) \land B(y, z)\), a size \(s\) CP proof of \(\Pi\) of \(F\), and \(\alpha \in \{0,1\}^y\) outputs \(I_{F}(\alpha)\)

**Claim:** For each inequality \(ax + by + cz \geq d\) in \(\Pi\) there are constants \(\delta_0, \delta_1\) 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. \(\delta_0, \delta_1\) are constructible in poly\((s)\) time from \(\Pi\) and \(\alpha\)
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 $F = A(x, y) \land B(y, z)$, a size $s$ CP proof of $\Pi$ of $F$, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$.

**Claim:** For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly(s) time from $\Pi$ and $\alpha$
3. $\delta_0 + \delta_1 \geq d - b\alpha$
Lemma: There is a time \( \text{poly}(s) \) algorithm which given a split formula \( F = A(x, y) \land B(y, z) \), a size \( s \) CP proof of \( \Pi \) of \( F \), and \( \alpha \in \{0,1\}^y \) outputs \( I_F(\alpha) \).

Claim: For each inequality \( ax + by + cz \geq d \) in \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in \( \text{poly}(s) \) time from \( \Pi \) and \( \alpha \).
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 $F = A(x, y) \land B(y, z)$, a size $s$ CP proof of $\Pi$ of $F$, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$.

Claim: For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly(s) time from $\Pi$ and $\alpha$.
3. $\delta_0 + \delta_1 \geq d - b\alpha$.

Proof of Lemma: Claim allows us to extract from $\Pi$ a proof of

- $A(x, \alpha)$ if $A(x, \alpha)$ is unsatisfiable.
- $B(\alpha, z)$ if $B(\alpha, z)$ is unsatisfiable.
Feasible Interpolation For CP

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

**Claim:** For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly(s) time from $\Pi$ and $\alpha$
3. $\delta_0 + \delta_1 \geq d - b\alpha$

**Proof of Lemma:** Claim allows us to extract from $\Pi$ 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 $\text{poly}(s)$ algorithm which given a split formula $F = A(x, y) \land B(y, z)$, a size $s$ CP proof of $\Pi$ of $F$, and $\alpha \in \{0, 1\}^y$ outputs $I_F(\alpha)$.

Claim: For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in $\text{poly}(s)$ time from $\Pi$ and $\alpha$
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof of Lemma: Applying claim to the last line $0 \geq 1$ of $\Pi$, we get
Feasible Interpolation For CP

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

**Claim:** For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in $\text{poly}(s)$ time from $\Pi$ and $\alpha$.
3. $\delta_0 + \delta_1 \geq d - b\alpha$.

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

- Derivation of $0 \geq \delta_0$ from $A(x, \alpha)$.
- Derivation of $0 \geq \delta_1$ from $B(\alpha, z)$. 
Lemma: There is a time \text{poly}(s) algorithm which given a split formula
\[ F = A(x, y) \land B(y, z) \], a size \( s \) CP proof of \( \Pi \) of \( F \), and \( \alpha \in \{0,1\}^y \) outputs \( I_F(\alpha) \).

Claim: For each inequality \( ax + by + cz \geq d \) in \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in \( \text{poly}(s) \) time from \( \Pi \) and \( \alpha \)
3. \( \delta_0 + \delta_1 \geq d - b\alpha \)

Proof of Lemma: Applying claim to the last line \( 0 \geq 1 \) of \( \Pi \), we get
- Derivation of \( 0 \geq \delta_0 \) from \( A(x, \alpha) \) with \( \delta_0 + \delta_1 \geq 1 \)
- Derivation of \( 0 \geq \delta_1 \) from \( B(\alpha, z) \)
Lemma: There is a time \( \text{poly}(s) \) algorithm which given a split formula 
\[ F = A(x, y) \land B(y, z), \] 
a size \( s \) CP proof of \( \Pi \) of \( F \), and \( \alpha \in \{0,1\}^y \) outputs \( I_F(\alpha) \).

Claim: For each inequality \( ax + by + cz \geq d \) in \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in \( \text{poly}(s) \) time from \( \Pi \) and \( \alpha \)
3. \( \delta_0 + \delta_1 \geq d - b\alpha \)

Proof of Lemma: Applying claim to the last line \( 0 \geq 1 \) of \( \Pi \), we get
- Derivation of \( 0 \geq \delta_0 \) from \( A(x, \alpha) \) with \( \delta_0 + \delta_1 \geq 1 \)
- Derivation of \( 0 \geq \delta_1 \) from \( B(\alpha, z) \)

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 \( \text{poly}(s) \) algorithm which given a split formula \( F = A(x, y) \land B(y, z) \), a size \( s \) CP proof of \( \Pi \) of \( F \), and \( \alpha \in \{0,1\}^y \) outputs \( I_F(\alpha) \).

Claim: For each inequality \( ax + by + cz \geq d \) in \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in \( \text{poly}(s) \) time from \( \Pi \) and \( \alpha \)
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 $\text{poly}(s)$ algorithm which given a split formula $F = A(x, y) \land B(y, z)$, a size $s$ CP proof of $\Pi$ of $F$, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$.

**Claim:** For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in $\text{poly}(s)$ time from $\Pi$ and $\alpha$
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 $\text{poly}(s)$ algorithm which given a split formula $F = A(x, y) \land B(y, z)$, a size $s$ CP proof of $\Pi$ of $F$, and $\alpha \in \{0,1\}^y$ outputs $I_F(\alpha)$.

**Claim:** For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in $\text{poly}(s)$ time from $\Pi$ and $\alpha$.
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 $\delta_0$ and $\delta_1$ in time $\text{poly}(s)$. 


Feasible Interpolation For CP

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

**Claim:** For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in $\text{poly}(s)$ time from $\Pi$ and $\alpha$.
3. $\delta_0 + \delta_1 \geq d - b\alpha$.

**Proof of Lemma:** The poly-time algorithm:

1. Constructs $\delta_0$ and $\delta_1$ in time $\text{poly}(s)$.
2. If $\delta_0 > 0$ then $A(x, \alpha)$ is unsatisfiable and we output 0.
Feasible Interpolation For CP

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

Claim: For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in $\text{poly}(s)$ time from $\Pi$ and $\alpha$
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 $\delta_0$ and $\delta_1$ in time $\text{poly}(s)$
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 $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly($s$) time from $\Pi$ and $\alpha$
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 $\Pi_0$ be the axiom $ax \geq d - b\alpha$ of $A(x, \alpha)$
  $\rightarrow$ Let $\delta_1 = 0$ and the proof $\Pi_1$ be the trivial axiom $0 \geq 0$
- If $ax + by + cz \geq d$ is an axiom of $B(y, z)$ then $a = 0$
  $\rightarrow$ Let $\delta_0 = 0$ and $\Pi_0$ be $0 \geq 0$
  $\rightarrow$ Let $\delta_1 = d - b\alpha$ and $\Pi_1$ 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 \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in \( \text{poly}(s) \) time from \( \Pi \) and \( \alpha \)
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 $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly$(s)$ time from $\Pi$ and $\alpha$
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 $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly$(s)$ time from $\Pi$ and $\alpha$
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 $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly$(s)$ time from $\Pi$ and $\alpha$
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 $\Pi_0$ 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 \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in poly\((s)\) time from \( \Pi \) and \( \alpha \)
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 \( \Pi_0 \) be the axiom \( ax \geq d - b\alpha \) of \( A(x, \alpha) \)
  - \( \rightarrow \) Let \( \delta_1 = 0 \) and the proof \( \Pi_1 \) be the trivial axiom \( 0 \geq 0 \)
Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly$(s)$ time from $\Pi$ and $\alpha$
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 $\Pi_0$ be the axiom $ax \geq d - b\alpha$ of $A(x, \alpha)$
  - $\rightarrow$ Let $\delta_1 = 0$ and the proof $\Pi_1$ 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 \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in \( \text{poly}(s) \) time from \( \Pi \) and \( \alpha \)
3. \( \delta_0 + \delta_1 \geq d - b\alpha \)

Proof: Cut: Suppose that \( ax + by + cz \geq d \) is deduced by cut in \( \Pi \)

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

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 $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly($s$) time from $\Pi$ and $\alpha$
3. $\delta_0 + \delta_1 \geq d - b\alpha$

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

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

For $t$ dividing $a', b', c'$

And by induction we have derived

From $A(x, \alpha)$: $a'x \geq \delta'_0$

From $B(\alpha, z)$: $c'z \geq \delta'_1$
Feasible Interpolation For CP

Claim: For each inequality \( ax + by + cz \geq d \) in \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in \( \text{poly}(s) \) time from \( \Pi \) and \( \alpha \)
3. \( \delta_0 + \delta_1 \geq d - b\alpha \)

Proof: Cut: Suppose that \( ax + by + cz \geq d \) is deduced by cut in \( \Pi \)

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

For \( t \) dividing \( a', b', c' \)

And by induction we have derived
- From \( A(x, \alpha) \): \( a'x \geq \delta'_0 \)
- From \( B(\alpha, z) \): \( c'z \geq \delta'_1 \)

With \( \delta'_0 + \delta'_1 \geq d' - b'\alpha \)
Feasible Interpolation For CP

Claim: For each inequality \( ax + by + cz \geq d \) in \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in poly(s) time from \( \Pi \) and \( \alpha \)
3. \( \delta_0 + \delta_1 \geq d - b\alpha \)

Proof: Cut: Suppose that \( ax + by + cz \geq d \) is deduced by cut in \( \Pi \)

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

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 \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in poly(s) time from \( \Pi \) and \( \alpha \)
3. \( \delta_0 + \delta_1 \geq d - b\alpha \)

**Proof:** **Cut:** Suppose that \( ax + by + cz \geq d \) is deduced by cut in \( \Pi \)

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

For \( t \) dividing \( a', b', c' \)

And by induction we have derived

From \( A(x, \alpha) \):
\[
a'x \geq \delta_0' \quad \rightarrow \text{Cut} \rightarrow \quad (a'/t)x \geq \lceil \delta_0'/t \rceil = \delta_0
\]

From \( B(\alpha, z) \):
\[
c'z \geq \delta_1' \quad \rightarrow \text{Cut} \rightarrow \quad (c'/t)z \geq \lceil \delta_1'/t \rceil = \delta_1
\]

With \( \delta_0' + \delta_1' \geq d' - b'\alpha \)
Feasible Interpolation For CP

Claim: For each inequality \( ax + by + cz \geq d \) in \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in poly(s) time from \( \Pi \) and \( \alpha \)
3. \( \delta_0 + \delta_1 \geq d - b\alpha \)

Proof: Cut: Suppose that \( ax + by + cz \geq d \) is deduced by cut in \( \Pi \)

\[
\begin{align*}
\frac{a'x + b'y + c'z \geq d'}{(a'/t)x + (b'/t)y + (c'/t)z \geq \lceil d'/t \rceil}
\end{align*}
\]

For \( t \) dividing \( a', b', c' \)

And by induction we have derived

\[
\begin{align*}
(a'/t)x \geq \lceil \delta'_0/t \rceil = \delta_0 \\
(c'/t)z \geq \lceil \delta'_1/t \rceil = \delta_1
\end{align*}
\]

Invariant:

\[
\delta_0 + \delta_1
\]
Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly($s$) time from $\Pi$ and $\alpha$
3. $\delta_0 + \delta_1 \geq d - b\alpha$

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

$$a'x + b'y + c'z \geq d'$$

For $t$ dividing $a', b', c'$

And by induction we have derived

$$a'x \geq \delta'_0 \quad \rightarrow \quad \text{Cut} \rightarrow \quad (a'/t)x \geq \lceil \delta'_0/t \rceil = \delta_0$$

$$c'z \geq \delta'_1 \quad \rightarrow \quad \text{Cut} \rightarrow \quad (c'/t)z \geq \lceil \delta'_1/t \rceil = \delta_1$$

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 $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly$(s)$ time from $\Pi$ and $\alpha$
3. $\delta_0 + \delta_1 \geq d - b\alpha$

**Proof:** If $ax + by + cz \geq d$ is deduced by cut in $\Pi$

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

For $t$ dividing $a', b', c'$

And by induction we have derived

\[
\begin{align*}
a'x & \geq \delta'_0 \quad \rightarrow \text{Cut} \rightarrow \quad (a'/t)x \geq \lceil \delta'_0/t \rceil = \delta_0 \\
c'z & \geq \delta'_1 \quad \rightarrow \text{Cut} \rightarrow \quad (c'/t)z \geq \lceil \delta'_1/t \rceil = \delta_1
\end{align*}
\]

**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 $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly($s$) time from $\Pi$ and $\alpha$
3. $\delta_0 + \delta_1 \geq d - b\alpha$

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

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

For $t$ dividing $a', b', c'$

And by induction we have derived

\[
\begin{align*}
    a'x \geq \delta'_0 & \quad \rightarrow \text{Cut} \rightarrow \quad (a'/t)x \geq \lceil \delta'_0/t \rceil = \delta_0 \\
c'z \geq \delta'_1 & \quad \rightarrow \text{Cut} \rightarrow \quad (c'/t)z \geq \lceil \delta'_1/t \rceil = \delta_1
\end{align*}
\]

**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 \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in poly(s) time from \( \Pi \) and \( \alpha \)
3. \( \delta_0 + \delta_1 \geq d - b\alpha \)

Proof: Cut: Suppose that \( ax + by + cz \geq d \) is deduced by cut in \( \Pi \)

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

For \( t \) dividing \( a', b', c' \)

And by induction we have derived

\[
\begin{align*}
(a'/t)x & \geq \lceil \delta_0/t \rceil = \delta_0 \\
(c'/t)z & \geq \lceil \delta_1/t \rceil = \delta_1
\end{align*}
\]

Invariant:

\[
\delta_0 + \delta_1 = \lceil \delta_0/t \rceil + \lceil \delta_1/t \rceil \geq \lceil (\delta_0' + \delta_1')/t \rceil \geq \lceil (d - b\alpha)/t \rceil = \lfloor d/t \rfloor - b\alpha/t
\]
Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly$(s)$ time from $\Pi$ and $\alpha$
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 \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in \( \text{poly}(s) \) time from \( \Pi \) and \( \alpha \)
3. \( \delta_0 + \delta_1 \geq d - b\alpha \)

Proof: Non-negative Linear Combination:

\[
\frac{a'x + b'y + c'z \geq d', \quad a''x + b''y + c''z \geq d''}{\gamma'a' + \gamma'a''}x + (\gamma'b' + \gamma'b'')y + (\gamma'c' + \gamma'c'')z \geq \gamma'd' + \gamma''d''}
\]

For \( \gamma', \gamma'' \geq 0 \)

\[ax + by + cz \geq d\]
Feasible Interpolation For CP

Claim: For each inequality \( ax + by + cz \geq d \) in \( \Pi \) there are constants \( \delta_0, \delta_1 \) 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. \( \delta_0, \delta_1 \) are constructible in poly(\( s \)) time from \( \Pi \) and \( \alpha \)
3. \( \delta_0 + \delta_1 \geq d - b\alpha \)

Proof: Non-negative Linear Combination:

\[
\begin{align*}
\gamma'a'x + \gamma'b'y + \gamma'c'z & \geq \gamma'd' \\
\gamma'a''x + \gamma'b''y + \gamma'c''z & \geq \gamma'd''
\end{align*}
\]

\[
(\gamma'a' + \gamma'a'')x + (\gamma'b' + \gamma'b'')y + (\gamma'c' + \gamma'c'')z \geq \gamma'd' + \gamma''d''
\]

For \( \gamma', \gamma'' \geq 0 \)

And by induction we have derived

\[
\begin{align*}
a'x & \geq \delta'_0, \quad a''x \geq \delta''_0 \quad \text{From} \ A(x, \alpha) \\
c'z & \geq \delta'_1, \quad c''z \geq \delta''_1 \quad \text{From} \ B(\alpha, z)
\end{align*}
\]
Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly($s$) time from $\Pi$ and $\alpha$
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Non-negative Linear Combination:

$$a'x + b'y + c'z \geq d', \quad a''x + b''y + c''z \geq d''$$

$$(\gamma'a' + \gamma''a'')x + (\gamma'b' + \gamma''b'')y + (\gamma'c' + \gamma''c'')z \geq \gamma'd' + \gamma''d''$$

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

And by induction we have derived

- $a'x \geq \delta'_0, \quad a''x \geq \delta''_0 \rightarrow$ non-neg combo $\rightarrow (\gamma'a' + \gamma''a'')x \geq \gamma'\delta'_0 + \gamma''\delta''_0$
- $c'z \geq \delta'_1, \quad c''z \geq \delta''_1 \rightarrow$ non-neg combo $\rightarrow (\gamma'c' + \gamma''c'')z \geq \gamma'\delta'_1 + \gamma''\delta''_1$
Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly$(s)$ time from $\Pi$ and $\alpha$
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Non-negative Linear Combination:
\[
\frac{a'x + b'y + c'z \geq d'}{a''x + b''y + c''z \geq d''} \quad \Rightarrow \quad (\gamma'a' + \gamma''a'')x + (\gamma'b' + \gamma''b'')y + (\gamma'c' + \gamma''c'')z \geq \gamma'd' + \gamma''d''
\]
For $\gamma', \gamma'' \geq 0$

And by induction we have derived
\[
\begin{align*}
a'x &\geq \delta_0', \quad a''x \geq \delta_0'' \quad \Rightarrow \quad \text{non-neg combo} \quad \Rightarrow \quad (\gamma'a' + \gamma''a'')x \geq \gamma'\delta_0' + \gamma''\delta_0'' = \delta_0 \\
c'z &\geq \delta_1', \quad c''z \geq \delta_1'' \quad \Rightarrow \quad \text{non-neg combo} \quad \Rightarrow \quad (\gamma'c' + \gamma''c'')z \geq \gamma'\delta_1' + \gamma''\delta_1'' = \delta_1
\end{align*}
\]
Feasible Interpolation For CP

Claim: For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly($s$) time from $\Pi$ and $\alpha$
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Non-negative Linear Combination:

\[
\begin{align*}
 a'x + b'y + c'z & \geq d', \\
 a''x + b''y + c''z & \geq d''
\end{align*}
\]

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

\[
(\gamma'a' + \gamma''a'')x + (\gamma'b' + \gamma''b'')y + (\gamma'c' + \gamma''c'')z \geq \gamma'd' + \gamma''d''
\]

And by induction we have derived

\[
\begin{align*}
 a'x & \geq \delta'_0, \quad a''x \geq \delta''_0 \rightarrow \text{non-neg combo} \rightarrow (\gamma'a' + \gamma''a'')x \geq \gamma'\delta'_0 + \gamma''\delta''_0 = \delta_0 \\
 c'z & \geq \delta'_1, \quad c''z \geq \delta''_1 \rightarrow \text{non-neg combo} \rightarrow (\gamma'c' + \gamma''c'')z \geq \gamma'\delta'_1 + \gamma''\delta''_1 = \delta_1
\end{align*}
\]

Invariant: $\delta_0 + \delta_1$
Feasible Interpolation For CP

**Claim:** For each inequality $ax + by + cz \geq d$ in $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly($s$) time from $\Pi$ and $\alpha$
3. $\delta_0 + \delta_1 \geq d - ba$

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

\[
\begin{align*}
(a'x + b'y + c'z) & \geq d', \\
(a''x + b''y + c''z) & \geq d''
\end{align*}
\]
\[
(\gamma'a' + \gamma''a'')x + (\gamma'b' + \gamma''b'')y + (\gamma'c' + \gamma''c'')z \geq \gamma'd' + \gamma''d''
\]

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

And by induction we have derived

\[
\begin{align*}
(a'x & \geq \delta_0', \ a''x \geq \delta_0'') \rightarrow \text{non-neg combo} \rightarrow (\gamma'a' + \gamma''a'')x \geq \gamma'\delta_0' + \gamma''\delta_0'' = \delta_0 \\
(c'z & \geq \delta_1', \ c''z \geq \delta_1'') \rightarrow \text{non-neg combo} \rightarrow (\gamma'c' + \gamma''c'')z \geq \gamma'\delta_1' + \gamma''\delta_1'' = \delta_1
\end{align*}
\]

**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 $\Pi$ there are constants $\delta_0, \delta_1$ 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. $\delta_0, \delta_1$ are constructible in poly$(s)$ time from $\Pi$ and $\alpha$
3. $\delta_0 + \delta_1 \geq d - b\alpha$

Proof: Non-negative Linear Combination:

$$a'x + b'y + c'z \geq d', \quad a''x + b''y + c''z \geq d''$$

$$(\gamma'a' + \gamma''a'')x + (\gamma'b' + \gamma''b'')y + (\gamma'c' + \gamma''c'')z \geq \gamma'd' + \gamma''d''$$

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

And by induction we have derived

$$a'x \geq \delta_0', \quad a''x \geq \delta_0'' \rightarrow \text{non-neg combo} \rightarrow (\gamma'a' + \gamma''a'')x \geq \gamma'\delta_0' + \gamma''\delta_0'' = \delta_0$$

$$c'z \geq \delta_1', \quad c''z \geq \delta_1'' \rightarrow \text{non-neg combo} \rightarrow (\gamma'c' + \gamma''c'')z \geq \gamma'\delta_1' + \gamma''\delta_1'' = \delta_1$$

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 $\text{poly}(s)$ algorithm which given a split formula $F = A(x, y) \land B(y, z)$, a size $s$ CP proof of $\Pi$ 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 \( \text{poly}(s) \) algorithm which given a split formula 
\[ F = A(x, y) \land B(y, z), \] 
a size \( s \) CP proof of \( \Pi \) of \( F \), and \( \alpha \in \{0,1\}^y \) outputs \( I_F(\alpha) \).

This lemma is overkill!

→ Don’t need the full power of poly-time algorithms to construct \( \delta_0, \delta_1 \).
Feasible Interpolation by Real Circuits

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

This lemma is overkill!

→ Don’t need the full power of poly-time algorithms to construct $\delta_0, \delta_1$.
→ In order to calculate $\delta_0, \delta_1$, only need a computational model which supports addition, multiplication, division, ceiling.
Feasible Interpolation by Real Circuits

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

**This lemma is overkill!**

→ Don’t need the full power of \text{poly-time algorithms} to construct \( \delta_0, \delta_1 \).

→ In order to calculate \( \delta_0, \delta_1 \), 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 $\land$ and $\lor$ gates — no $\neg$
Feasible Interpolation by Real Circuits

Monotone Circuits: boolean circuits using only \( \land \) and \( \lor \) gates — no \( \neg \)
Feasible Interpolation by Real Circuits

Monotone Circuits: boolean circuits using only $\land$ and $\lor$ 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

Monotone Circuits: boolean circuits using only $\land$ and $\lor$ gates — no $\lnot$

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

**Thm:** If there is a size $s$ CP proof $\Pi$ of $F = A(x, y) \land B(y, z)$ then there is a size $\text{poly}(s)$ 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 $\Pi$ of $F = A(x, y) \land B(y, z)$ then there is a size $\text{poly}(s)$ 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 $\Pi$ of $F = A(x, y) \land B(y, z)$ then there is a size $\text{poly}(s)$ 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 $\Pi$
Feasible Interpolation by Real Circuits

**Thm:** If there is a size $s$ CP proof $\Pi$ of $F = A(x, y) \land B(y, z)$ then there is a size $\text{poly}(s)$ 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 $\Pi$
$\rightarrow$ Axiom of $A(x, \alpha)$: then $-\delta_0 = b\alpha - d$. Monotone in $\alpha$ as only positive $y$-vars.
Feasible Interpolation by Real Circuits

**Thm:** If there is a size \( s \) CP proof \( \Pi \) of \( F = A(x, y) \land B(y, z) \) then there is a size \( \text{poly}(s) \) 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 \( \Pi \)
\( \rightarrow \) Axiom of \( A(x, \alpha) \): then \(-\delta_0 = b\alpha - d\). Monotone in \( \alpha \) 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 $\Pi$ of $F = A(x, y) \land B(y, z)$ then there is a size $\text{poly}(s)$ 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 $\Pi$

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

$\rightarrow$ Non-neg combo: From $-\delta'_0$ and $-\delta''_0$ derive $-\delta_0 = \gamma'(-\delta'_0) + \gamma''(-\delta''_0)$

$\rightarrow$ Cut: From $-\delta'_0$ derive $[-\delta'_0/t]$
Feasible Interpolation by Real Circuits

**Thm:** If there is a size $s$ CP proof $\Pi$ of $F = A(x, y) \land B(y, z)$ then there is a size $\text{poly}(s)$ monotone real circuit computing $I_F(y)$.

**Proof:** Suppose we have calculated $-\delta_0$ for the last line in $\Pi$. What do we output?
Feasible Interpolation by Real Circuits

**Thm:** If there is a size $s$ CP proof $\Pi$ of $F = A(x, y) \land B(y, z)$ then there is a size $\text{poly}(s)$ monotone real circuit computing $I_F(y)$.

**Proof:** Suppose we have calculated $-\delta_0$ for the last line in $\Pi$. What do we output?
Feasible Interpolation by Real Circuits

**Thm:** If there is a size $s$ CP proof $\Pi$ of $F = A(x, y) \land B(y, z)$ then there is a size $\text{poly}(s)$ monotone real circuit computing $I_F(y)$

**Proof:** Suppose we have calculated $-\delta_0$ for the last line in $\Pi$. 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 $\Pi$ of $F = A(x, y) \land B(y, z)$ then there is a size $\text{poly}(s)$ monotone real circuit computing $I_F(y)$

**Proof:** Suppose we have calculated $-\delta_0$ for the last line in $\Pi$. 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 $\Pi$ of $F = A(x, y) \land B(y, z)$ 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 $\Pi$. 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

$\implies$ Let the output gate of the circuit be $-\delta_0 \geq 0$. 
Thm: If there is a size $s$ CP proof $\Pi$ of $F = A(x, y) \land B(y, z)$ then there is a size $\text{poly}(s)$ monotone real circuit computing $I_F(y)$.
Feasible Interpolation by Real Circuits

**Thm:** If there is a size $s$ CP proof $\Pi$ of $F = A(x, y) \land B(y, z)$ then there is a size $\text{poly}(s)$ 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 $\Pi$ of $F = A(x, y) \land B(y, z)$ then there is a size $\text{poly}(s)$ monotone real circuit computing $I_F(y)$.

Lower bounds on the size of monotone real circuits computing $I_F$ imply cutting planes lower bounds on split formula $F$!

Recall *Clique — Color* formula

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

**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 \( \Pi \) of \( F = A(x, y) \land B(y, z) \) then there is a size \( \text{poly}(s) \) monotone real circuit computing \( I_F(y) \)

Lower bounds on the size of monotone real circuits computing \( I_F \) imply lower bounds on Cutting Planes lower bounds on split formula \( F \)!

Recall *Clique – Color* formula

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

**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 $\Pi$ of $F = A(x, y) \land B(y, z)$ then there is a size $\text{poly}(s)$ monotone real circuit computing $I_F(y)$