On the Power and Limitations of Branch and Cut

Noah Fleming

Joint work with
Göös, Tan, Impagliazzo, Pitassi, Robere, Wigderson

In This Talk...

- Algorithm analysis from proof complexity
- The proof complexity of integer programming
  - Branch-and-cut & Stabbing Planes
- Cutting Planes vs Stabbing Planes
- Short proofs of $\mathbb{F}_q$ linear equations
- Deep Cutting Planes Proofs

Algorithm Analysis from Proofs

Idea: Formalize the techniques used in a class of algorithms A as a proof system P

Algorithm Analysis from Proofs

Idea: Formalize the techniques used in a class of algorithms A as a proof system P

▷ Hides practical details of algorithms

Algorithm Analysis from Proofs

Idea: Formalize the techniques used in a class of algorithms A as a proof system P

- ▷ Hides practical details of algorithms
- ▷ Lower bounds on P-proofs → lower bounds on runtime of A

Algorithm Analysis from Proofs

Idea: Formalize the techniques used in a class of algorithms A as a proof system P

- e.g. Algorithms for SAT
  - CDCL and Resolution

Algorithm Analysis from Proofs

Idea: Formalize the techniques used in a class of algorithms A as a proof system P

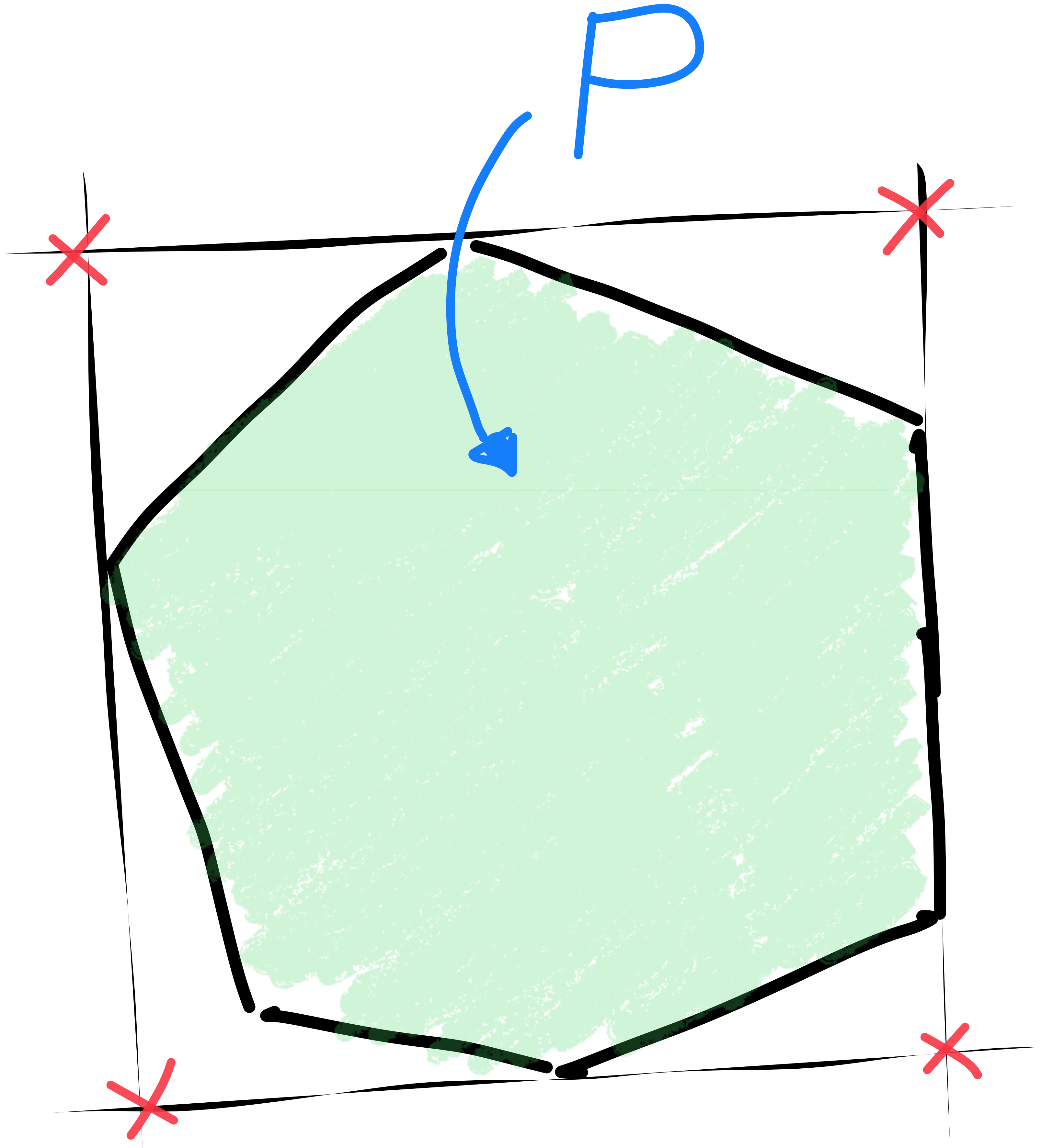
- e.g. Algorithms for SAT
  - CDCL and Resolution
- e.g. Algorithms for Integer Programming
  - Chvátal-Gomory Cutting Planes and Cutting Planes

Integer Programming

Integer-programming: Given $Ax \geq b$ find $x \in \mathbb{Z}^n$, $Ax \geq b$

Integer Programming

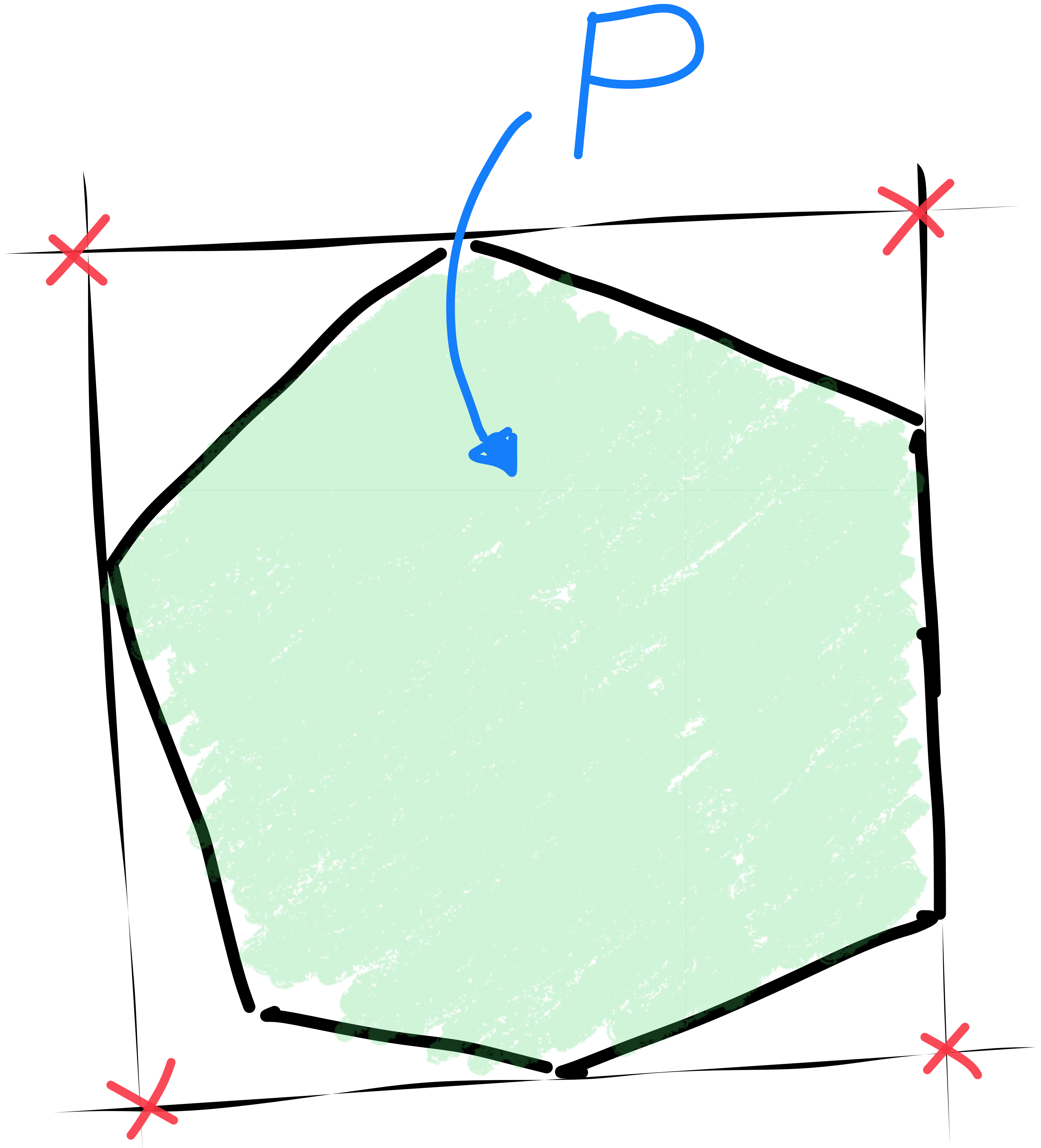
Integer-programming: Given $P = \{x : Ax \geq b\}$ find $x \in P \cap \mathbb{Z}^n$



Integer Programming

Integer-programming: Given $P = \{x : Ax \geq b\}$ find $x \in P \cap \mathbb{Z}^n$

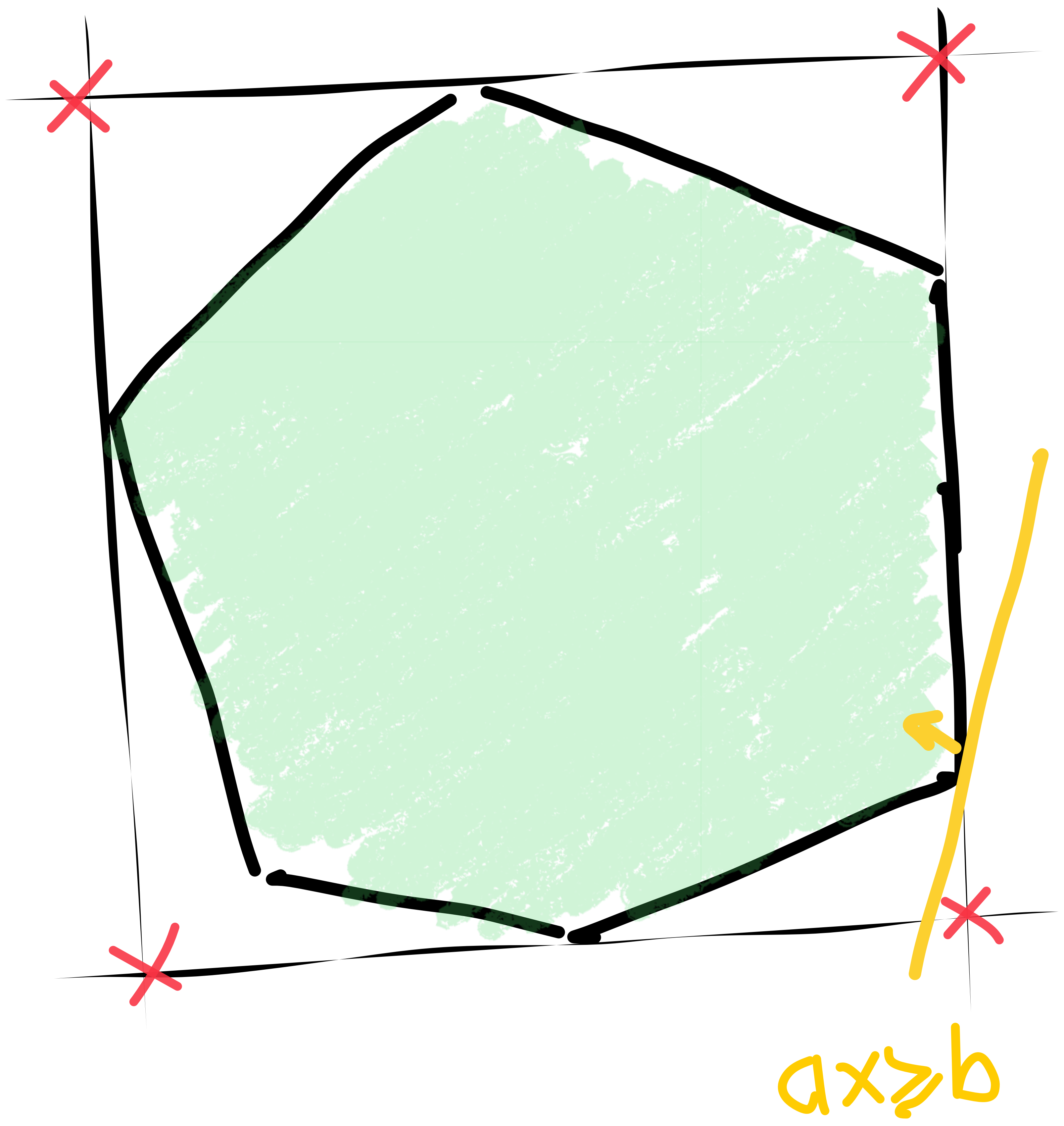
A Classic approach: Chvátal-Gomory Cutting Planes



Chvátal-Gomory Cutting Planes

Integer-programming: Given $P = \{x : Ax \geq b\}$ find $x \in P \cap \mathbb{Z}^n$

A Classic approach: Chvátal-Gomory Cutting Planes



CG-Cut: If $ax \geq b$ is valid for P

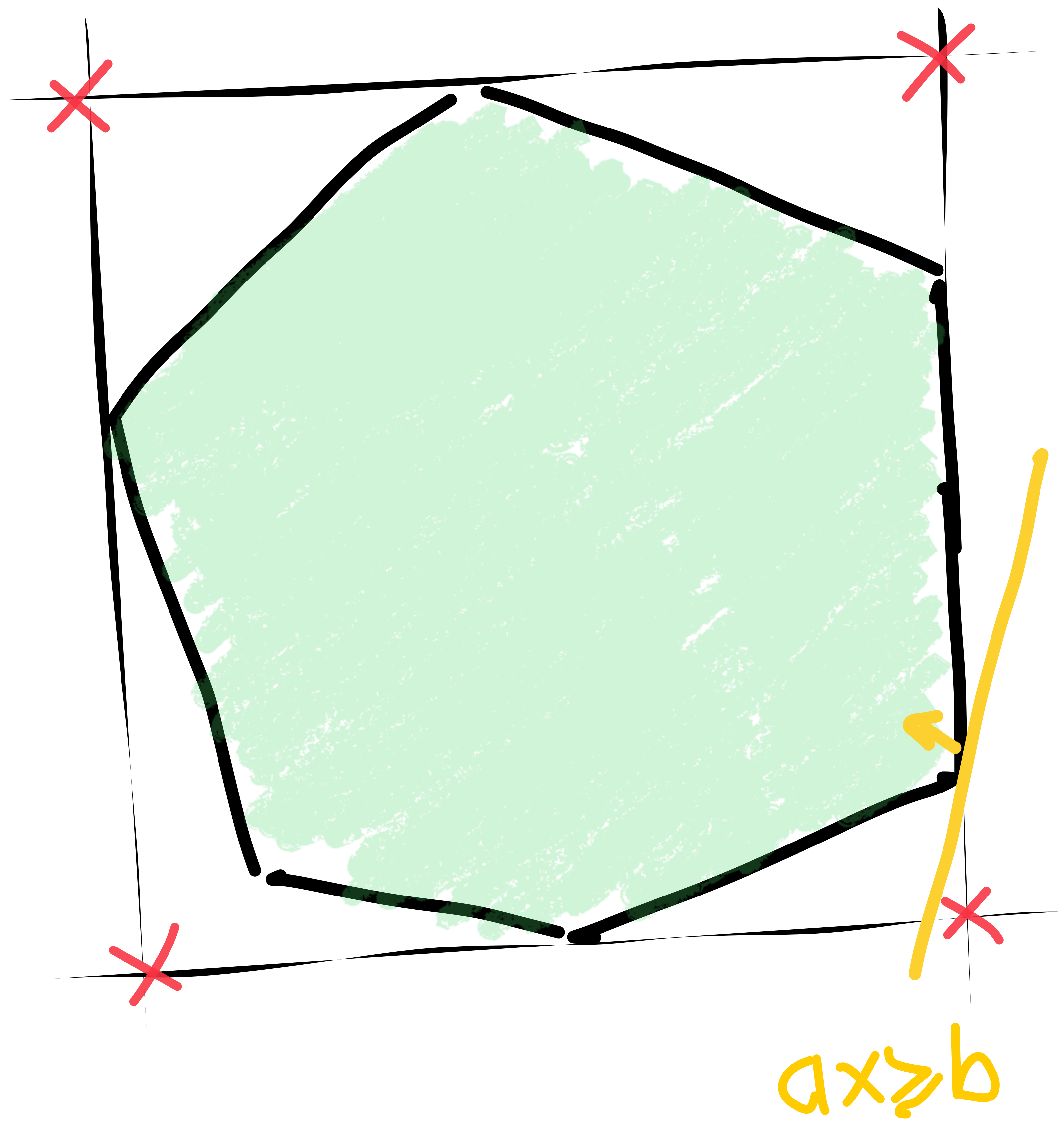
Chvátal-Gomory Cutting Planes

Integer-programming: Given $P=\{x: Ax\geq b\}$ find $x\in P\cap\mathbb{Z}^n$

A Classic approach: Chvátal-Gomory Cutting Planes

$a\in\mathbb{Z}^n, b\in\mathbb{R}$

CG-Cut: If $ax\geq b$ is valid for P



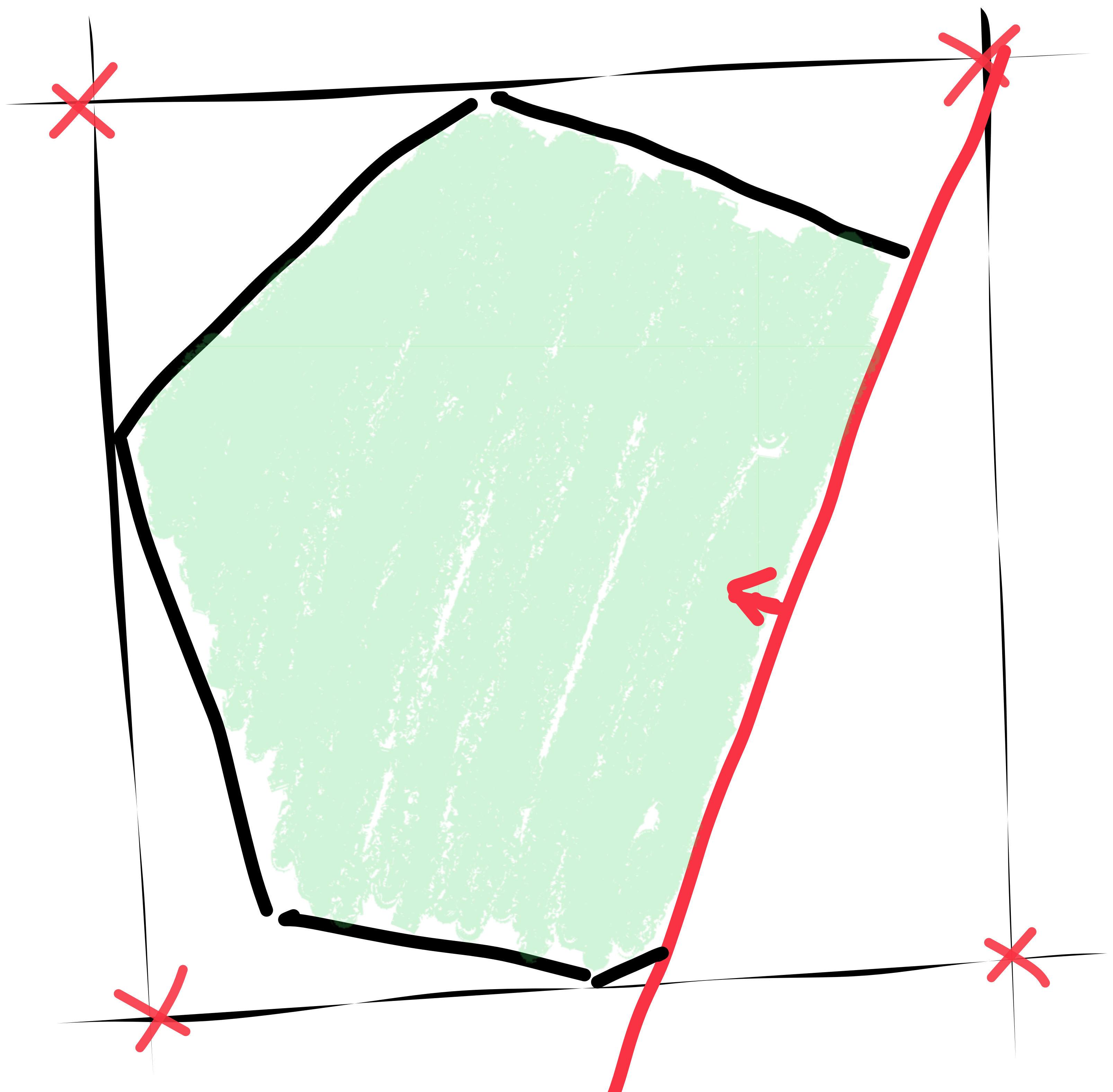
Chvátal-Gomory Cutting Planes

Integer-programming: Given $P = \{x : Ax \geq b\}$ find $x \in P \cap \mathbb{Z}^n$

A Classic approach: Chvátal-Gomory Cutting Planes

$a \in \mathbb{Z}^n$, $b \in \mathbb{R}$

CG-Cut: If $ax \geq b$ is valid for P then $ax \geq \lceil b \rceil$ is a CG-cut



Chvátal-Gomory Cutting Planes

Integer-programming: Given $P = \{x : Ax \geq b\}$ find $x \in P \cap \mathbb{Z}^n$

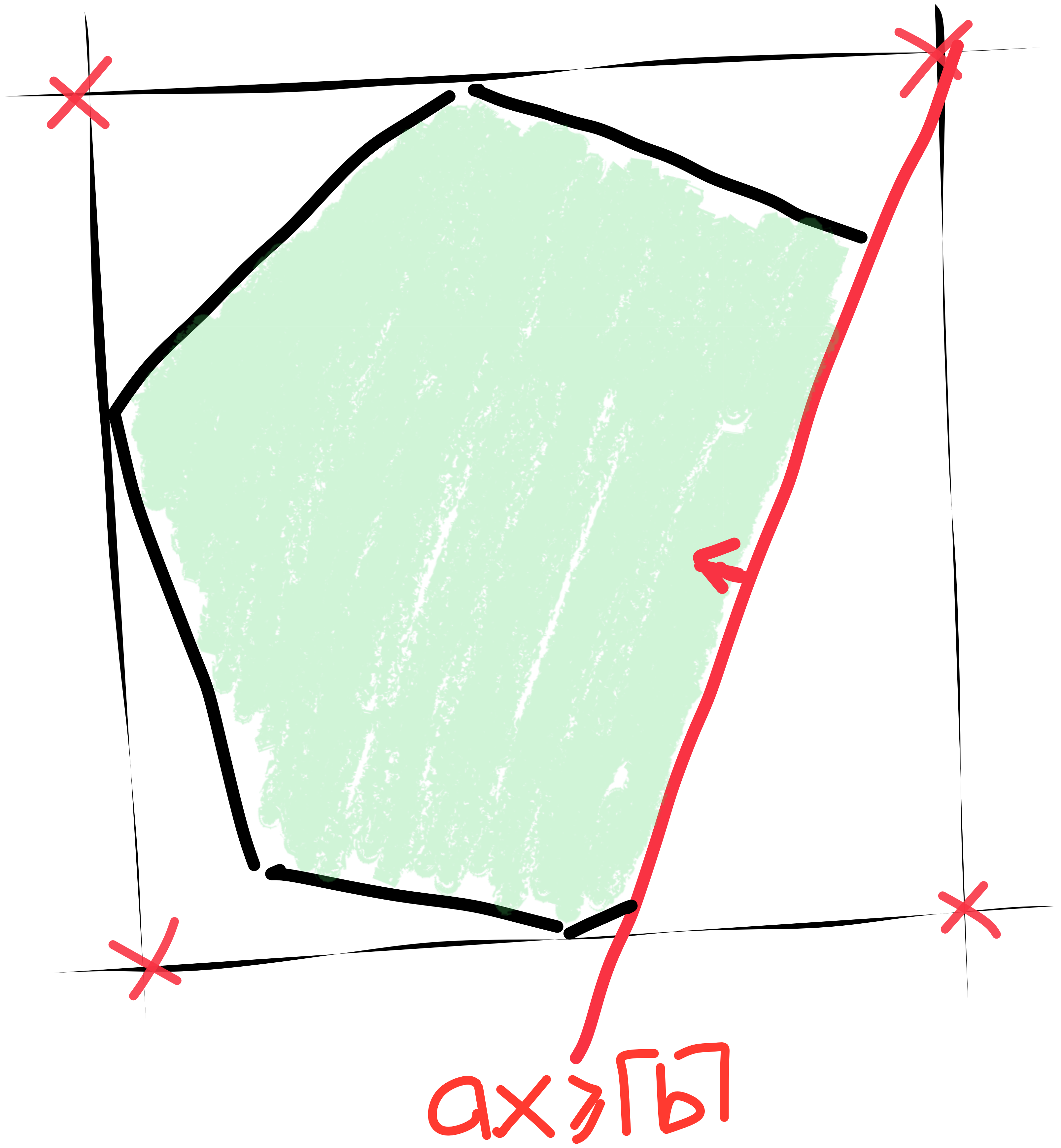
A Classic approach: Chvátal-Gomory Cutting Planes

$a \in \mathbb{Z}^n, b \in \mathbb{R}$

CG-Cut: If $ax \geq b$ is valid for P

then $ax \geq \lceil b \rceil$ is a CG-cut

→ Preserves integer solutions to P



Chvátal-Gomory Cutting Planes

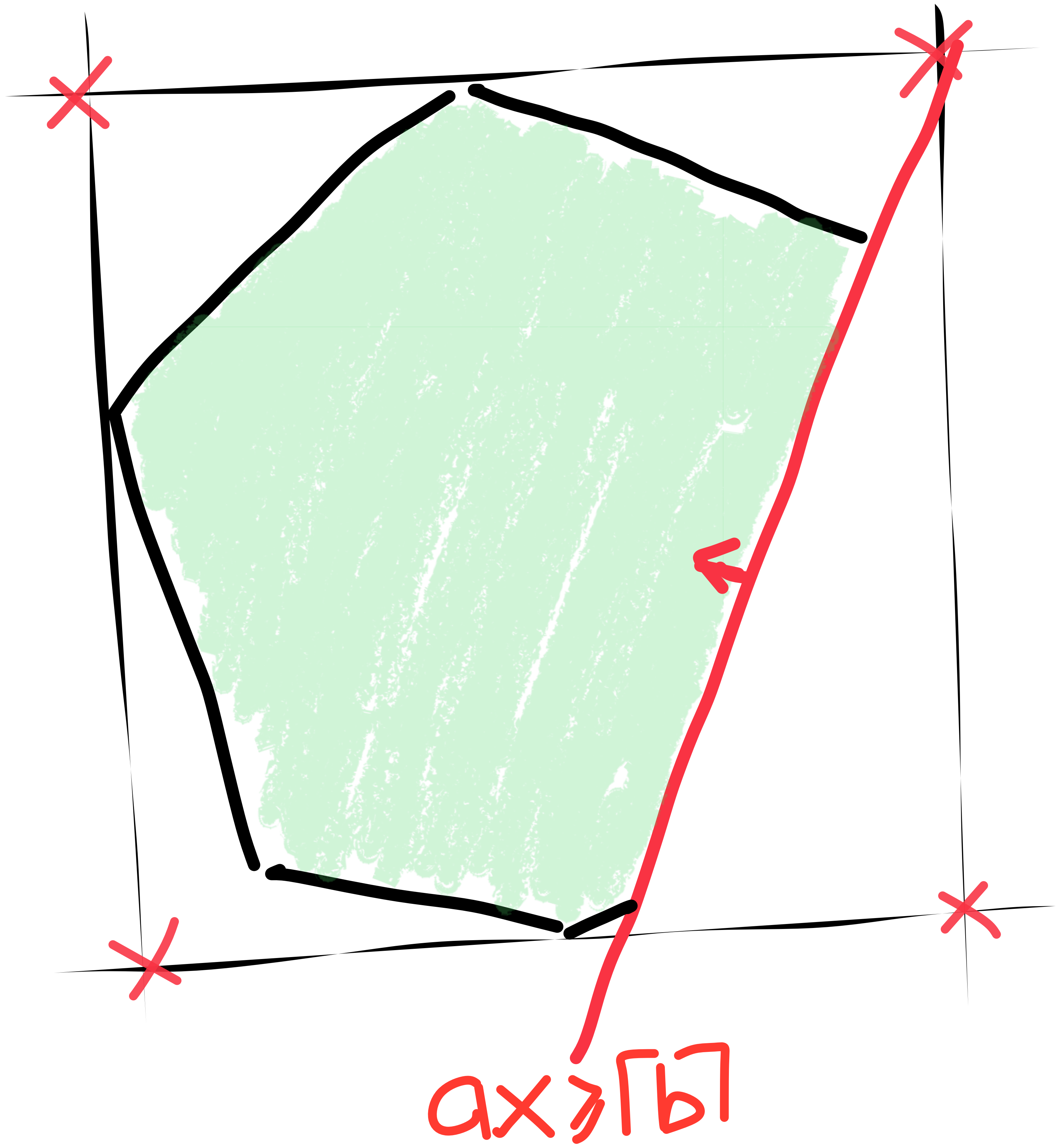
Integer-programming: Given $P = \{x : Ax \geq b\}$ find $x \in P \cap \mathbb{Z}^n$

CG-Cutting Planes

Heuristically add CG-cuts to P

CG-Cut: If $ax \geq b$ is valid for P then $ax \geq \lceil b \rceil$ is a CG-cut

→ Preserves integer solutions to P



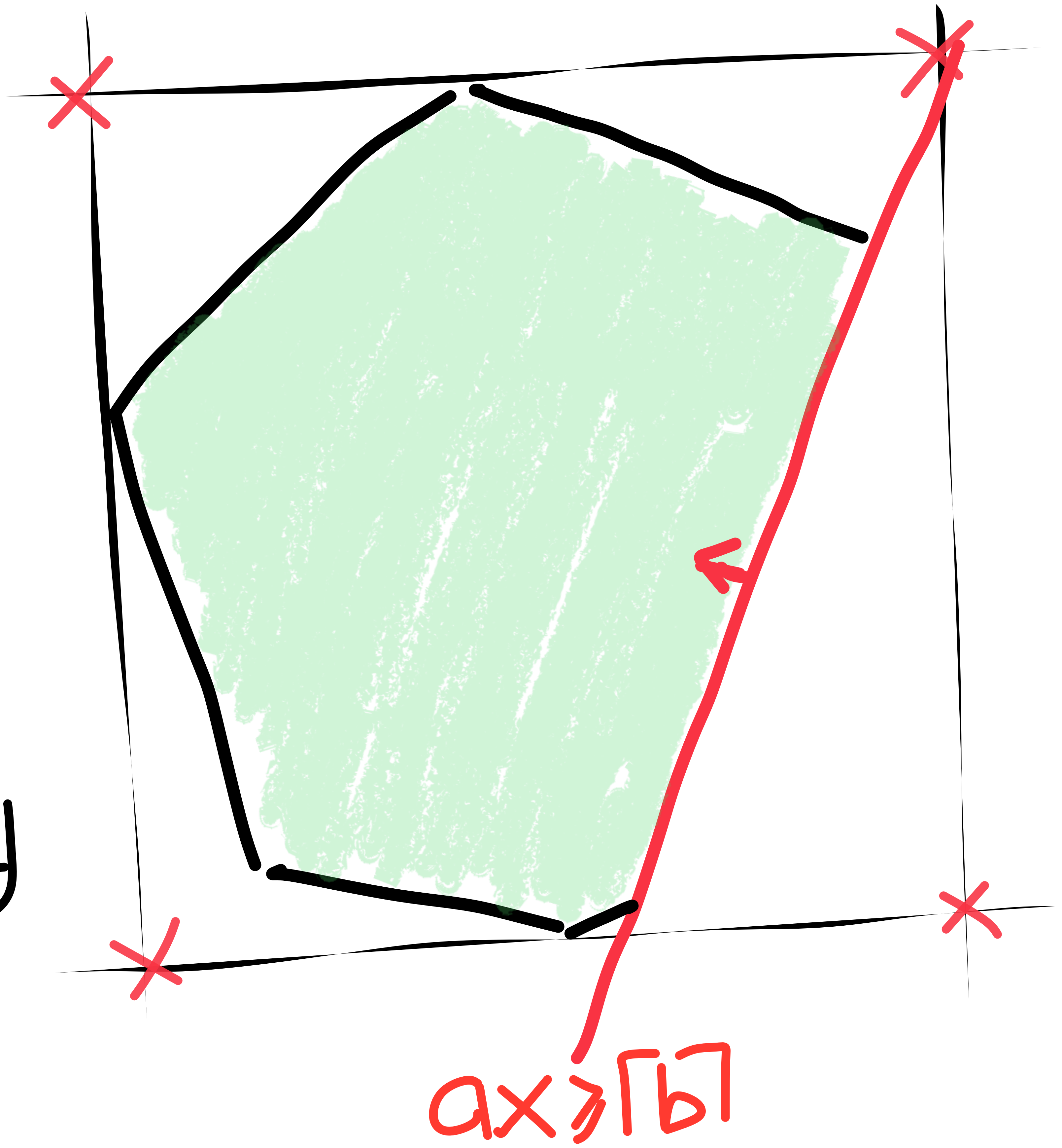
Chvátal-Gomory Cutting Planes

Integer-programming: Given $P = \{x : Ax \geq b\}$ find $x \in P \cap \mathbb{Z}^n$

CG-Cutting Planes

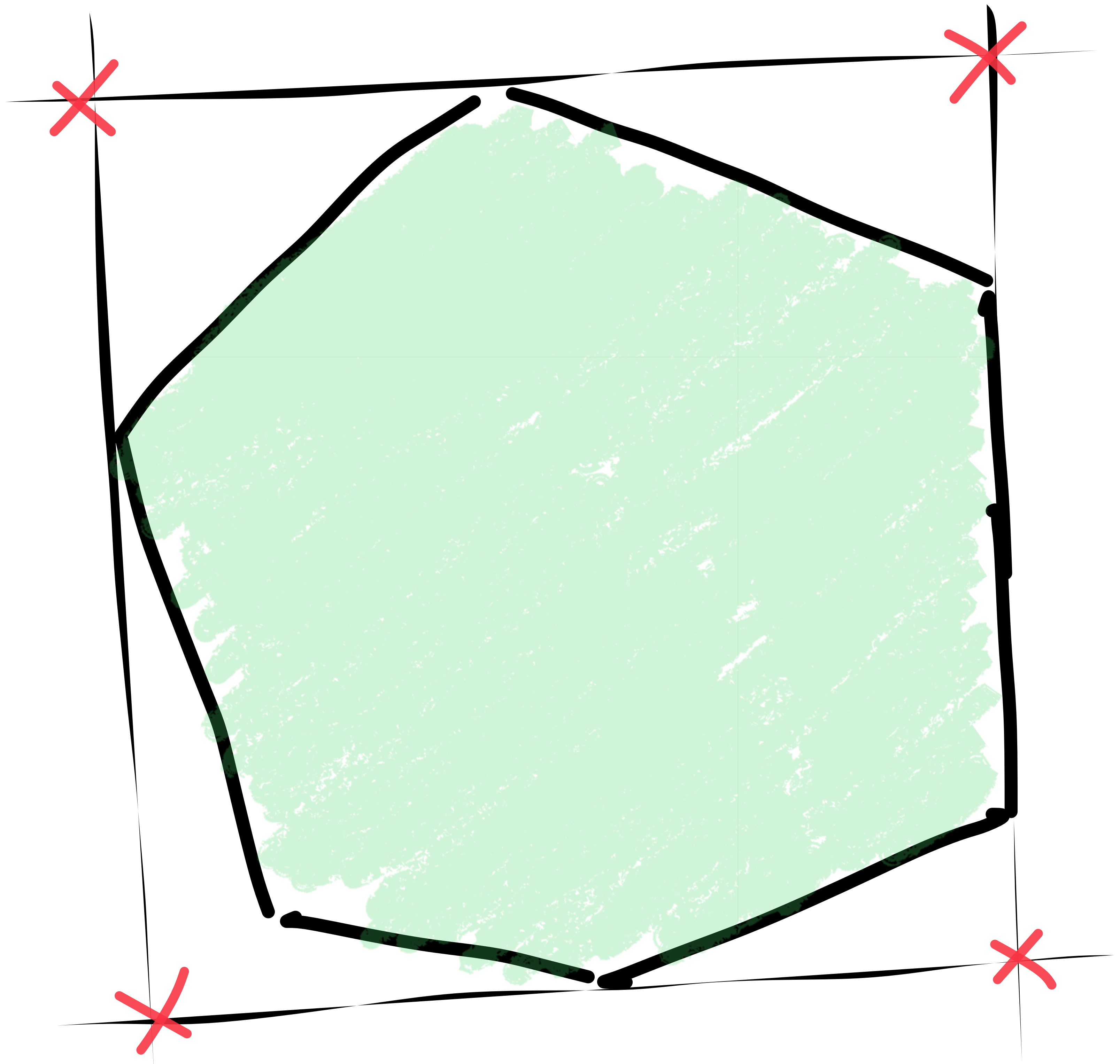
Heuristically add CG-cuts to P until:

- an integer solution is found
- the empty polytope is deduced



Cutting Planes [CCT87]

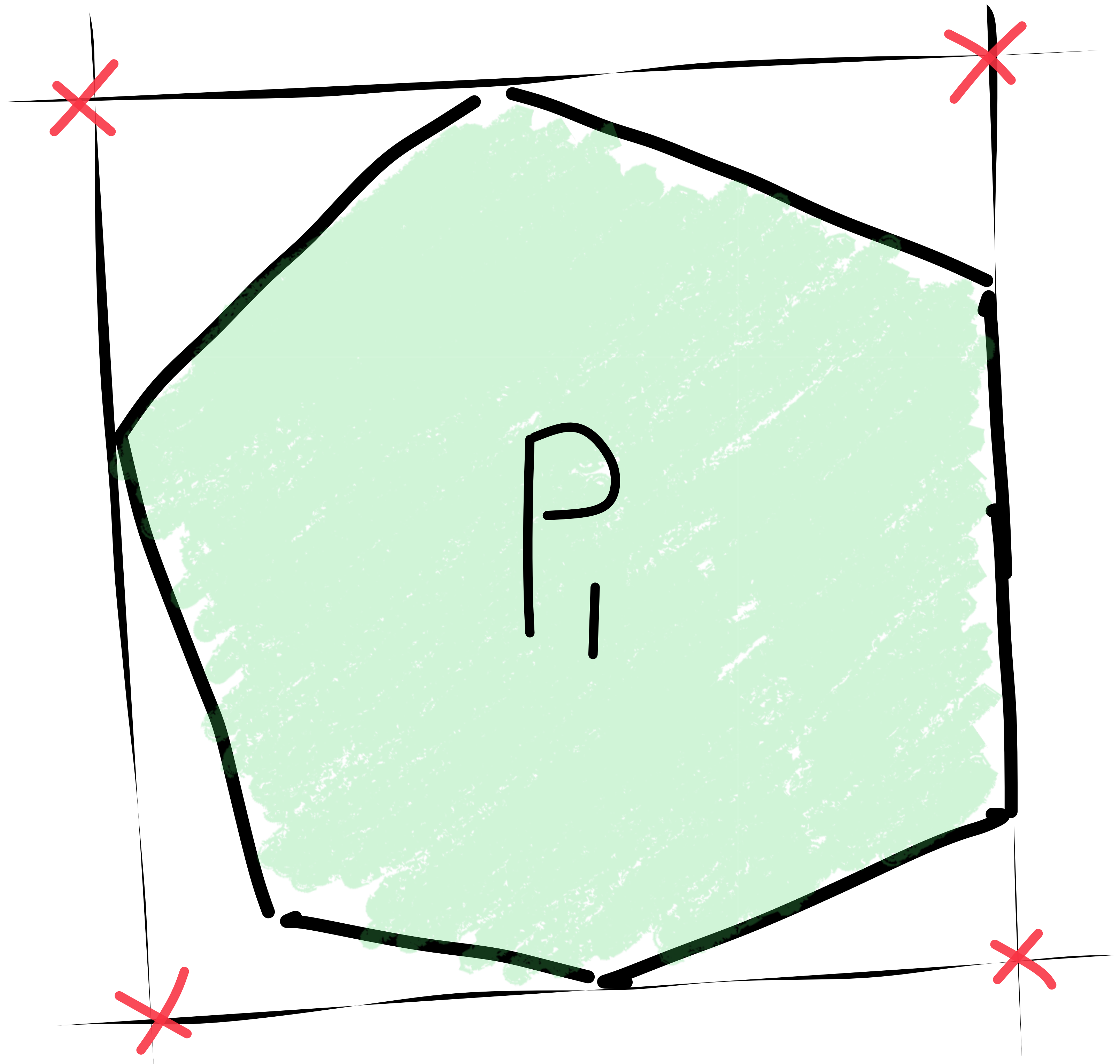
Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$



Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$

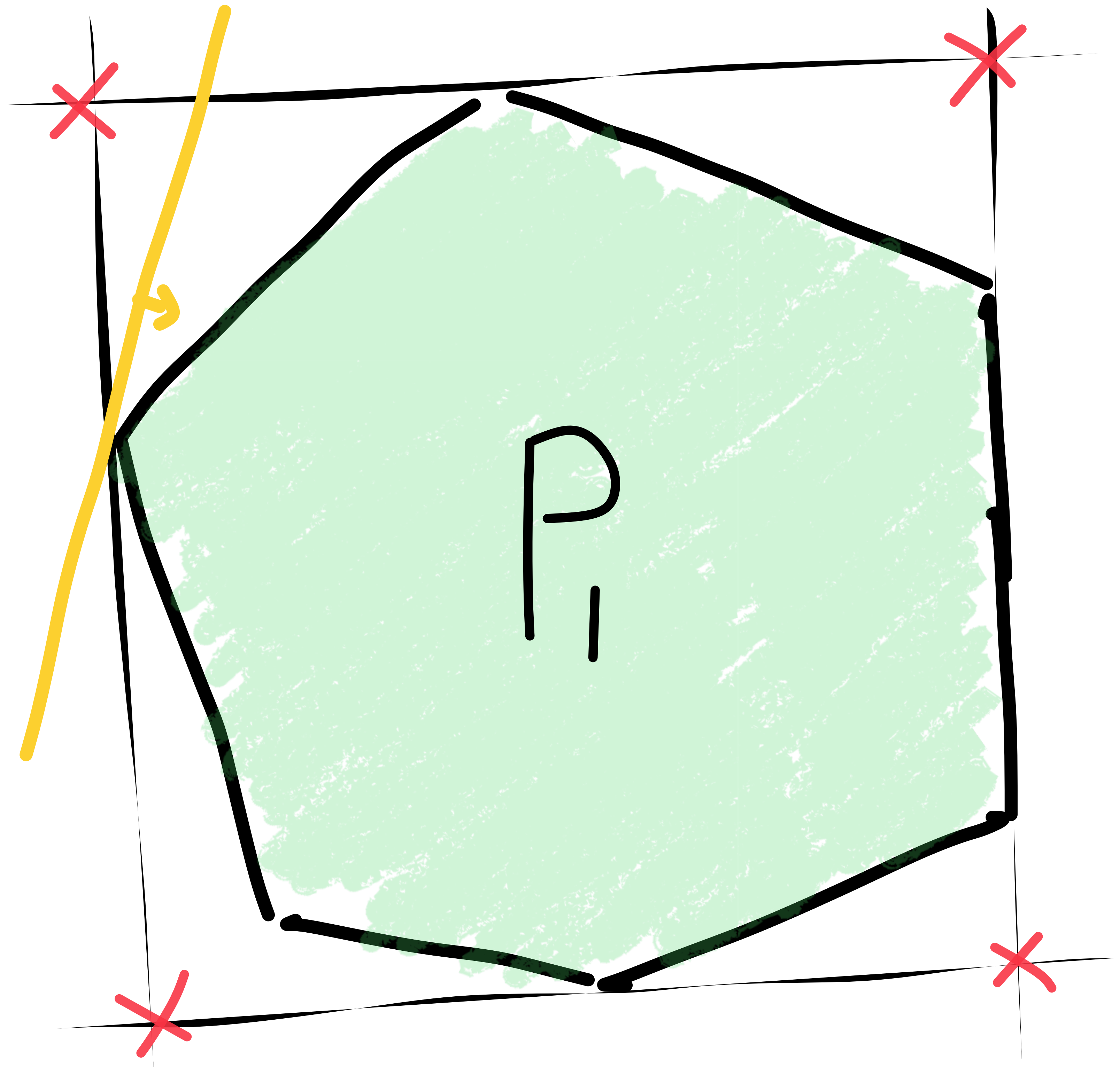
A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$ s.t. P_{i+1} is deduced from P_i by a CG-cut



Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$

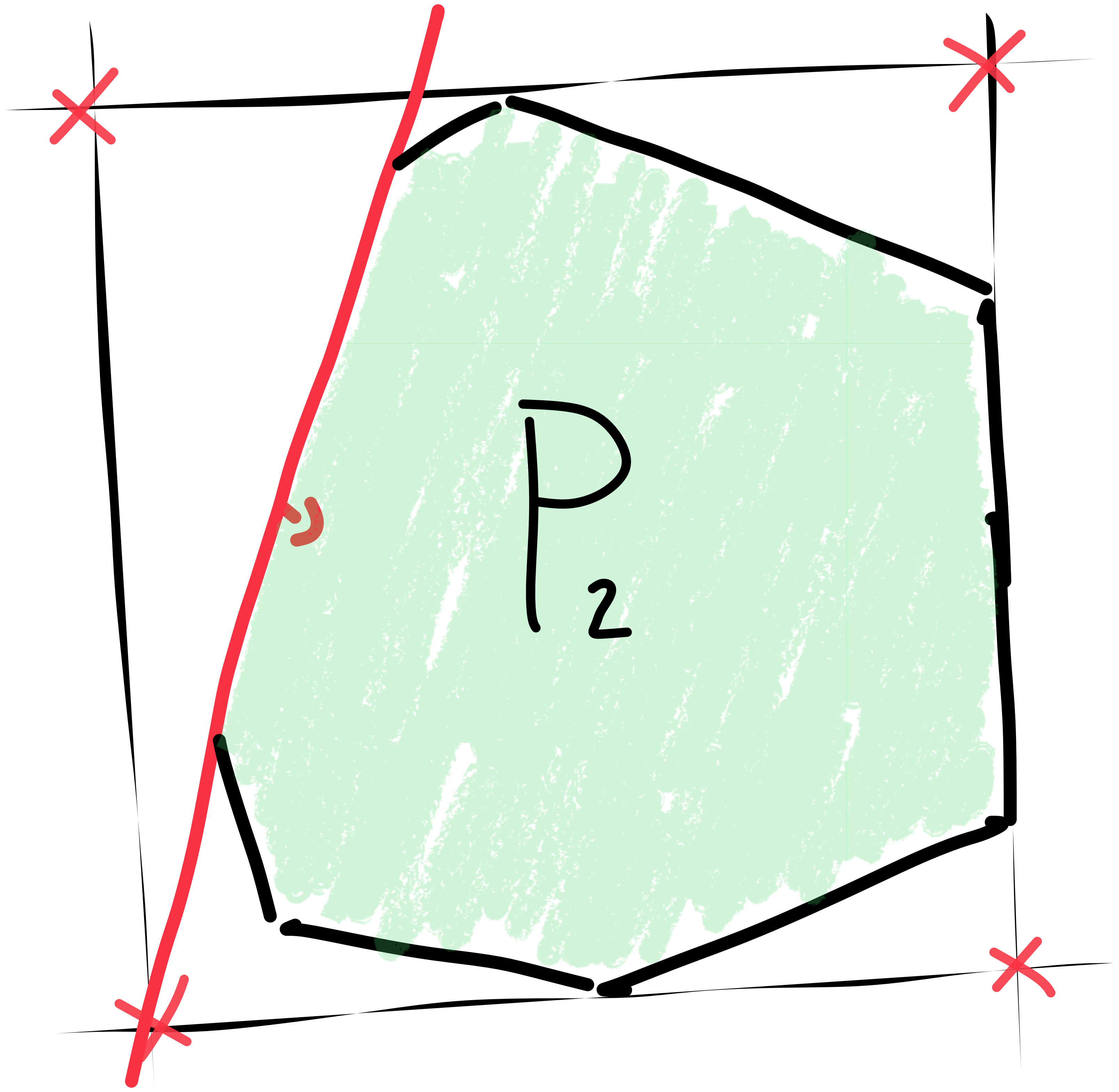
A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$ s.t. P_{i+1} is deduced from P_i by a CG-cut



Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$

A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \dots, P_s = \emptyset$ s.t. P_{i+1} is deduced from P_i by a CG-cut

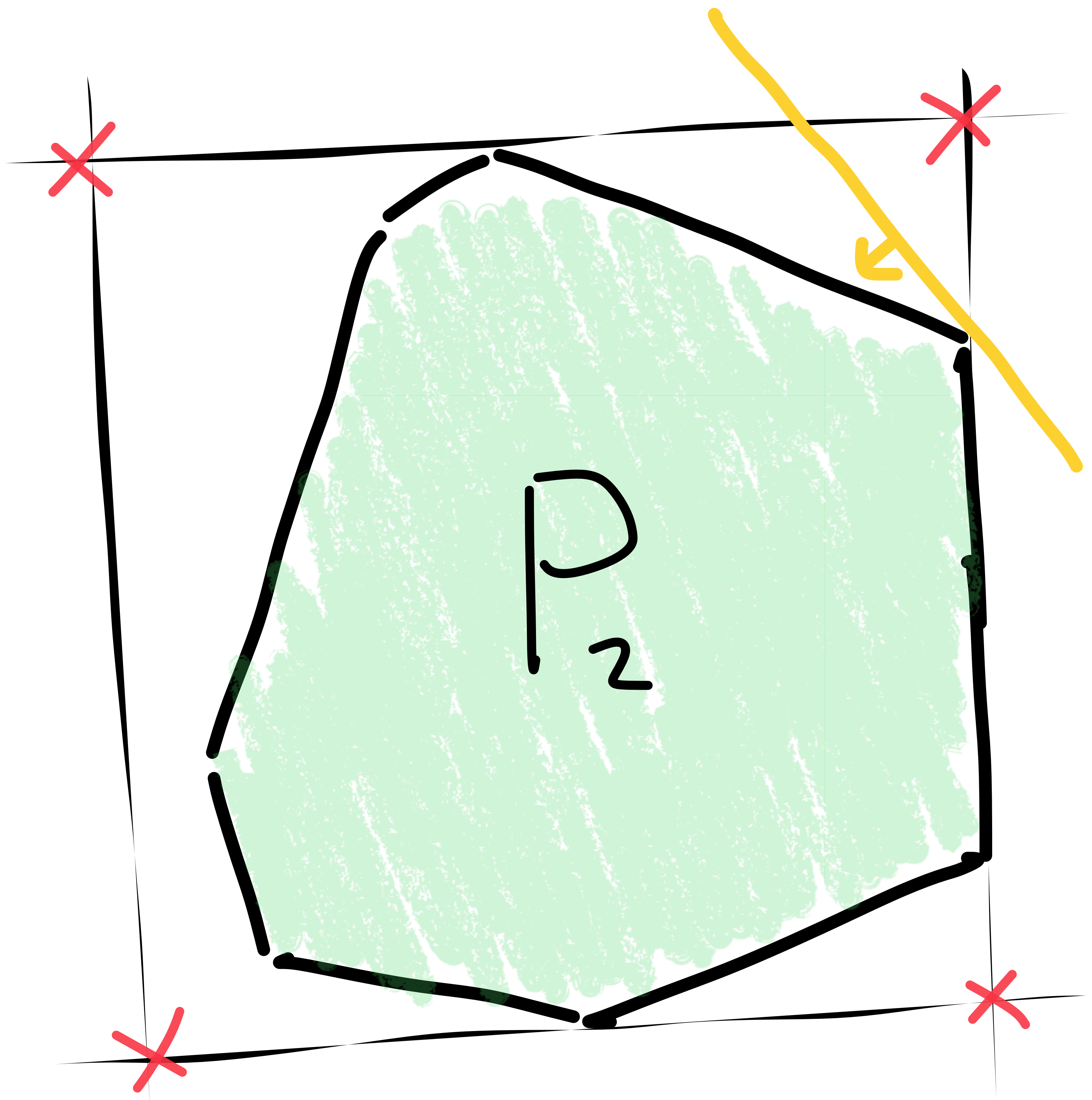


Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$

A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$

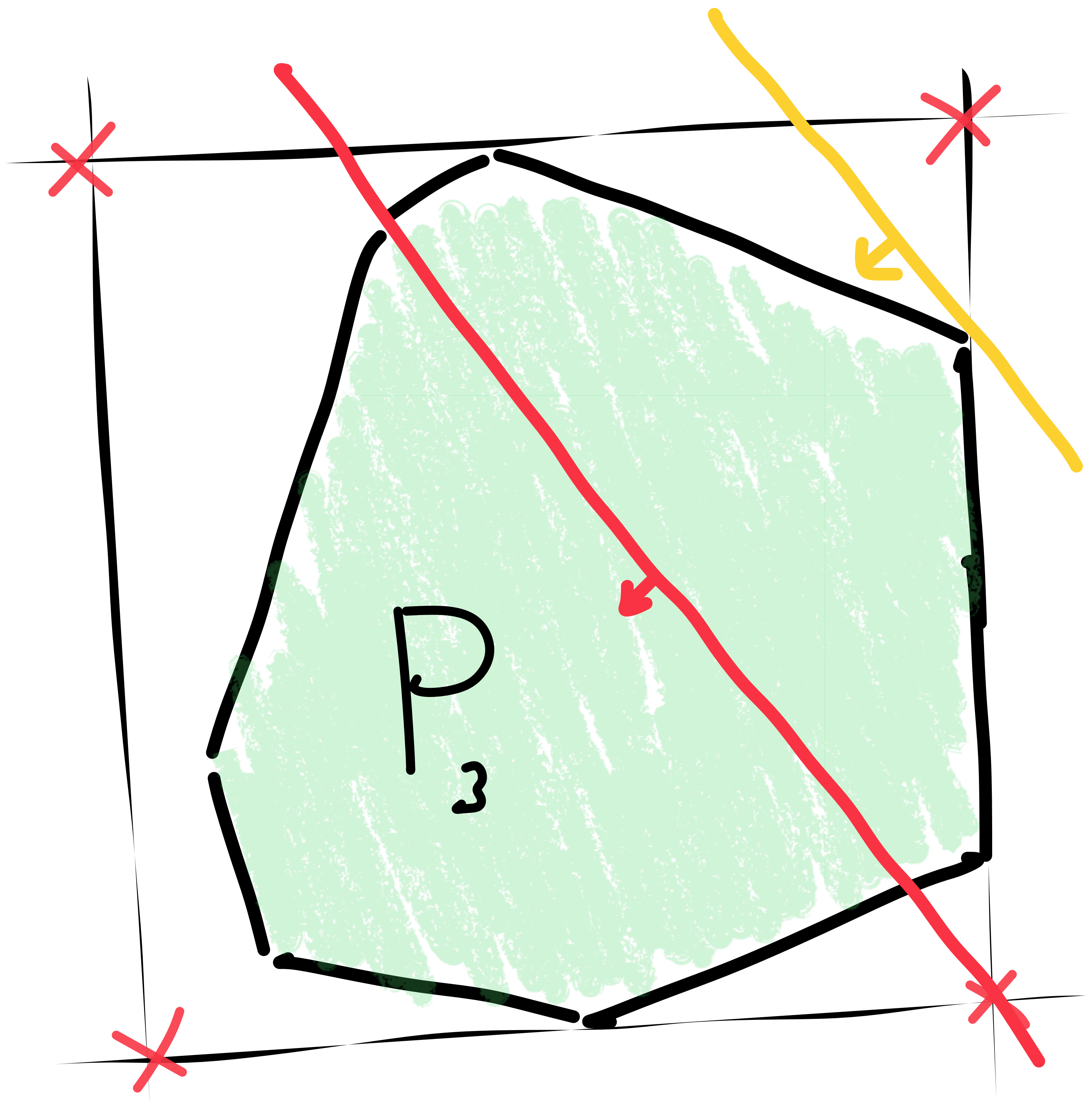
s.t. P_{i+1} is deduced from P_i by a CG-cut



Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$

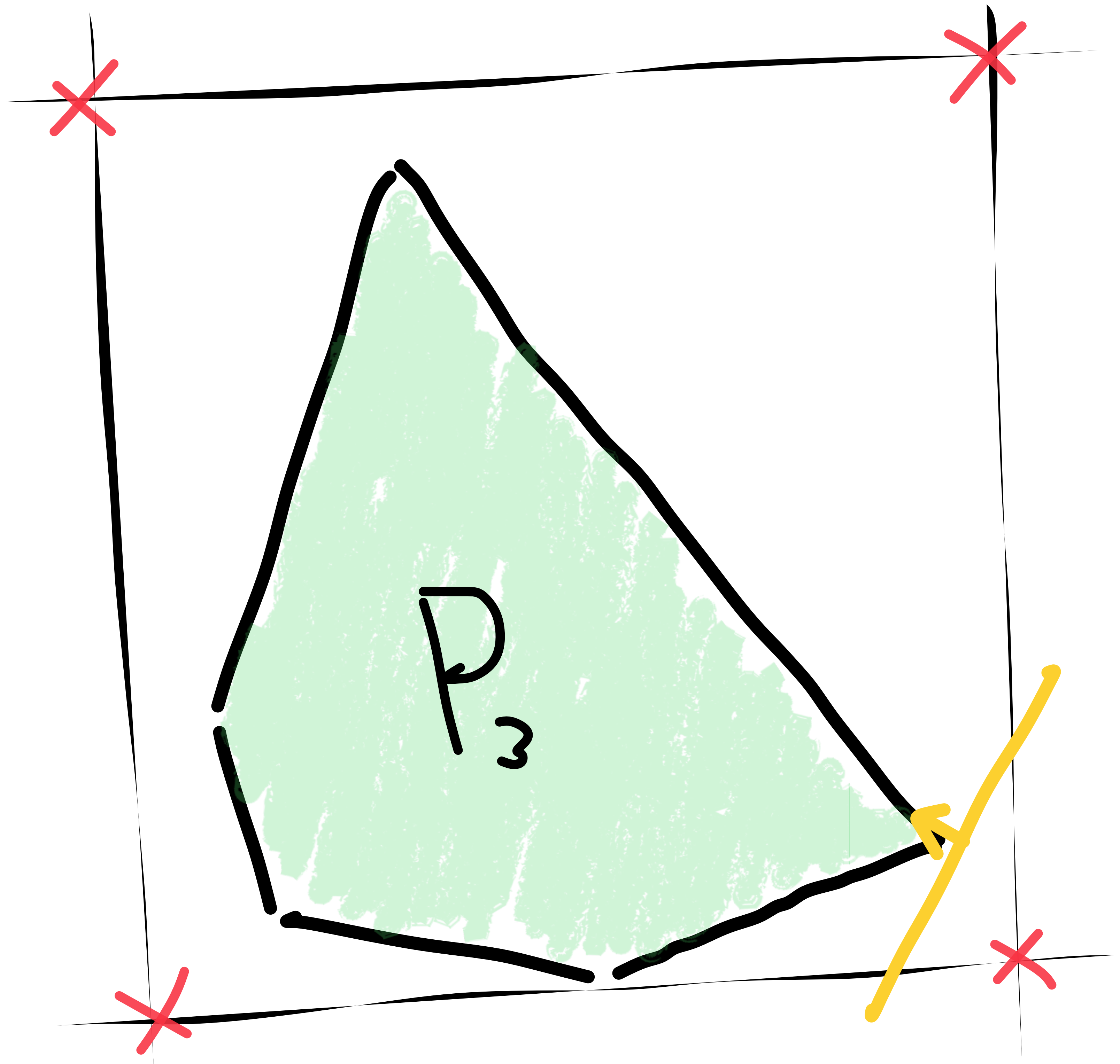
A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$ s.t. P_{i+1} is deduced from P_i by a CG-cut



Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$

A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$ s.t. P_{i+1} is deduced from P_i by a CG-cut

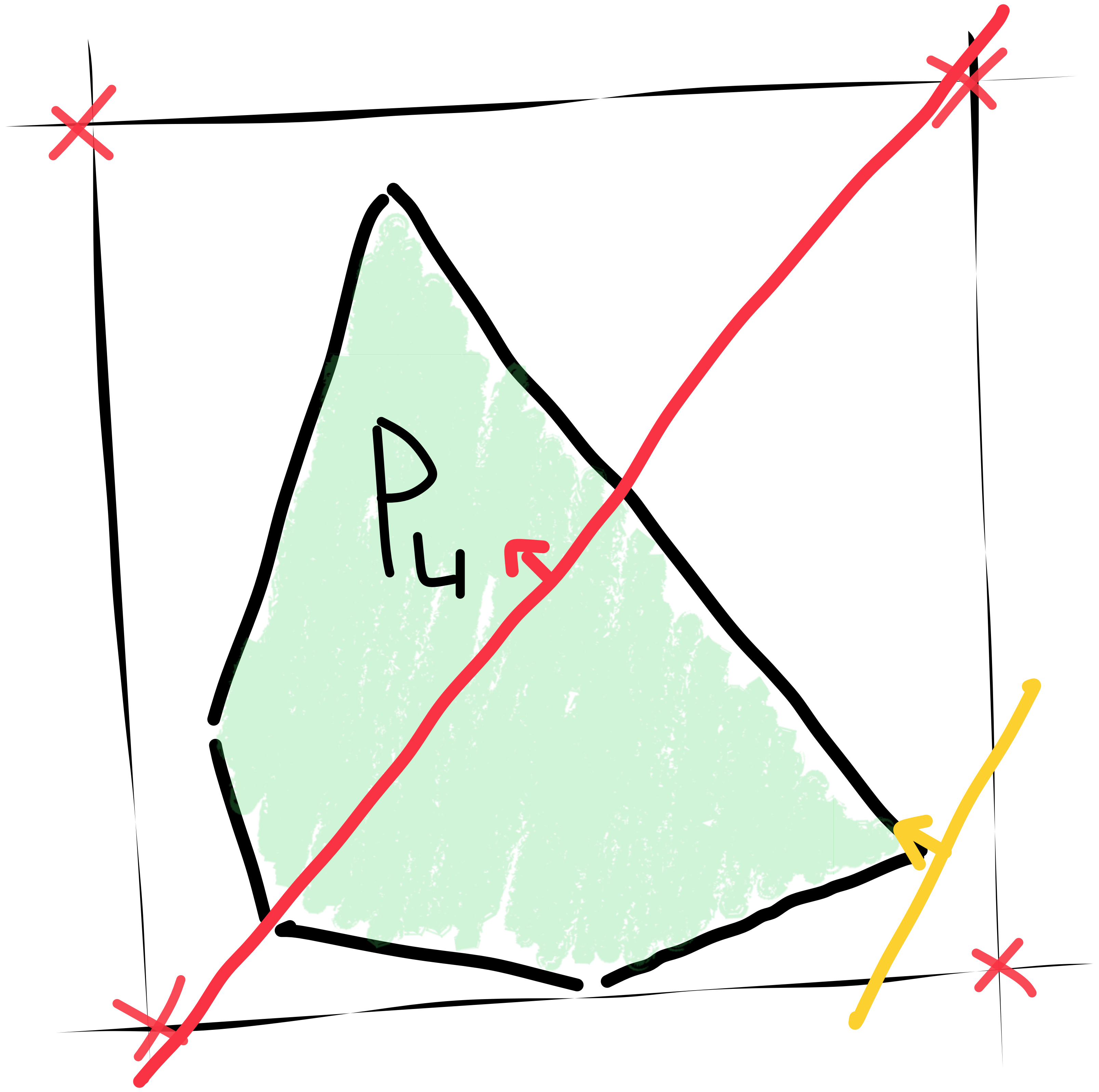


Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$

A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \dots, P_s = \emptyset$

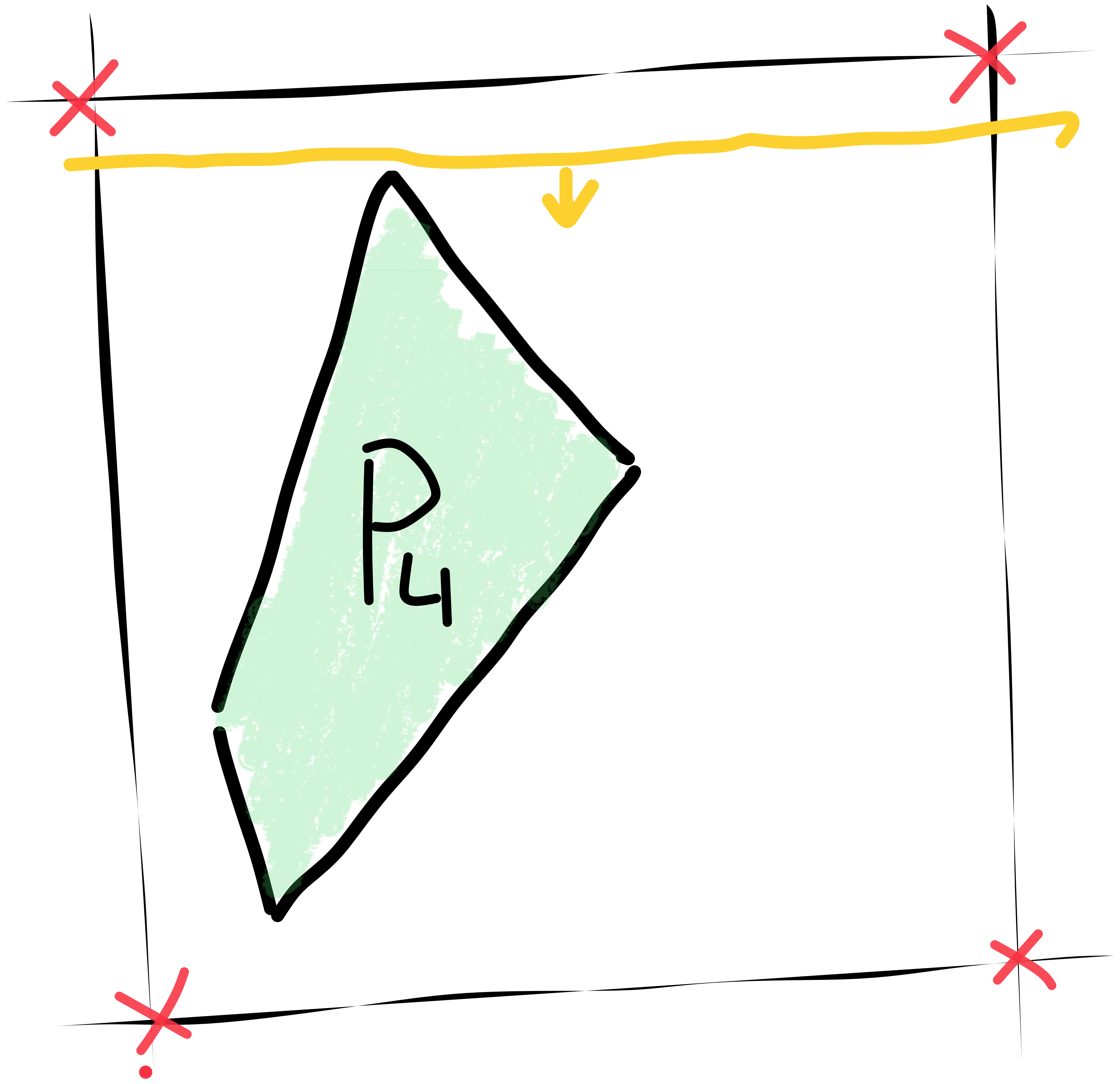
s.t. P_{i+1} is deduced from P_i by a CG-cut



Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$

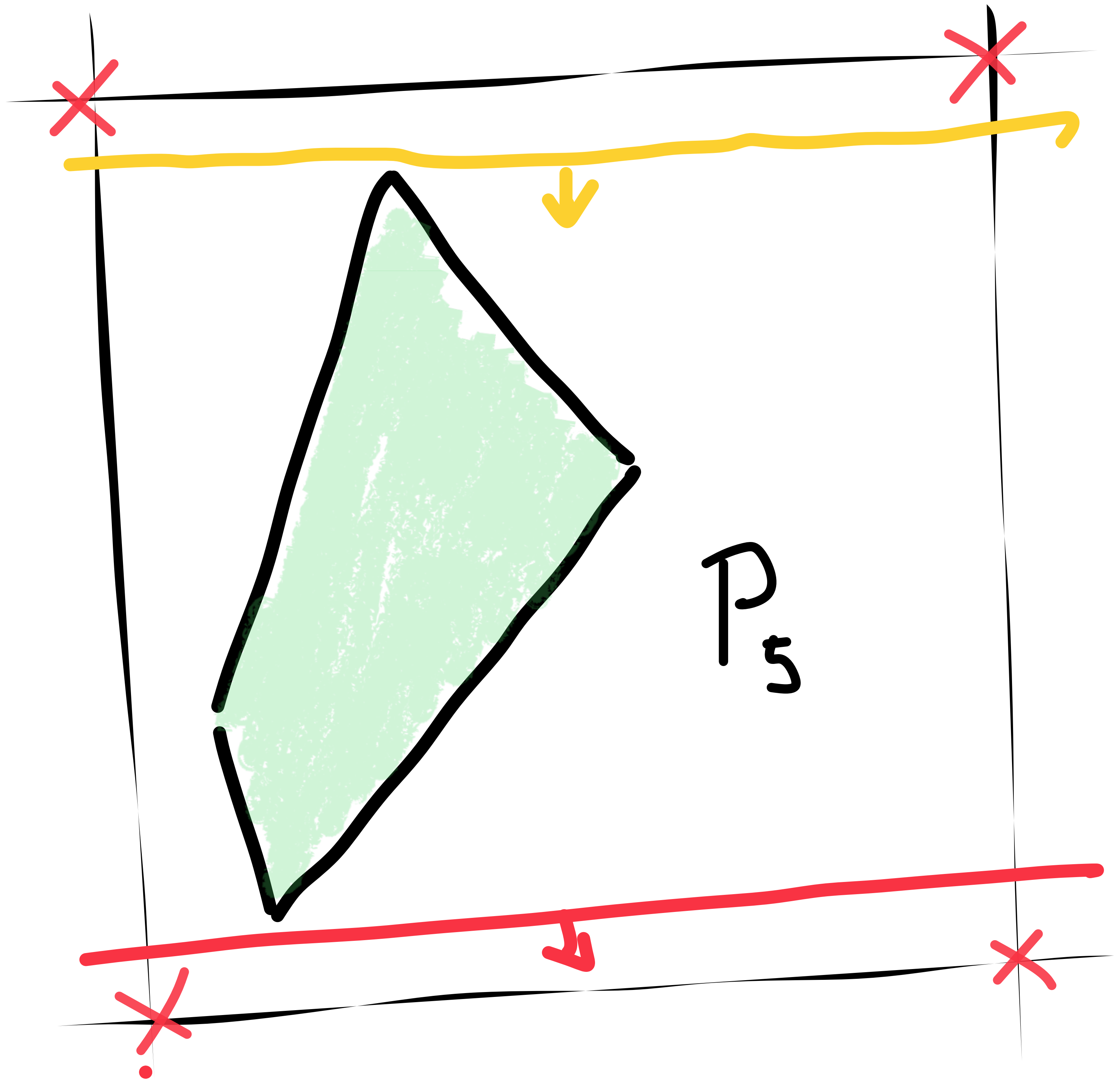
A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$ s.t. P_{i+1} is deduced from P_i by a CG-cut



Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$

A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$ s.t. P_{i+1} is deduced from P_i by a CG-cut

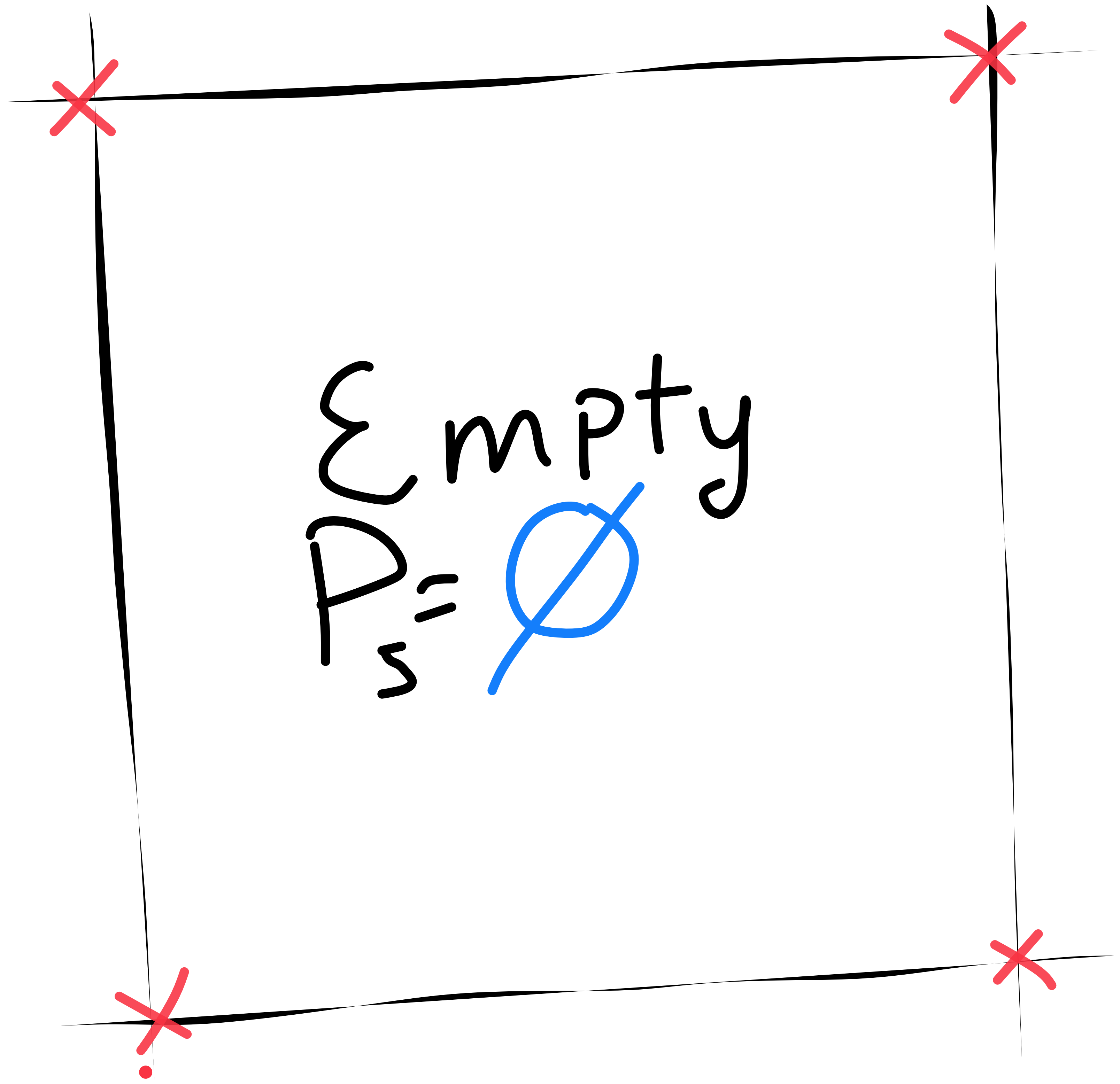


Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$

A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$

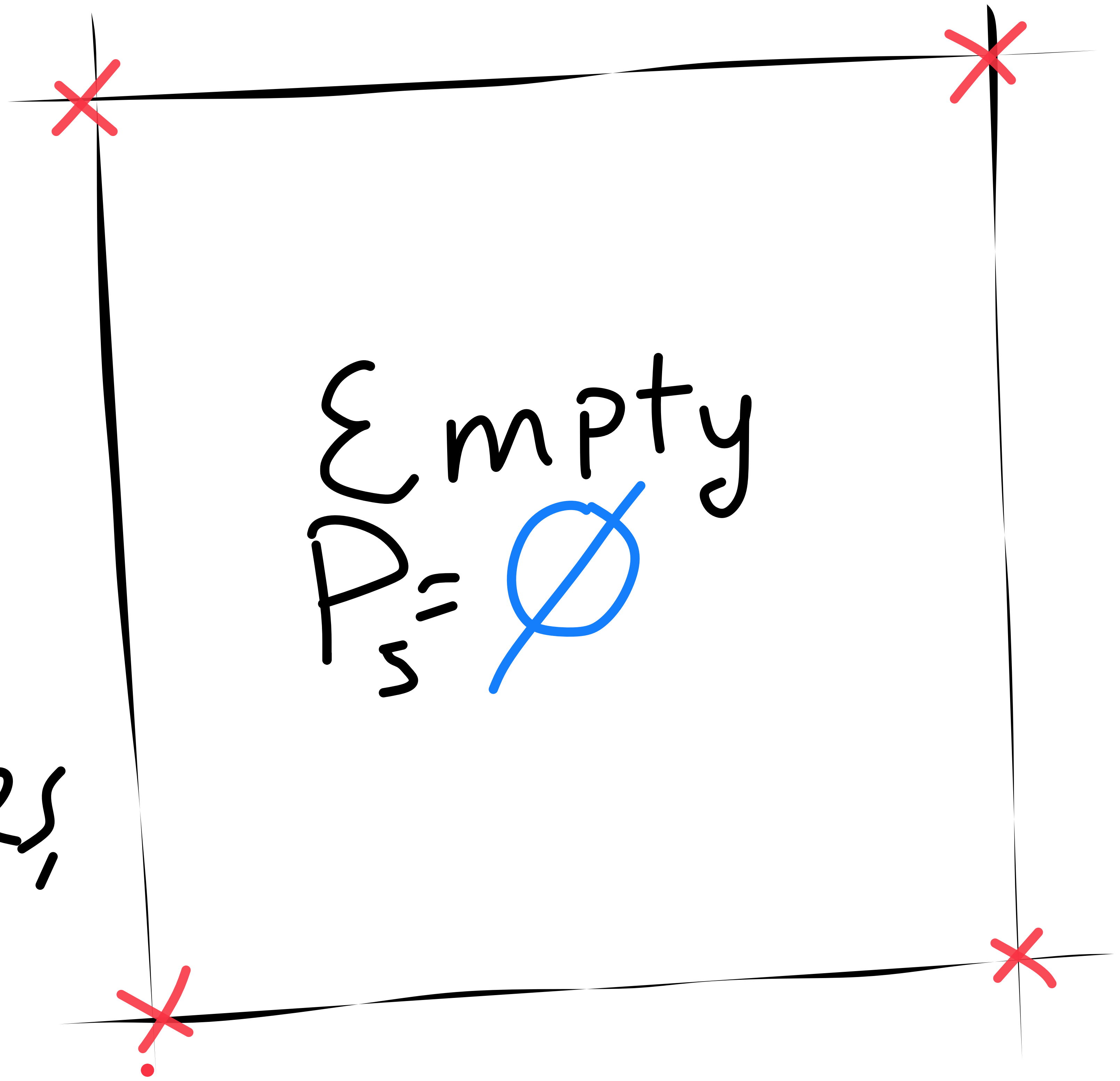
s.t. P_{i+1} is deduced from P_i by a CG-cut



Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$

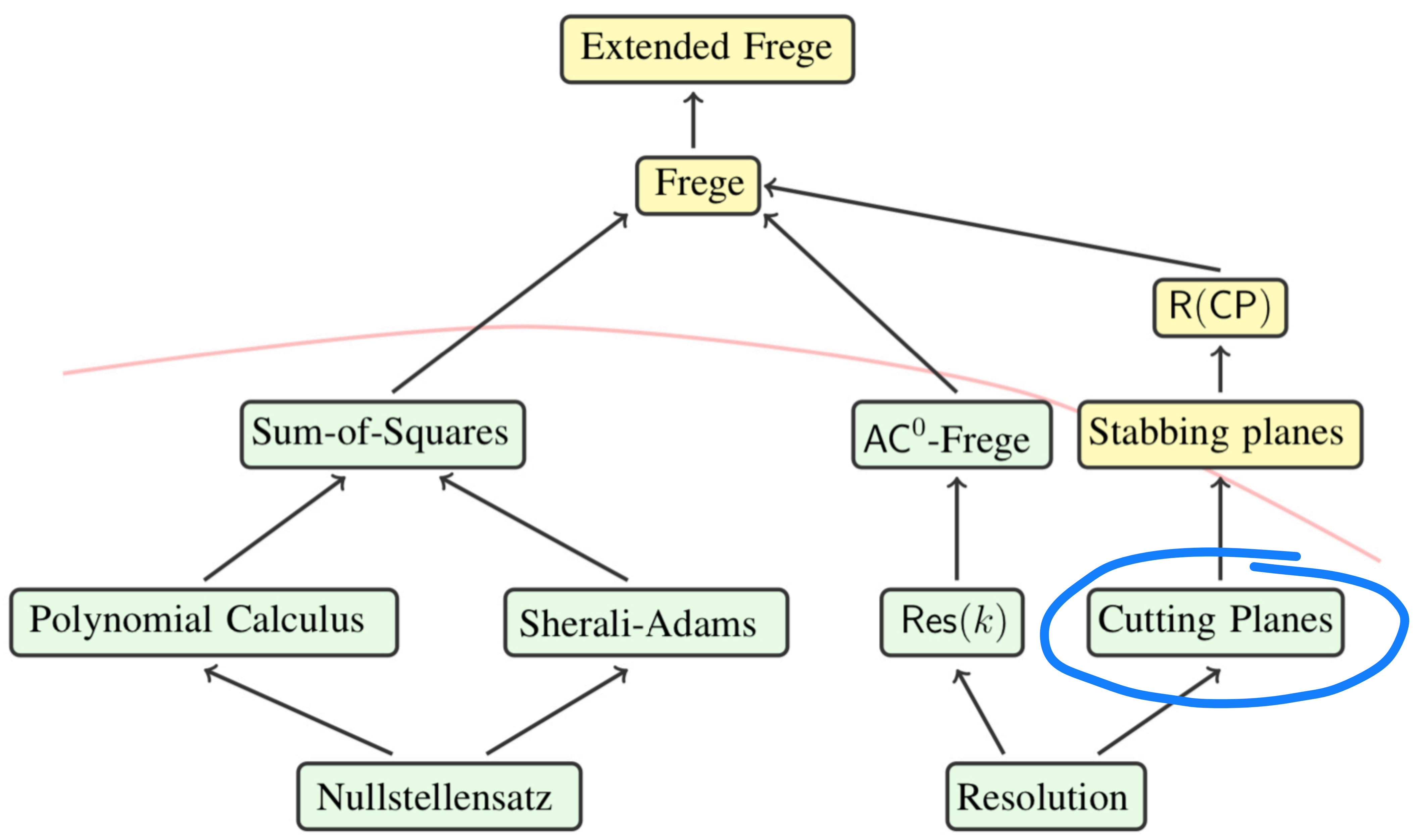
A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$ s.t. P_{i+1} is deduced from P_i by a CG-cut



Proof Size: the number of polytopes, s

Cutting Planes

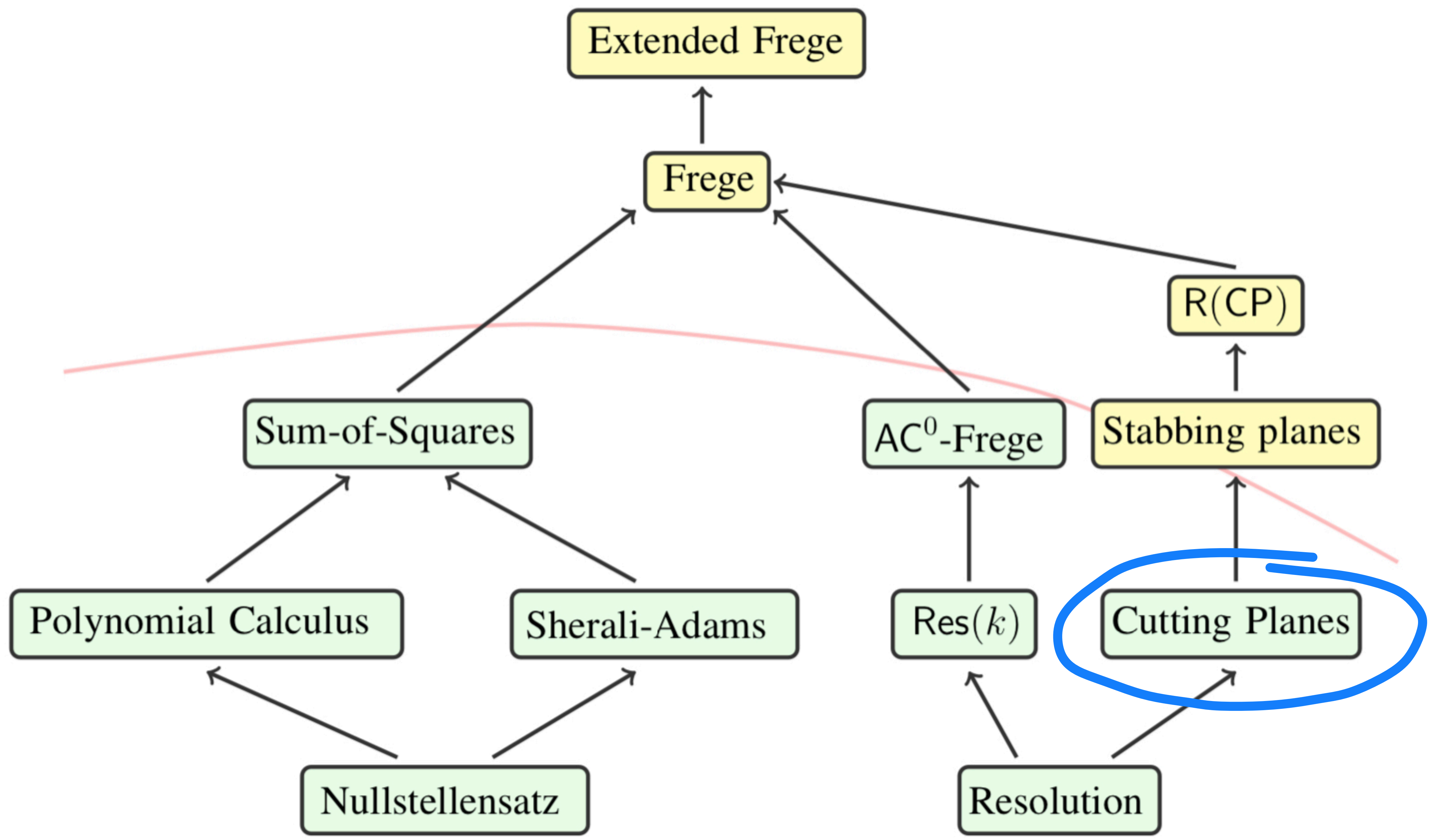
Powerful & Well-studied algebraic proof system



Cutting Planes

Powerful & Well-studied algebraic proof system

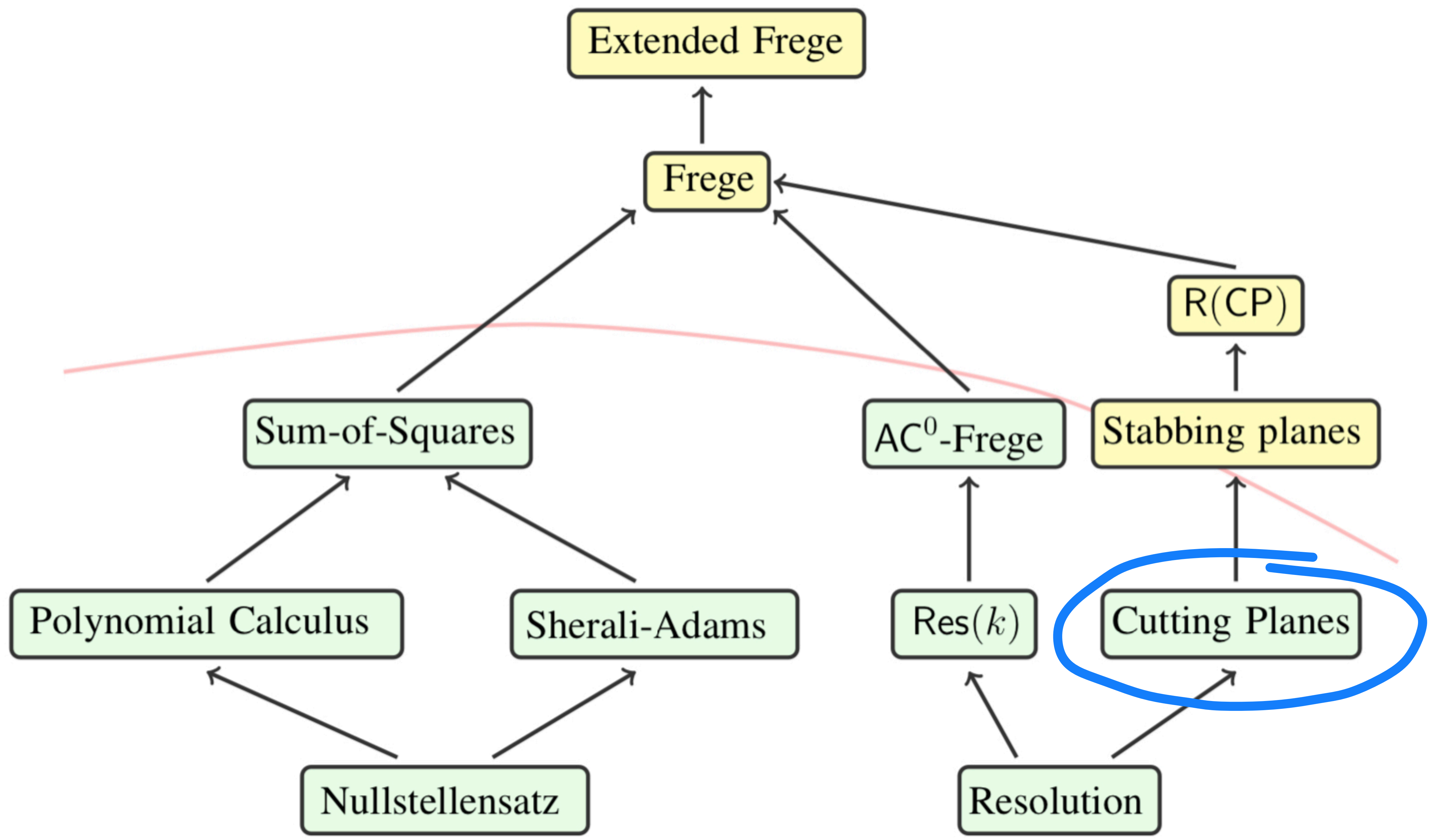
- Short proofs of PHP



Cutting Planes

Powerful & Well-studied algebraic proof system

- ▷ Short proofs of PHP
- ▷ Exponential lowerbounds [Pud97]



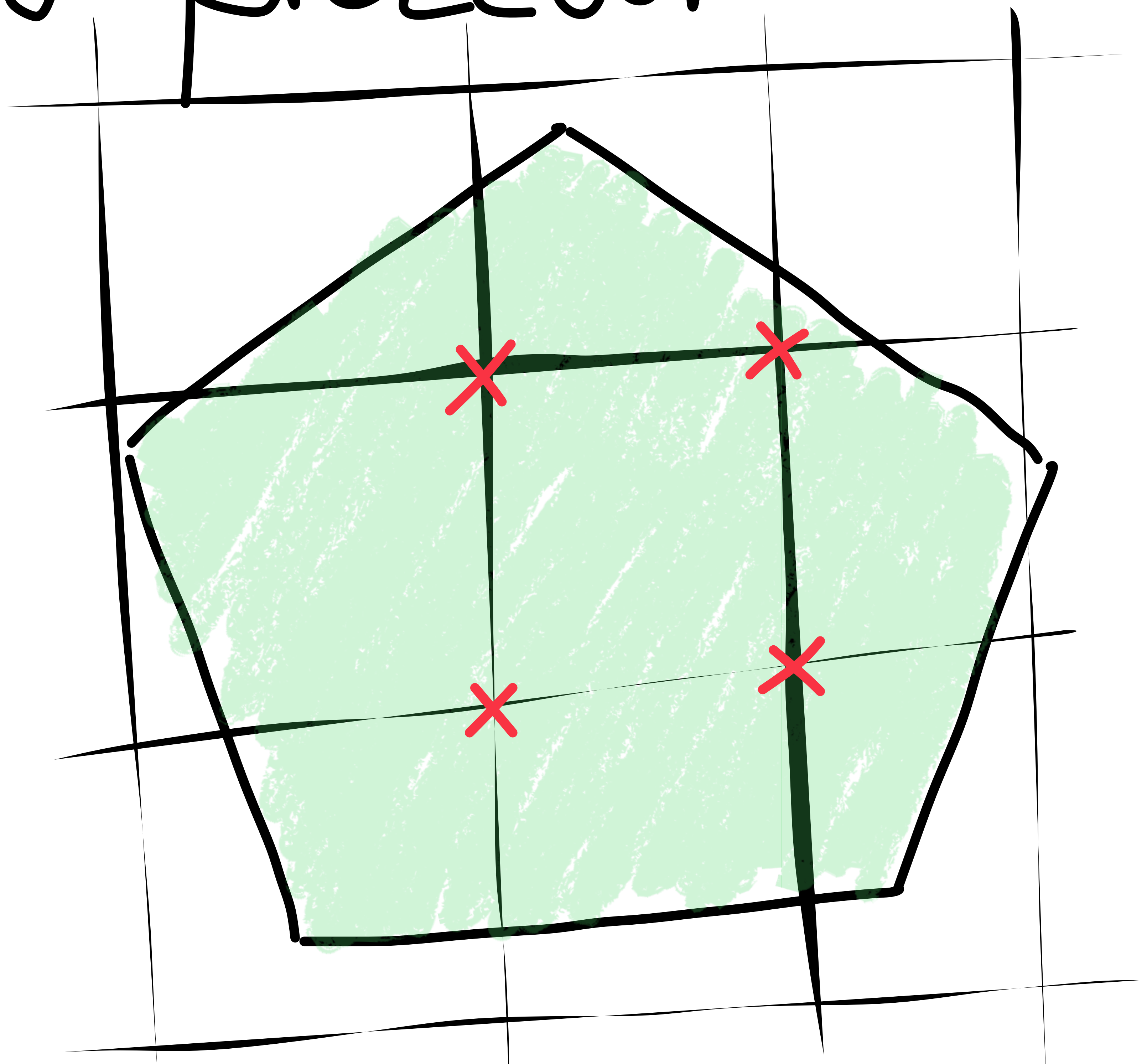
Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure

Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure

Branch-and-Cut on input P,

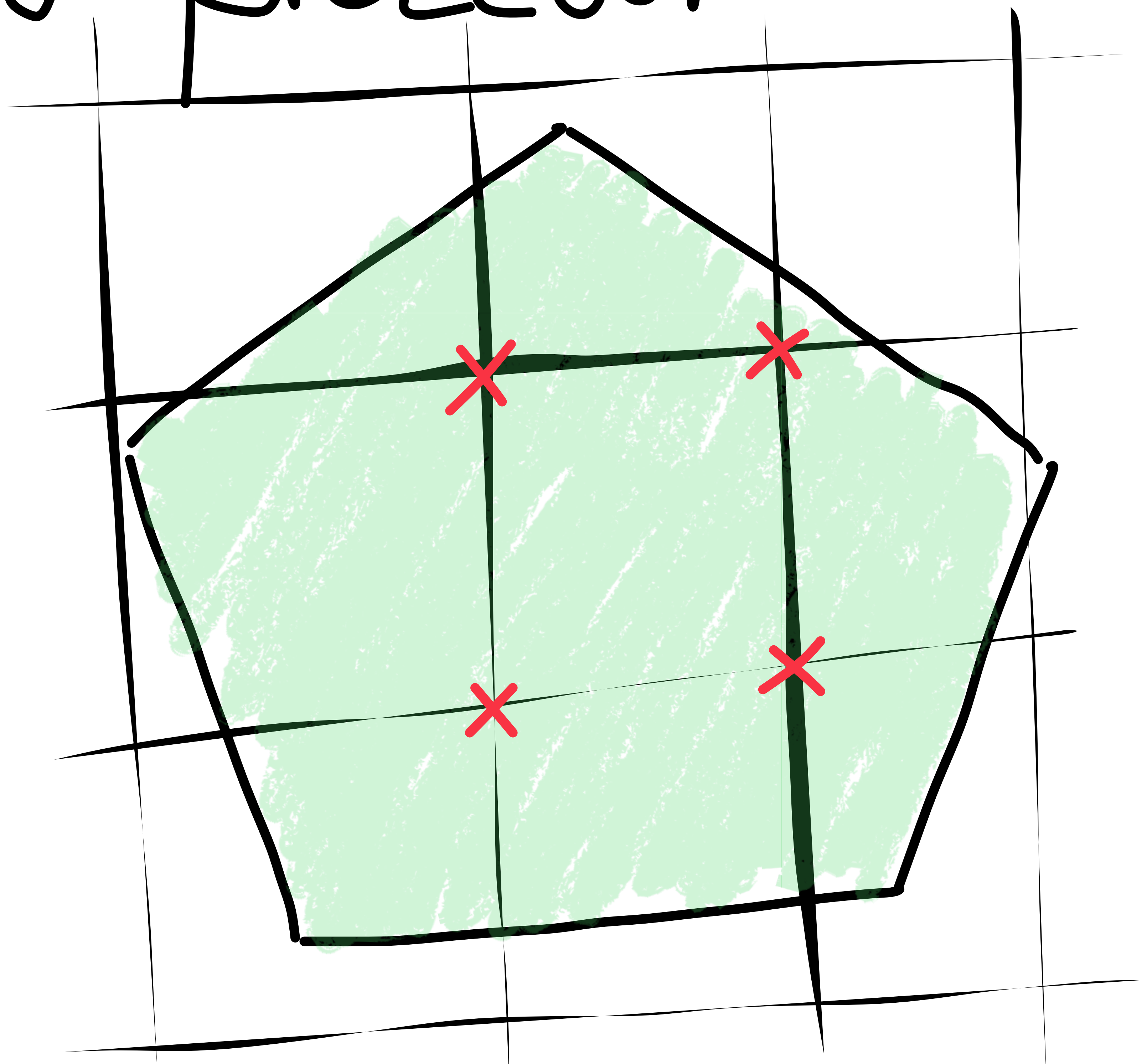


Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure

Branch-and-Cut on input P,

Branch: break P into $P_1, \dots, P_k$ s.t. $P \cap \mathbb{Z}^n \subseteq P_1 \cup \dots \cup P_k$

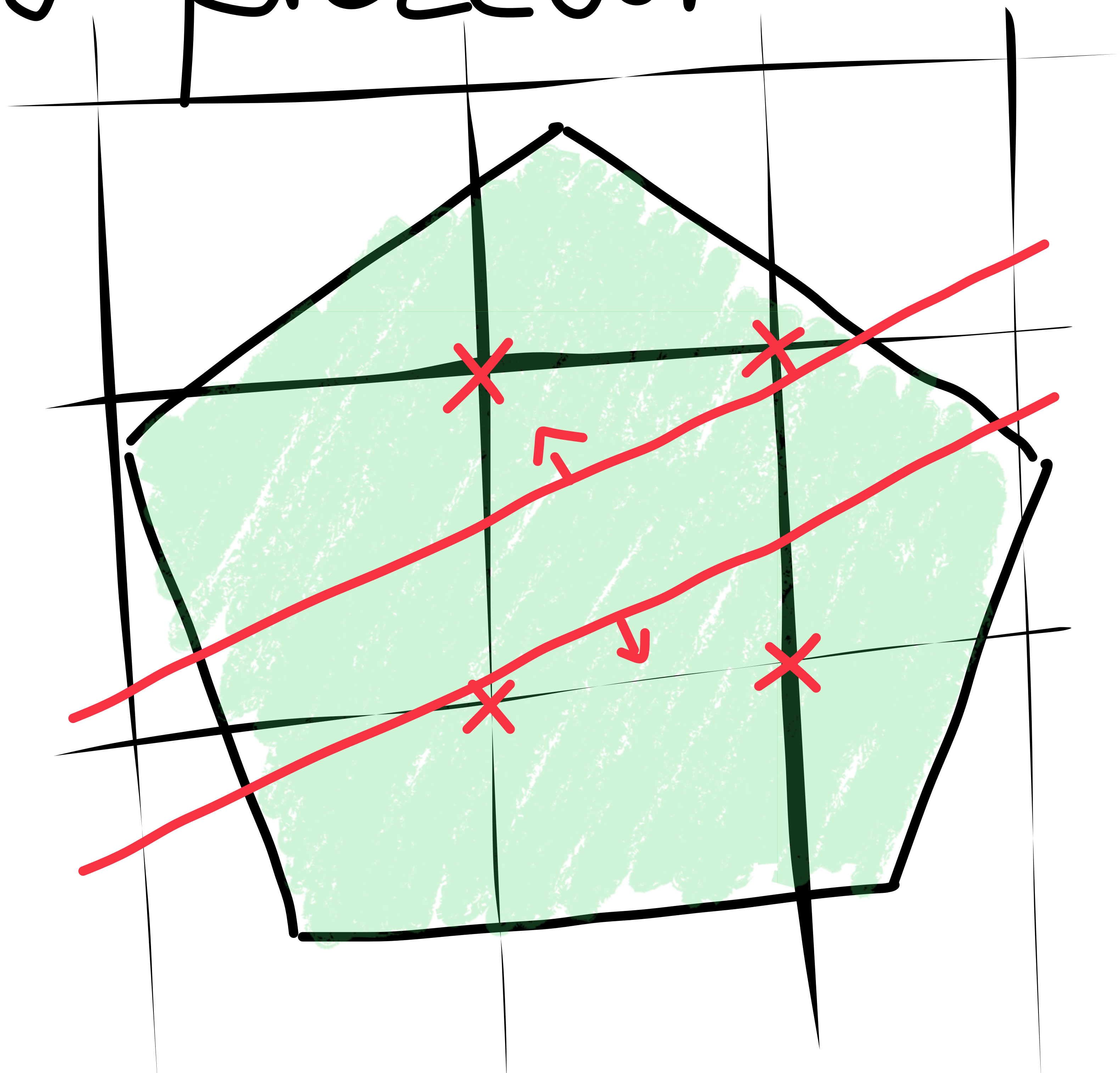


Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure

Branch-and-Cut on input P,

Branch: break P into $P_1, \ldots, P_k$ s.t. $P \cap \mathbb{Z}^n \subseteq P_1 \cup \ldots \cup P_k$

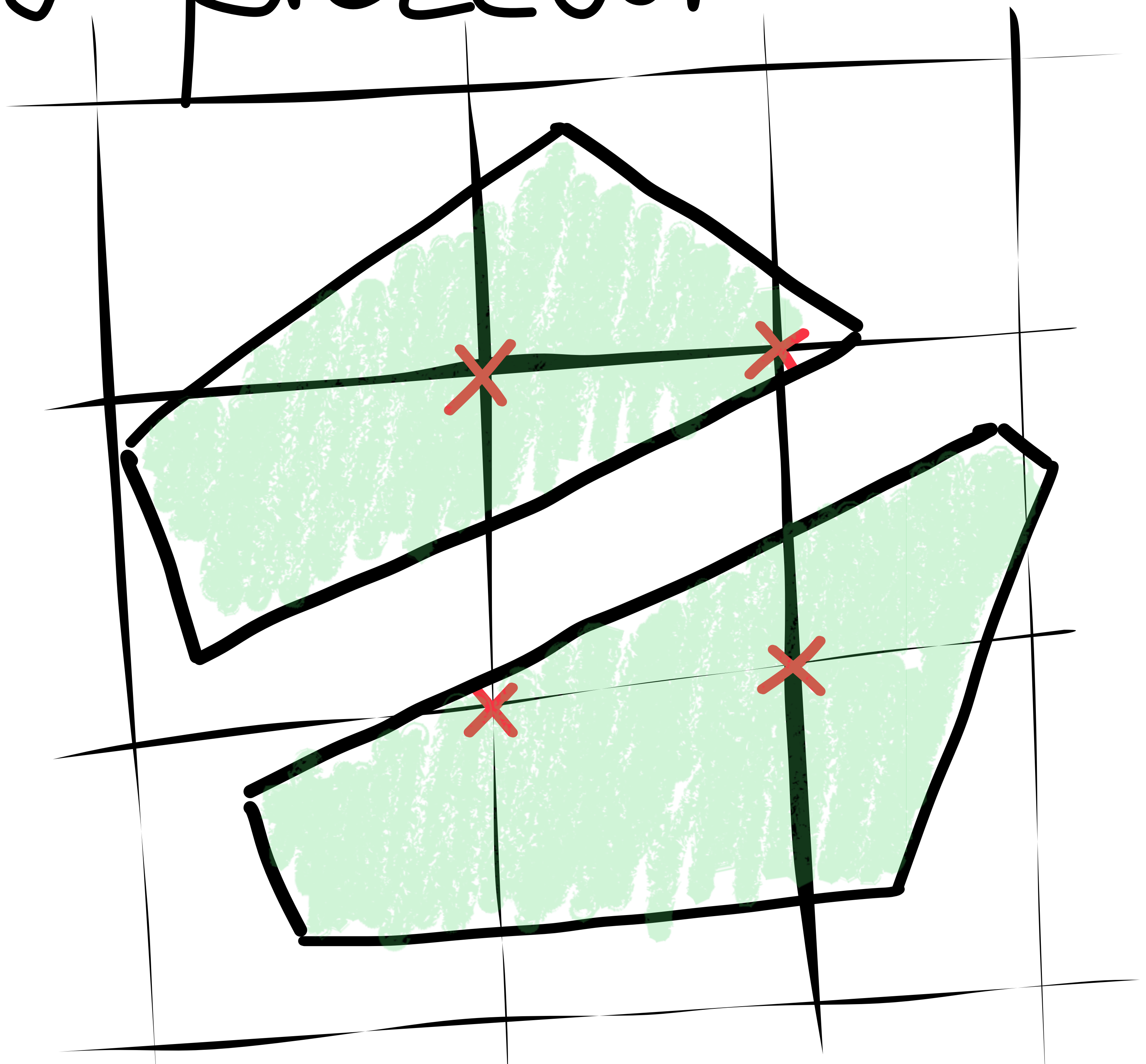


Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure

Branch-and-Cut on input P,

Branch: break P into $P_1, \ldots, P_k$ s.t. $P \cap \mathbb{Z}^n \subseteq P_1 \cup \ldots \cup P_k$



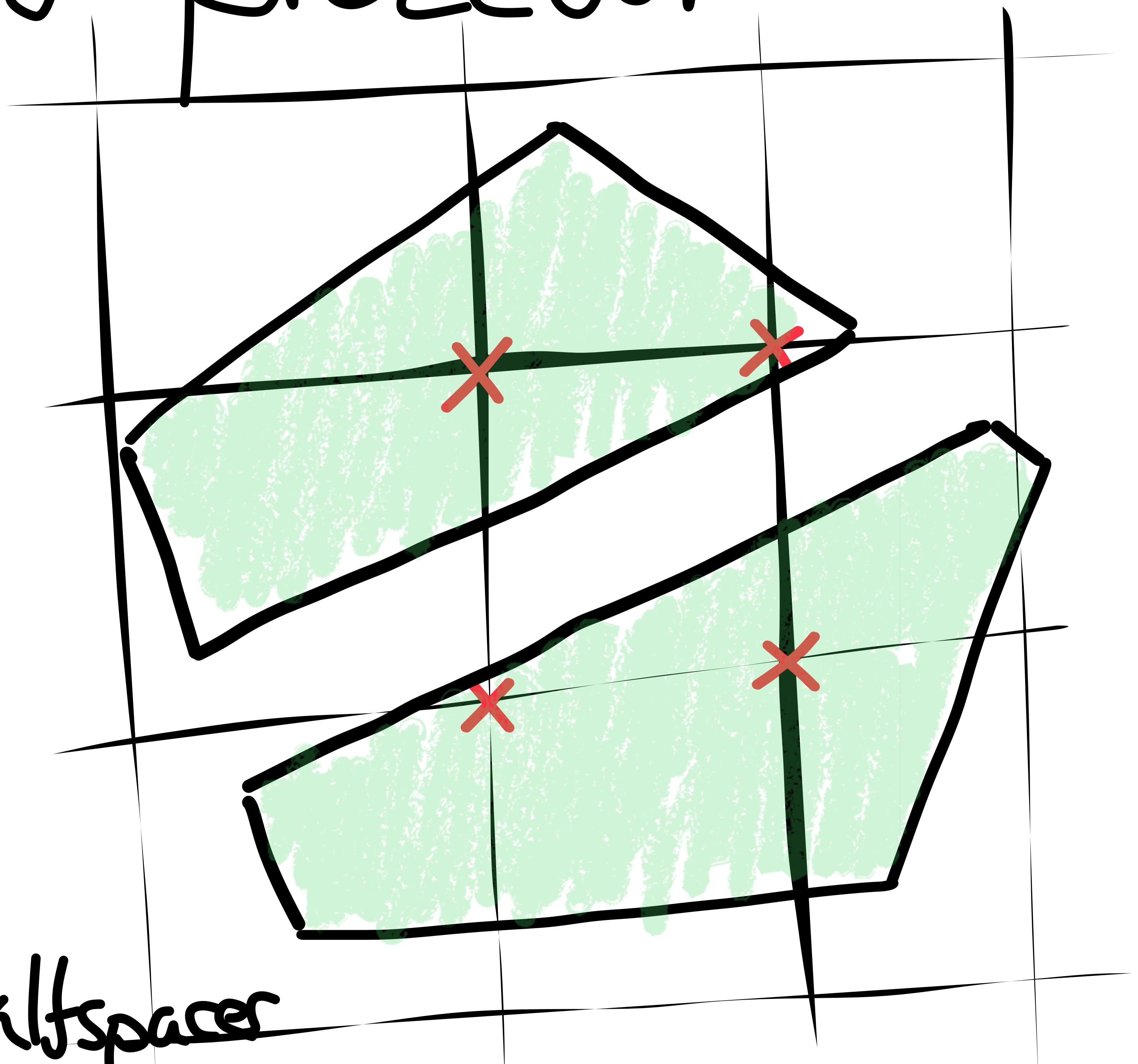
Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure

Branch-and-Cut On input P,

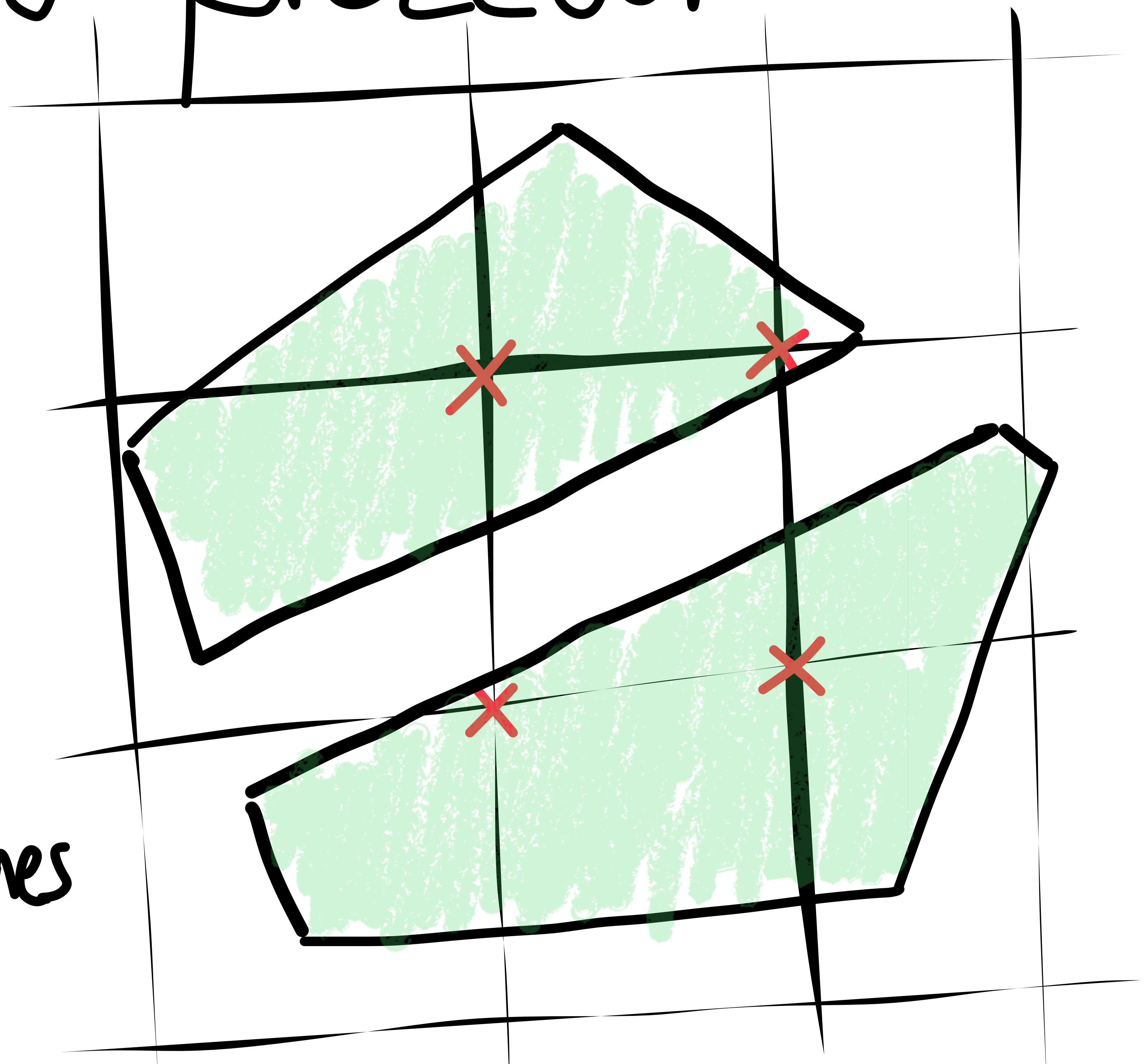
Branch: break P into $P_1, \dots, P_k$ s.t. $P \cap \mathbb{Z}^n \subseteq P_1 \cup \dots \cup P_k$

▷ In practice, P is broken into $P \cap \{x : ax \geq b\}$ and $P \cap \{x : ax \leq b-1\}$ for some class of halfspaces



Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure



Branch-and-Cut On input P,

Branch: break P into $P_1, \ldots, P_k$ s.t. $P \cap \mathbb{Z}^n \subseteq P_1 \cup \ldots \cup P_k$

Cut: Refine $P_1, \ldots, P_k$ with additional cutting planes

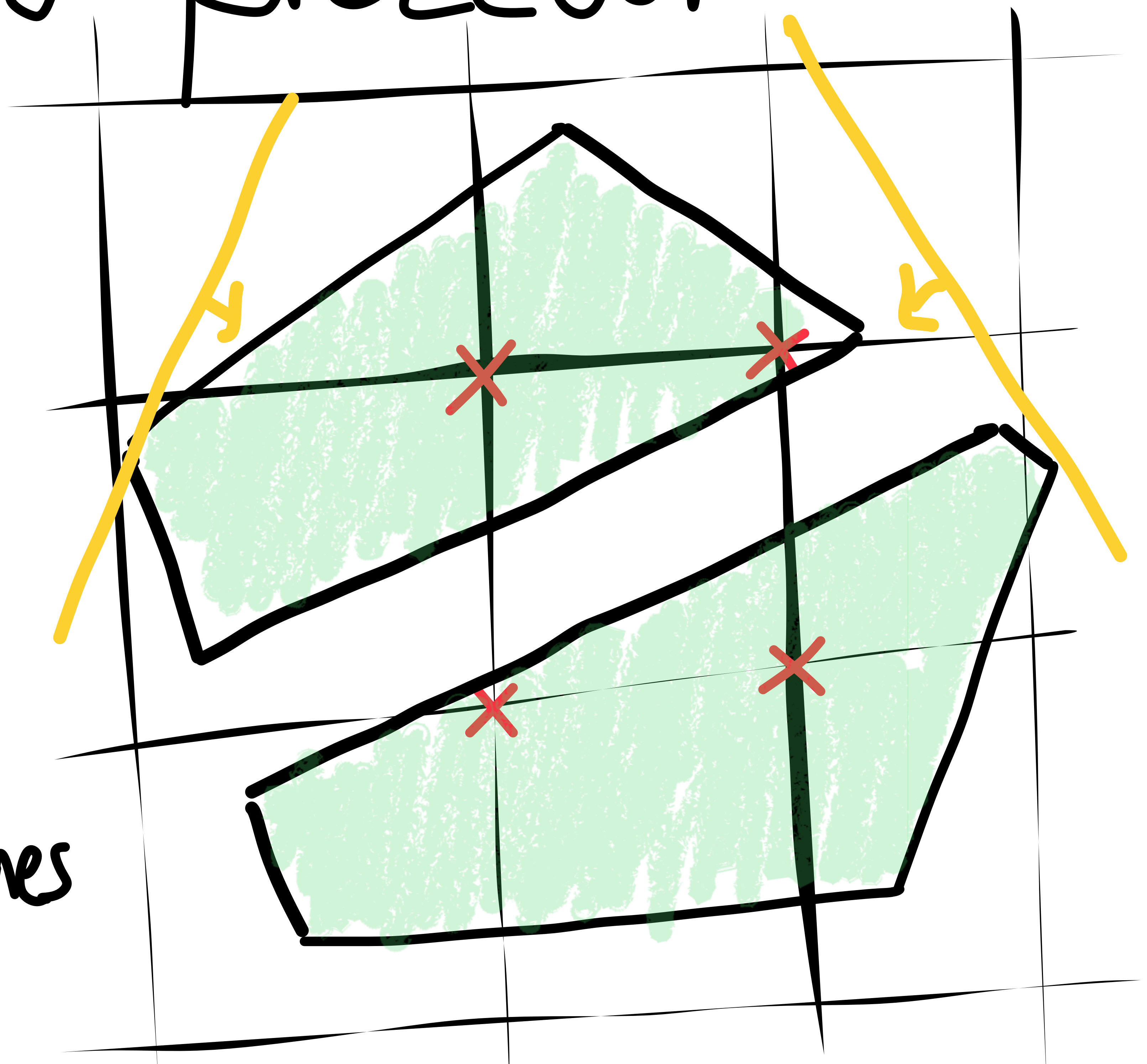
Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure

Branch-and-Cut On input P,

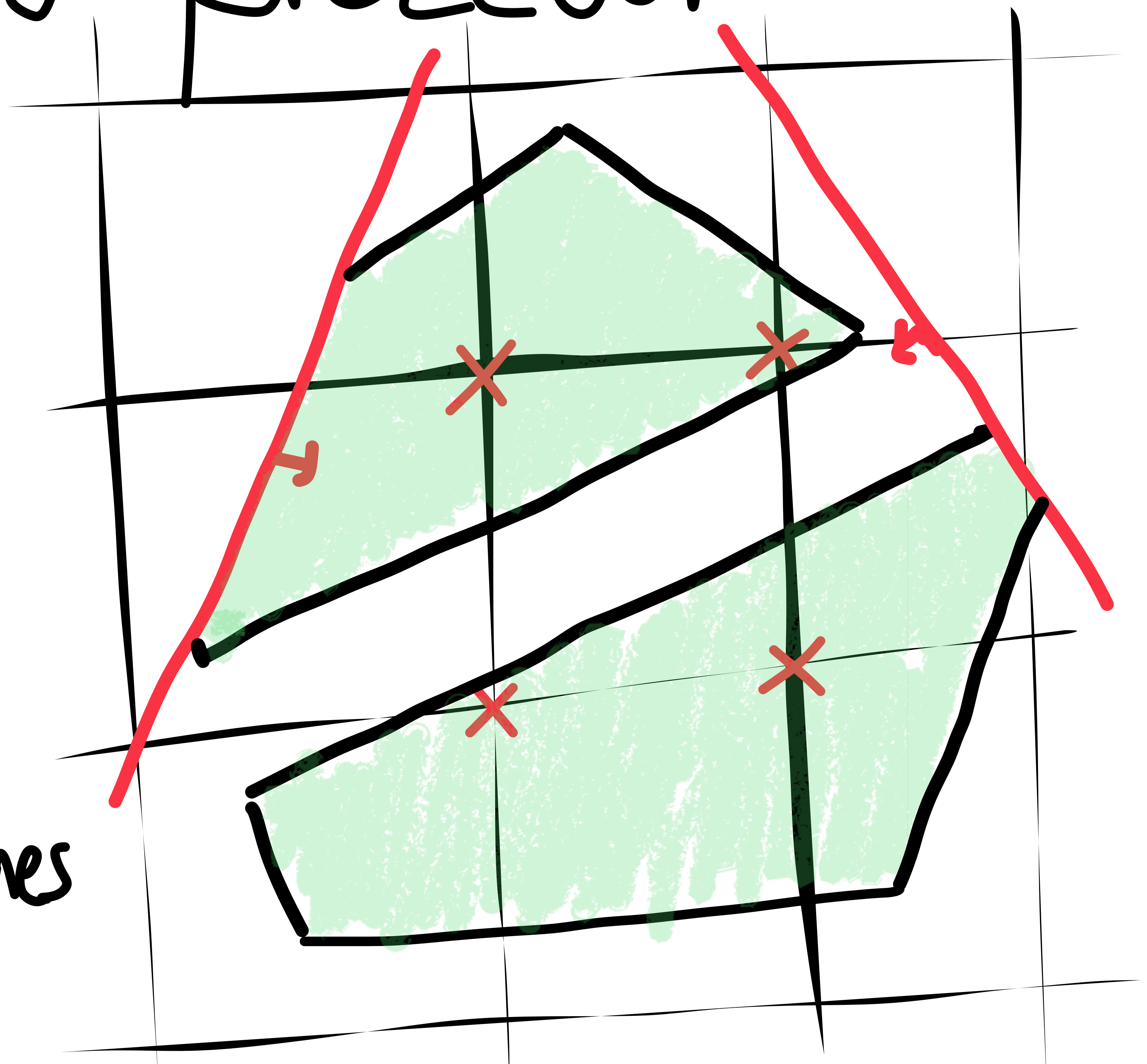
Branch: break P into $P_1, \dots, P_k$ s.t. $P \cap \mathbb{Z}^n \subseteq P_1 \cup \dots \cup P_k$

Cut: Refine $P_1, \dots, P_k$ with additional cutting planes



Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure



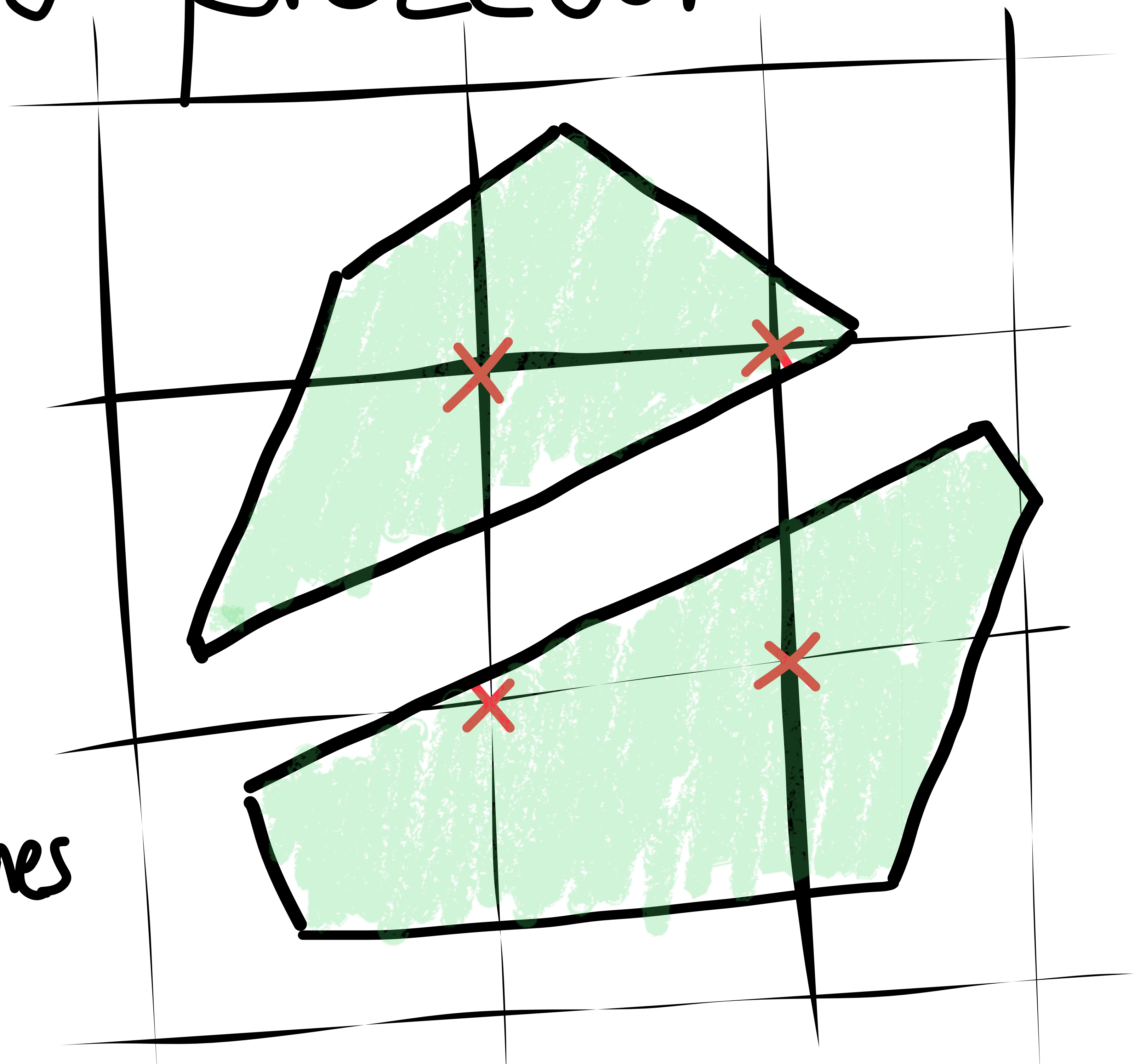
Branch-and-Cut On input P,

Branch: break P into $P_1, \dots, P_k$ s.t. $P \cap \mathbb{Z}^n \subseteq P_1 \cup \dots \cup P_k$

Cut: Refine $P_1, \dots, P_k$ with additional cutting planes

Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure



Branch-and-Cut On input P,

Branch: break P into $P_1, \dots, P_k$ s.t. $P \cap \mathbb{Z}^n \subseteq P_1 \cup \dots \cup P_k$

Cut: Refine $P_1, \dots, P_k$ with additional cutting planes

Repeat

Stabbing Planes [BFI+18]

- Formalizes practical branch-and-cut as a proof system

Stabbing Planes [BFI+18]

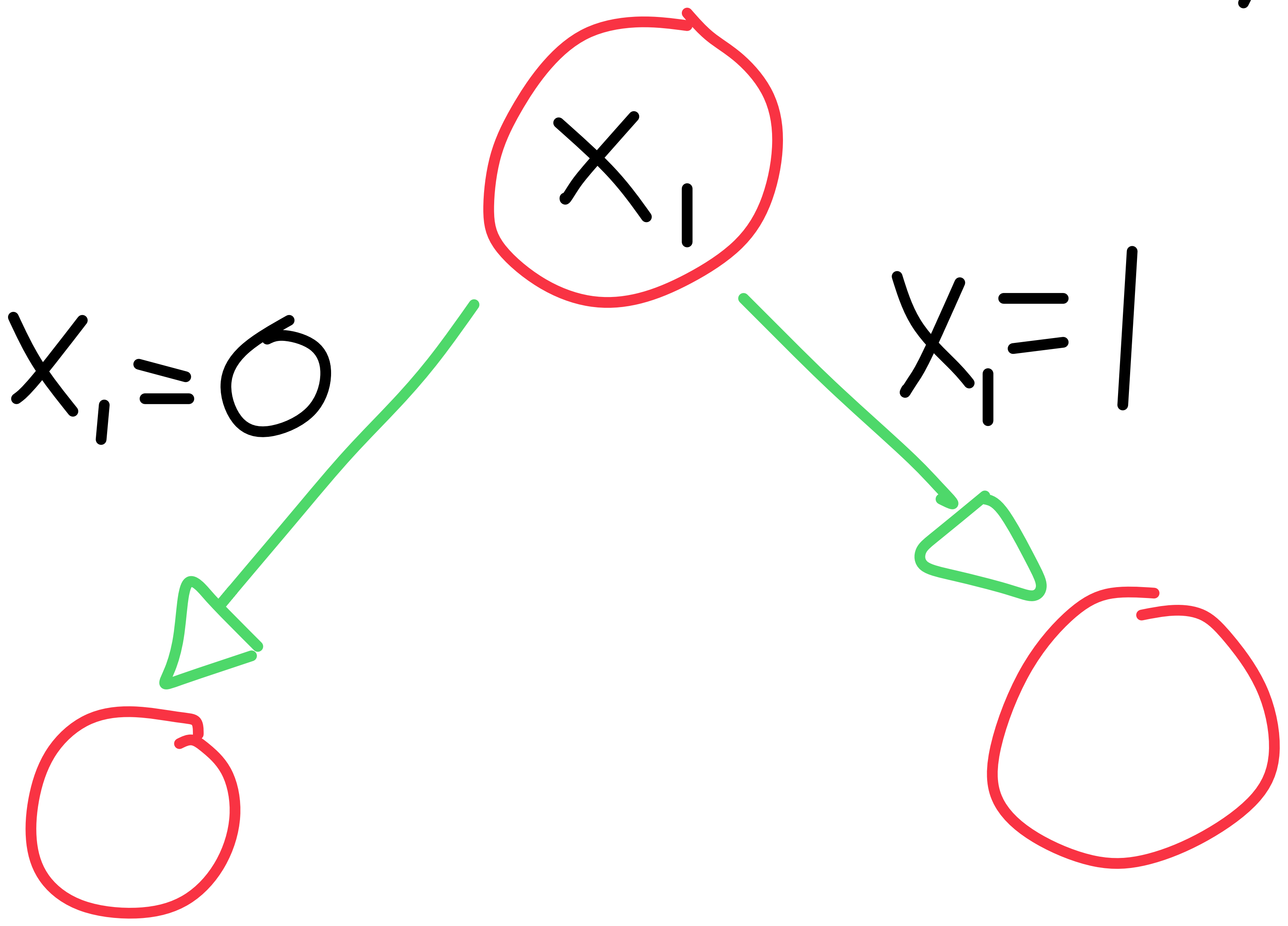
- Formalizes practical branch-and-cut as a proof system
- Extends DPLL to reason about linear inequalities

DPLL Refutation

$$\{x_1 \vee x_2,\ \bar{x}_1 \vee x_2,\ x_1 \vee \bar{x}_2,\ \bar{x}_1 \vee \bar{x}_2\}$$

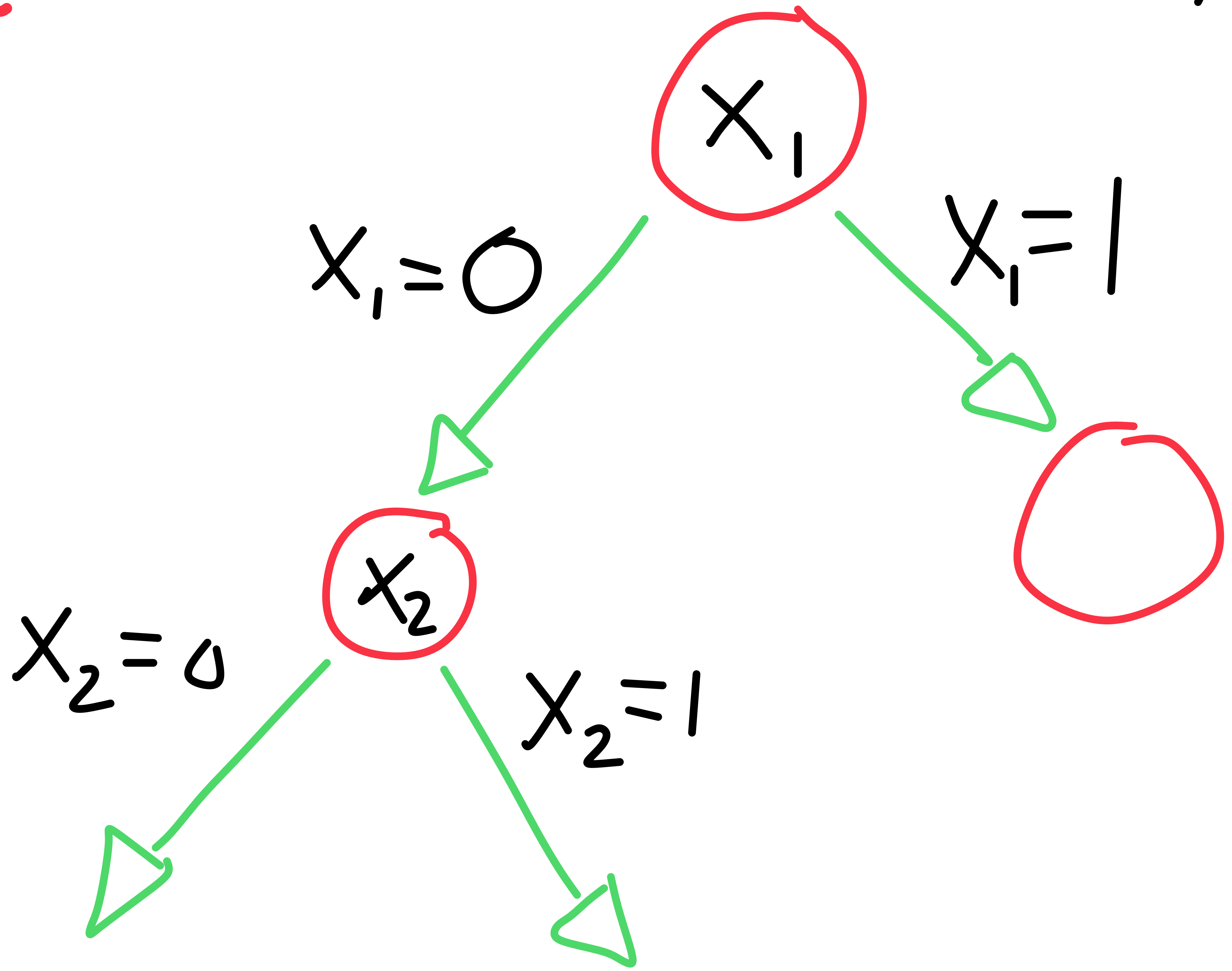
DPLL Refutation

$$\{x_1 \vee x_2,\ \bar{x}_1 \vee x_2,\ x_1 \vee \bar{x}_2,\ \bar{x}_1 \vee \bar{x}_2\}$$



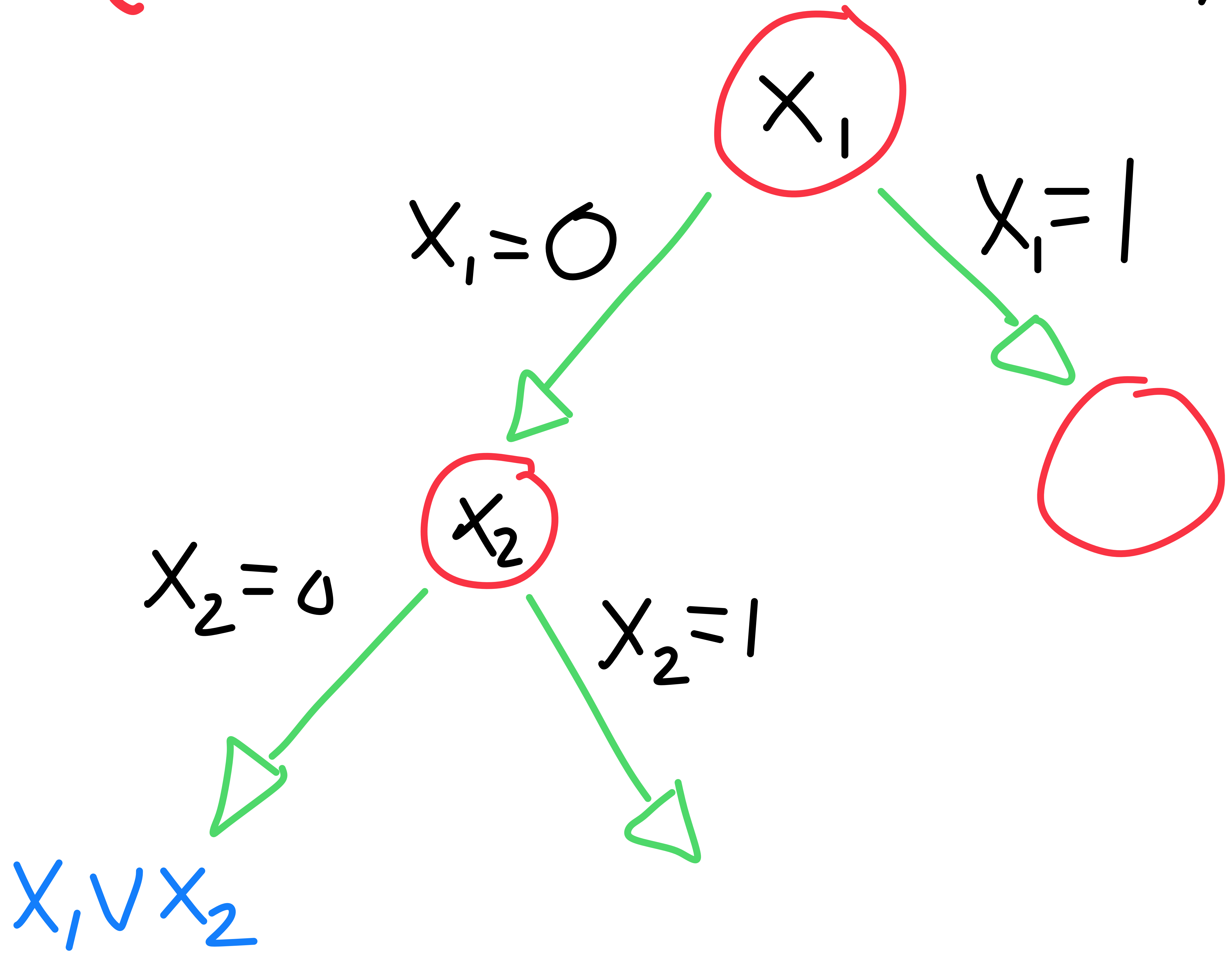
DPLL Refutation

$$\{x_1 \vee x_2,\ \bar{x}_1 \vee x_2,\ x_1 \vee \bar{x}_2,\ \bar{x}_1 \vee \bar{x}_2\}$$



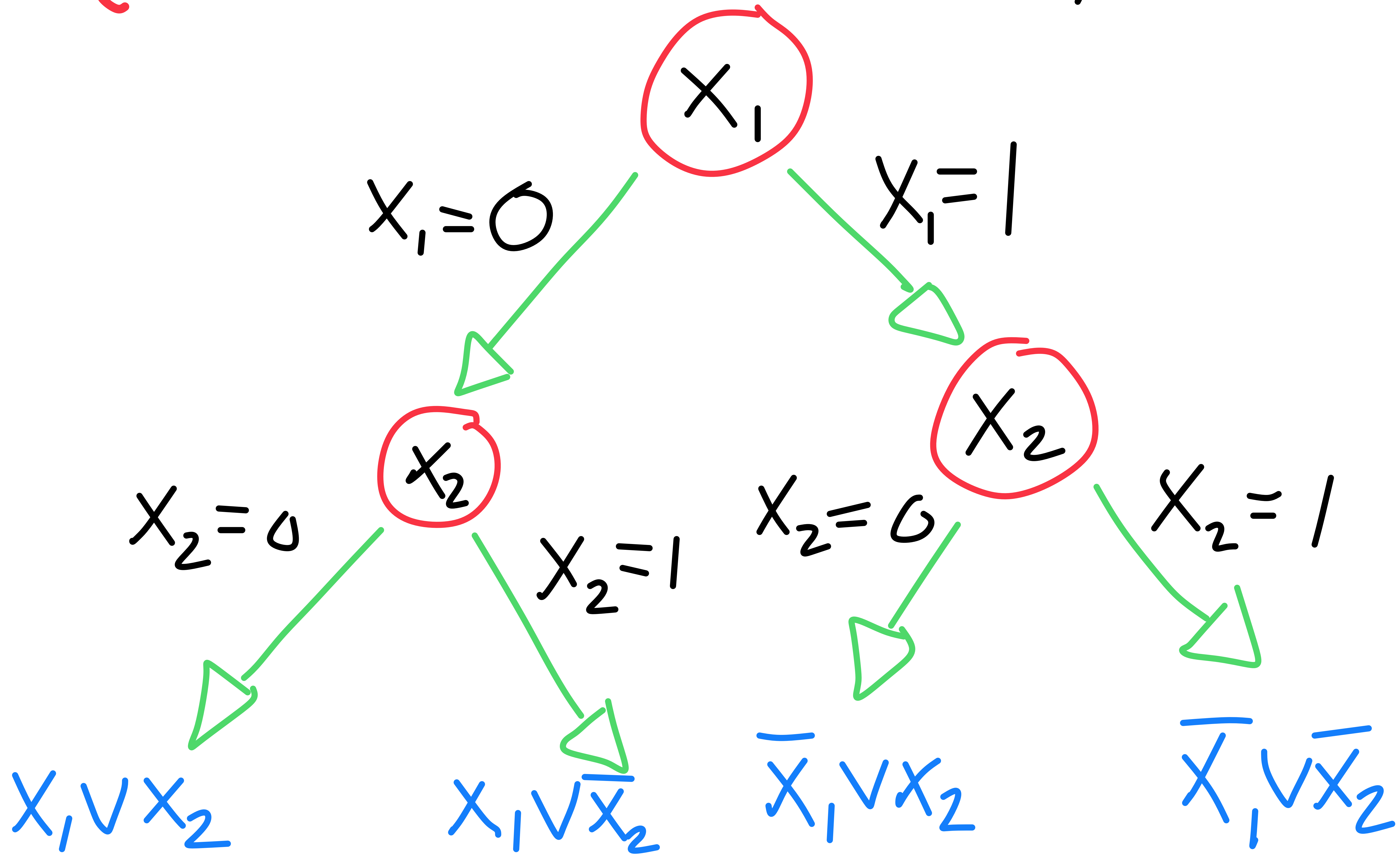
DPLL Refutation

$$\{x_1 \vee x_2,\ \bar{x}_1 \vee x_2,\ x_1 \vee \bar{x}_2,\ \bar{x}_1 \vee \bar{x}_2\}$$



DPLL Refutation

$$\{x_1 \vee x_2,\ \bar{x}_1 \vee x_2,\ x_1 \vee \bar{x}_2,\ \bar{x}_1 \vee \bar{x}_2\}$$



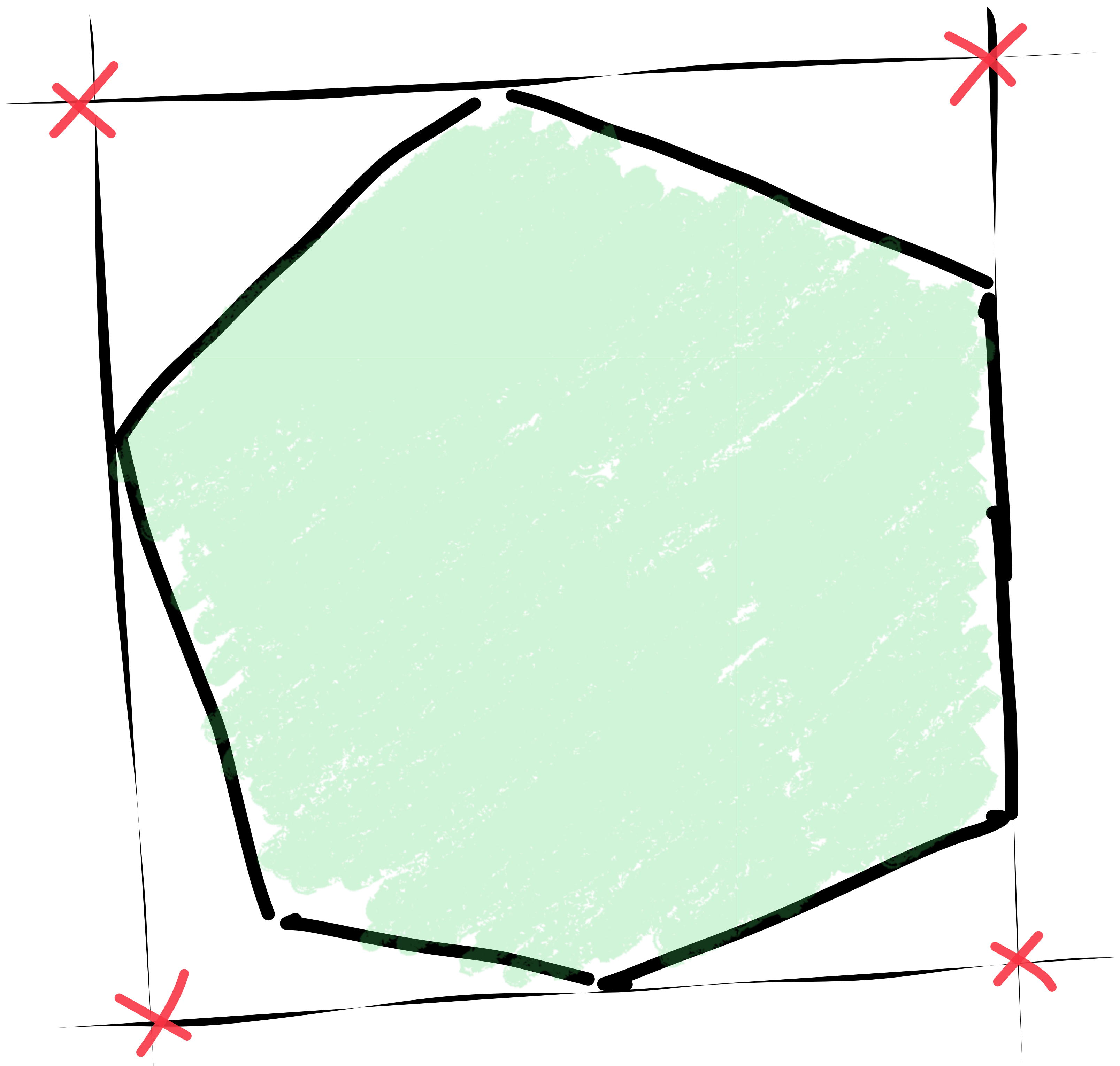
DPLL as Polytopes

$P = \{x_1 + x_2 \geq 1,\ x_1 - x_2 \geq 0,\ x_2 - x_1 \geq 0,\ -x_1 - x_2 \geq -1,\ 0 \leq x_i \leq 1\}$

DPLL as Polytopes

$$P = \{ x_1 + x_2 \geq 1,\ x_1 - x_2 \geq 0,\ x_2 - x_1 \geq 0,\ -x_1 - x_2 \geq -1,\ 0 \leq x_i \leq 1 \}$$

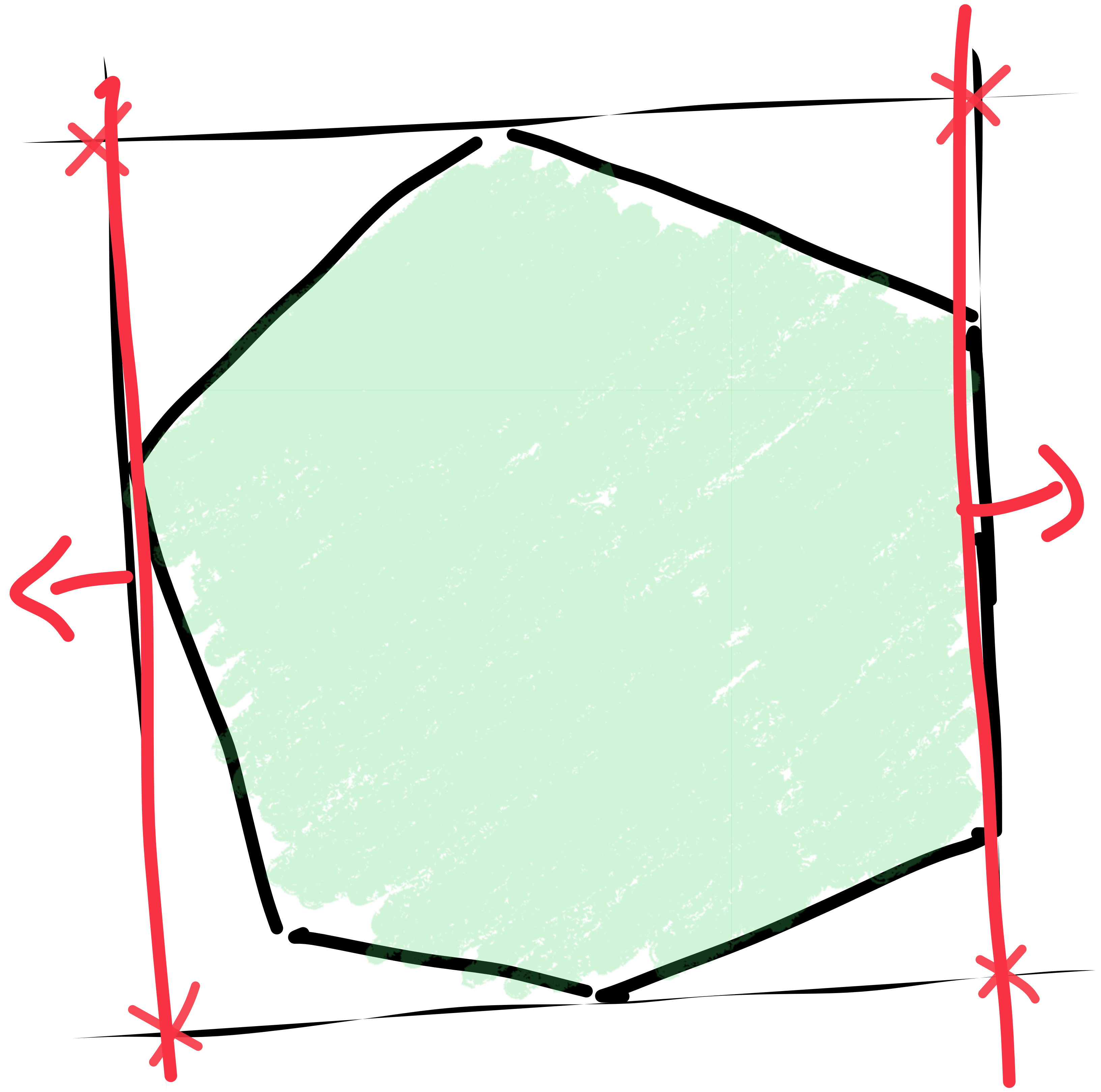
P



DPLL as Polytopes

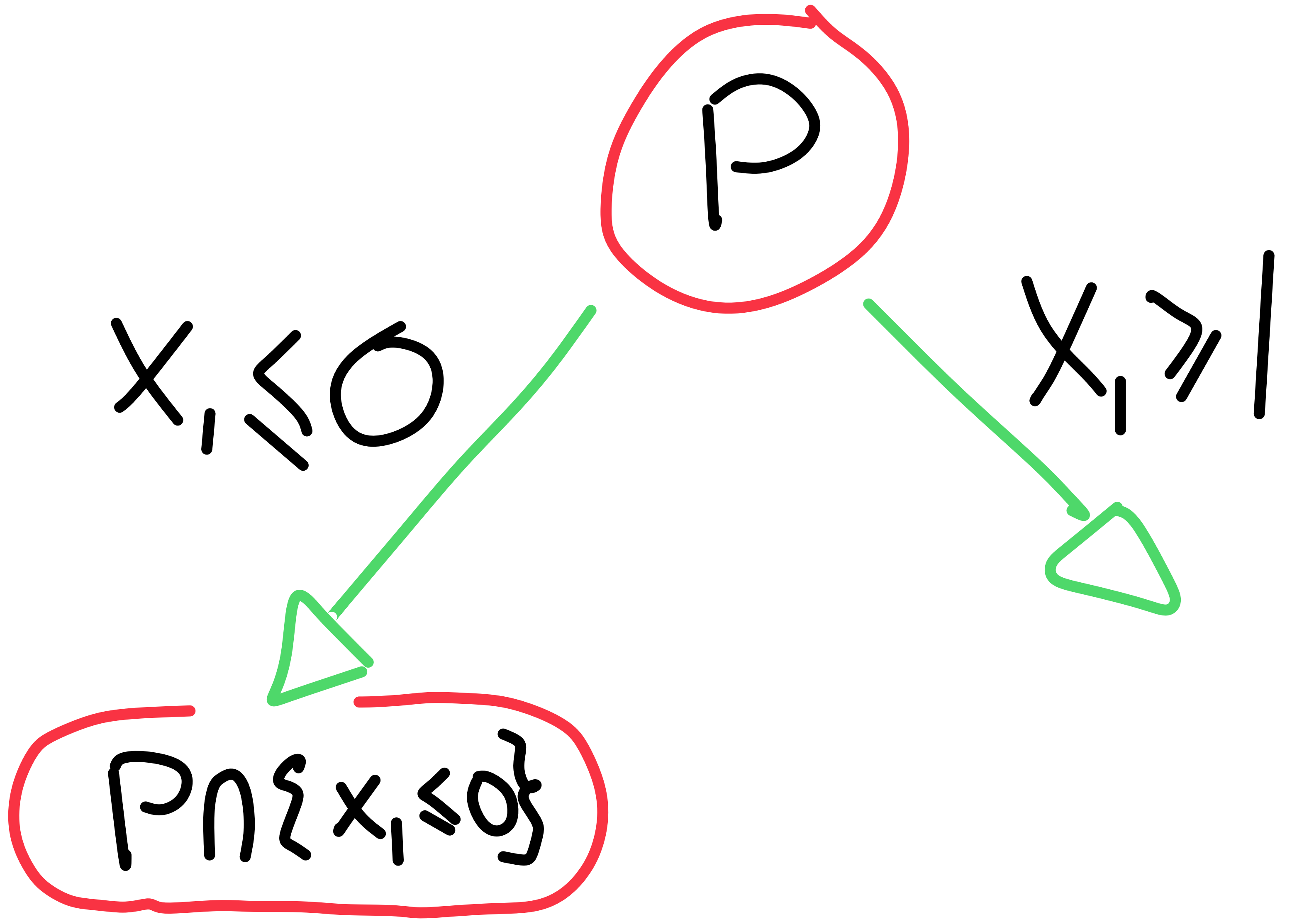
$P = \{x_1 + x_2 \geq 1,\ x_1 - x_2 \geq 0,\ x_2 - x_1 \geq 0,\ -x_1 - x_2 \geq -1,\ 0 \leq x_i \leq 1\}$

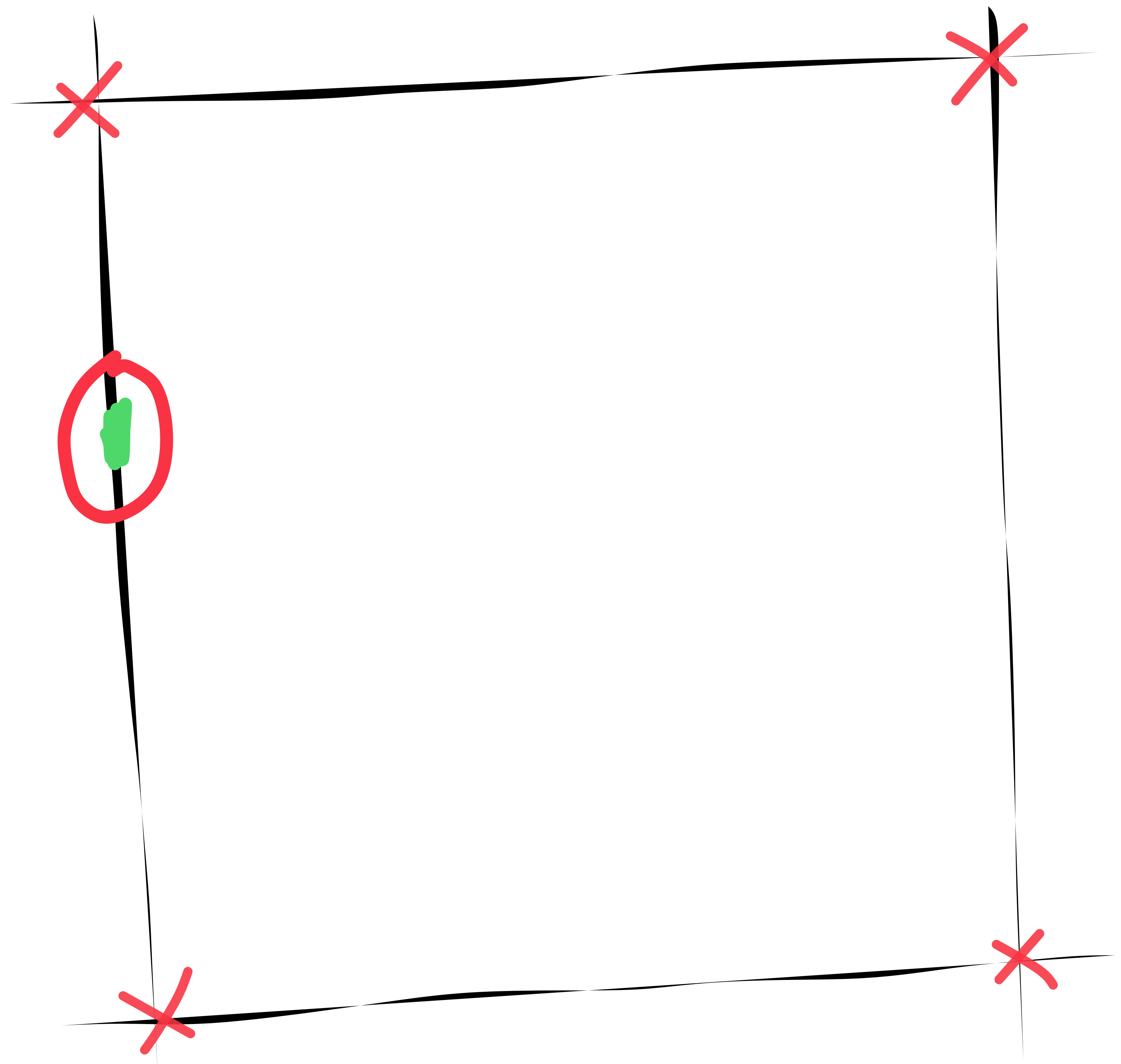
$$
\begin{array}{ccc}
 & P & \\
\swarrow^{x_1 \leq 0} & & \searrow^{x_1 \geq 1}
\end{array}
$$



DPLL as Polytopes

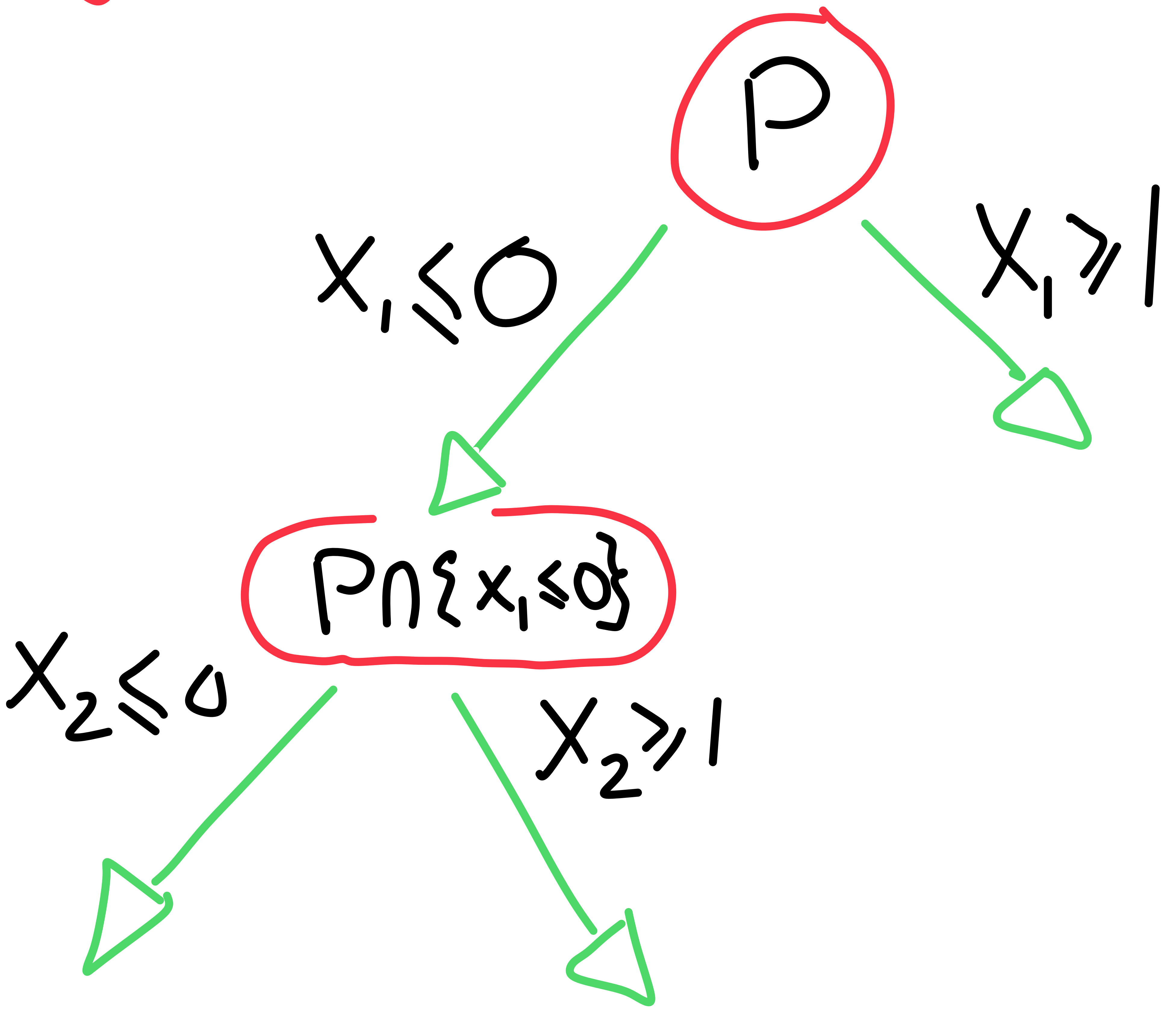
$$P = \{ x_1 + x_2 \geq 1,\ x_1 - x_2 \geq 0,\ x_2 - x_1 \geq 0,\ -x_1 - x_2 \geq -1,\ 0 \leq x_i \leq 1 \}$$

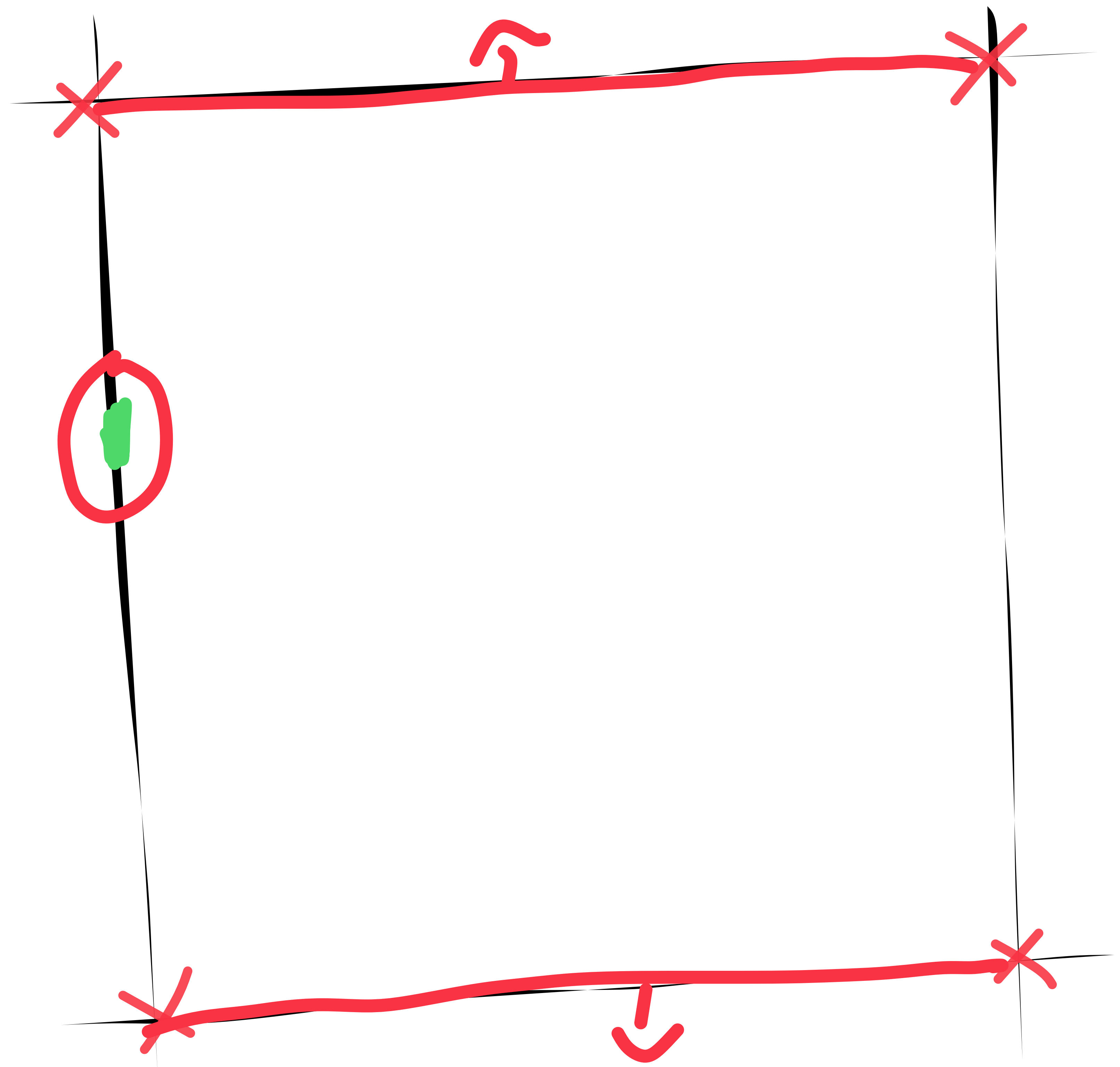




DPLL as Polytopes

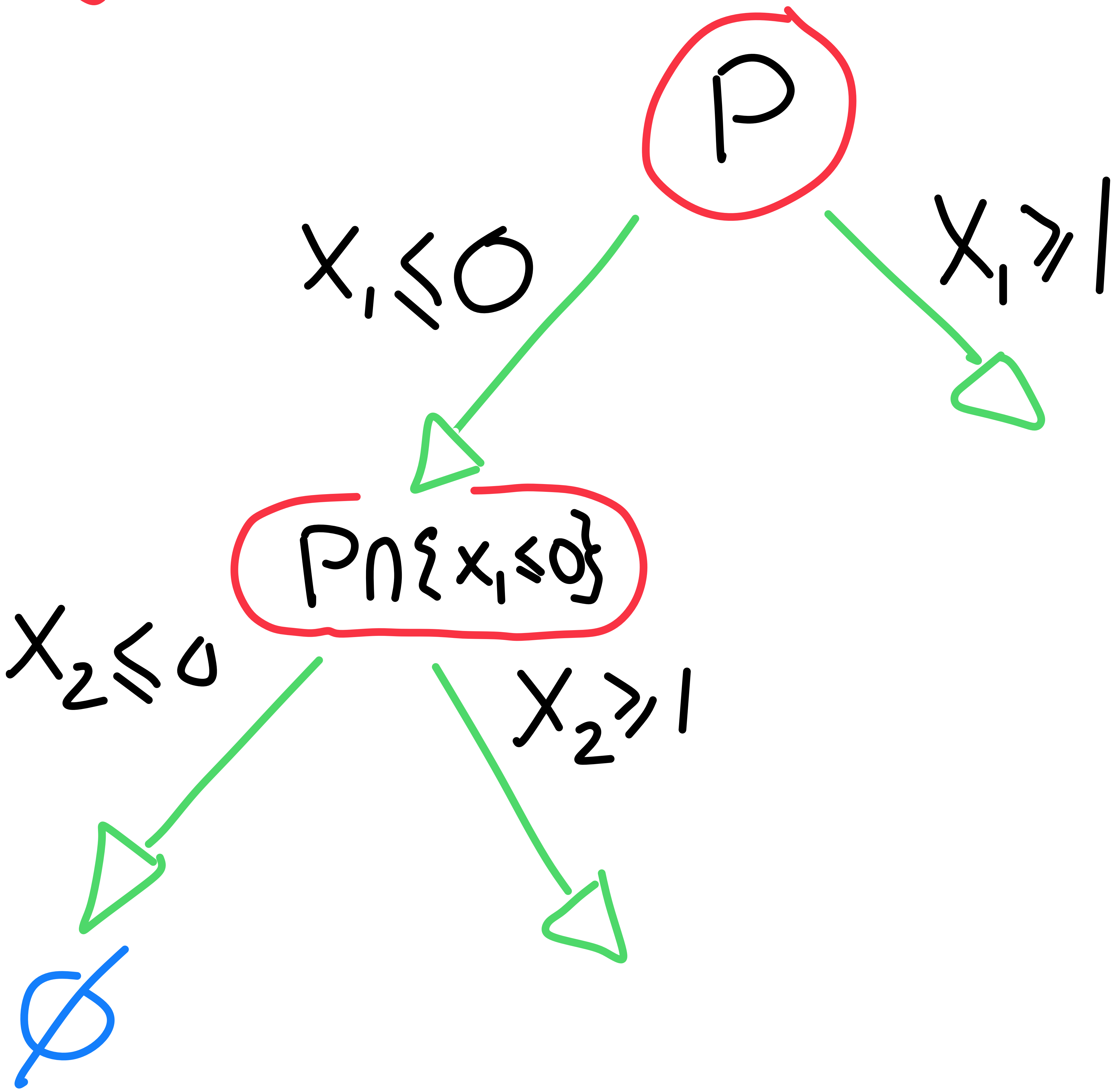
$$P = \{ x_1 + x_2 \geq 1,\ x_1 - x_2 \geq 0,\ x_2 - x_1 \geq 0,\ -x_1 - x_2 \geq -1,\ 0 \leq x_i \leq 1 \}$$

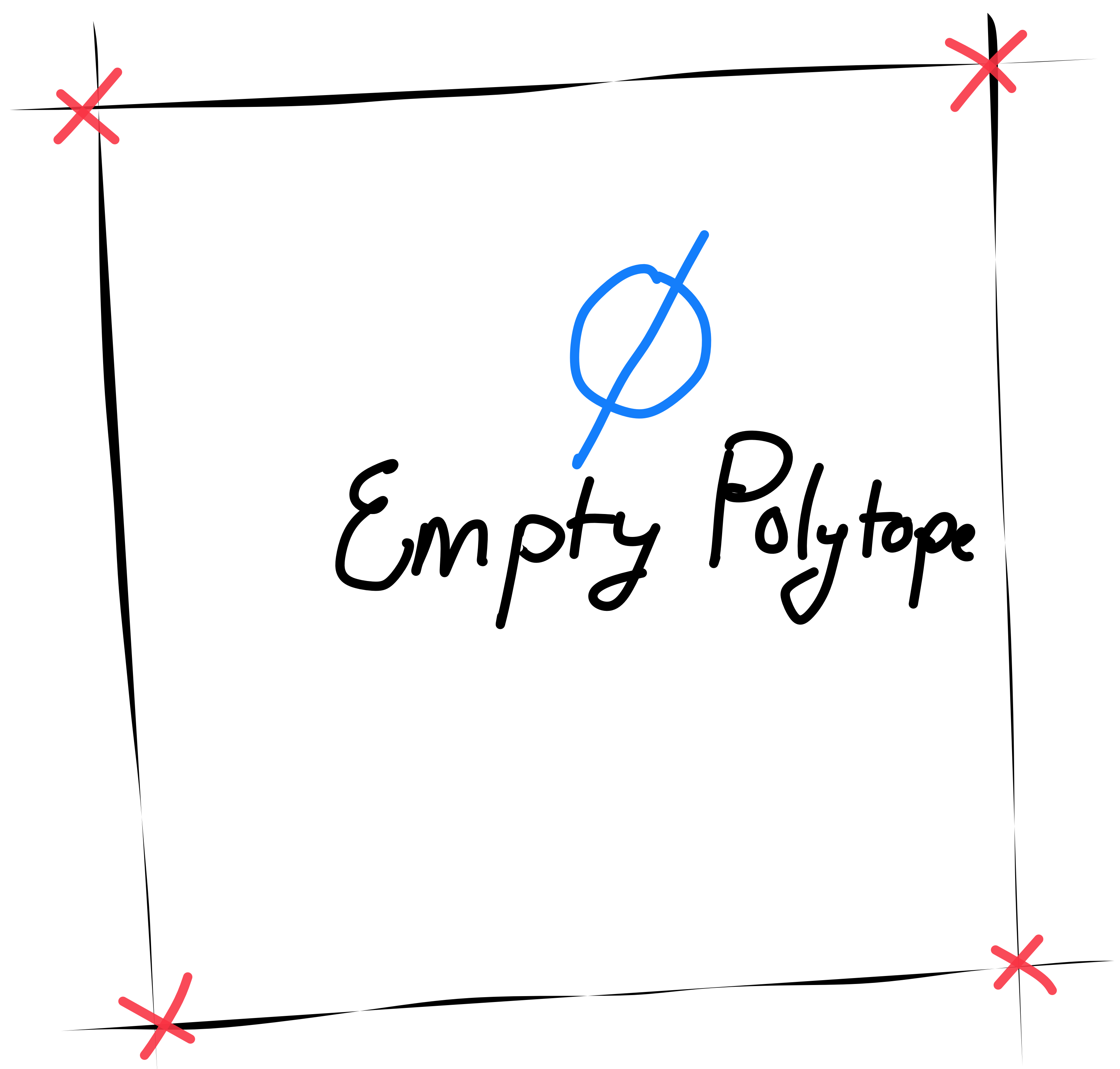




DPLL as Polytopes

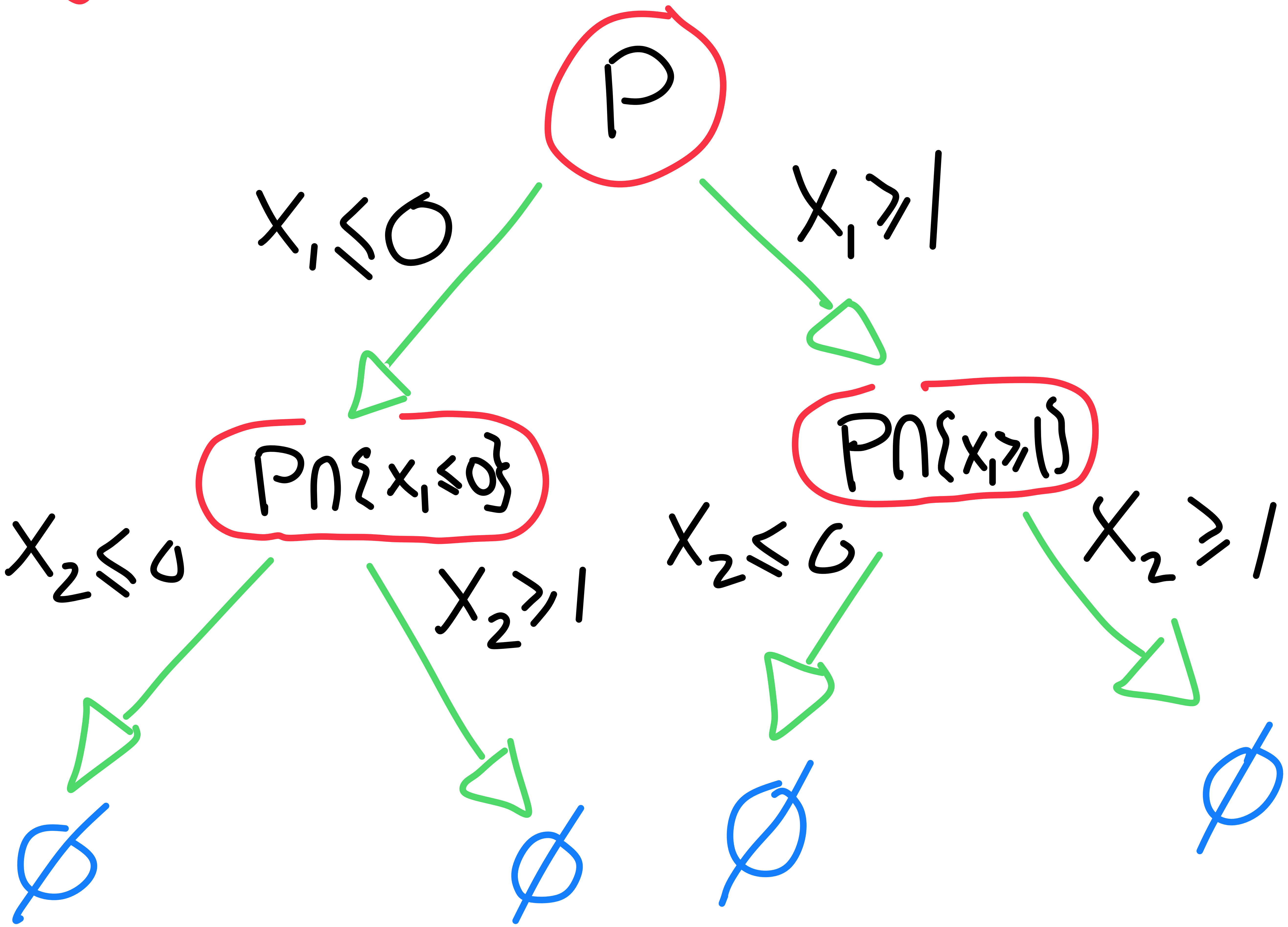
$$P = \{ x_1 + x_2 \geq 1,\ x_1 - x_2 \geq 0,\ x_2 - x_1 \geq 0,\ -x_1 - x_2 \geq -1,\ 0 \leq x_i \leq 1 \}$$

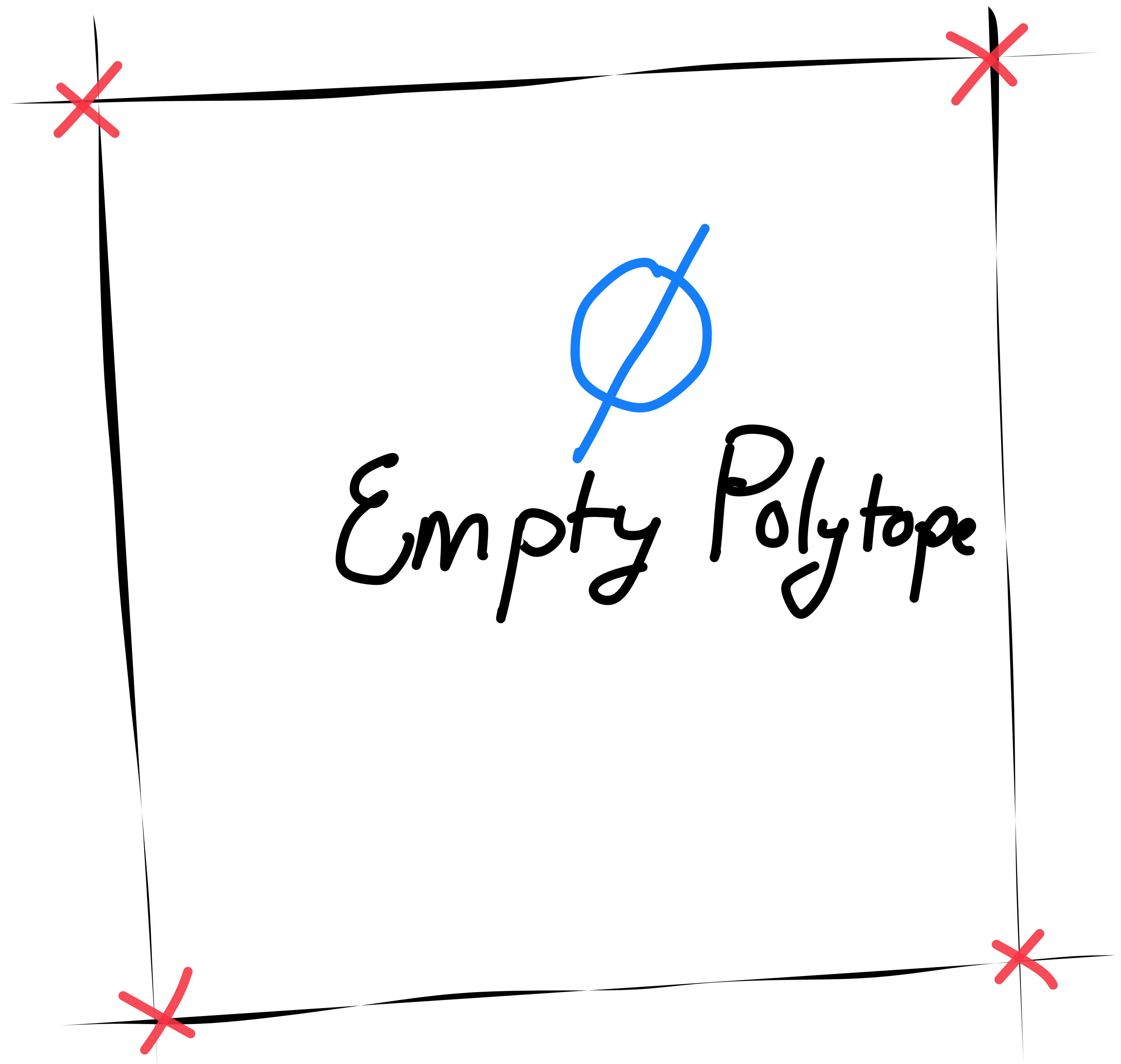




DPLL as Polytopes

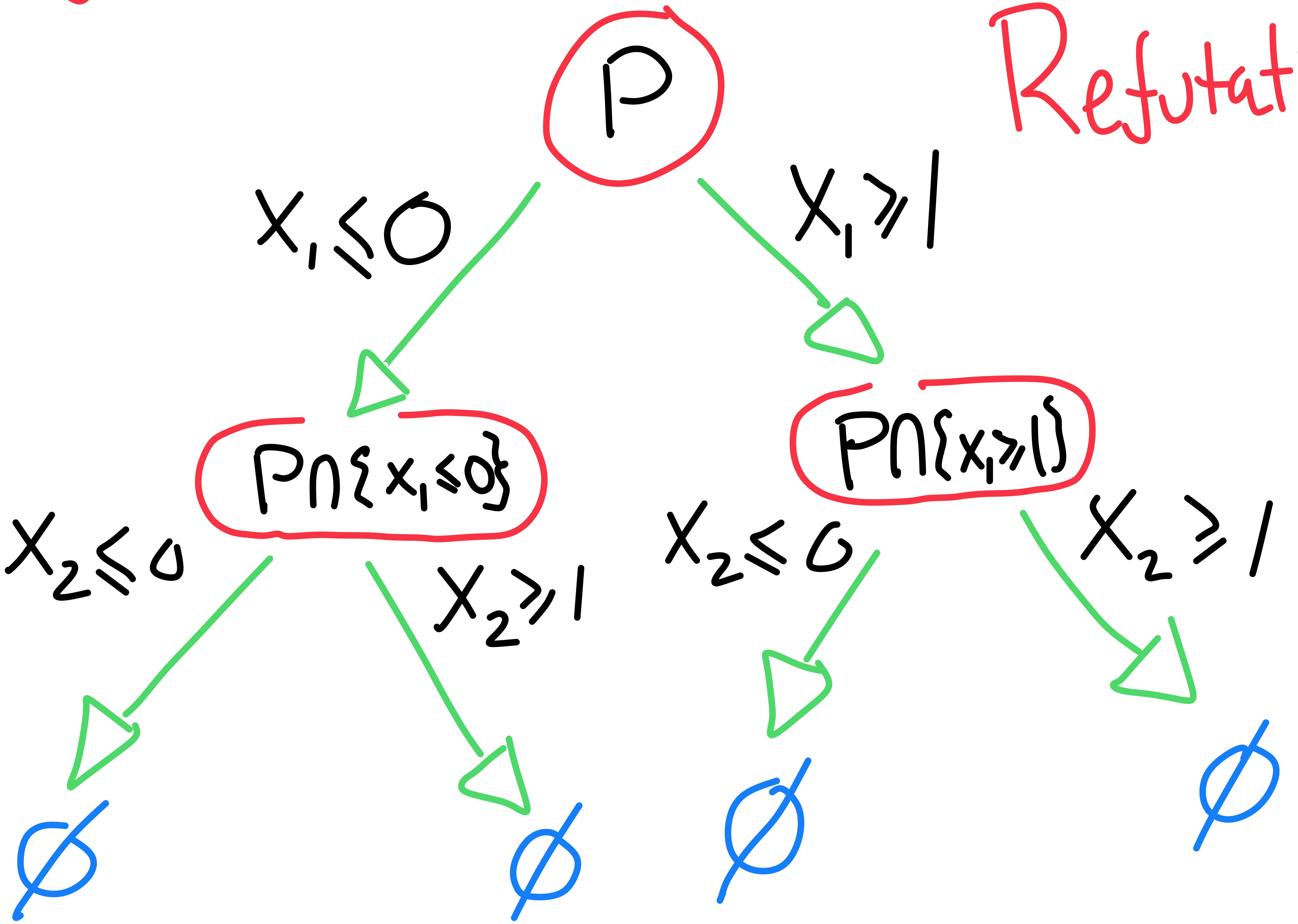
$$P = \{x_1 + x_2 \geq 1,\ x_1 - x_2 \geq 0,\ x_2 - x_1 \geq 0,\ -x_1 - x_2 \geq -1,\ 0 \leq x_i \leq 1\}$$





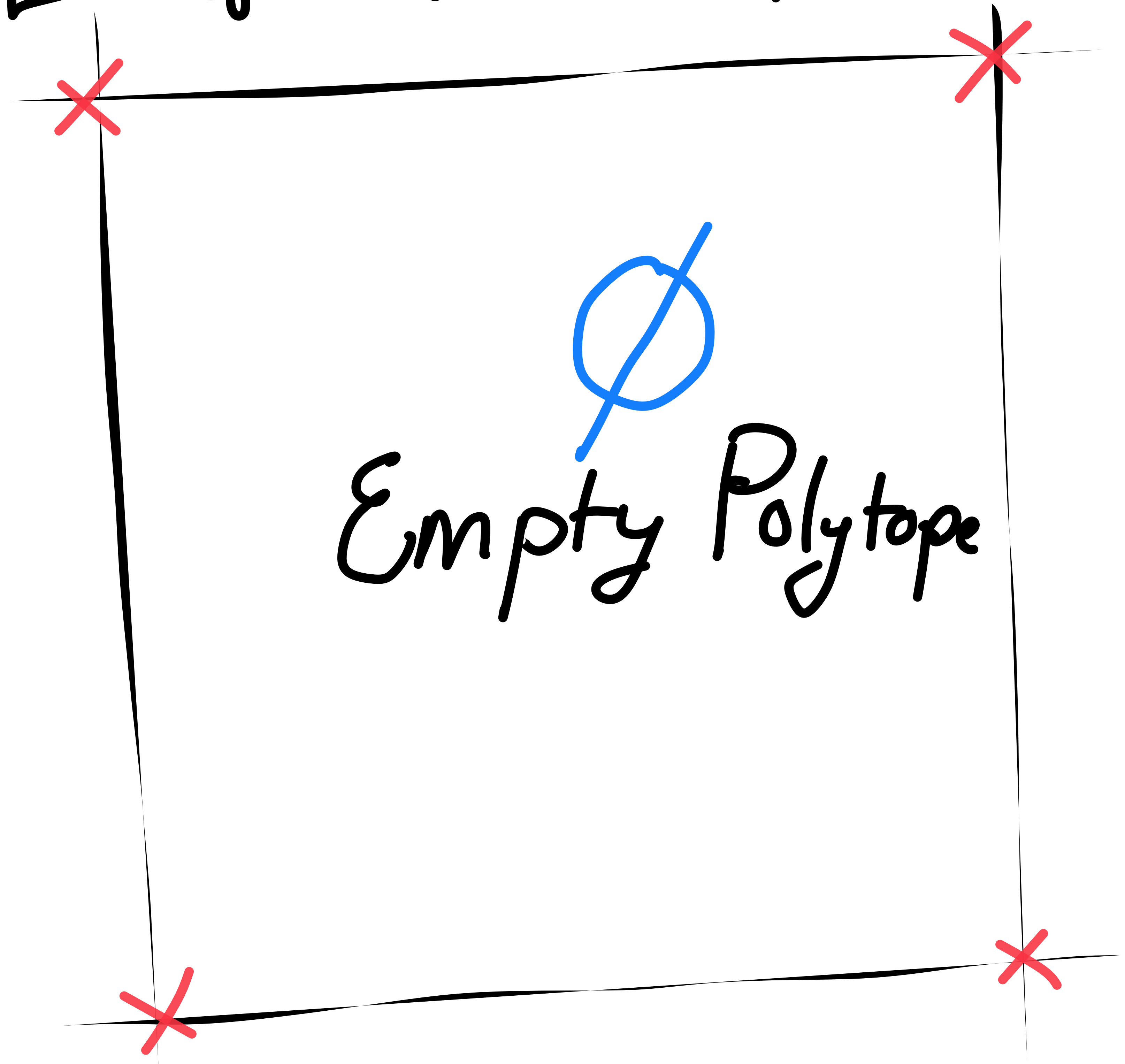
DPLL as Polytopes

$$P = \{ x_1 + x_2 \geq 1,\ x_1 - x_2 \geq 0,\ x_2 - x_1 \geq 0,\ -x_1 - x_2 \geq -1,\ 0 \leq x_i \leq 1 \}$$



Refutation!

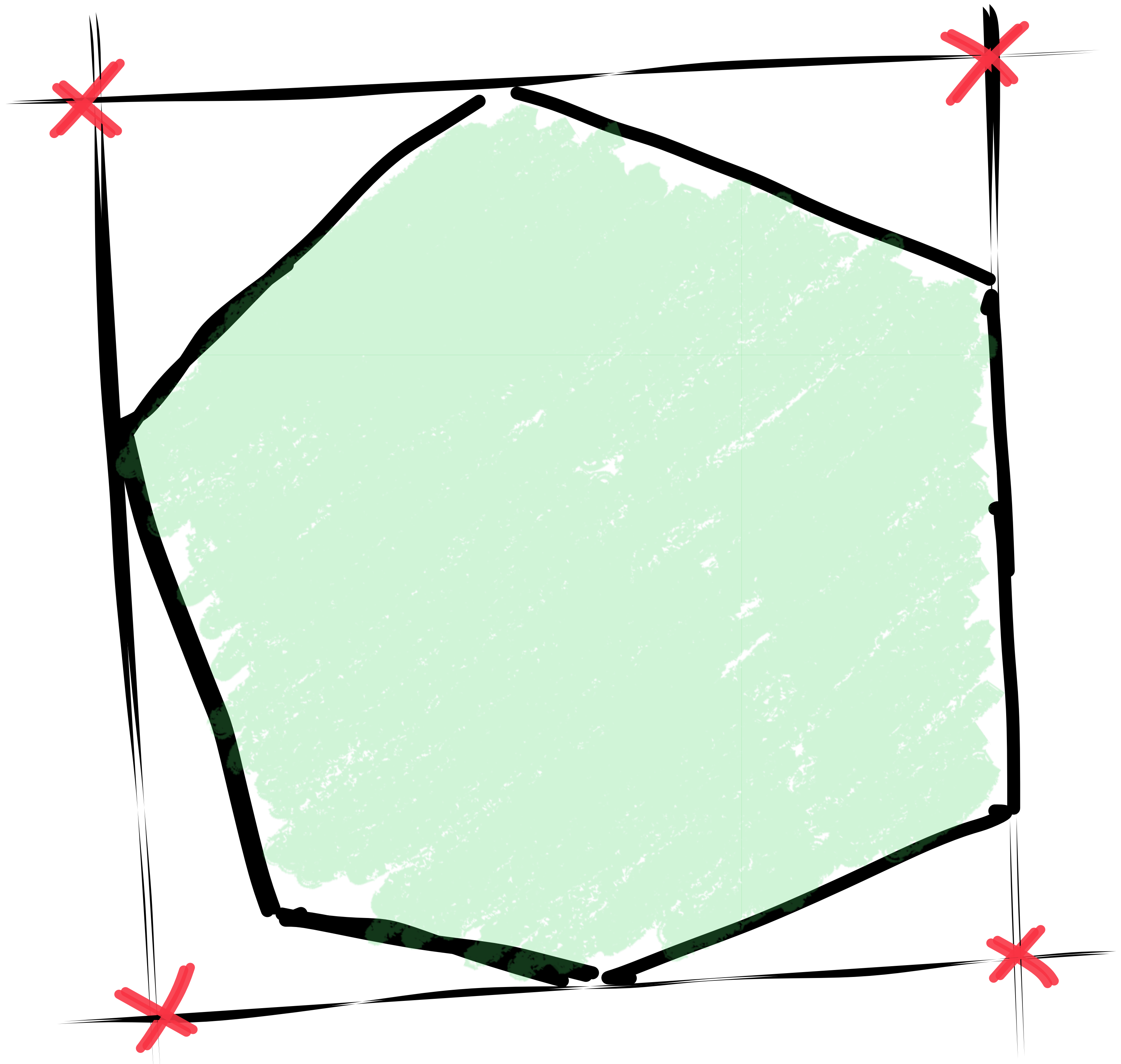
$\emptyset$ Derived at every Leaf of the tree



Stabbing Planes

$P = \{Ax \geq b\}$ s.t. $P \cap \mathbb{Z}^n = \emptyset$.

P

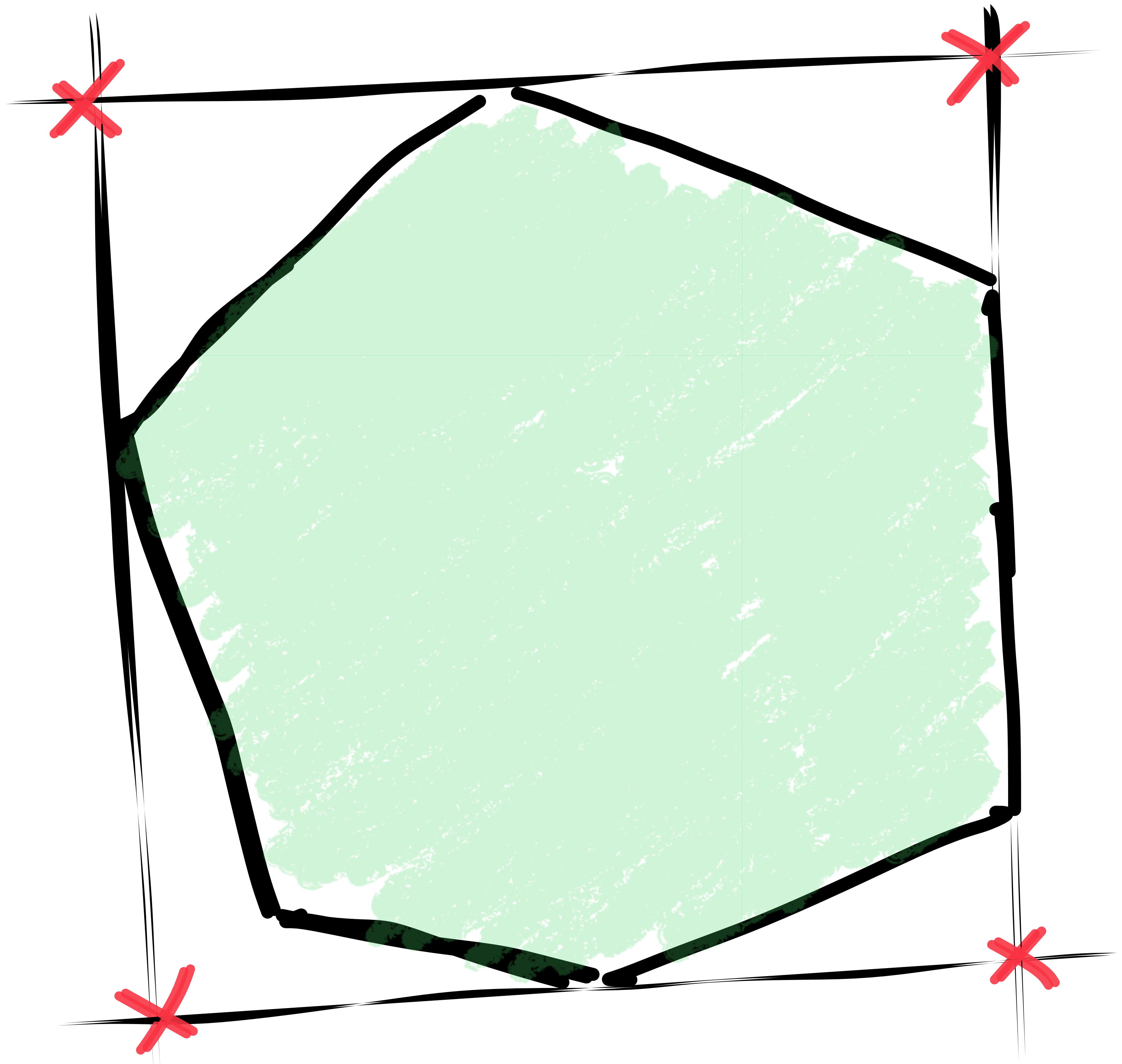


Stabbing Planes

$P = \{Ax \geq b\}$ s.t. $P \cap \mathbb{Z}^n = \emptyset$.

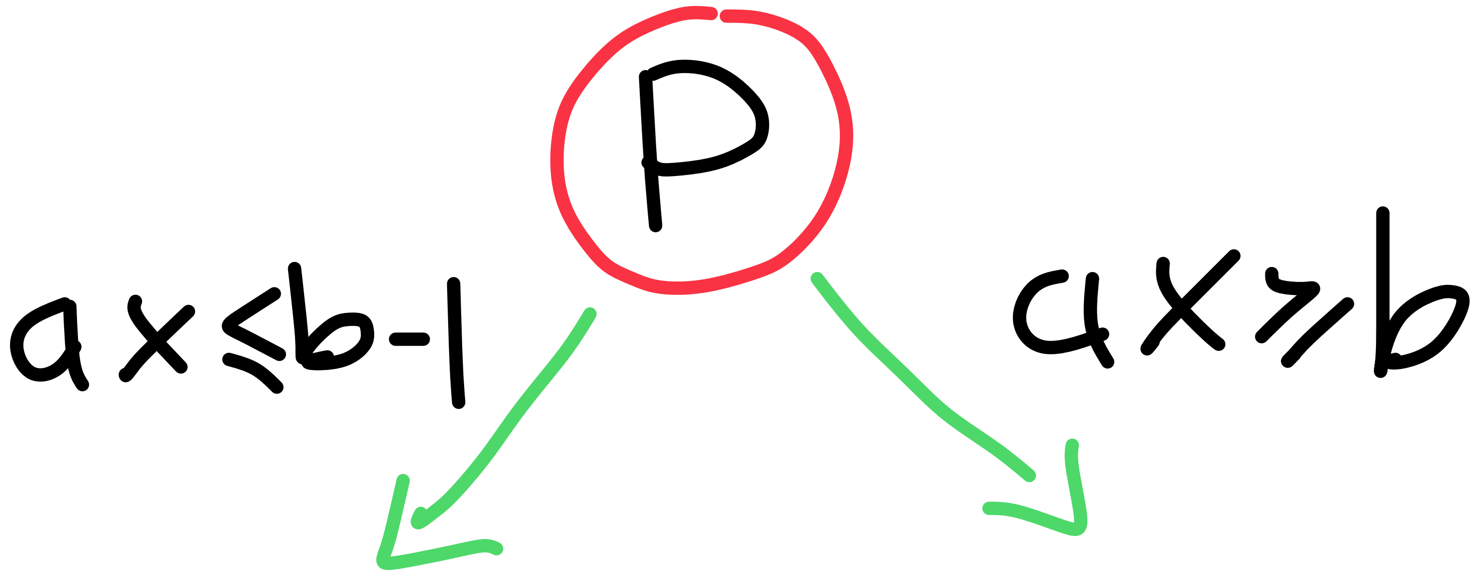
P

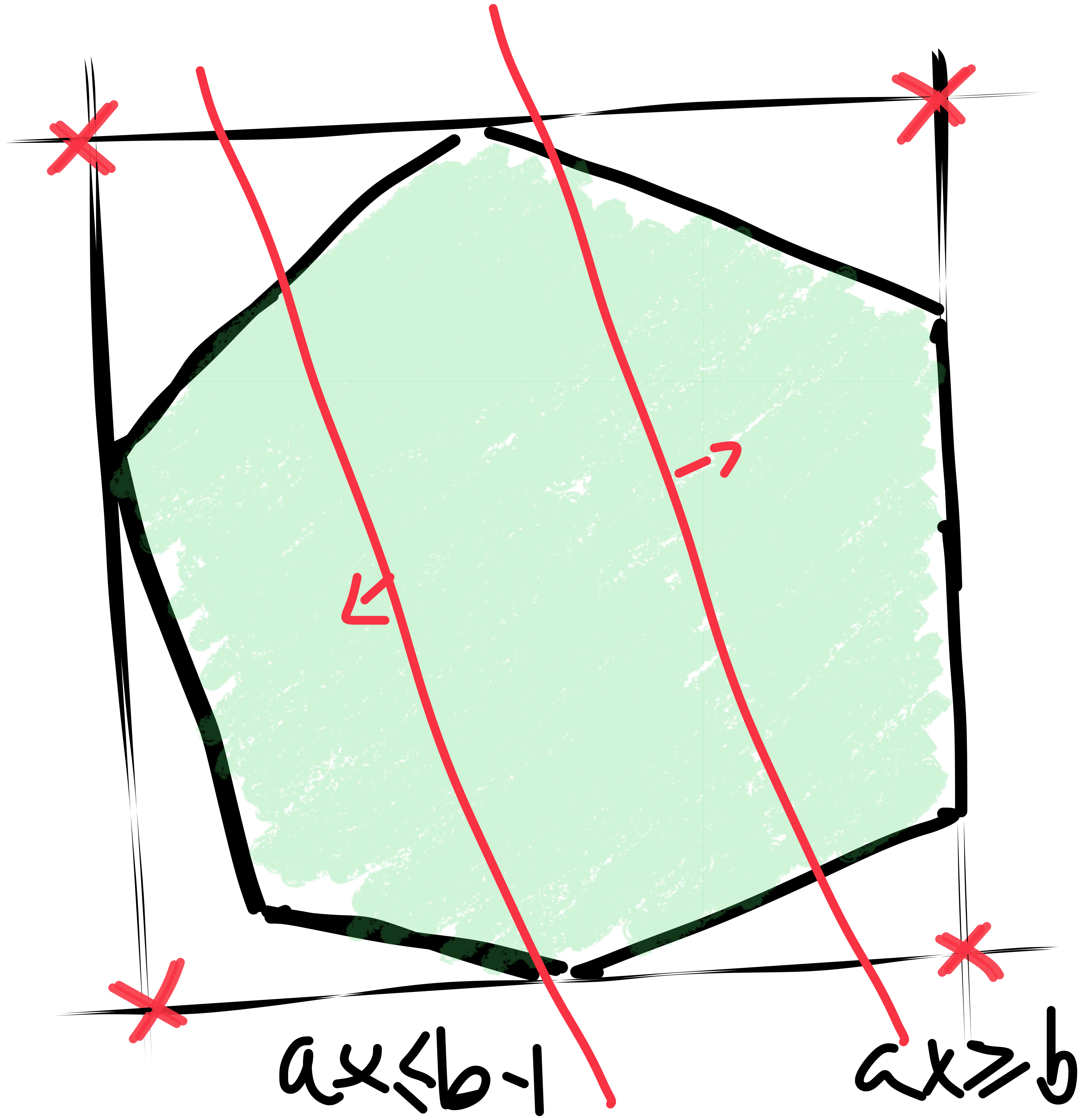
"query arbitrary linear inequalities"



Stabbing Planes

$P = \{Ax \geq b\}$ s.t. $P \cap \mathbb{Z}^n = \emptyset$.

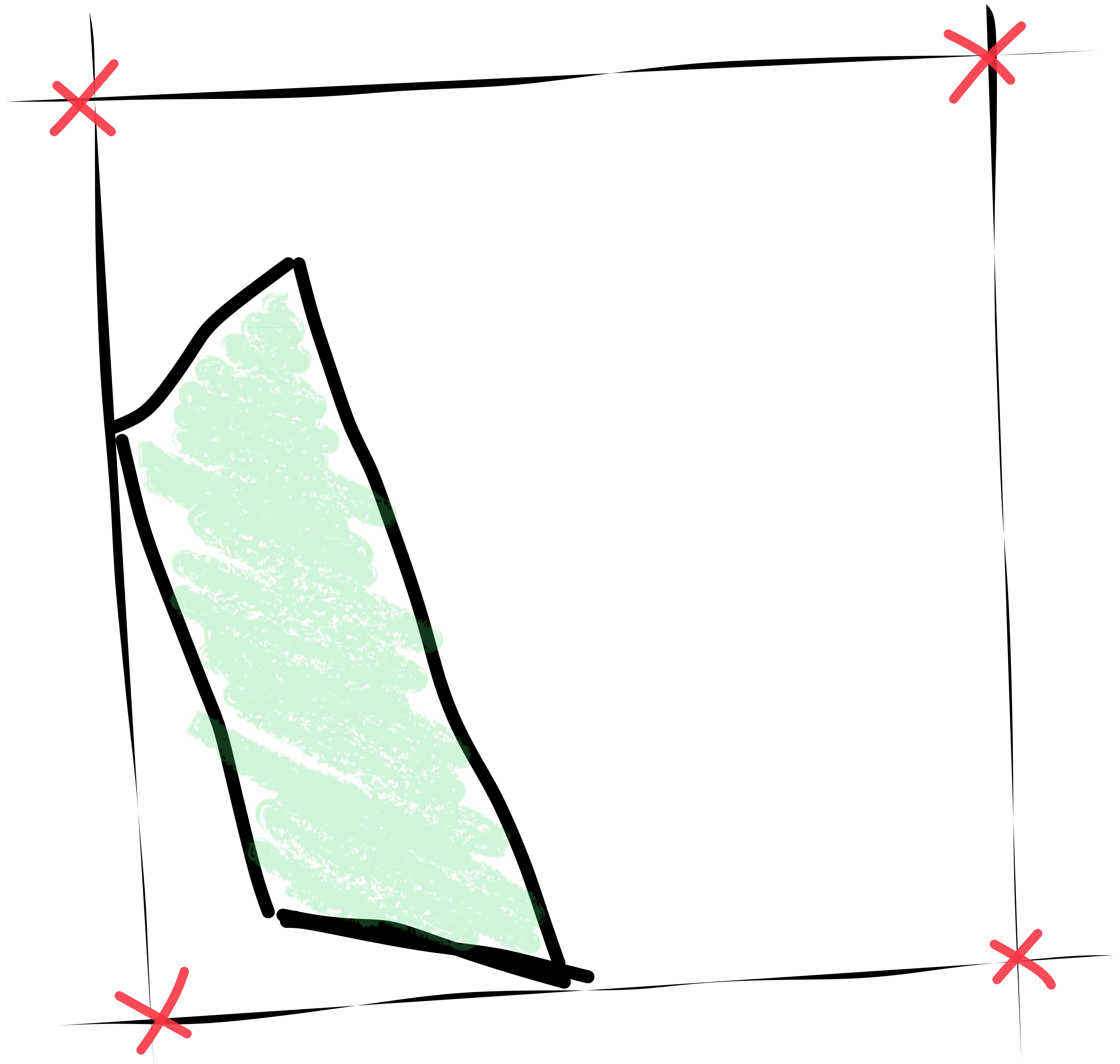




Stabbing Planes

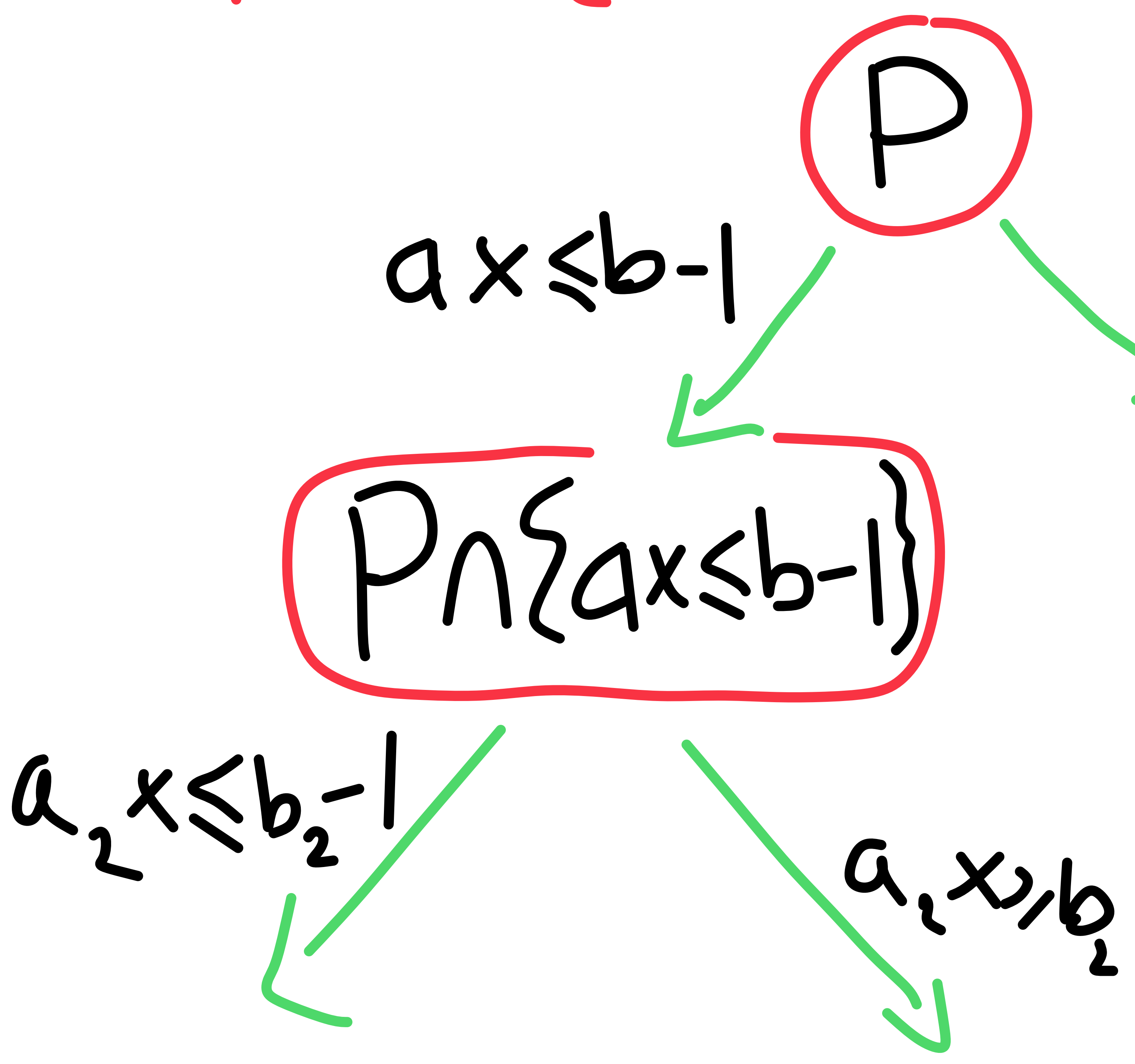
$P = \{Ax \geq b\}$ s.t. $P \cap \mathbb{Z}^n = \emptyset$.

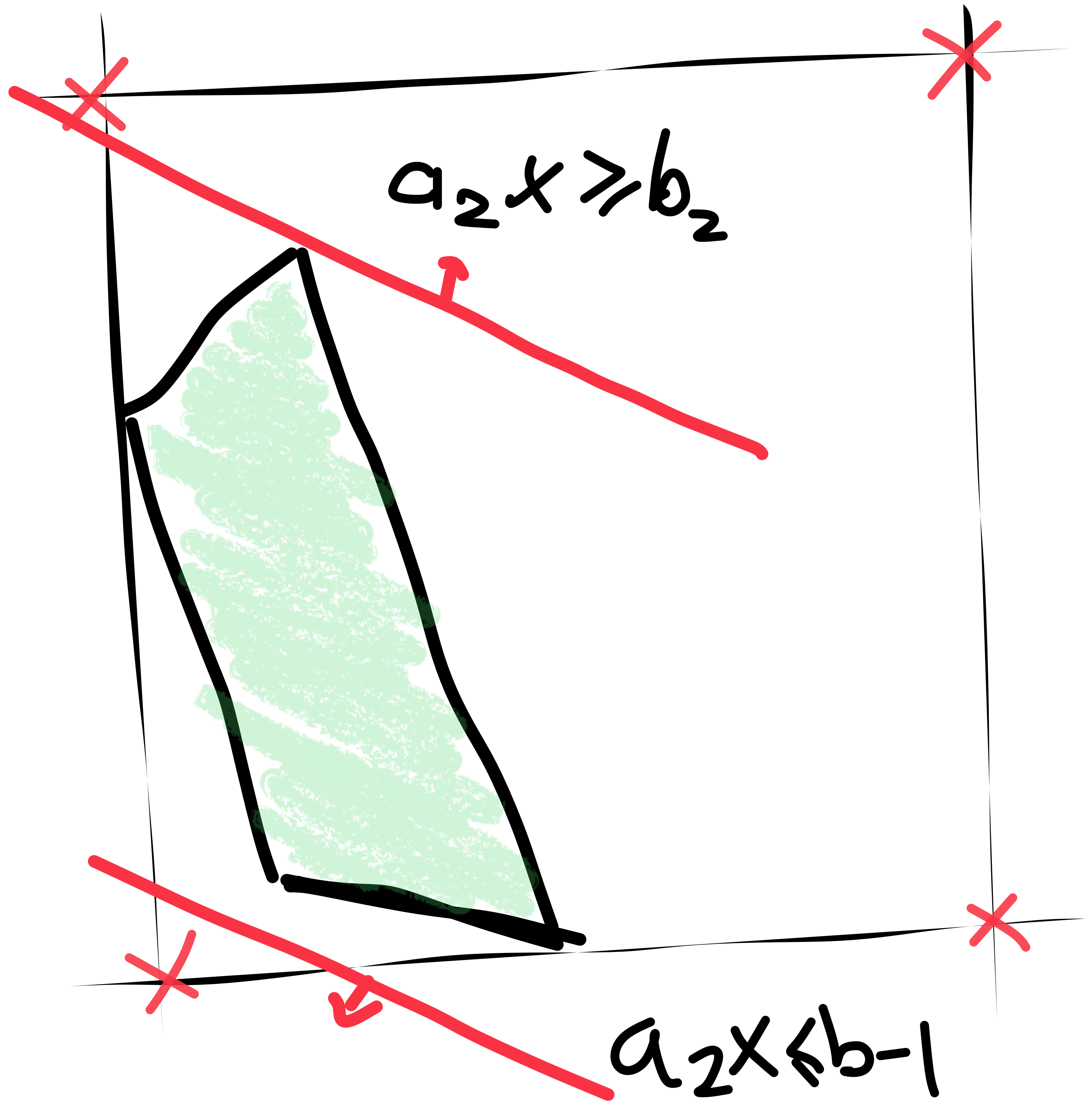
$$\begin{tikzcd} & P \arrow[dl, "ax \leq b-1"'] \arrow[dr, "ax \geq b"] & \\ \boxed{P \cap \{ax \leq b-1\}} & & {} \end{tikzcd}$$



Stabbing Planes

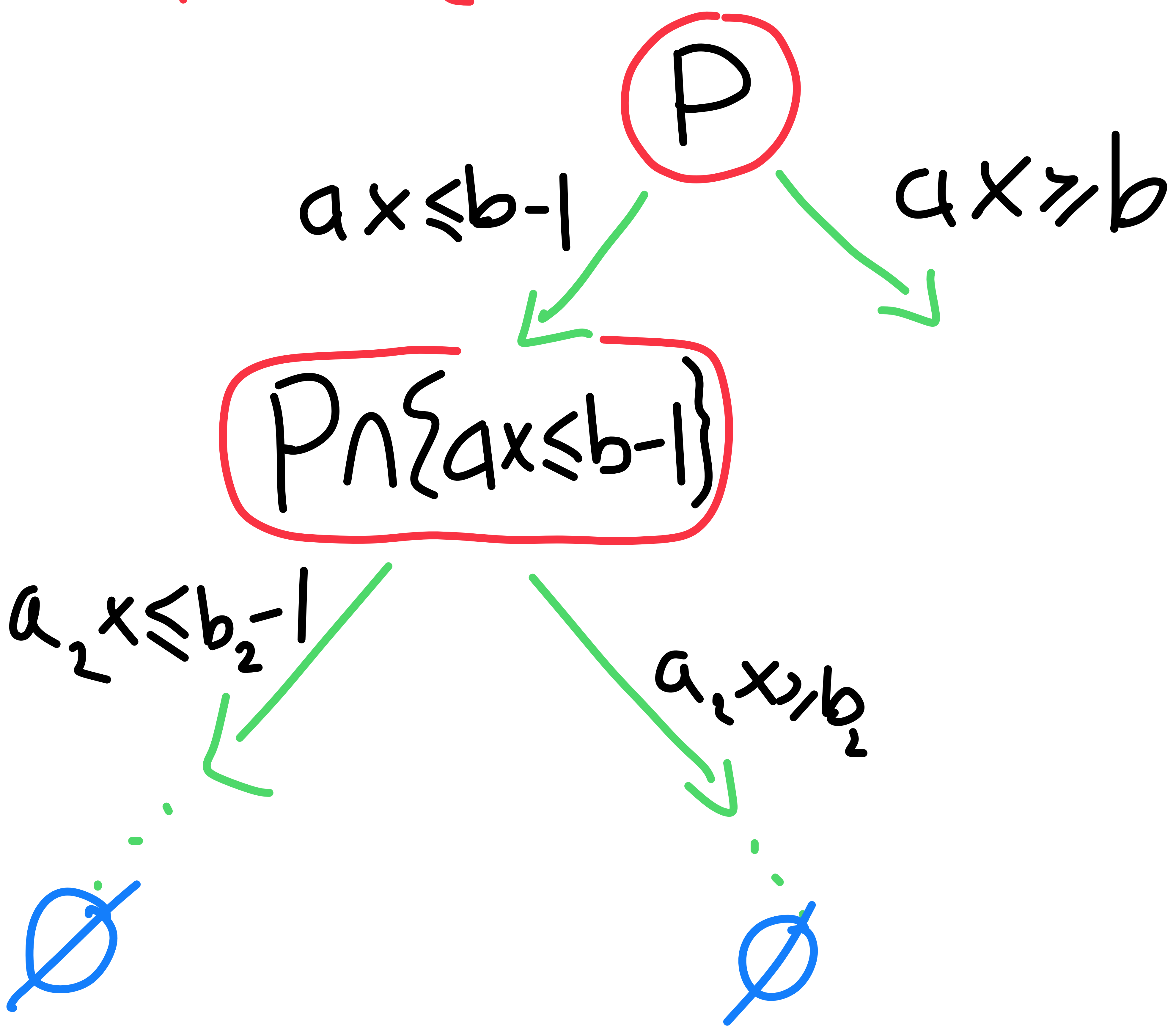
$P = \{Ax \geq b\}$ s.t. $P \cap \mathbb{Z}^n = \emptyset$.

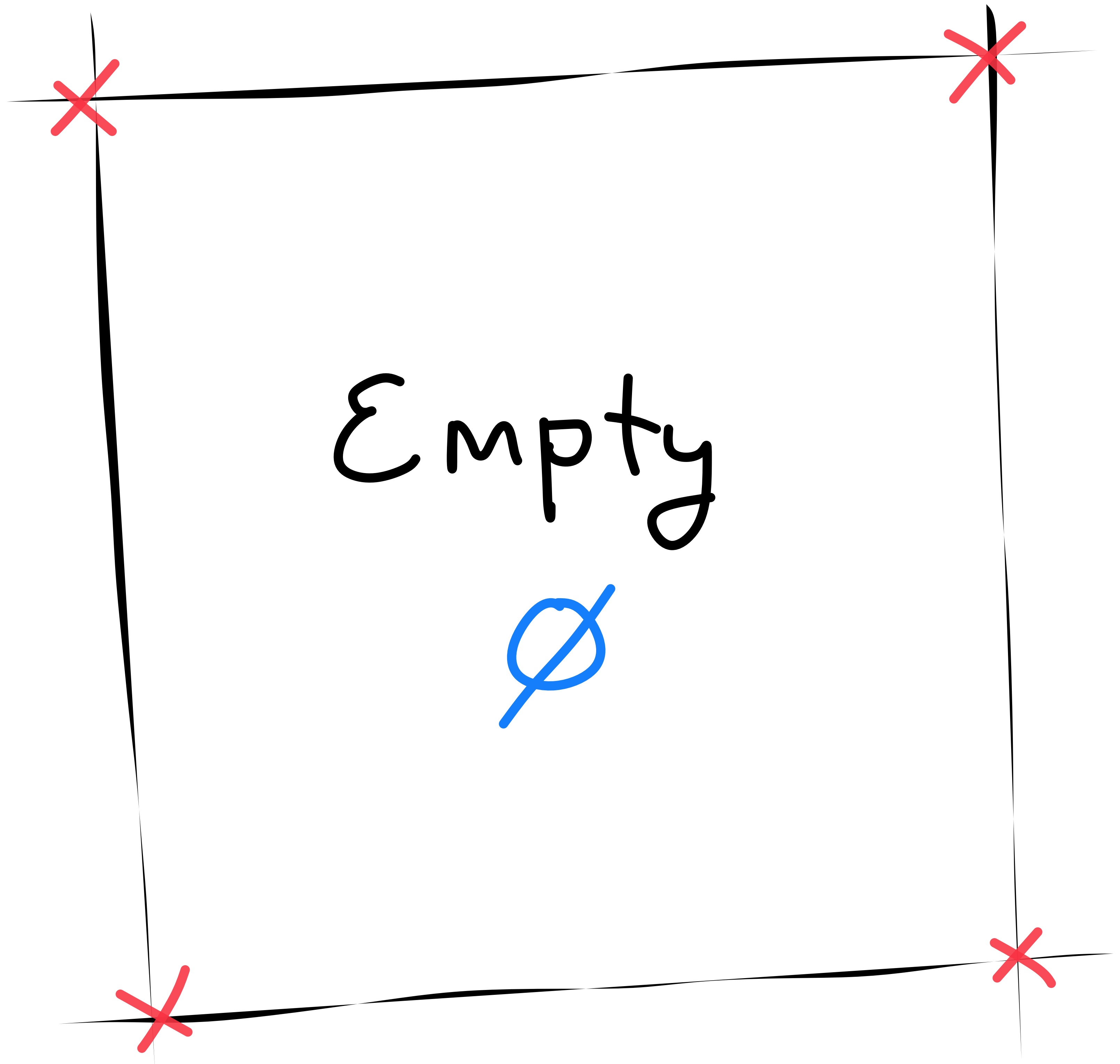




Stabbing Planes

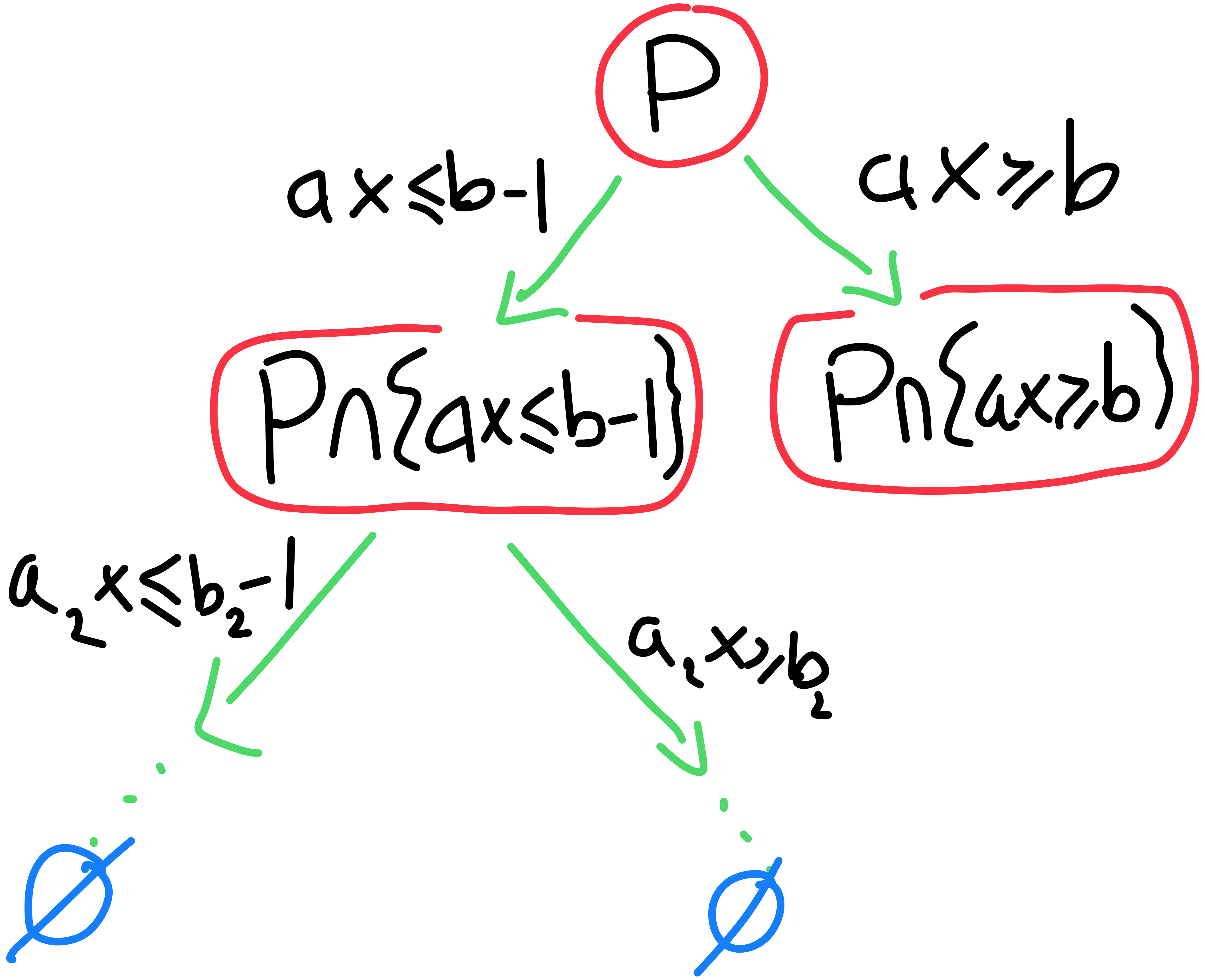
$P = \{Ax \geq b\}$ s.t. $P \cap \mathbb{Z}^n = \emptyset$.

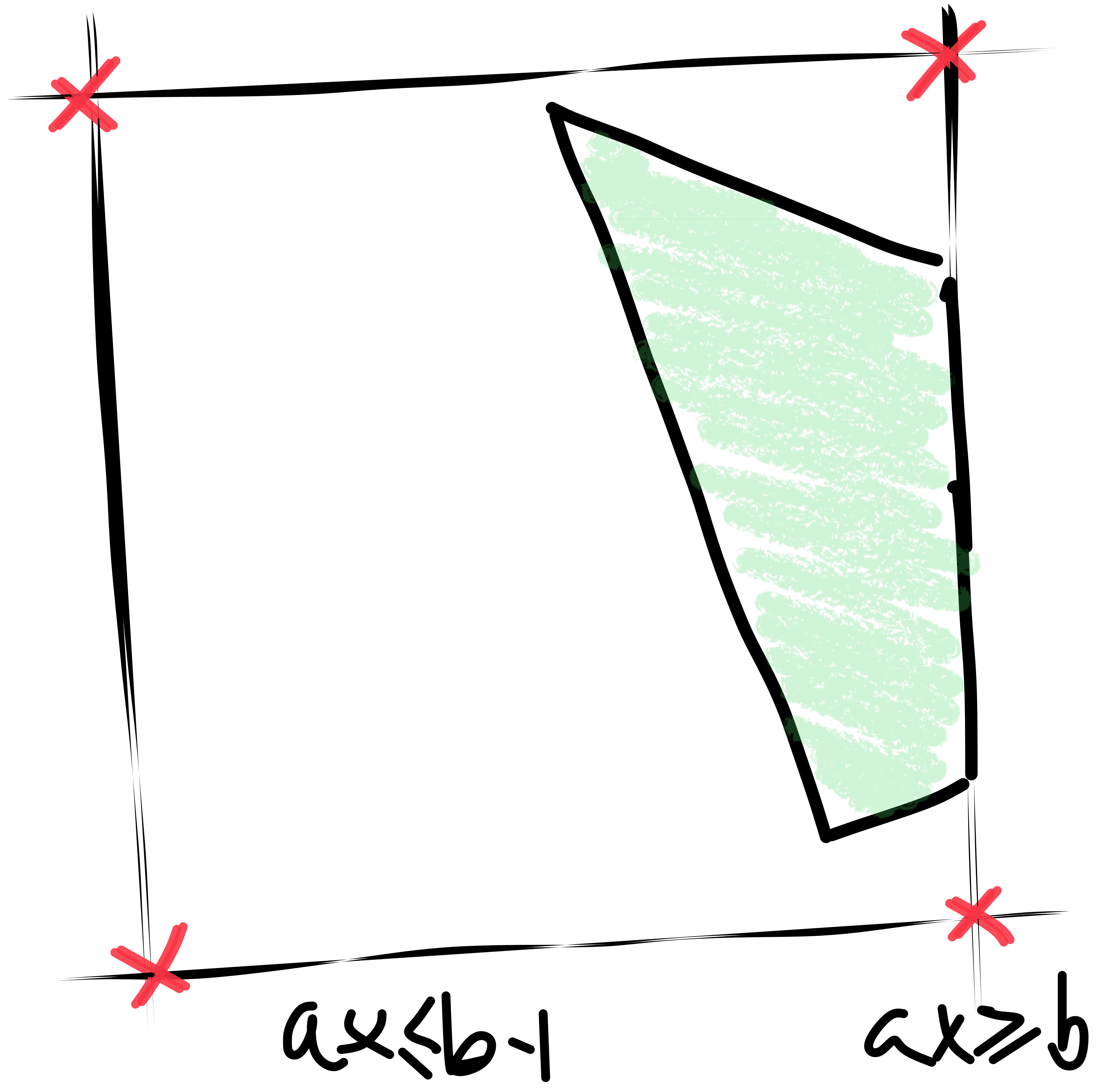




Stabbing Planes

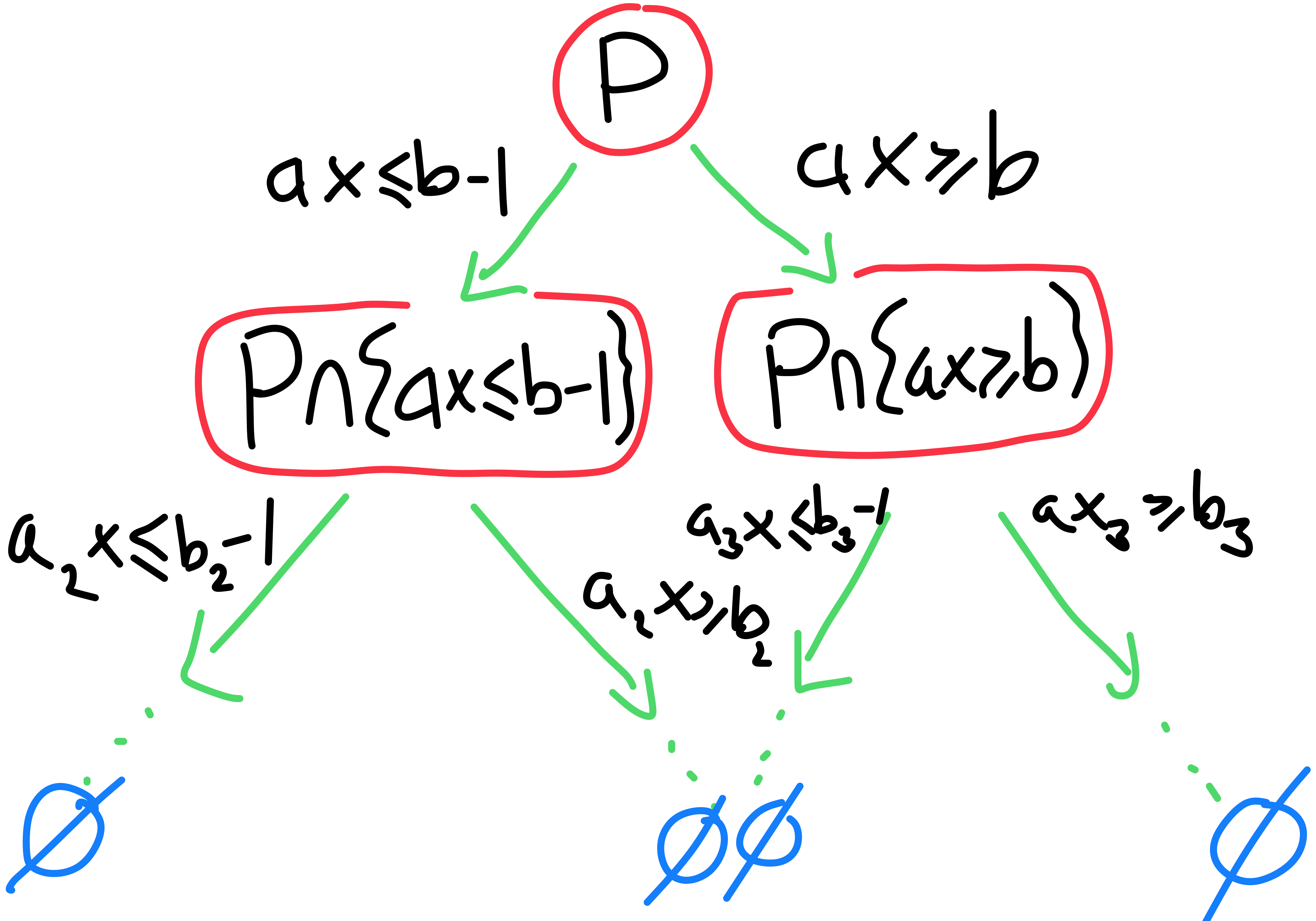
$P = \{Ax \geq b\}$ s.t. $P \cap \mathbb{Z}^n = \emptyset$.





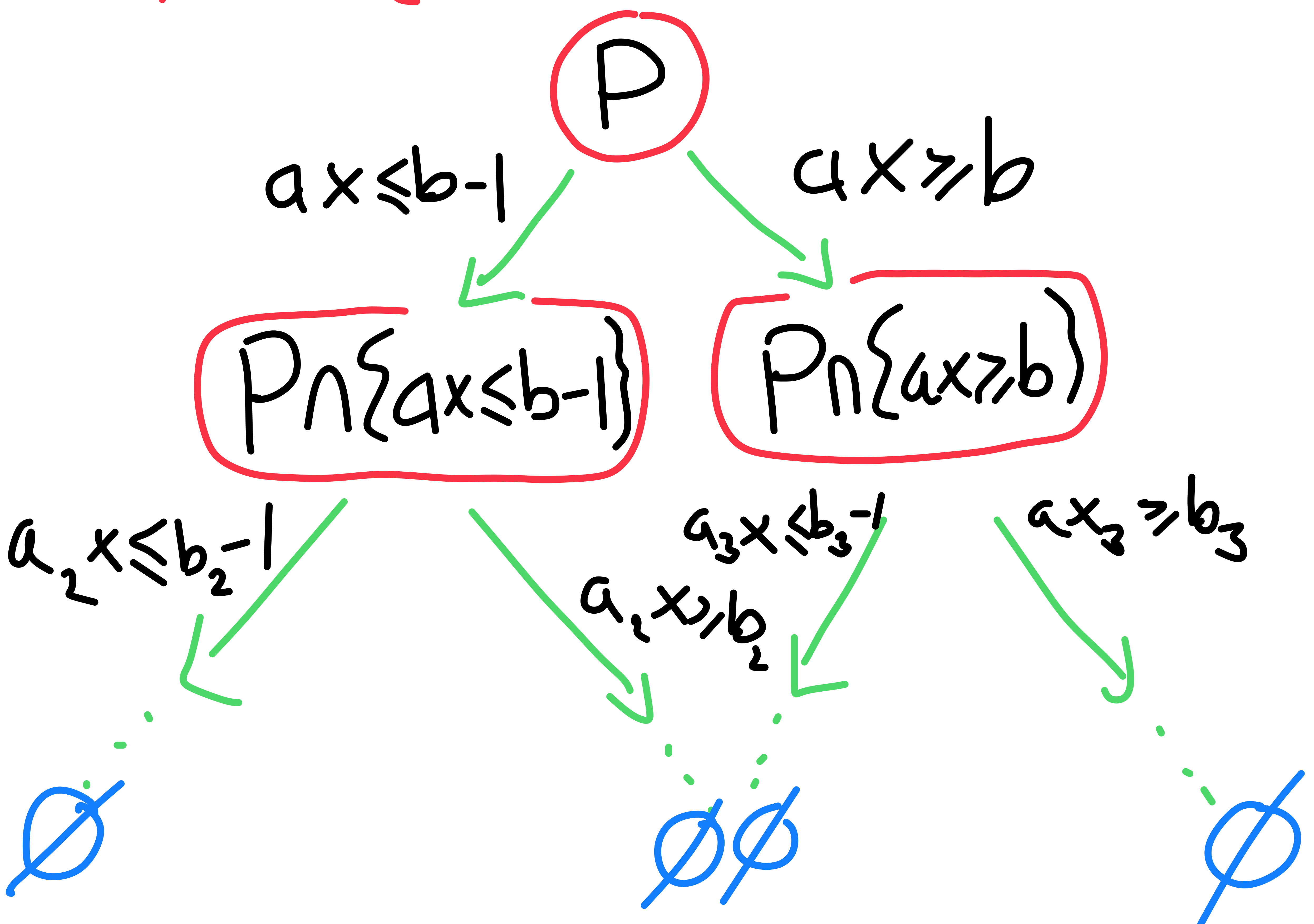
Stabbing Planes

$P = \{Ax \geq b\}$ s.t. $P \cap \mathbb{Z}^n = \emptyset$.



Stabbing Planes

$P = \{Ax \geq b\}$ s.t. $P \cap \mathbb{Z}^n = \emptyset$.

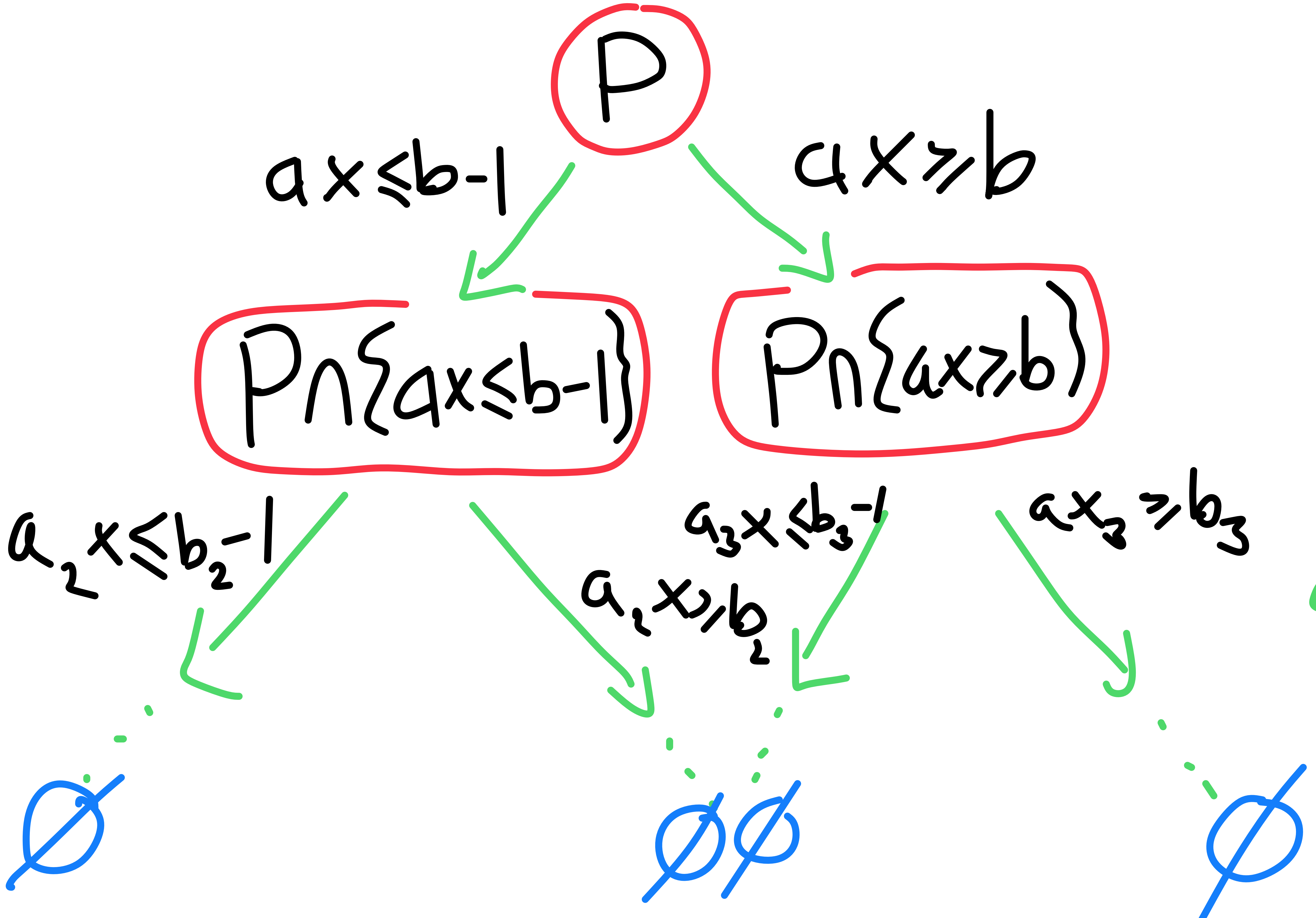


Refutation

Empty polytope $\emptyset$ deduced at every leaf proves $P \cap \mathbb{Z}^n = \emptyset$

Stabbing Planes

$P = \{Ax \geq b\}$ s.t. $P \cap \mathbb{Z}^n = \emptyset$.



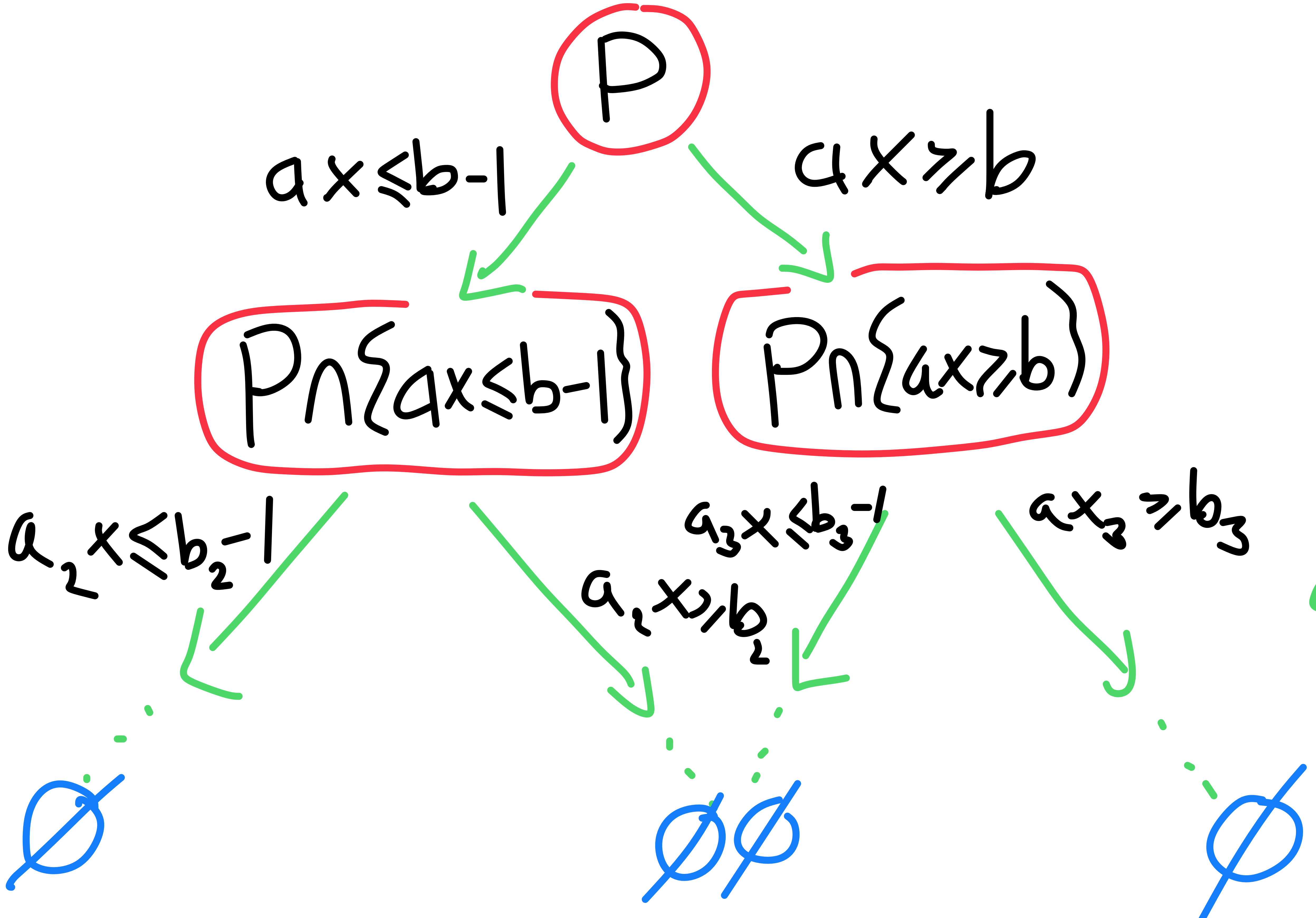
Refutation

Empty polytope $\emptyset$ deduced at every leaf proves $P \cap \mathbb{Z}^n = \emptyset$

Size: # of queries

Stabbing Planes

$P = \{Ax \geq b\}$ s.t. $P \cap \mathbb{Z}^n = \emptyset$.



Refutation

Empty polytope $\emptyset$ deduced at every leaf proves $P \cap \mathbb{Z}^n = \emptyset$

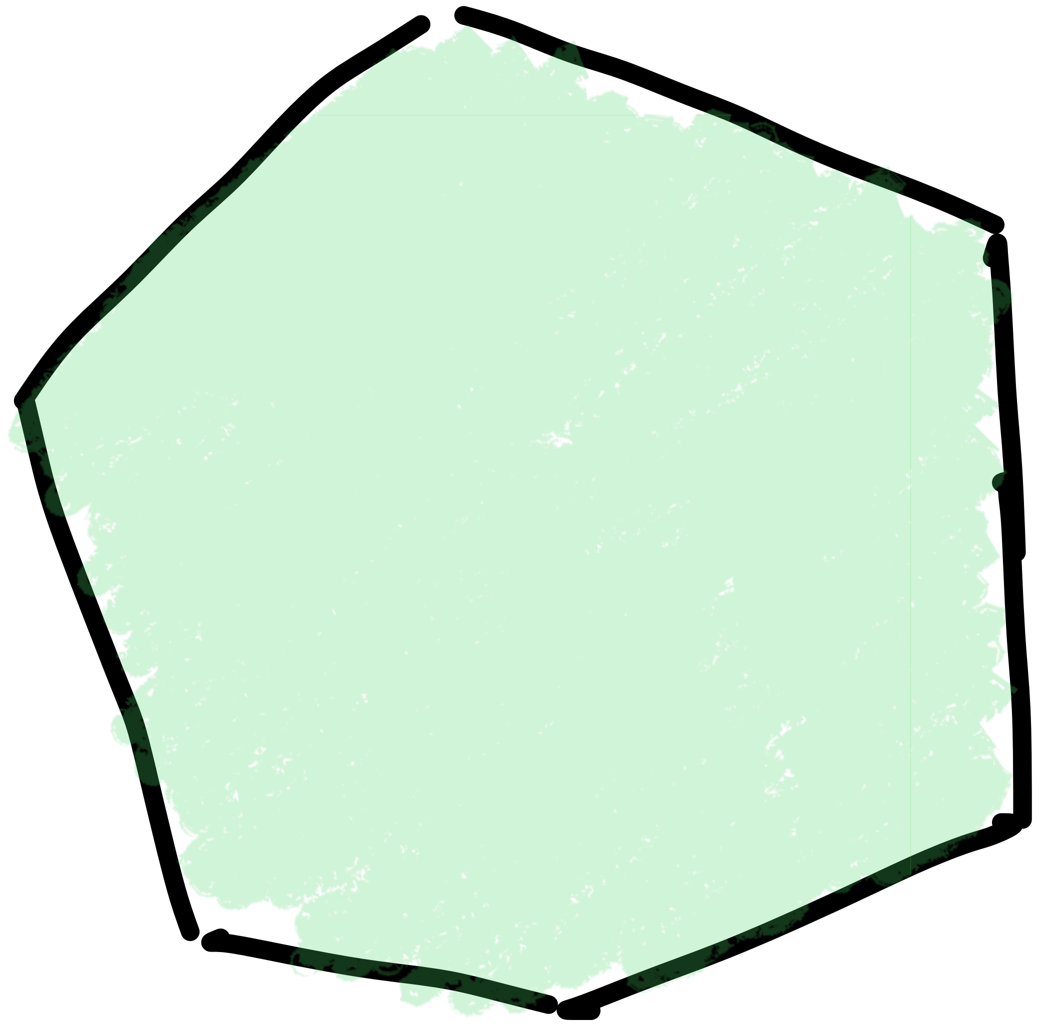
Size: # of queries $\approx$ bit-size [DT20]

Stabbing Planes vs Cutting Planes

- Branch-and-cut allows Cutting planes deductions

Stabbing Planes vs Cutting Planes

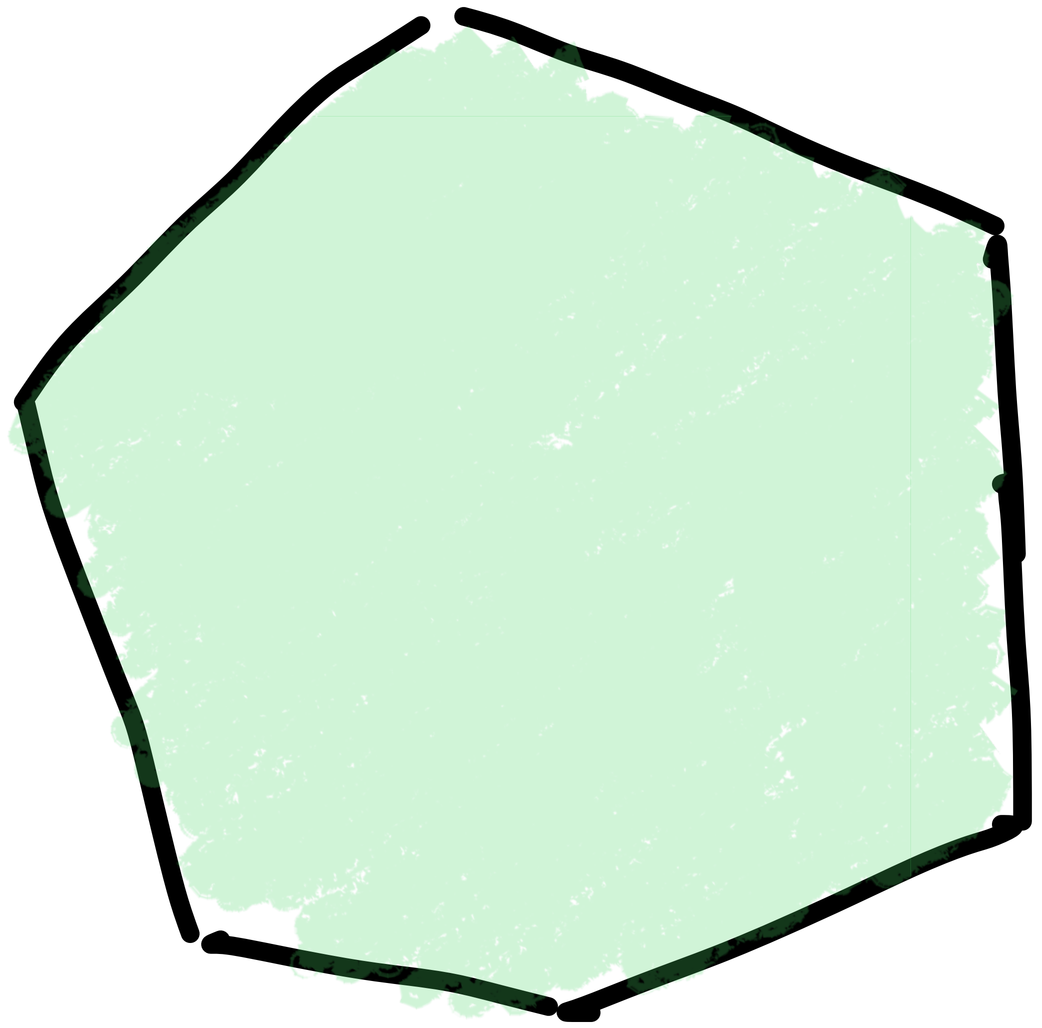
- Branch-and-cut allows Cutting planes deductions
  - Superfluous as SP can simulate CP



Stabbing Planes vs Cutting Planes

- Branch-and-cut allows cutting planes deductions
  - → Superfluous as SP can simulate CP

Clm: pathlike SP = CP

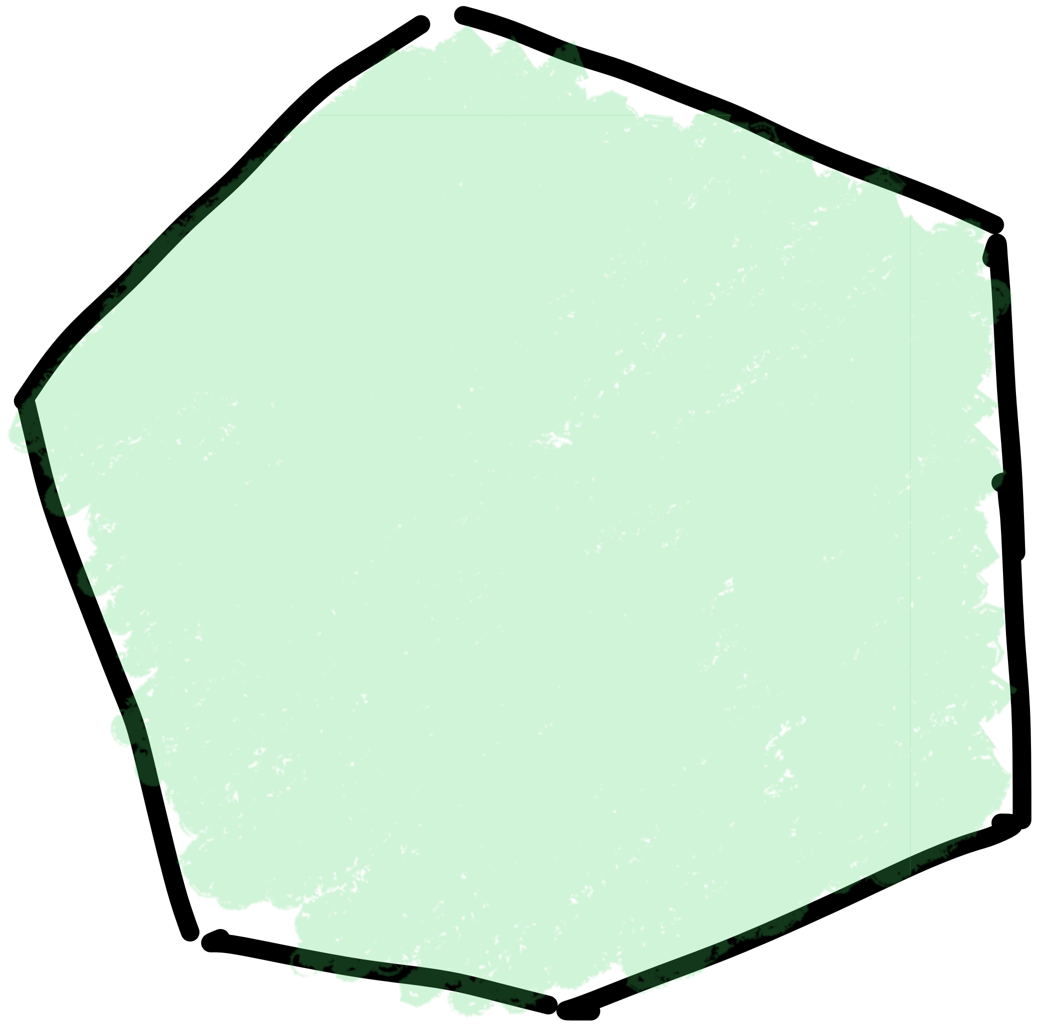


Stabbing Planes vs Cutting Planes

- Branch-and-cut allows Cutting planes deductions
  - → Superfluous as SP can simulate CP

Pathlike SP: A SP query $(ax \leq b-1, \; ax \geq b)$ at P is pathlike if $P \cap \{ax \leq b-1\} = \emptyset$ or $P \cap \{ax \geq b\} = \emptyset$

Clm: pathlike SP = CP

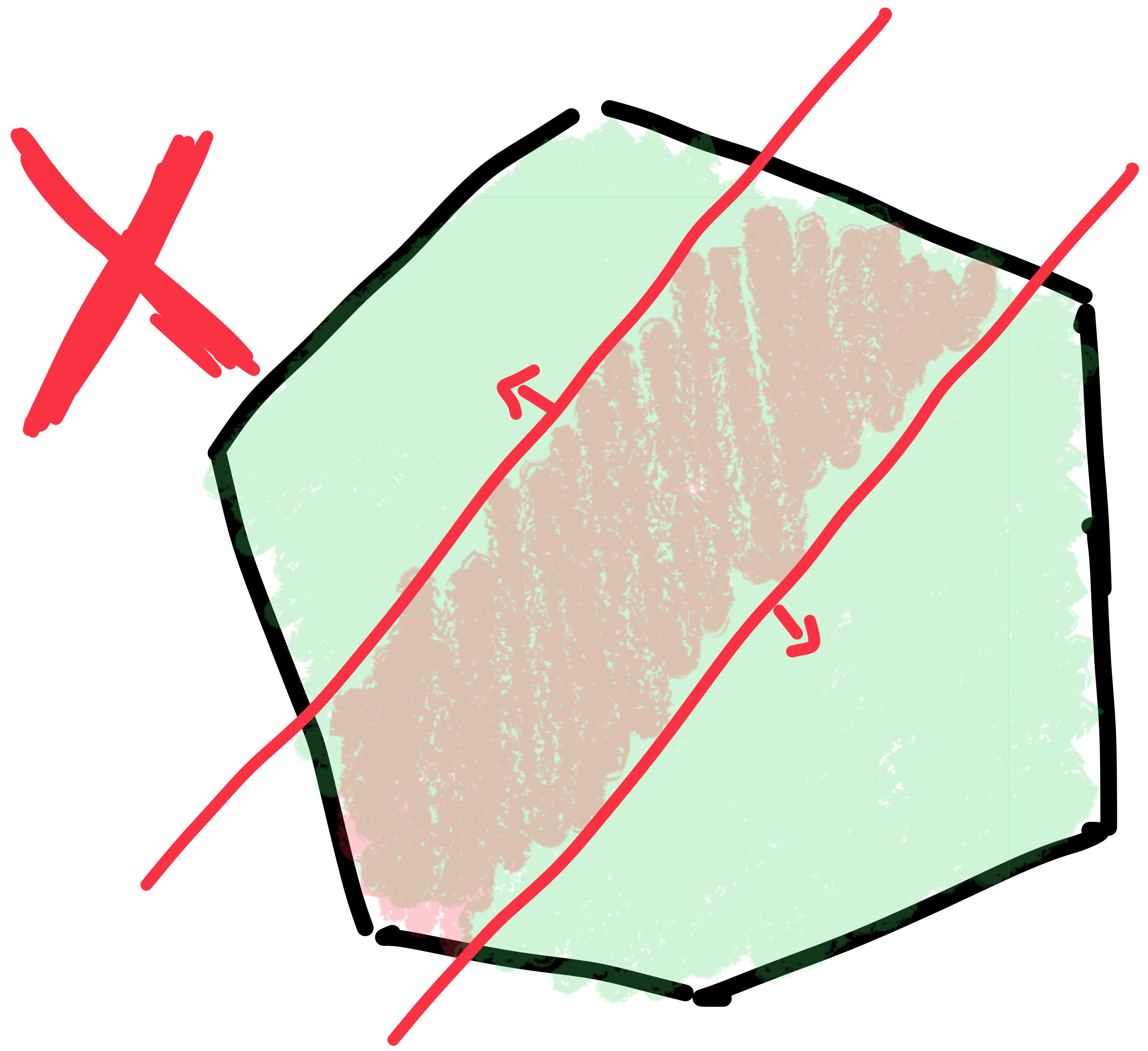


Stabbing Planes vs Cutting Planes

- Branch-and-cut allows Cutting planes deductions
  - → Superfluous as SP can simulate CP

Pathlike SP: A SP query $(ax \leq b-1, \; ax \geq b)$ at P is pathlike if $P \cap \{ax \leq b-1\} = \emptyset$ or $P \cap \{ax \geq b\} = \emptyset$

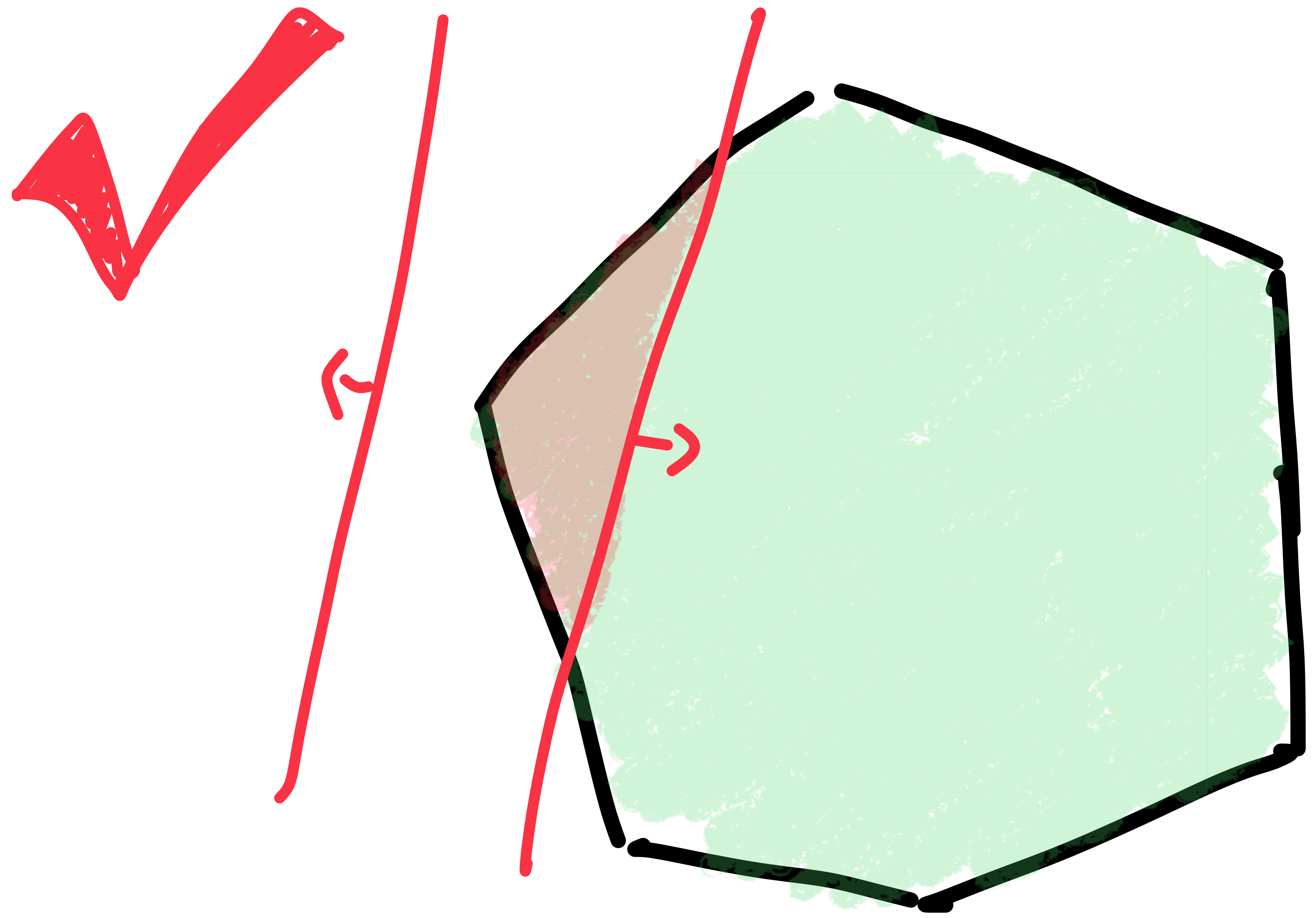
Clm: pathlike SP = CP



Stabbing Planes vs Cutting Planes

- Branch-and-cut allows Cutting planes deductions
  - → Superfluous as SP can simulate CP

Pathlike SP: A SP query $(ax \leq b-1,\ ax \geq b)$ at P is pathlike if $P \cap \{ax \leq b-1\} = \emptyset$ or $P \cap \{ax \geq b\} = \emptyset$



Clm: pathlike SP = CP

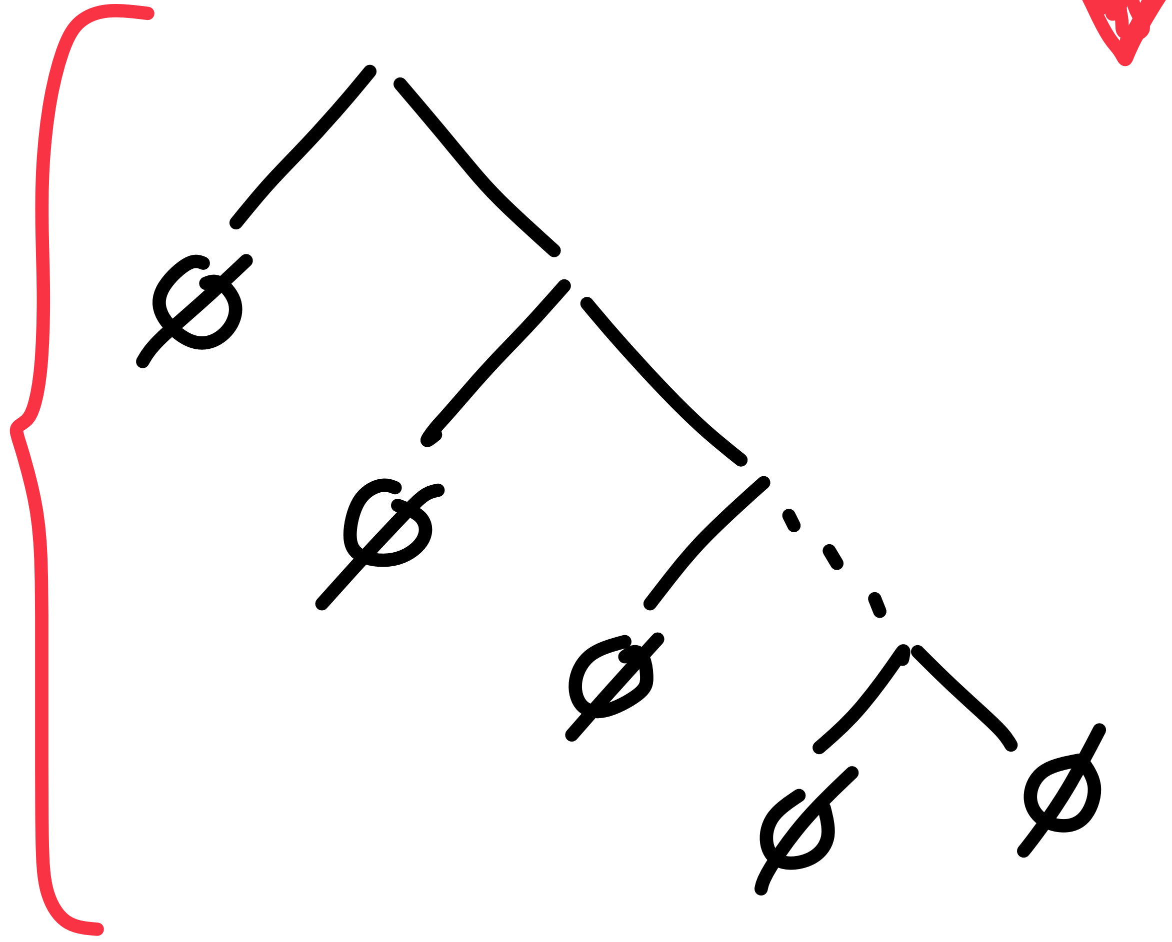
Stabbing Planes vs Cutting Planes

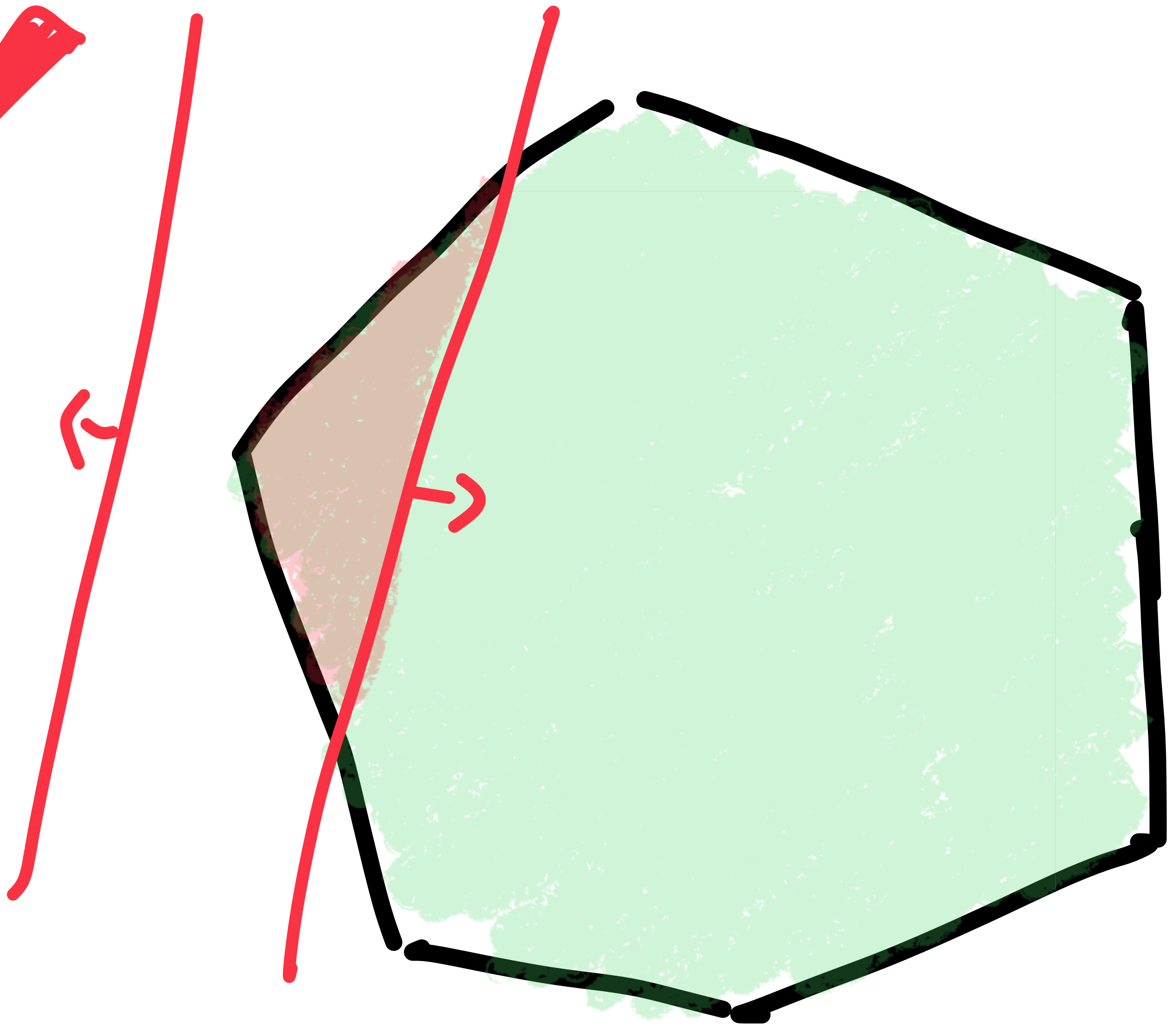
- Branch-and-cut allows Cutting planes deductions
  - → Superfluous as SP can simulate CP

Pathlike SP: A SP query $(ax \le b-1,\ ax \ge b)$ at P is pathlike if $P \cap \{ax \le b-1\} = \emptyset$ or $P \cap \{ax \ge b\} = \emptyset$

Clm: pathlike SP = CP

Pathlike SP Proof





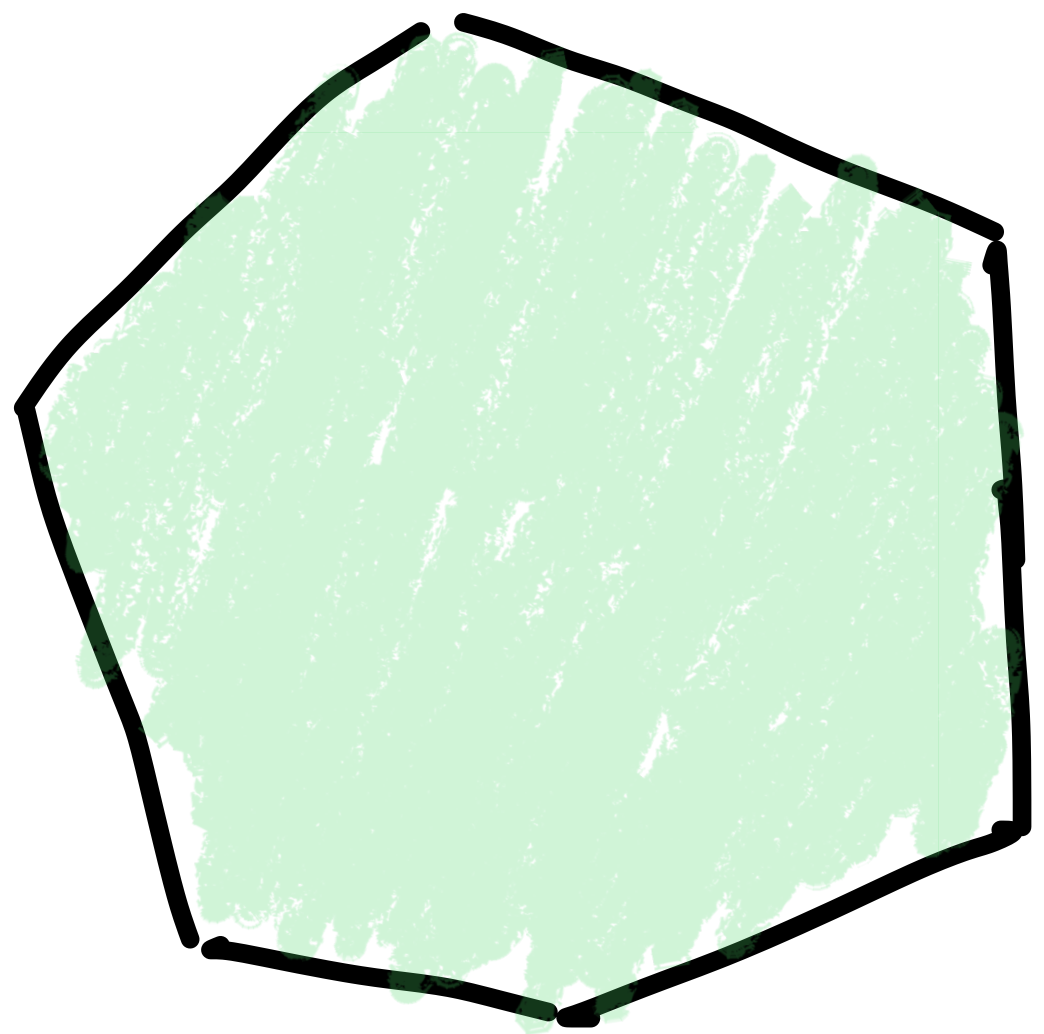
Stabbing Planes vs Cutting Planes

- Branch-and-cut allows Cutting planes deductions
  - $\to$ Superfluous as SP can simulate CP

Pathlike SP: A SP query $(ax \le b-1,\ ax \ge b)$ at P is pathlike if $P \cap \{ax \le b-1\} = \emptyset$ or $P \cap \{ax \ge b\} = \emptyset$

Clm: pathlike SP = CP

Pf: Show each CG-cut is a pathlike query



Stabbing Planes vs Cutting Planes

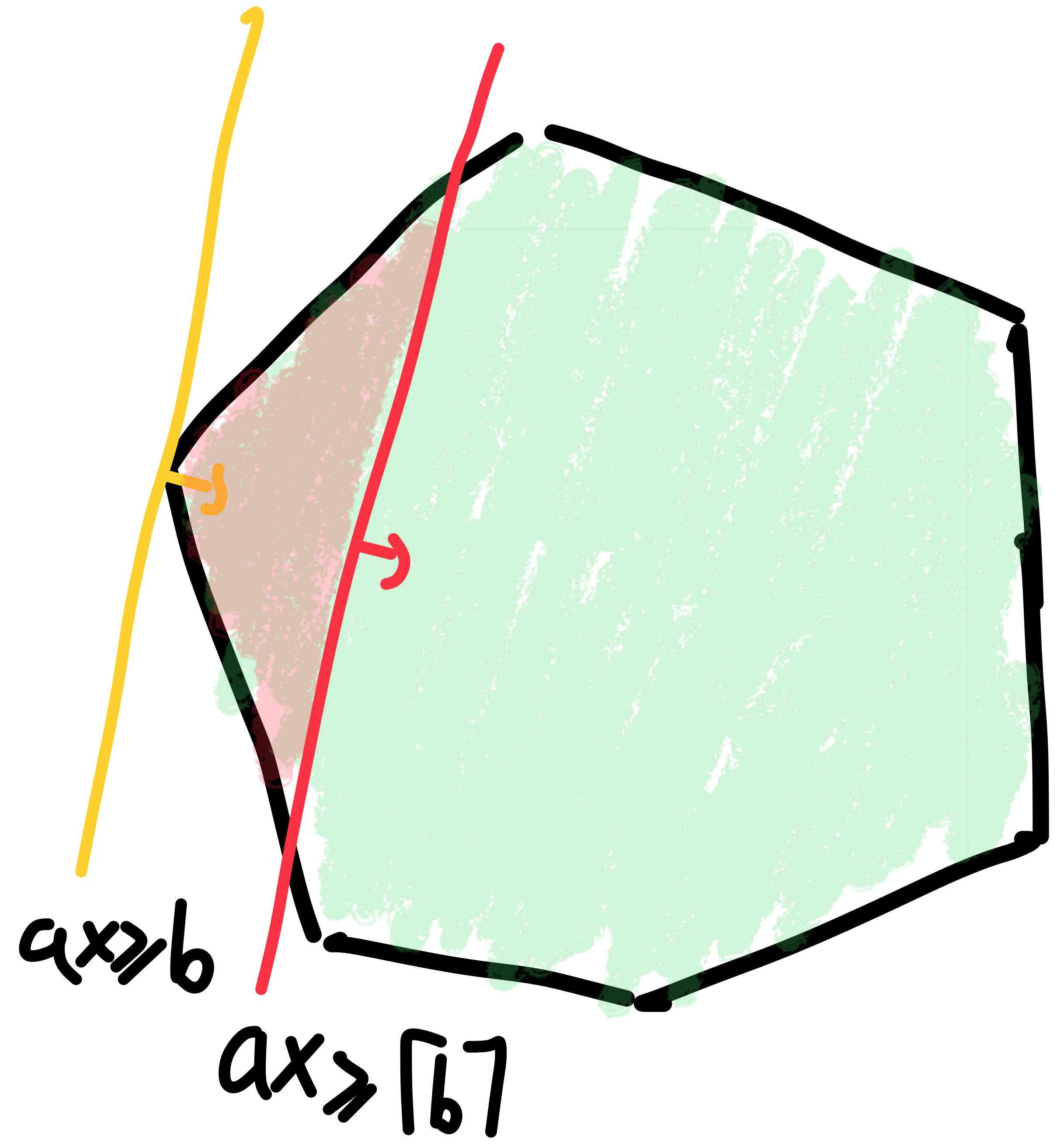
- ▷ Branch-and-cut allows Cutting planes deductions
  - → Superfluous as SP can simulate CP

Pathlike SP: A SP query $(ax \leq b-1,\ ax \geq b)$ at P is pathlike if $P \cap \{ax \leq b-1\} = \emptyset$ or $P \cap \{ax \geq b\} = \emptyset$

Clm: pathlike SP = CP

Pf: Show each CG-cut is a pathlike query

$ax \geq \lceil b \rceil$ is a CG-cut for P if $ax \geq b$ is valid for P



Stabbing Planes vs Cutting Planes

▷ Branch-and-cut allows Cutting planes deductions

→▷ Superfluous as SP can simulate CP

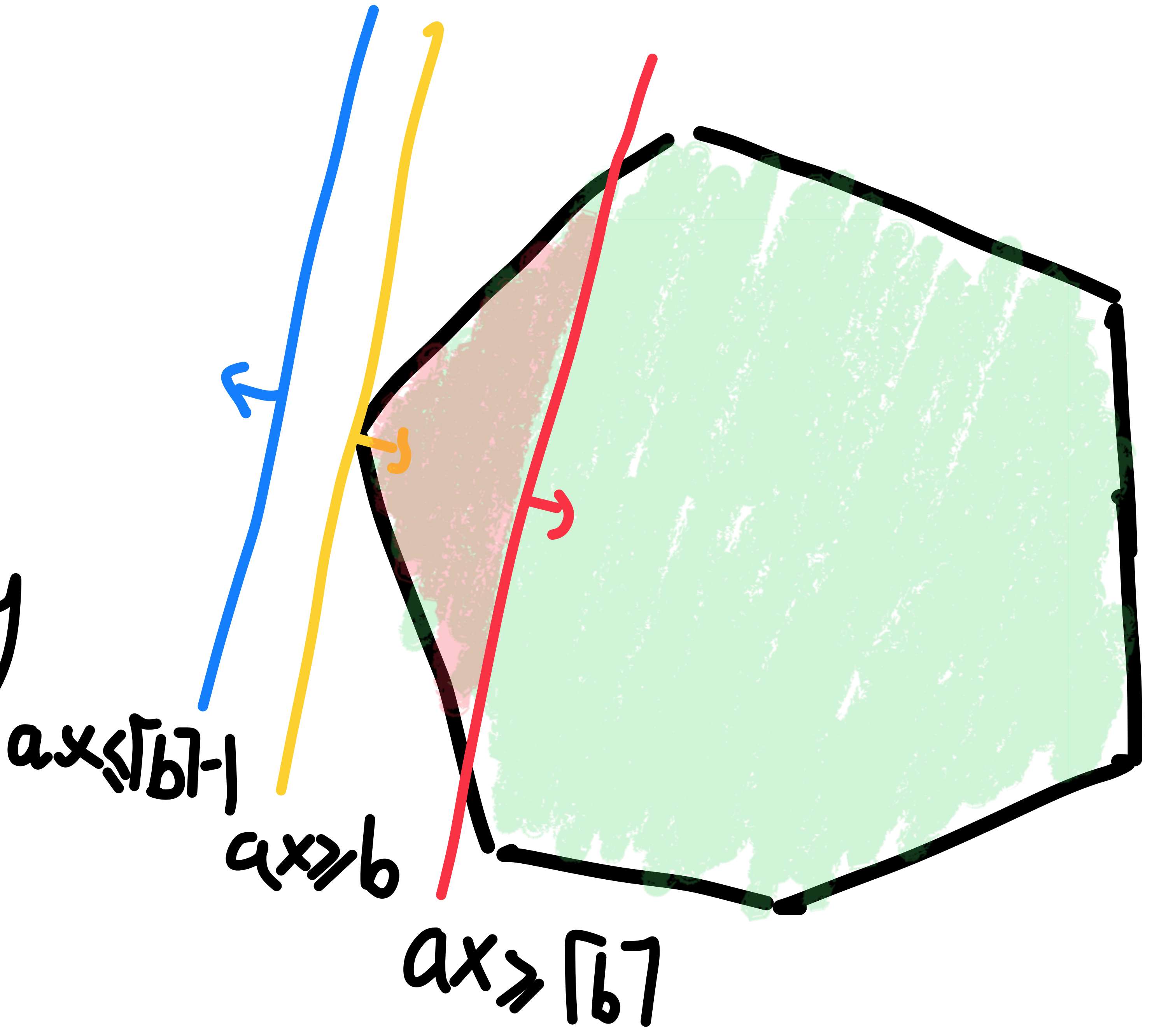
Pathlike SP: A SP query $(ax \leq b-1,\ ax \geq b)$ at P is pathlike if $P \cap \{ax \leq b-1\} = \emptyset$ or $P \cap \{ax \geq b\} = \emptyset$

Clm: pathlike SP = CP

Pf: Show each CG-cut is a pathlike query

$ax \geq \lceil b \rceil$ is a CG-cut for P if $ax \geq b$ is valid for P

$\Rightarrow P \cap \{ax \leq \lceil b \rceil - 1\} = \emptyset$



Stabbing Planes vs Cutting Planes

- Branch-and-cut allows Cutting planes deductions
  - → Superfluous as SP can simulate CP

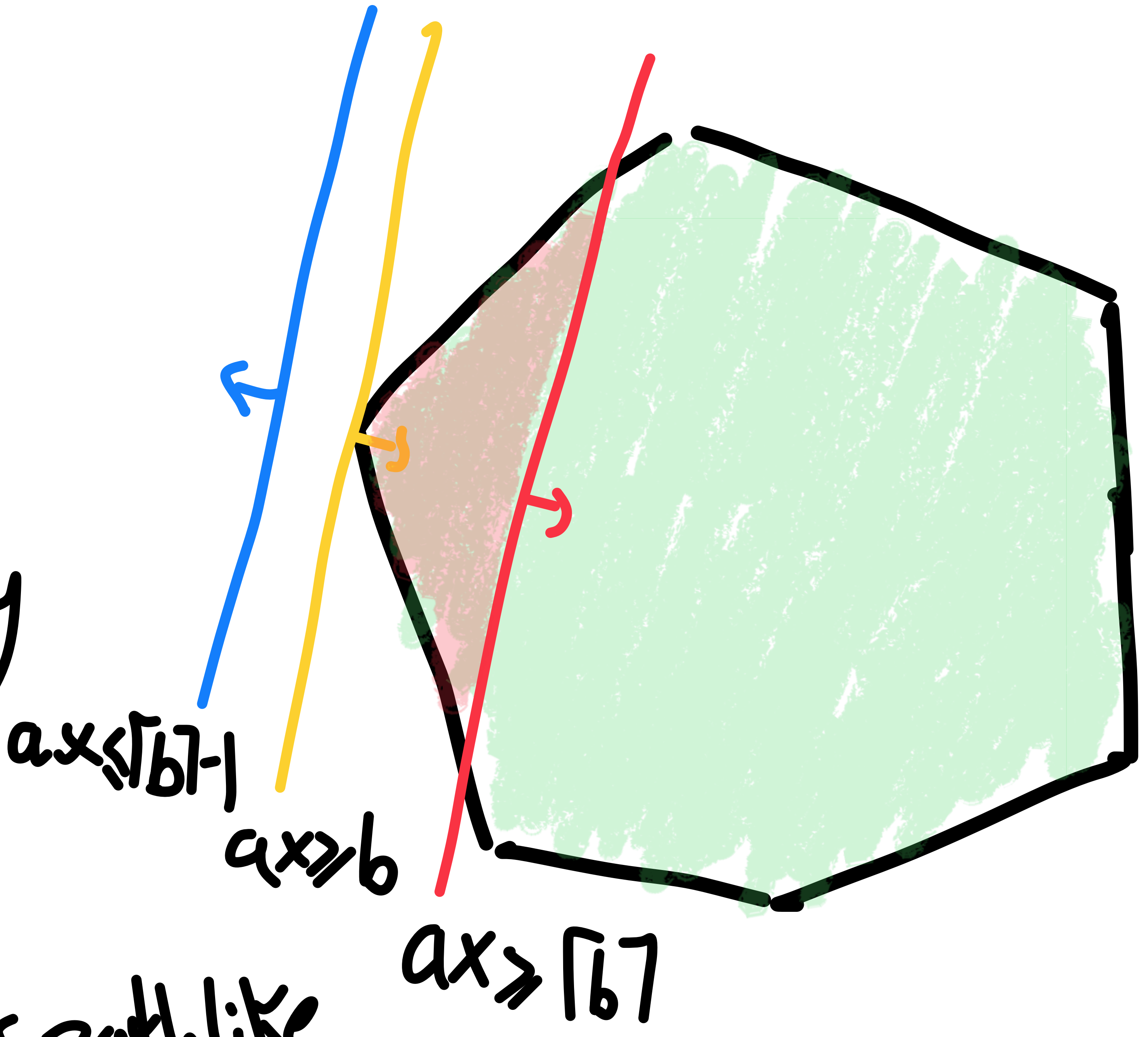
Pathlike SP: A SP query $(ax \leq b-1,\ ax \geq b)$ at P is pathlike if $P \cap \{ax \leq b-1\} = \emptyset$ or $P \cap \{ax \geq b\} = \emptyset$

Clm: pathlike SP = CP

Pf: Show each CG-cut is a pathlike query

$ax \geq \lceil b \rceil$ is a CG-cut for P if $ax \geq b$ is valid for P

$\Rightarrow P \cap \{ax \leq \lceil b \rceil - 1\} = \emptyset$ & $(ax \leq \lceil b \rceil - 1,\ ax \geq \lceil b \rceil)$ is pathlike



Stabbing Planes vs Cutting Planes

Q Can we separate CP from SP?

Stabbing Planes vs Cutting Planes

Q Can we separate CP from SP?

- Candidate: Tseitin formulas

Stabbing Planes vs Cutting Planes

Q Can we separate CP from SP?

- Candidate: Tseitin formulas
  - → quasipoly-size SP proofs of Tseitin [BFI+18]

Stabbing Planes vs Cutting Planes

Q Can we separate CP from SP?

- Candidate: Tseitin formulas
  - quasipoly-size SP proofs of Tseitin [BFI+18]
  - Conjectured to be hard for CP

Stabbing Planes vs Cutting Planes

Q Can we separate CP from SP?

- Candidate: Tseitin formulas
  - → quasipoly-size SP proofs of Tseitin [BFI+18]
  - → Conjectured to be hard for CP

[DT20] There are quasi-poly size CP proofs of Tseitin

Stabbing Planes vs Cutting Planes

Q Can we separate CP from SP?

- ▷ Candidate: Tseitin formulas
  - ⟶▷ quasipoly-size SP proofs of Tseitin [BFI+18]
  - ⟶▷ Conjectured to be hard for CP

[DT20] There are quasi-poly size CP proofs of Tseitin

- ▷ Translate the SP proof of Tseitin into CP

Stabbing Planes vs Cutting Planes

Q Can we separate CP from SP?

- Candidate: Tseitin formulas
  - → quasipoly-size SP proofs of Tseitin [BFI+18]
  - → Conjectured to be hard for CP

[DT20] There are quasi-poly size CP proofs of Tseitin

- Translate the SP proof of Tseitin into CP

Q Can every SP proof be translated into CP?

Stabbing Planes vs Cutting Planes

Thm: Every SP* proof can be quasipolynomially translated into CP

Stabbing Planes vs Cutting Planes

Thm: Every SP^* proof can be quasipolynomially translated into CP

Coefficients in queries are quasi-poly bounded

Stabbing Planes vs Cutting Planes

Thm: Every SP^* proof can be quasipolynomially translated into CP

→ SP^* is a "query" proof system (like DPLL) for CP

Stabbing Planes vs Cutting Planes

Thm: Every SP* proof can be quasipolynomially translated into CP

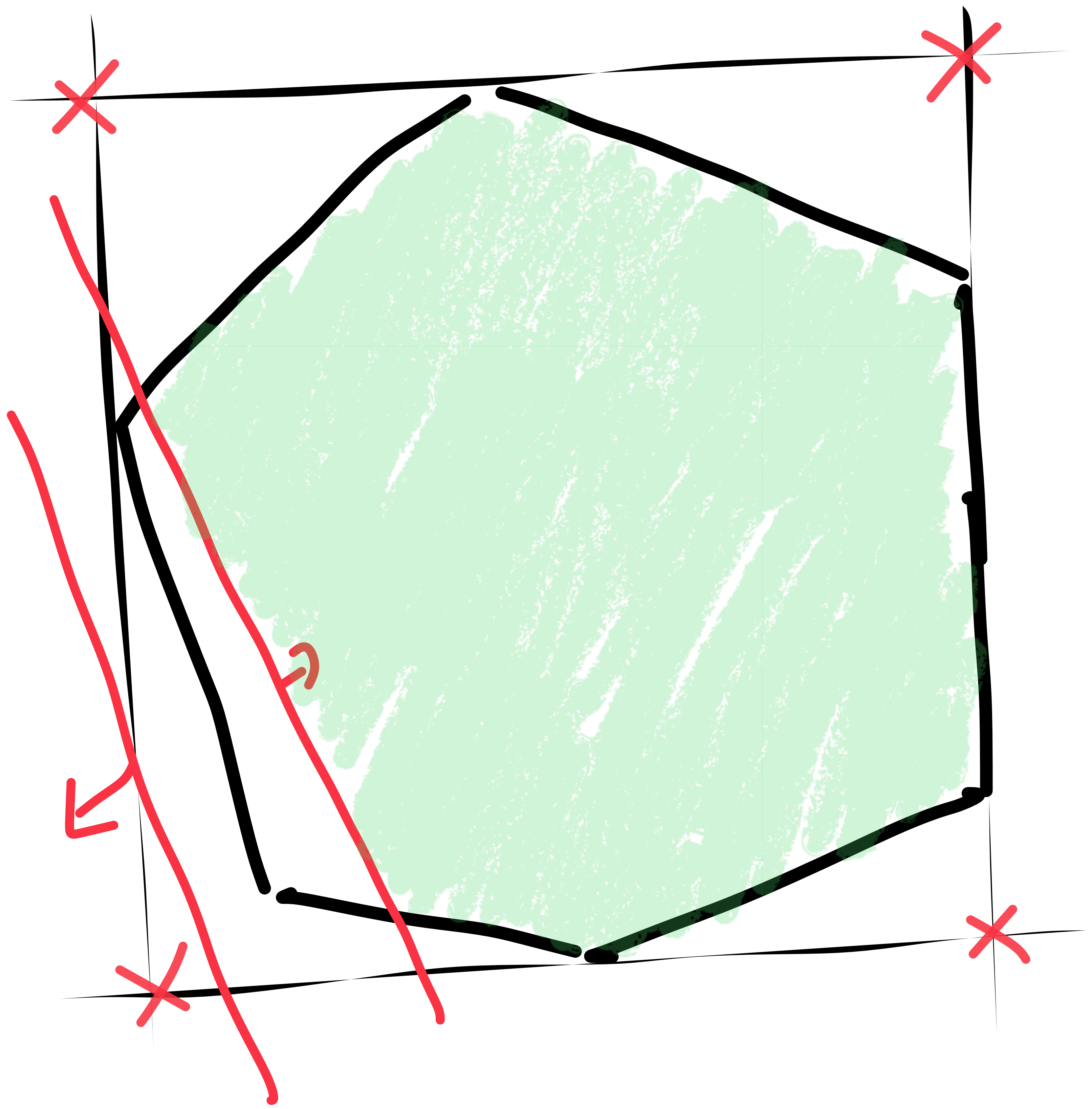
Pf: a) CP = pathlike SP = facelike SP

Stabbing Planes vs Cutting Planes

Thm: Every SP* proof can be quasipolynomially translated into CP

Pf: a) CP = pathlike SP = facelike SP

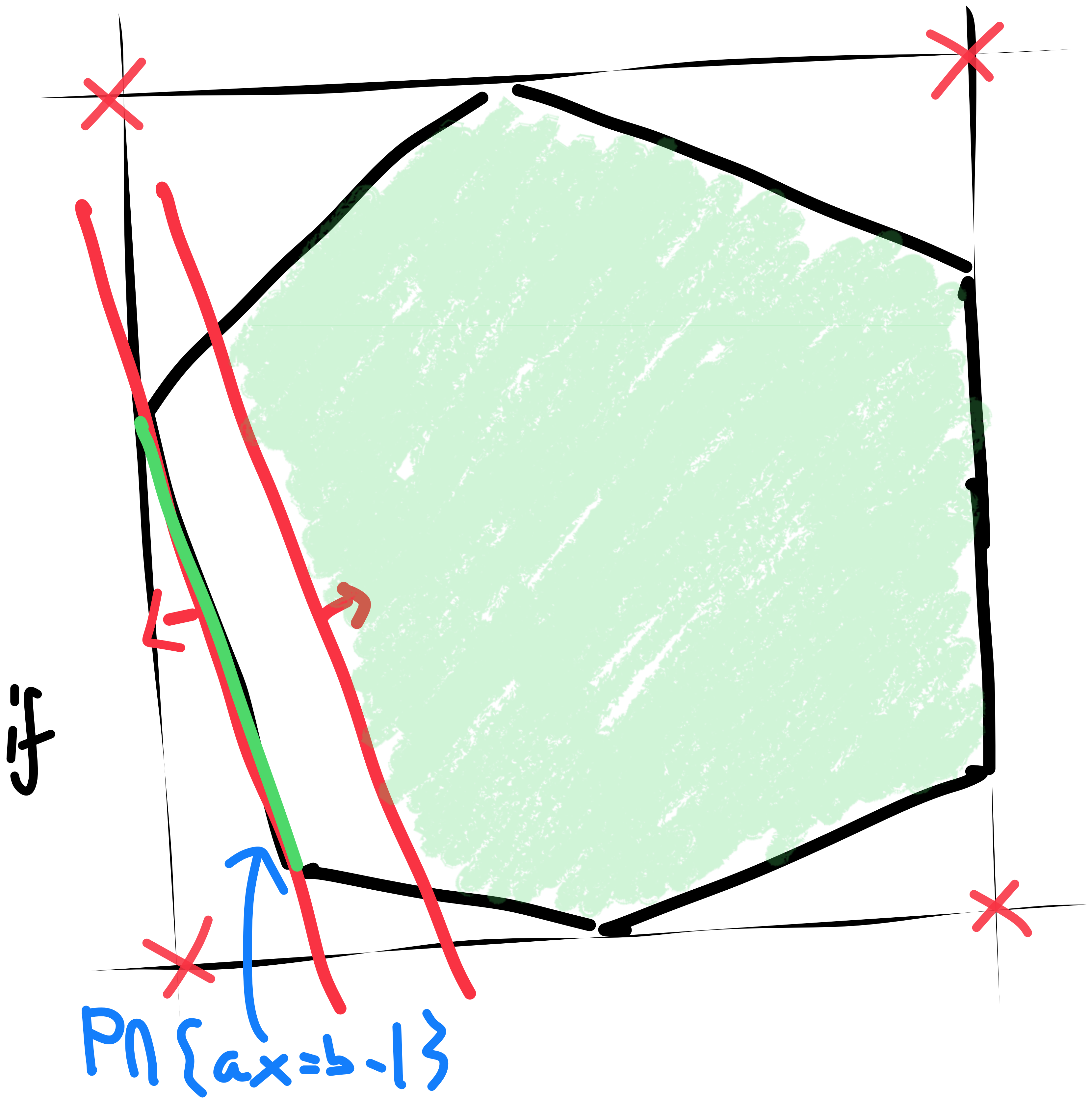
Pathlike SP: A query $(ax \leq b-1, ax \geq b)$ is pathlike if $P \cap \{ax \leq b-1\} = \emptyset$ or $P \cap \{ax \geq b\} = \emptyset$



Stubbing Planes vs Cutting Planes

Thm: Every SP* proof can be quasipolynomially translated into CP

Pf: a) CP = pathlike SP = facelike SP



Pathlike SP: A query $(ax \leq b-1, ax \geq b)$ is pathlike if
$P \cap \{ax \leq b-1\} = \emptyset$ or $P \cap \{ax \geq b\} = \emptyset$

Facelike SP: A query $(ax \leq b-1, ax \geq b)$ is facelike if
$P \cap \{ax < b-1\} = \emptyset$ or $P \cap \{ax > b\} = \emptyset$

Stabbing Planes vs Cutting Planes

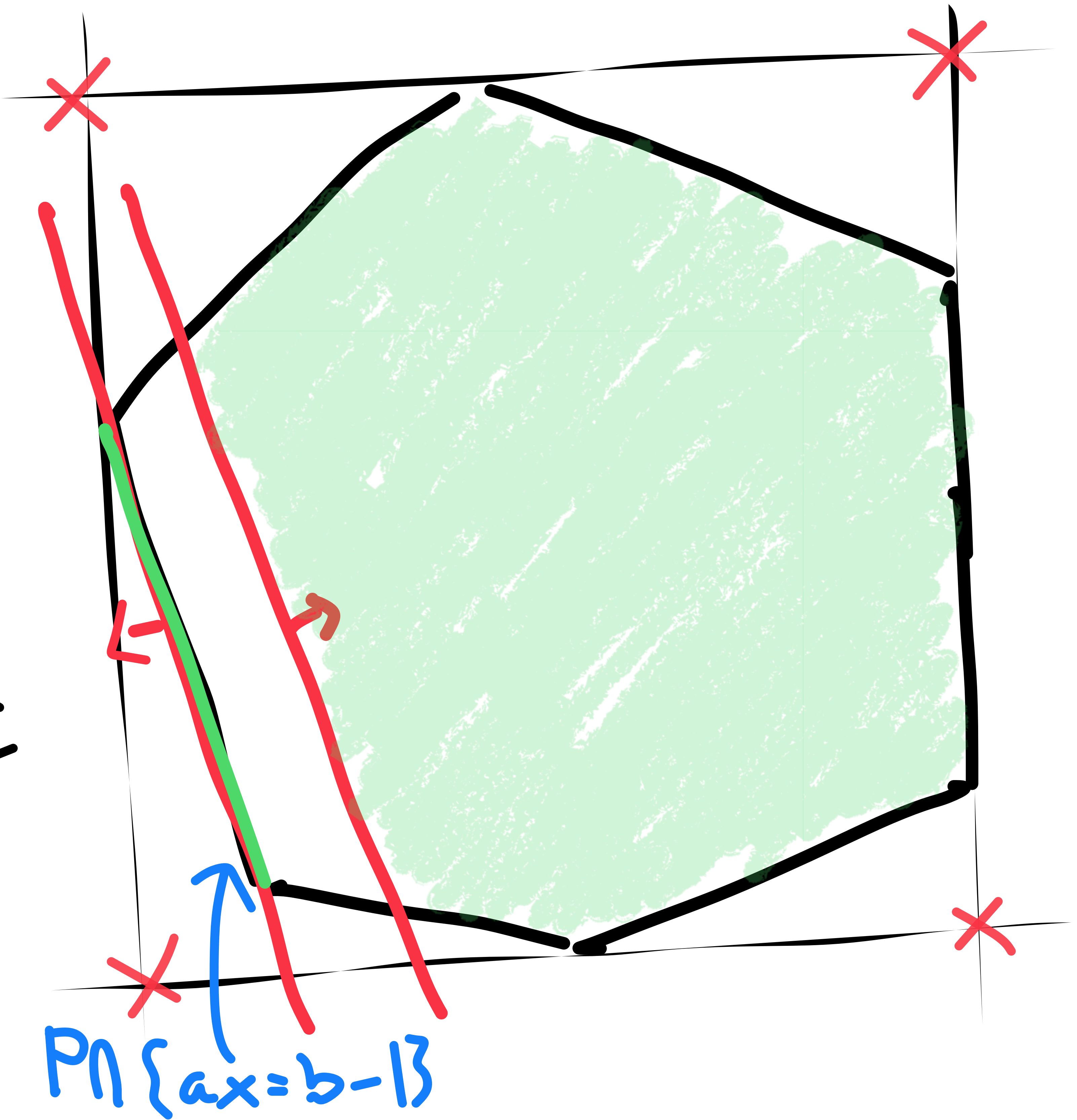
Thm: Every SP* proof can be quasipolynomially translated into CP

Pf: a) CP = pathlike SP = facelike SP

Pathlike SP: A query $(ax \leq b-1, ax \geq b)$ is pathlike if $P \cap \{ax \leq b-1\} = \emptyset$ or $P \cap \{ax \geq b\} = \emptyset$

Facelike SP: A query $(ax \leq b-1, ax \geq b)$ is facelike if $P \cap \{ax < b-1\} = \emptyset$ or $P \cap \{ax > b\} = \emptyset$

ie $P \cap \{ax \leq b-1\}$ or $P \cap \{ax \geq b\}$ is a face.



Stabbing Planes vs Cutting Planes

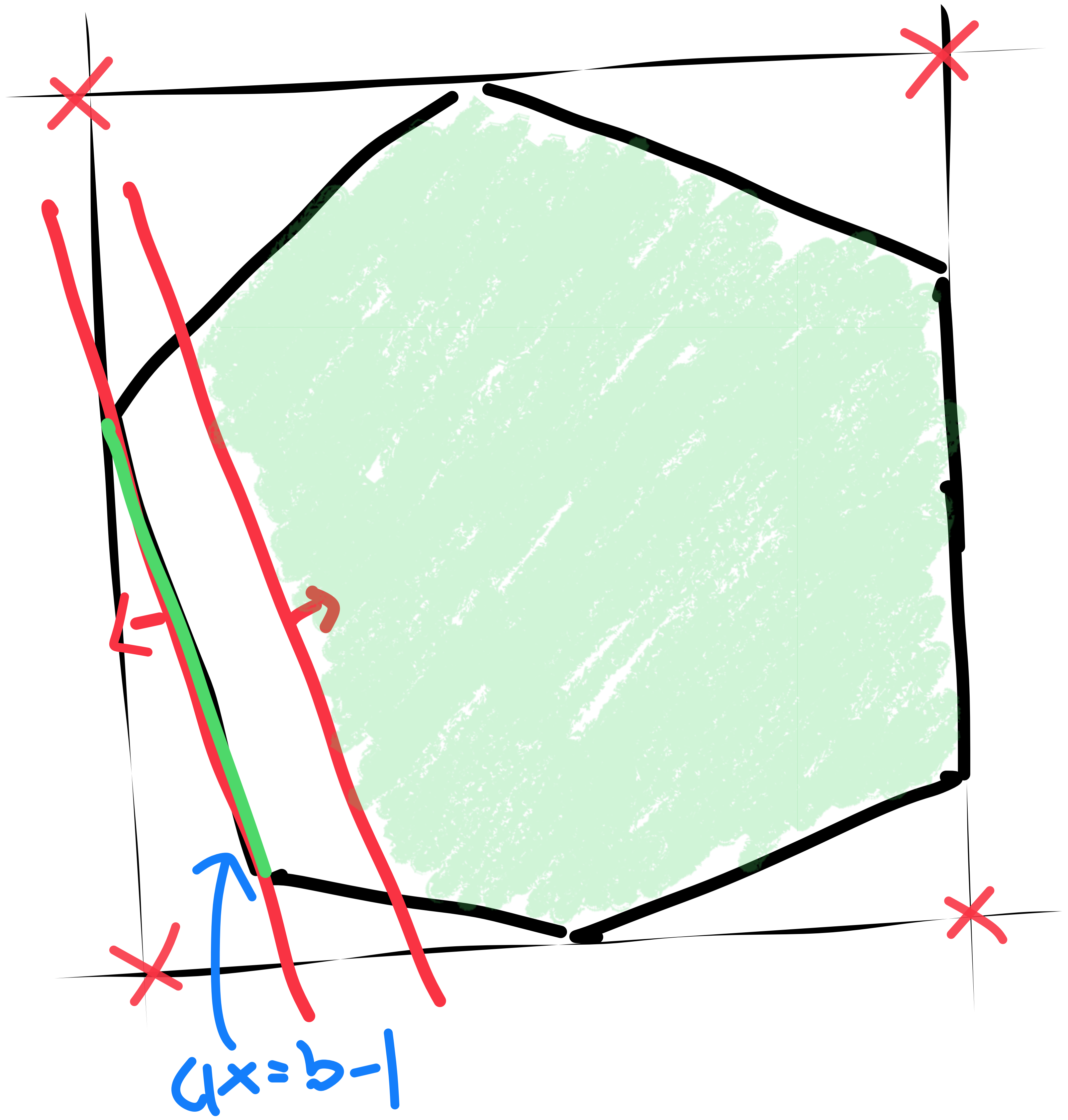
Thm: Every SP^* proof can be quasipolynomially translated into CP

Pf: a) CP = pathlike SP = facelike SP

b) Any SP^* proof can be made facelike with a quasi-poly blowup in size.

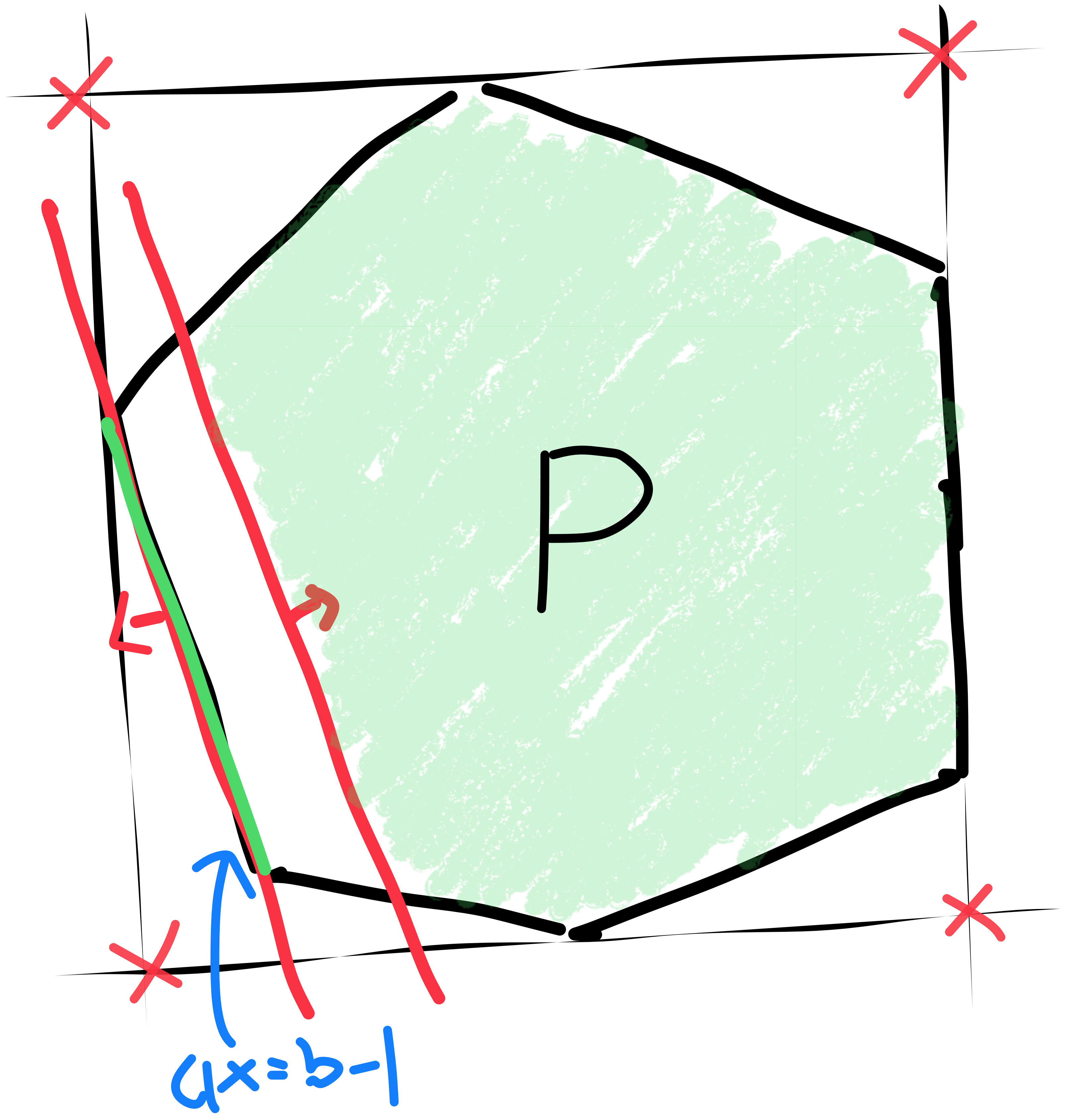
CP = Facelike SP

e.g.



CP = Facelike SP

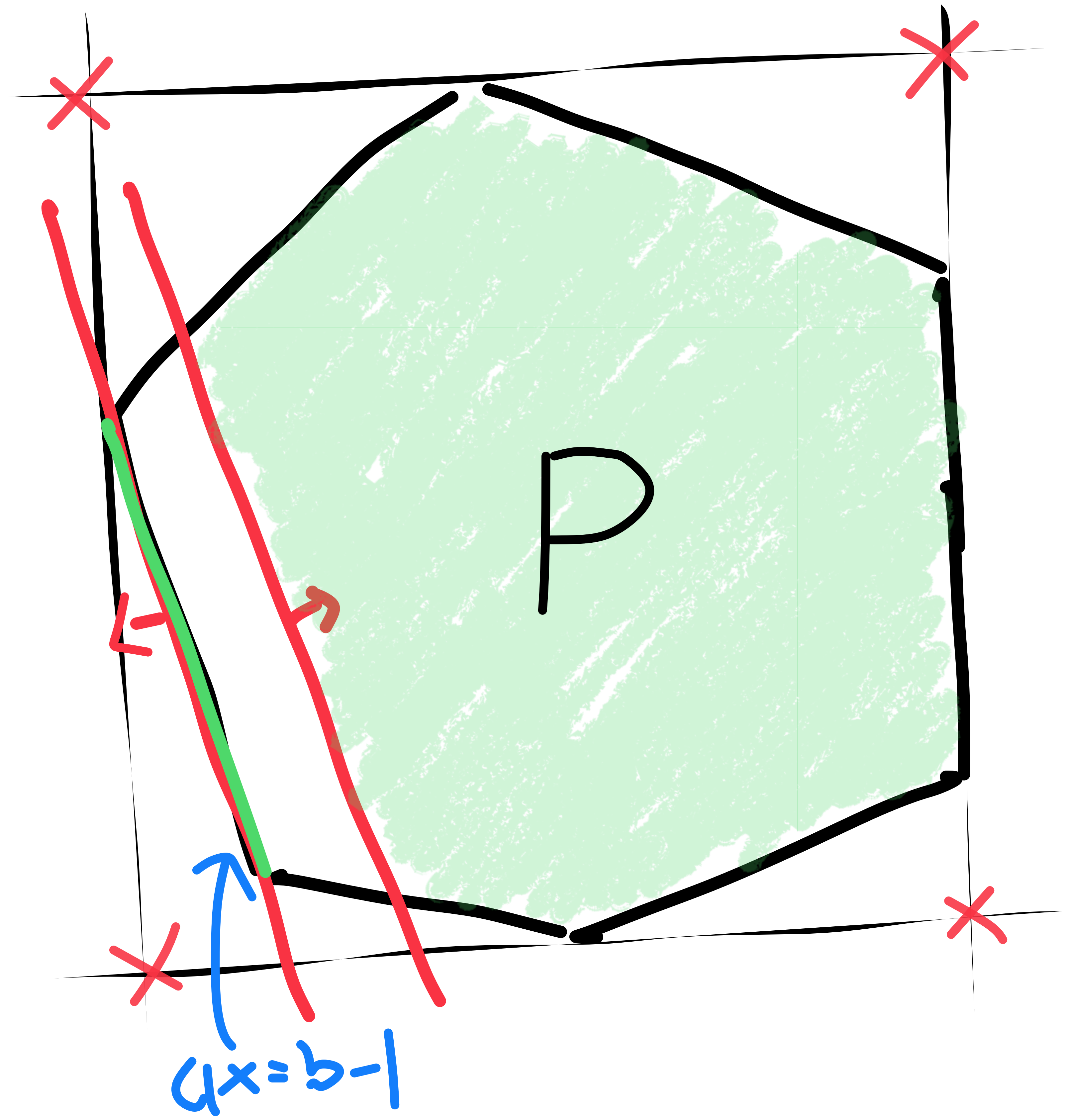
Observation: If we can refute $P \cap \{ax = b-1\}$ then we can deduce $P \cap \{ax \geq b\}$



CP = Facelike SP

Observation: If we can refute $P \cap \{ax = b-1\}$ then we can deduce $P \cap \{ax \geq b\}$

- $(ax \leq b-1, ax \geq b)$ would be a pathlike query

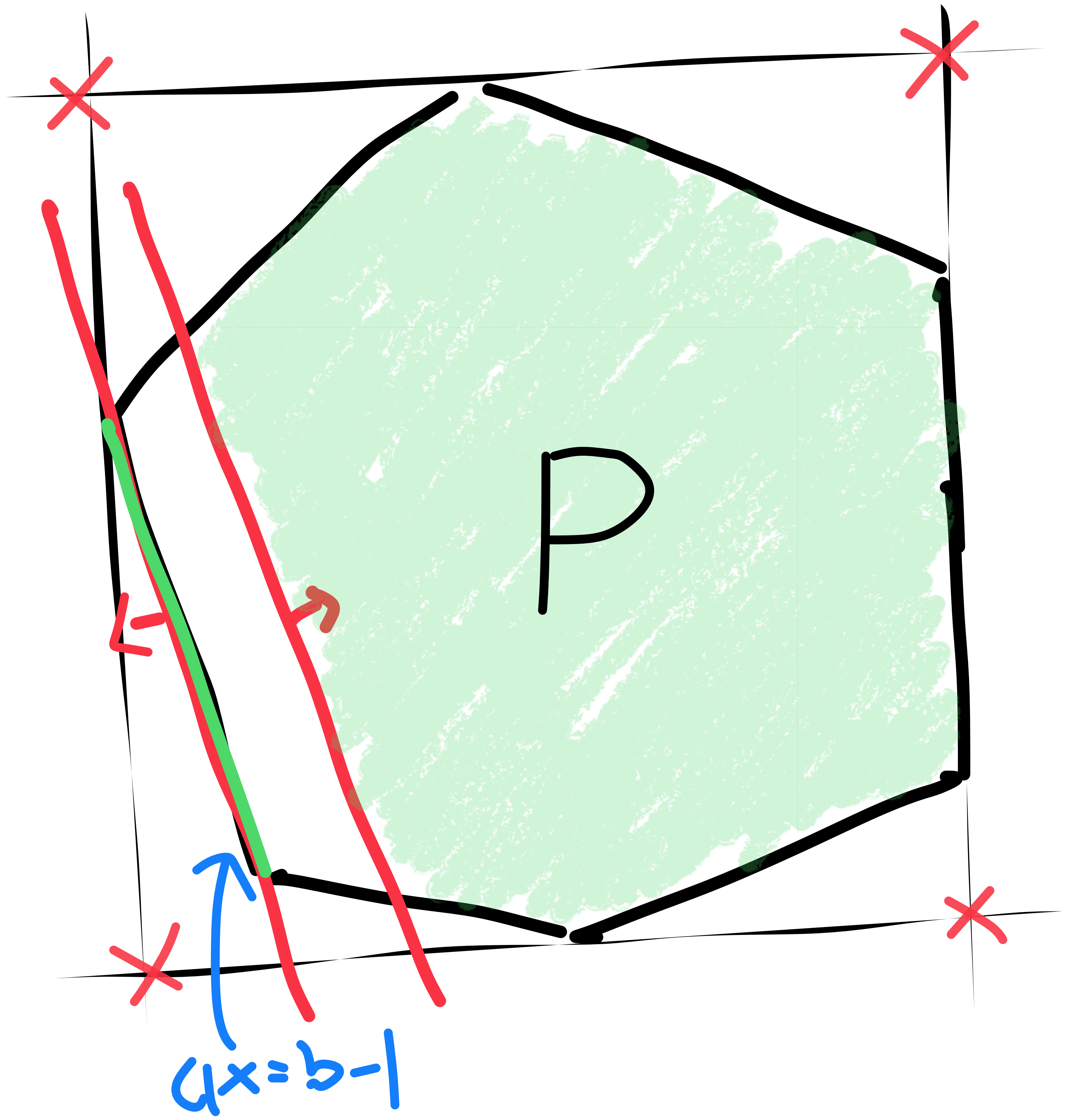


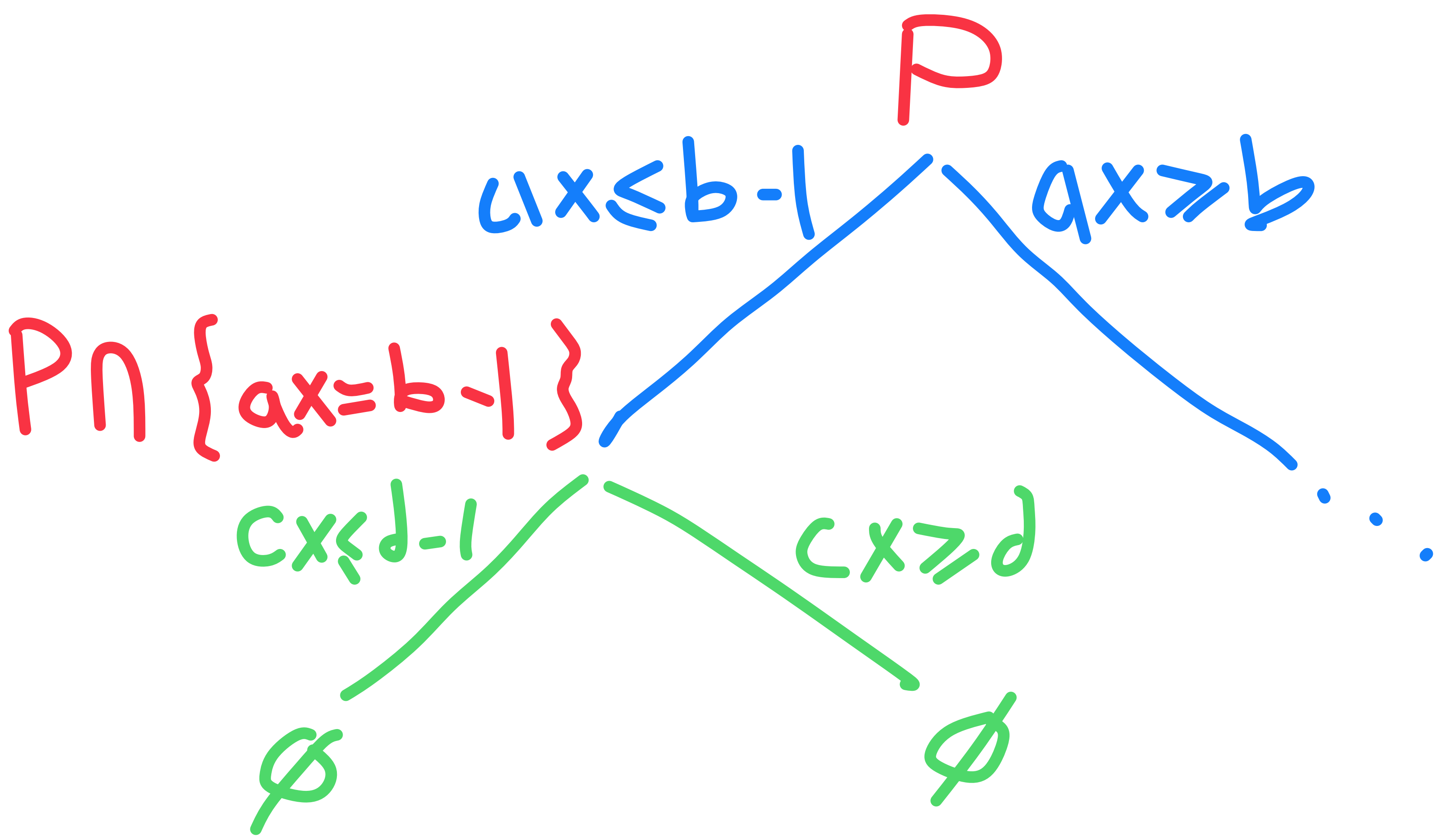
CP = Facelike SP

Observation: If we can refute $P \cap \{ax = b-1\}$ then we can deduce $P \cap \{ax \geq b\}$

- $(ax \leq b-1, ax \geq b)$ would be a pathlike query

Consider the partial Facelike SP proof



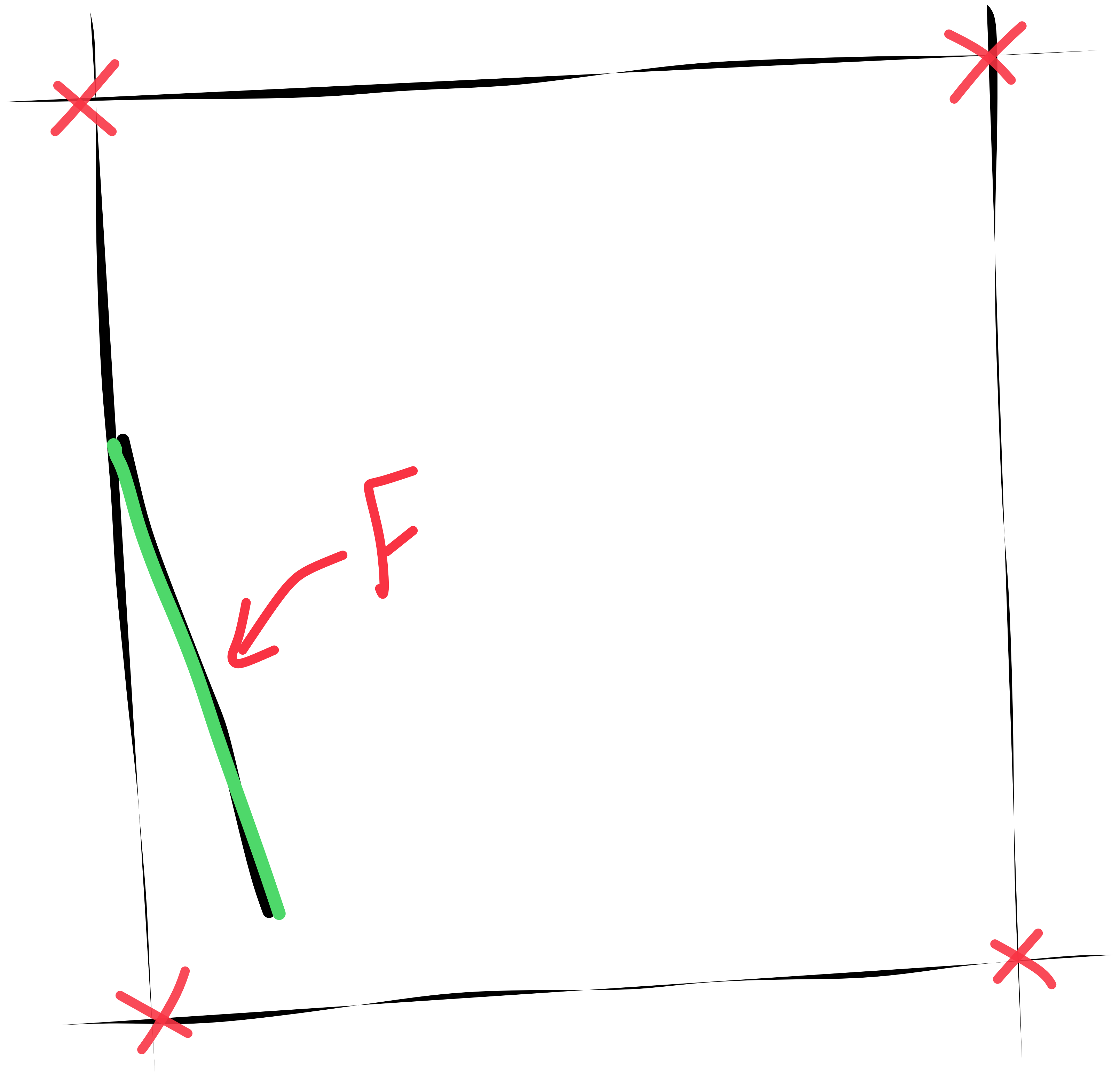


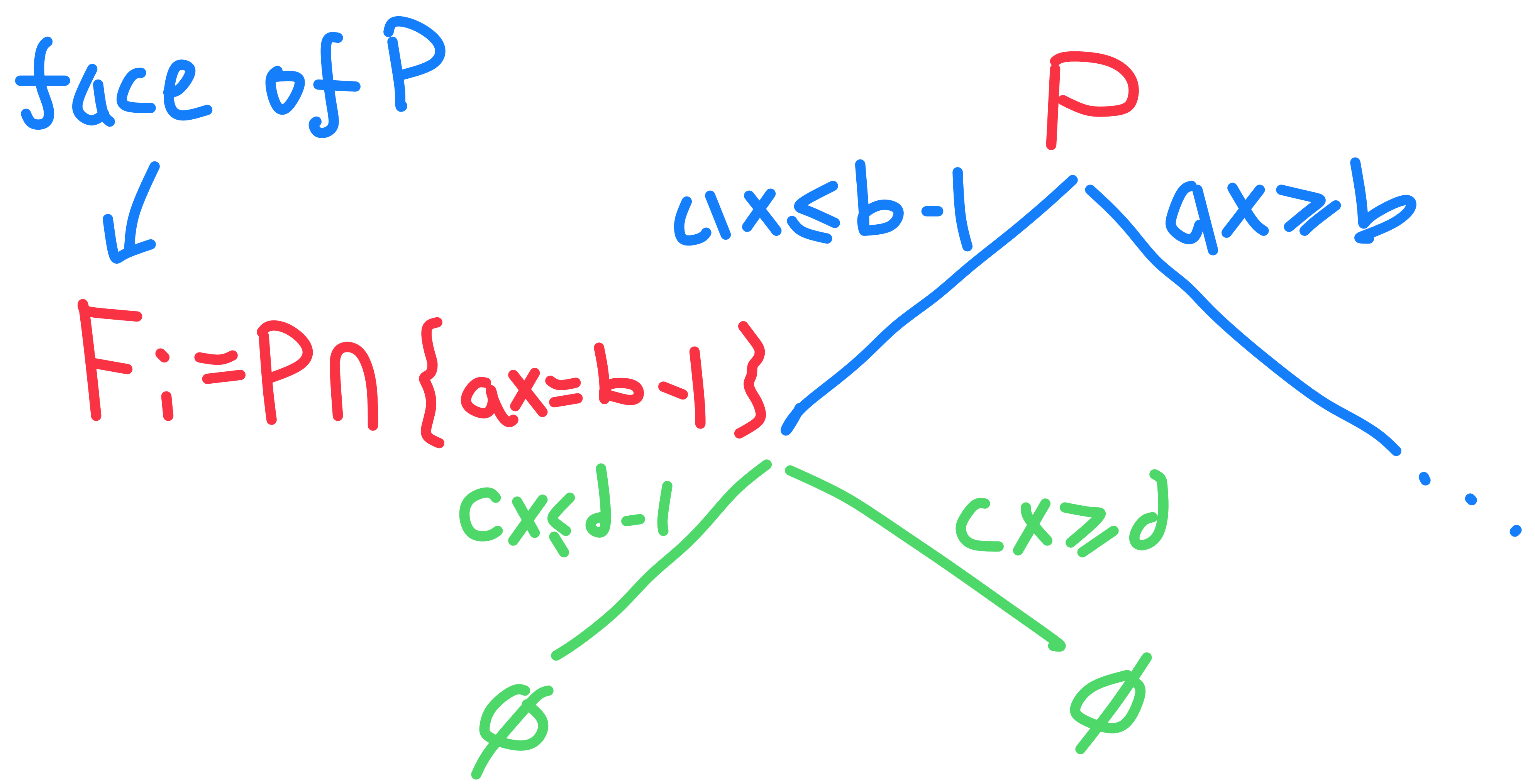
CP = Facelike SP

Observation: If we can refute $P \cap \{ax = b-1\}$ then we can deduce $P \cap \{ax \geq b\}$

- $(ax \leq b-1, ax \geq b)$ would be a pathlike query

Consider the partial Facelike SP proof



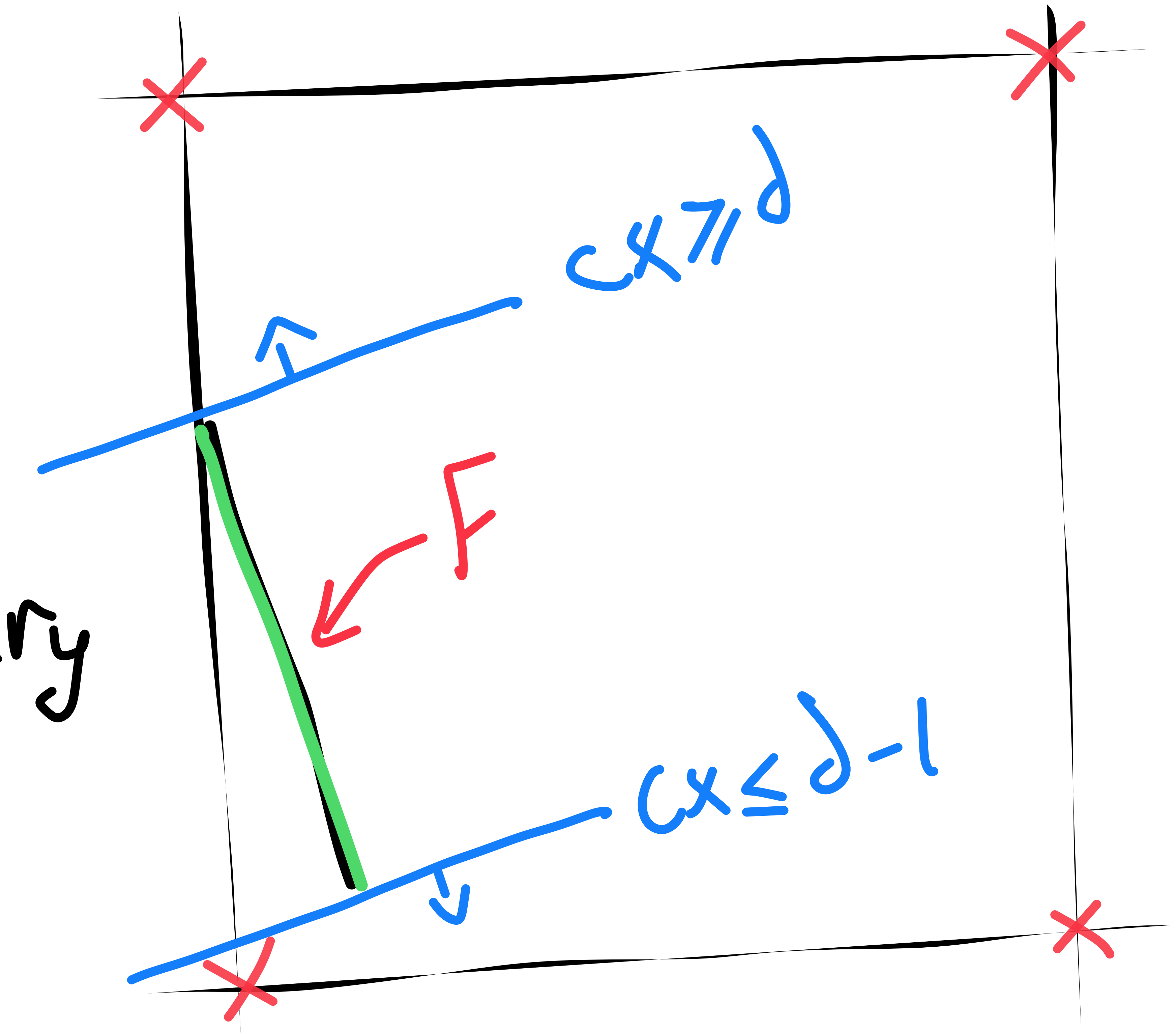


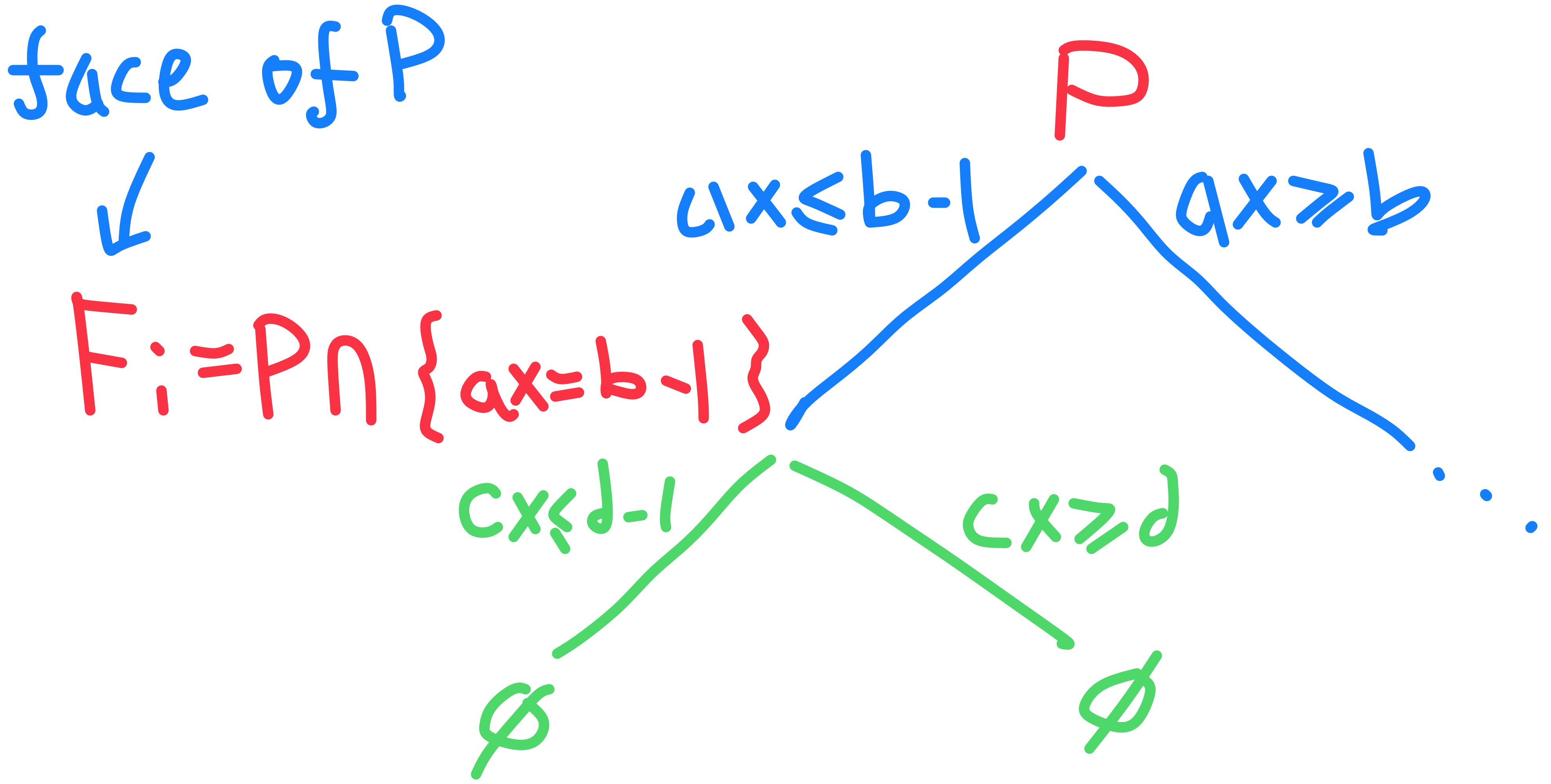
CP = Facelike SP

Observation: If we can refute $P\cap\{ax=b-1\}$ then we can deduce $P\cap\{ax\geq b\}$

- $(ax\leq b-1, ax\geq b)$ would be a pathlike query

Consider the partial Facelike SP proof





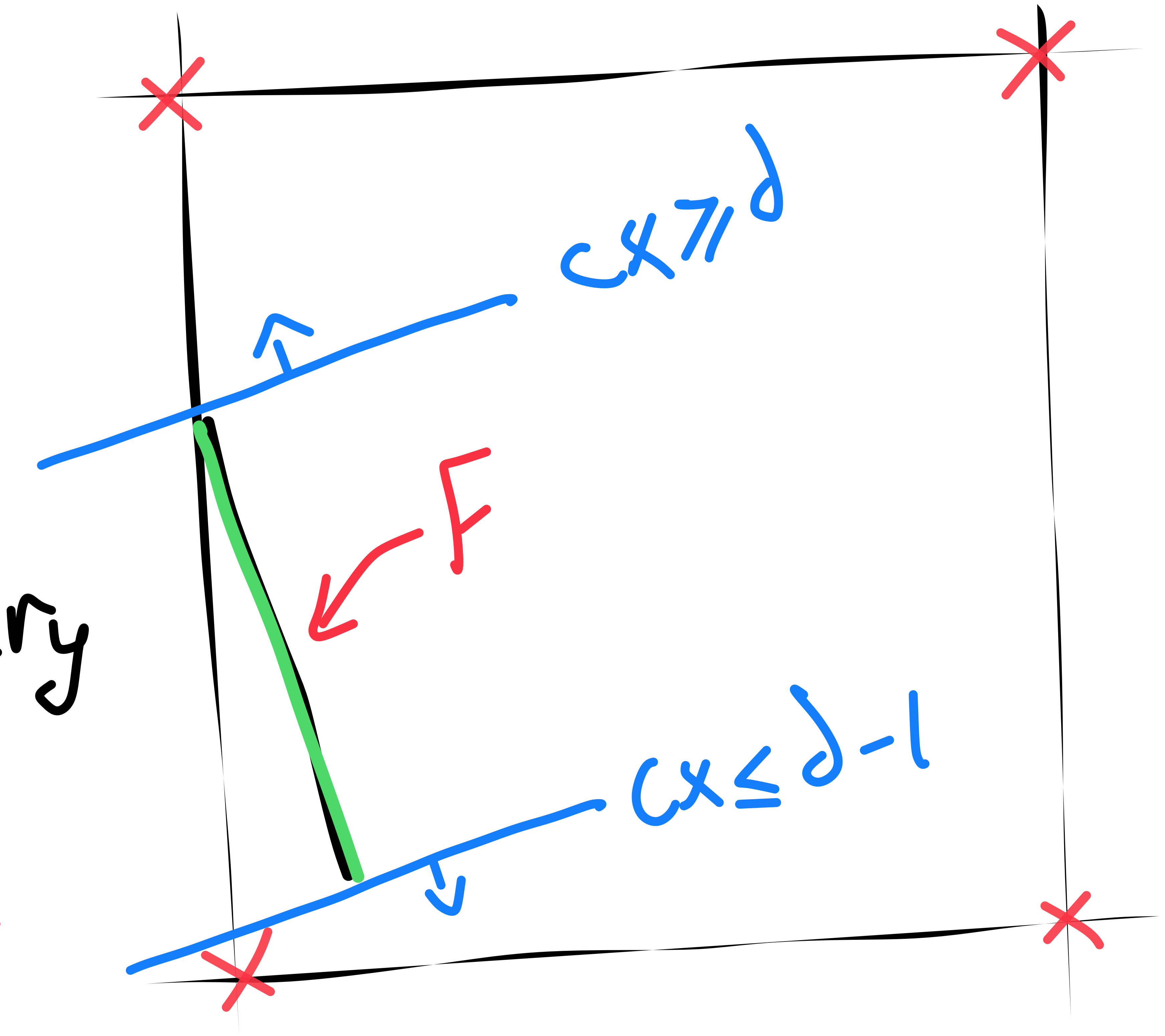
CP = Facelike SP

Observation: If we can refute $P \cap \{ax = b-1\}$ then we can deduce $P \cap \{ax \geq b\}$

- $(ax \leq b-1, ax \geq b)$ would be a pathlike query

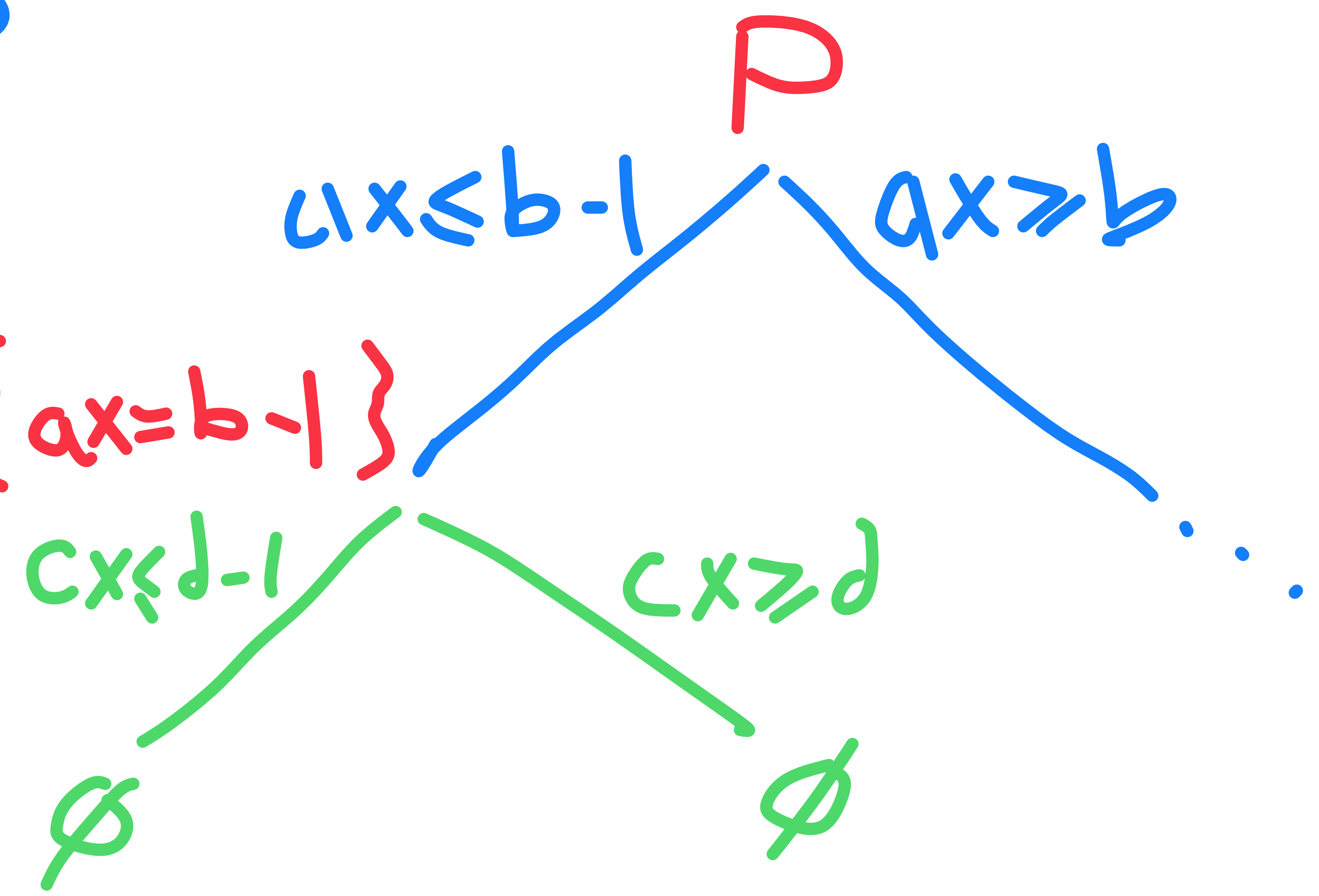
Consider the partial Facelike SP proof

Observe: $(cx \leq d-1, cx \geq d)$ is a pathlike query for F



face of P

$F := P \cap \{ax = b-1\}$



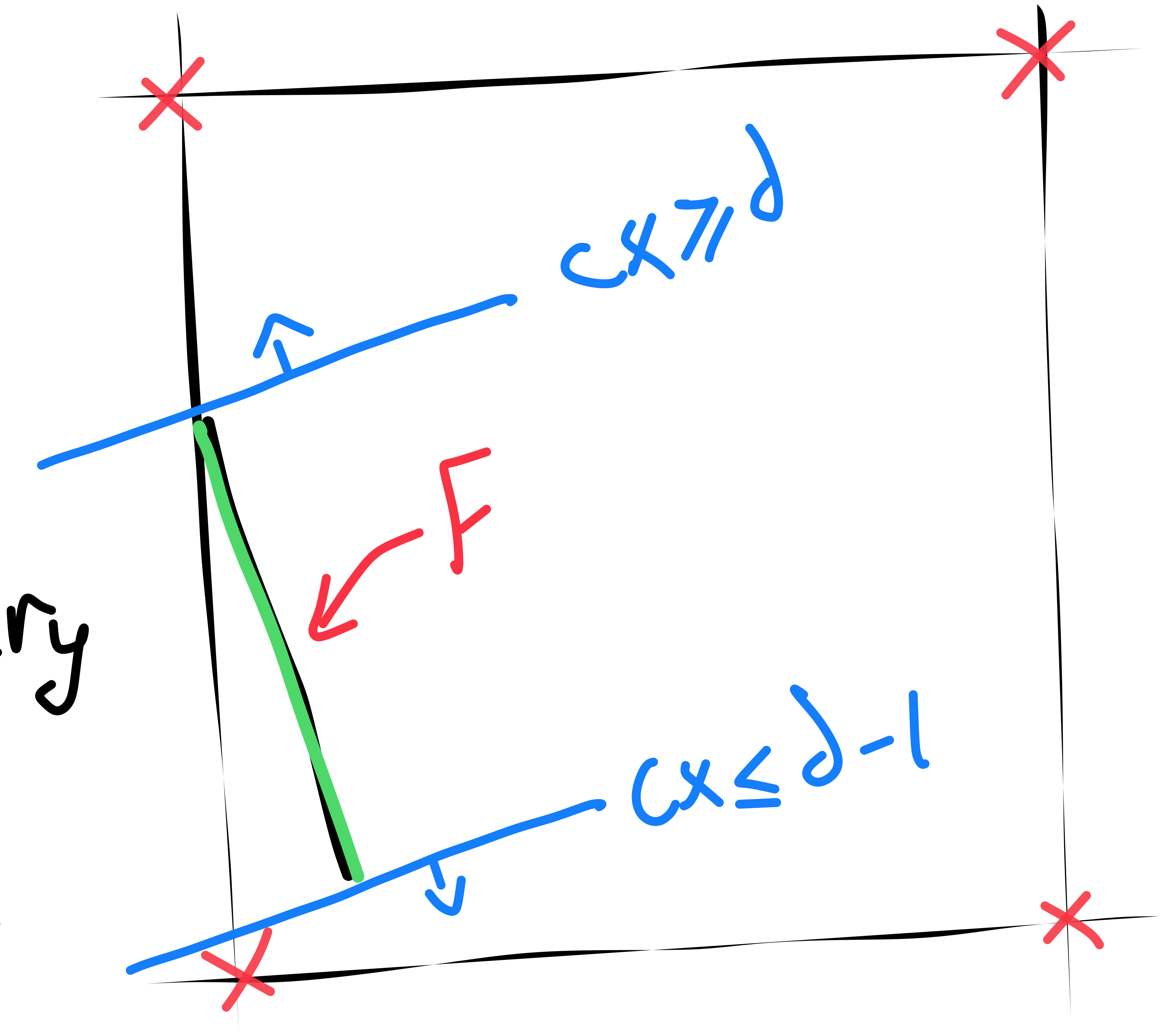
CP = Facelike SP

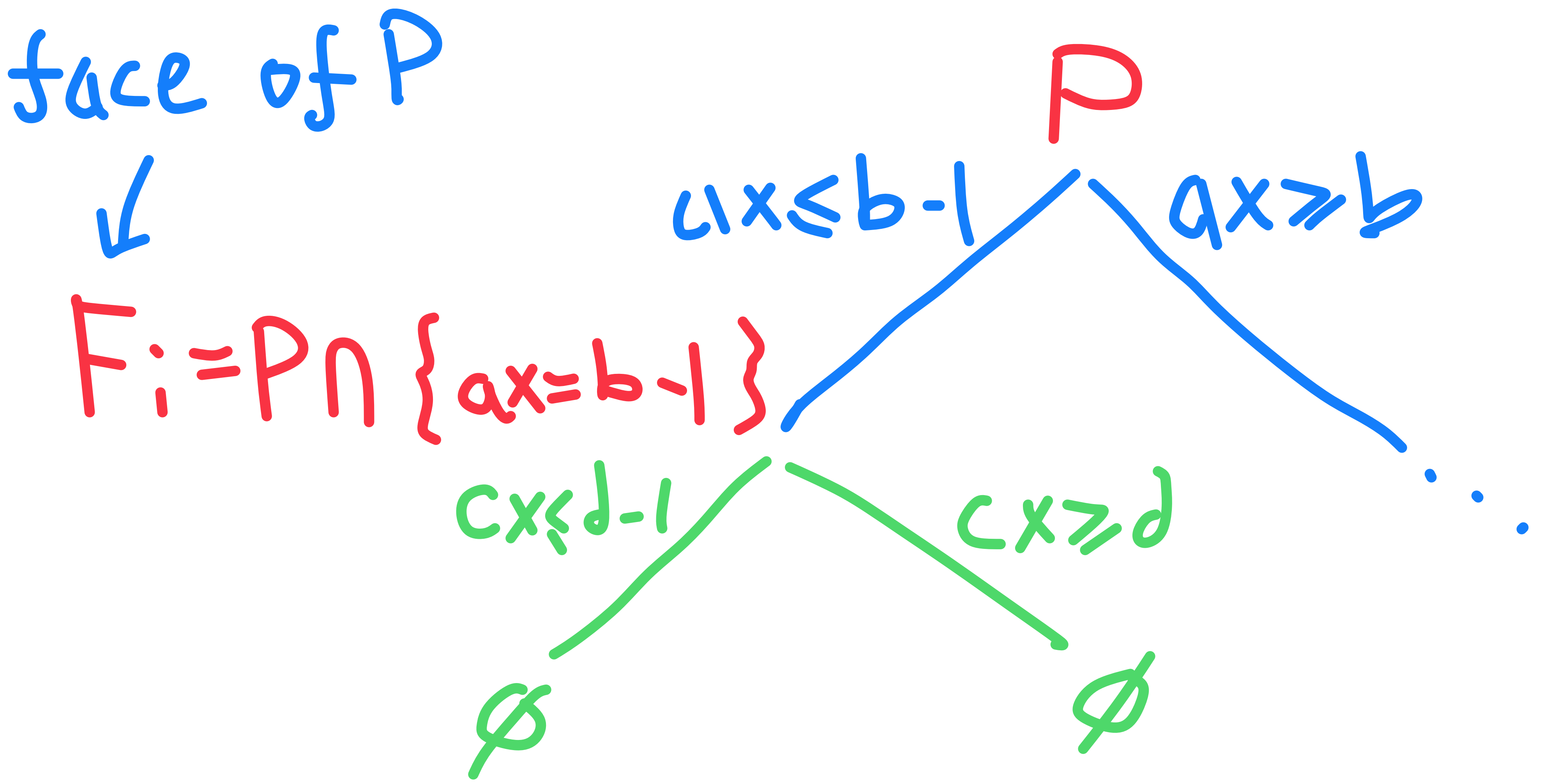
Observation: If we can refute $P \cap \{ax = b-1\}$ then we can deduce $P \cap \{ax \geq b\}$

- $(ax \leq b-1, ax \geq b)$ would be a pathlike query

Consider the partial Facelike SP proof

Observe: $(cx \leq d-1, cx \geq d)$ is a pathlike query for F





Q: Can we "simulate" the refutation of F when starting from P?
i.e. derive P' st $P' \cap F = \emptyset$?

CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

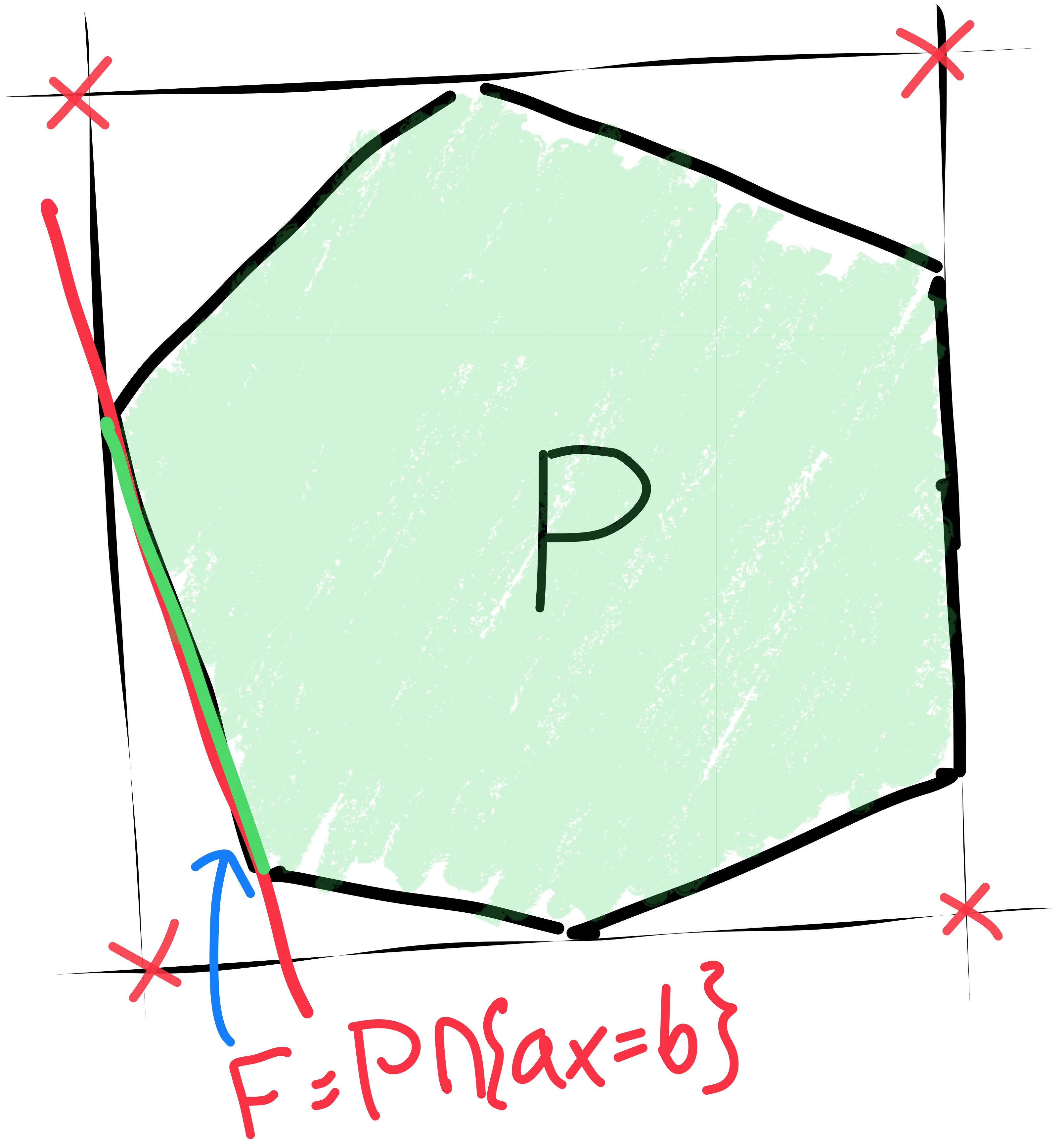
Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P' s.t. $P \vdash P'$ and $P' \cap F \subseteq F'$

CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P' s.t. $P \vdash P'$ and $P' \cap F \subseteq F'$

Proof Idea: Suppose $F = P \cap \{ax = b\}$

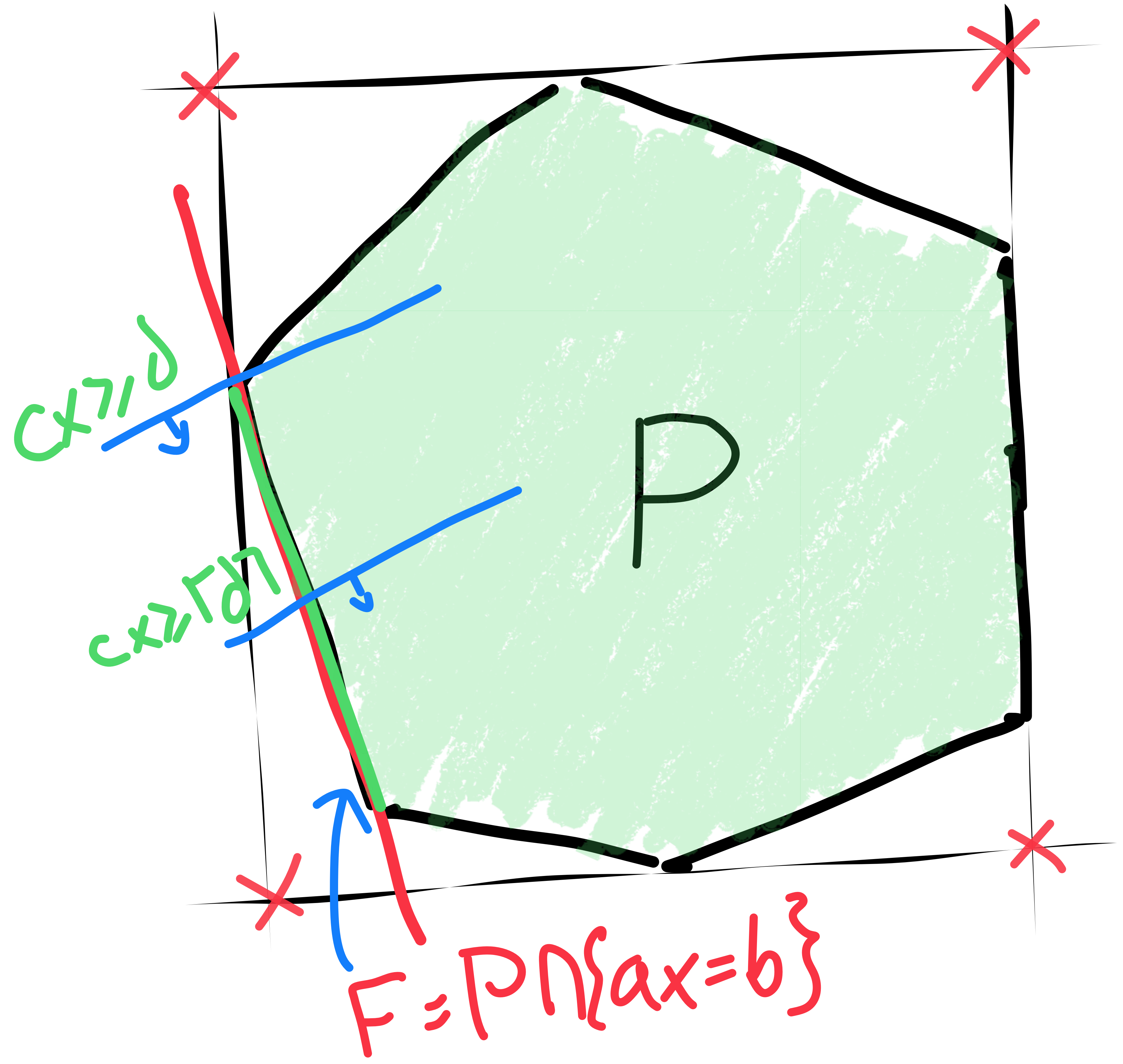


CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P'' s.t. $P \vdash P'$ and $P' \cap F \subseteq F'$

Proof Idea: Suppose $F = P \cap \{ax = b\}$ & $F \vdash F'$ by CG cut $cx \geq \lceil d \rceil$



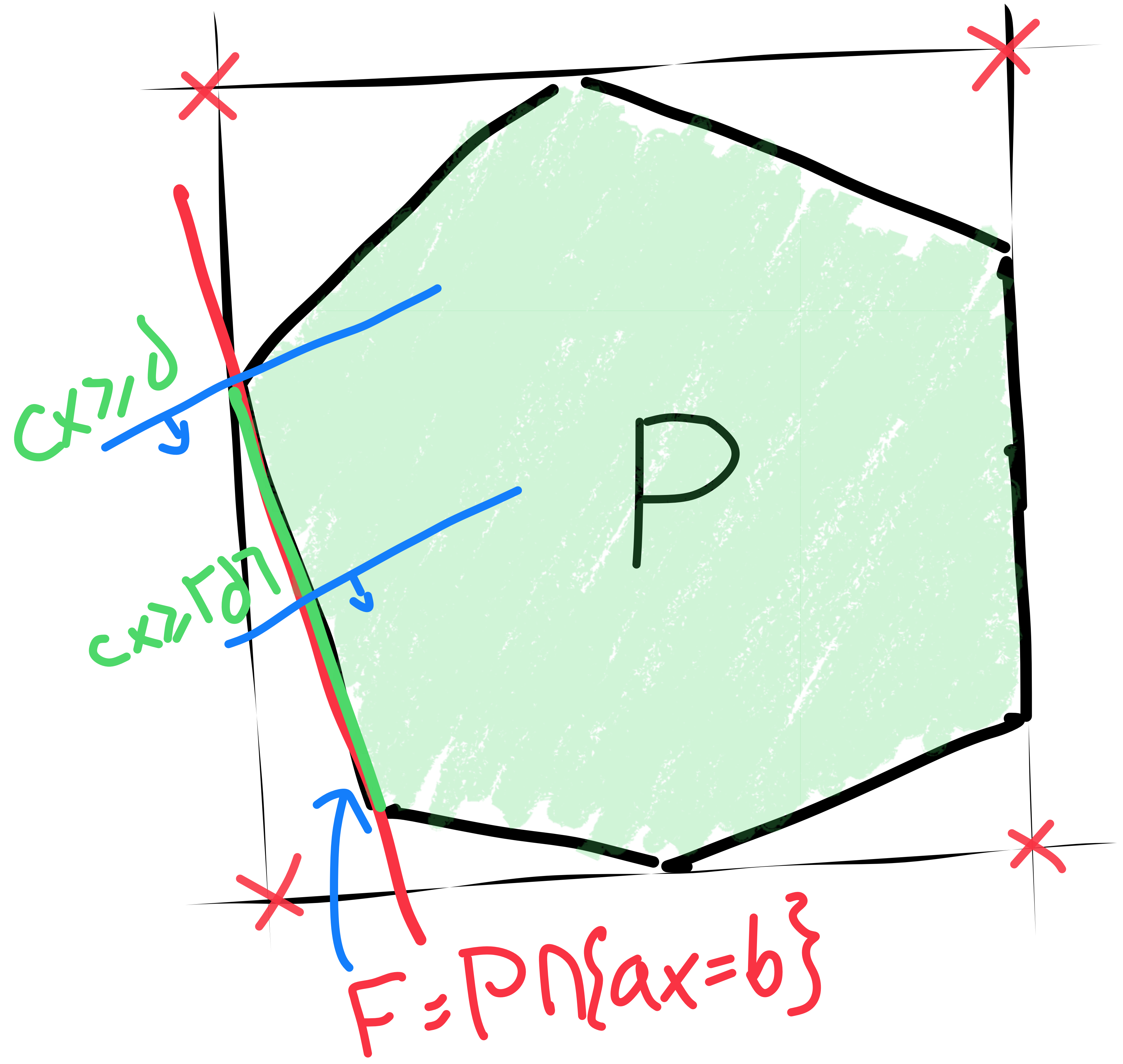
CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P'' s.t. $P \vdash P'$ and $P' \cap F \subseteq F'$

Proof Idea: Suppose $F = P \cap \{ax = b\}$ & $F \vdash F'$ by CG cut $cx \geq \lceil d \rceil$

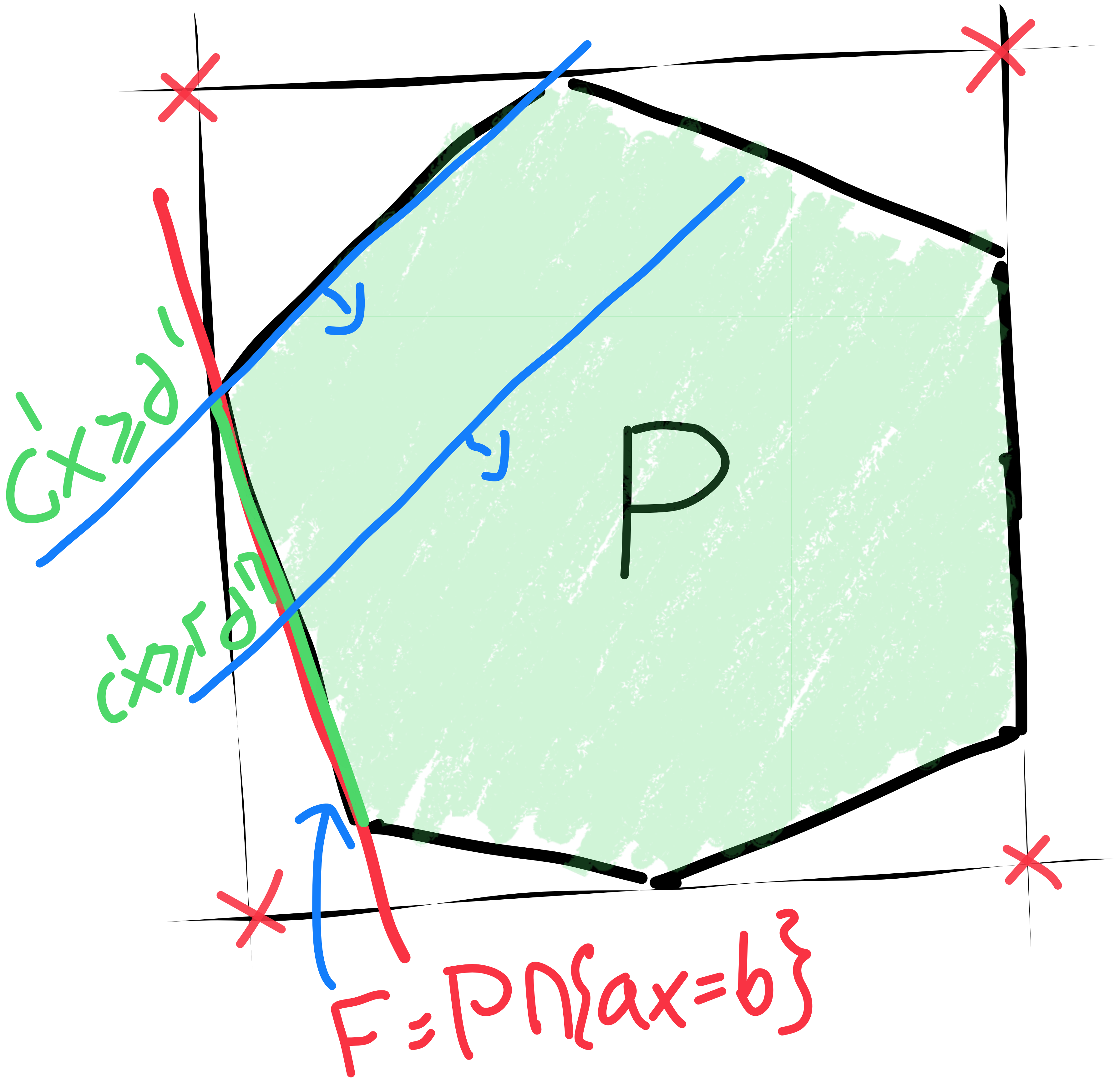
- $ax \leq b$ only inequality of F not valid for P



CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P'' s.t. $P \vdash P'$ and $P' \cap F \subseteq F'$



Proof Idea: Suppose $F = P \cap \{ax=b\}$ & $F \vdash F'$ by CG cut $cx \geq \lceil d \rceil$

- $ax \leq b$ only inequality of F not valid for P

$\therefore$ Shift $cx \geq d$ by $ax \geq b$ to $c'x \geq d'$ so it doesn't depend on $ax \leq b$

CP = Facelike SP

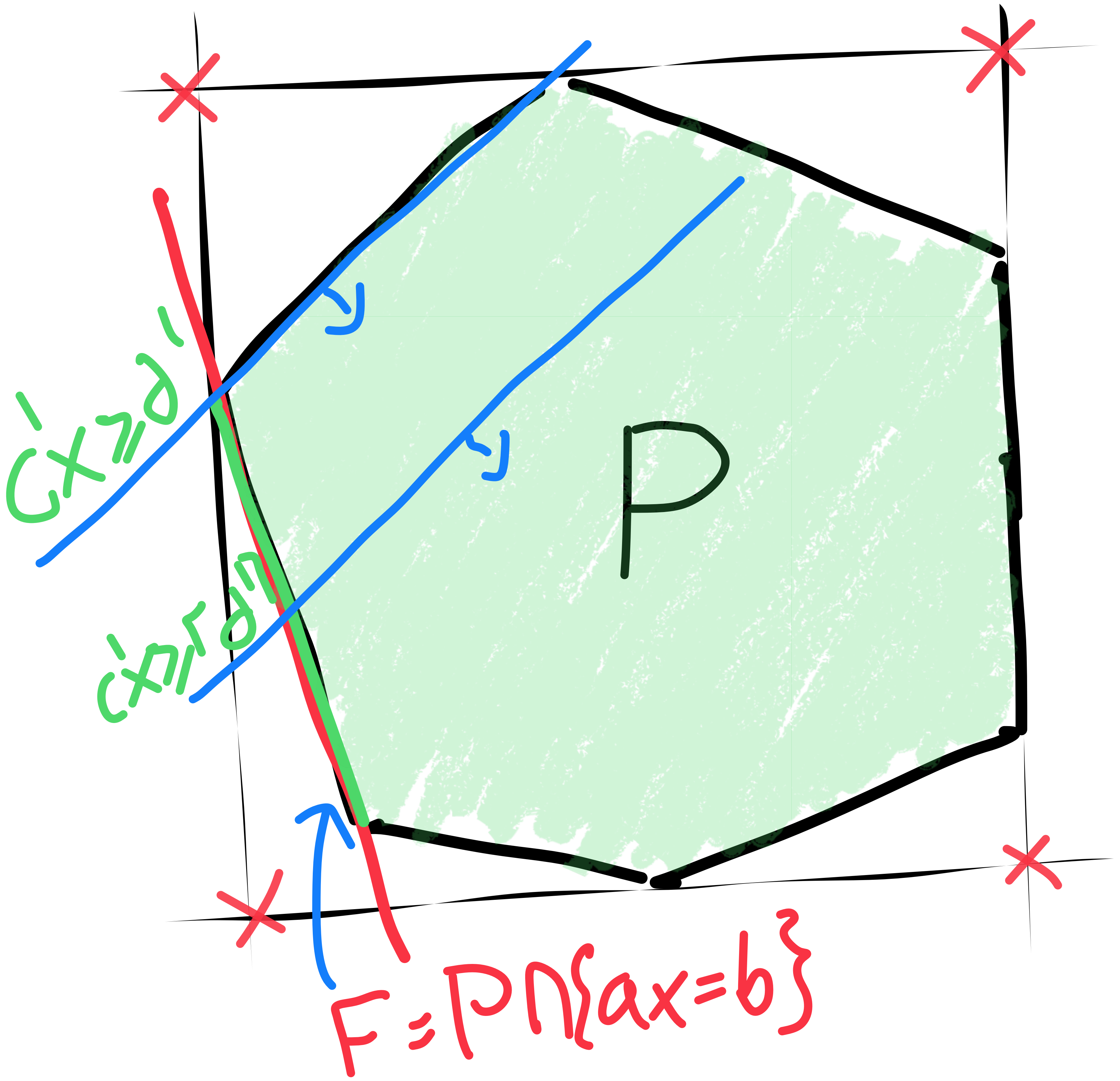
Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P'' s.t. $P \vdash P''$ and $P'' \cap F \subseteq F'$

Proof Idea: Suppose $F = P \cap \{ax = b\}$ & $F \vdash F'$ by CG cut $cx \geq \lceil d \rceil$

- $ax \leq b$ only inequality of F not valid for P
- ∴ Shift $cx \geq d$ by $ax \geq b$ to $c'x \geq d'$ so it doesn't depend on $ax \leq b$

$\Rightarrow P \cap \{c'x \geq \lceil d' \rceil\} \cap F \subseteq F'$



CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P' s.t. $P \vdash P'$ and $P' \cap F \subseteq F'$

Proof (CP = Facelike SP): In-order traversal of the SP proof using Shrijver

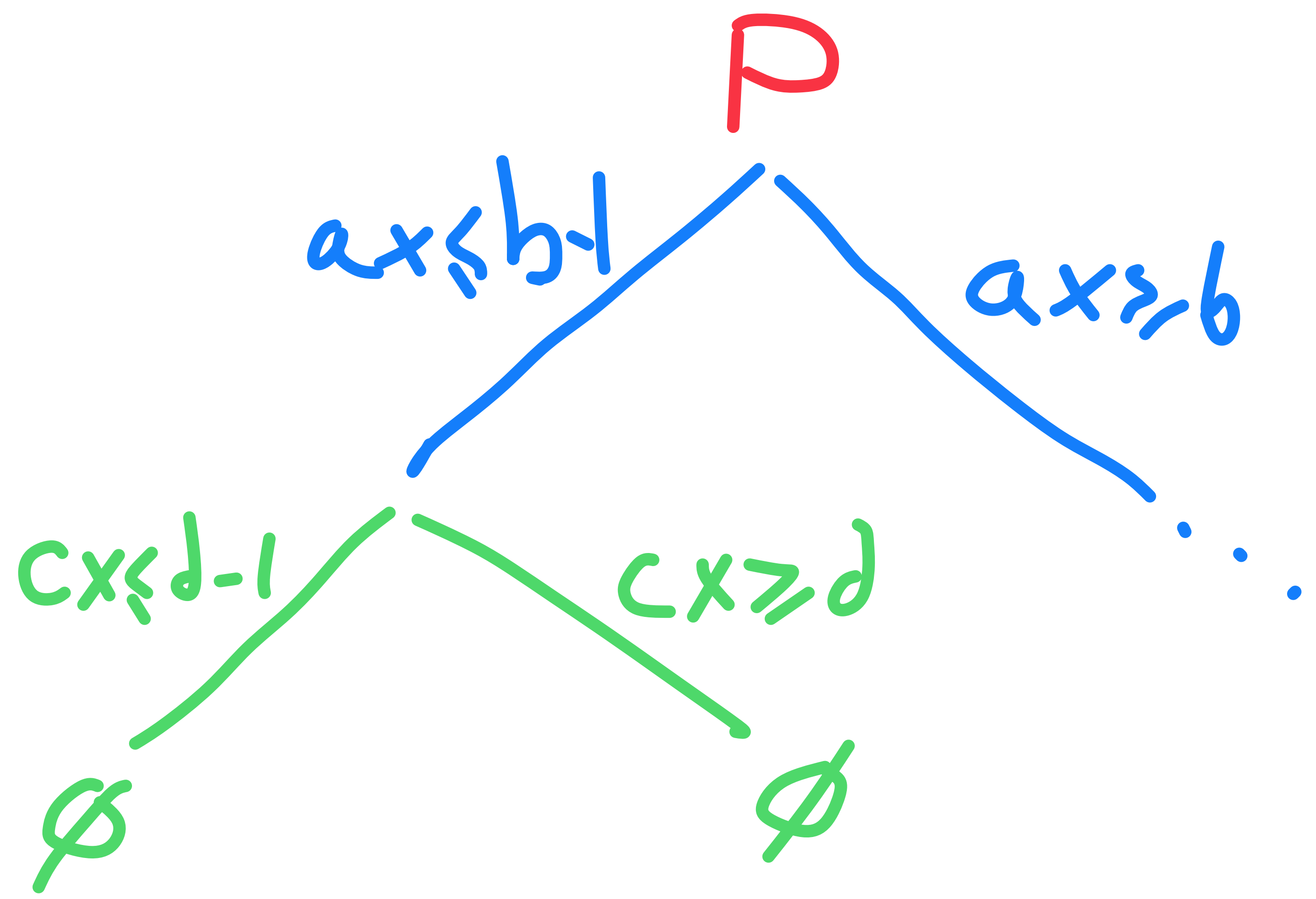
CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P'' s.t. $P \vdash P'$ and $P' \cap F \subseteq F'$

Proof (CP = Facelike SP): In-order traversal of the SP proof using Shrijver

- e.g.



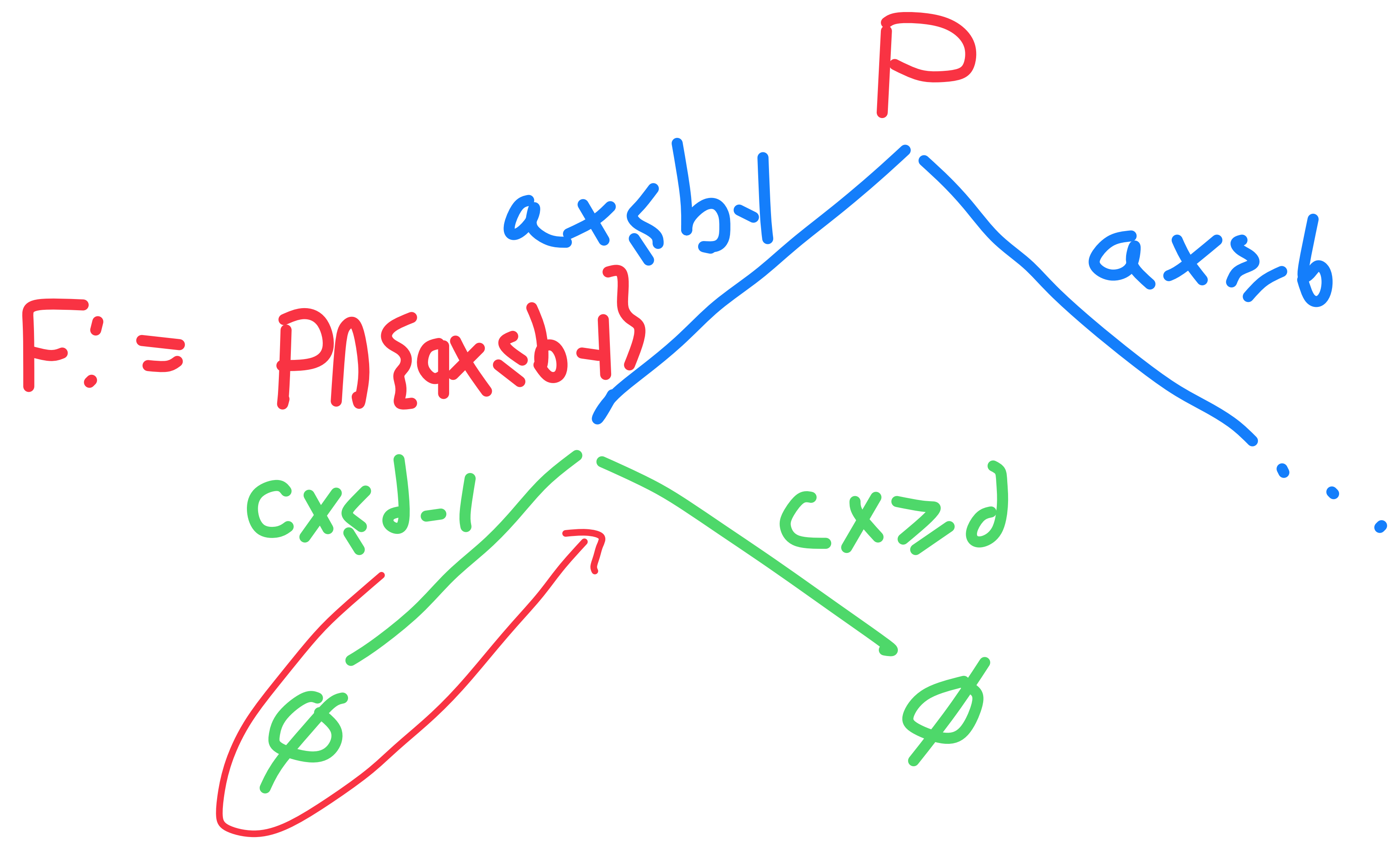
CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P'' s.t. $P \vdash P'$ and $P' \cap F \subseteq F'$

Proof (CP = Facelike SP): In-order traversal of the SP proof using Shrijver

- e.g.



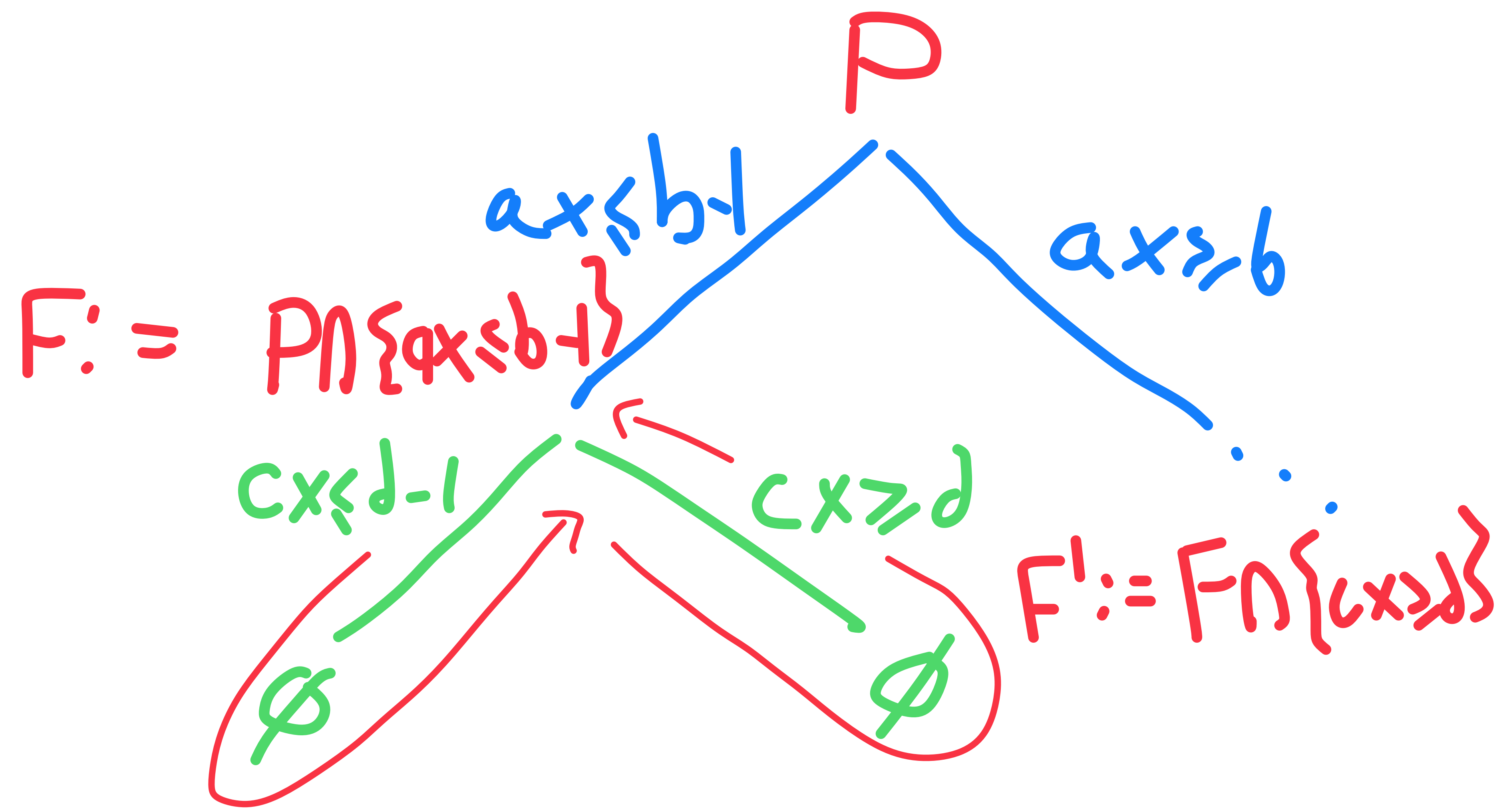
CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P'' s.t. $P \vdash P'$ and $P' \cap F \subseteq F'$

Proof (CP = Facelike SP): In-order traversal of the SP proof using Shrijver

- e.g.



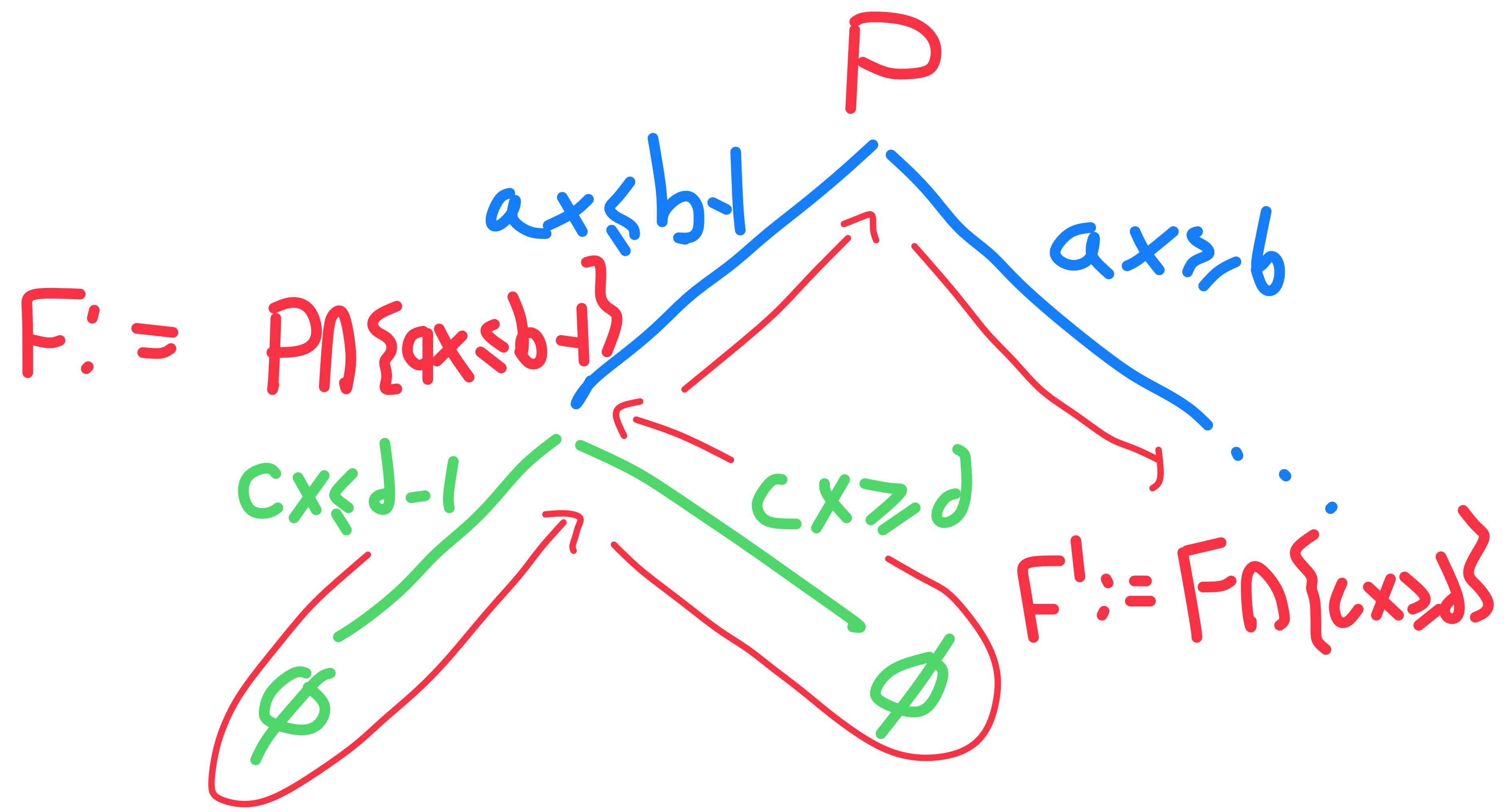
CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P' s.t. $P \vdash P'$ and $P' \cap F \subseteq F'$

Proof (CP = Facelike SP): In-order traversal of the SP proof using Shrijver

- e.g.



- By Schrijver there is P' s.t. $P \vdash P'$ and $P' \cap F \subseteq F \cap \{cx \geq d\} = \emptyset$

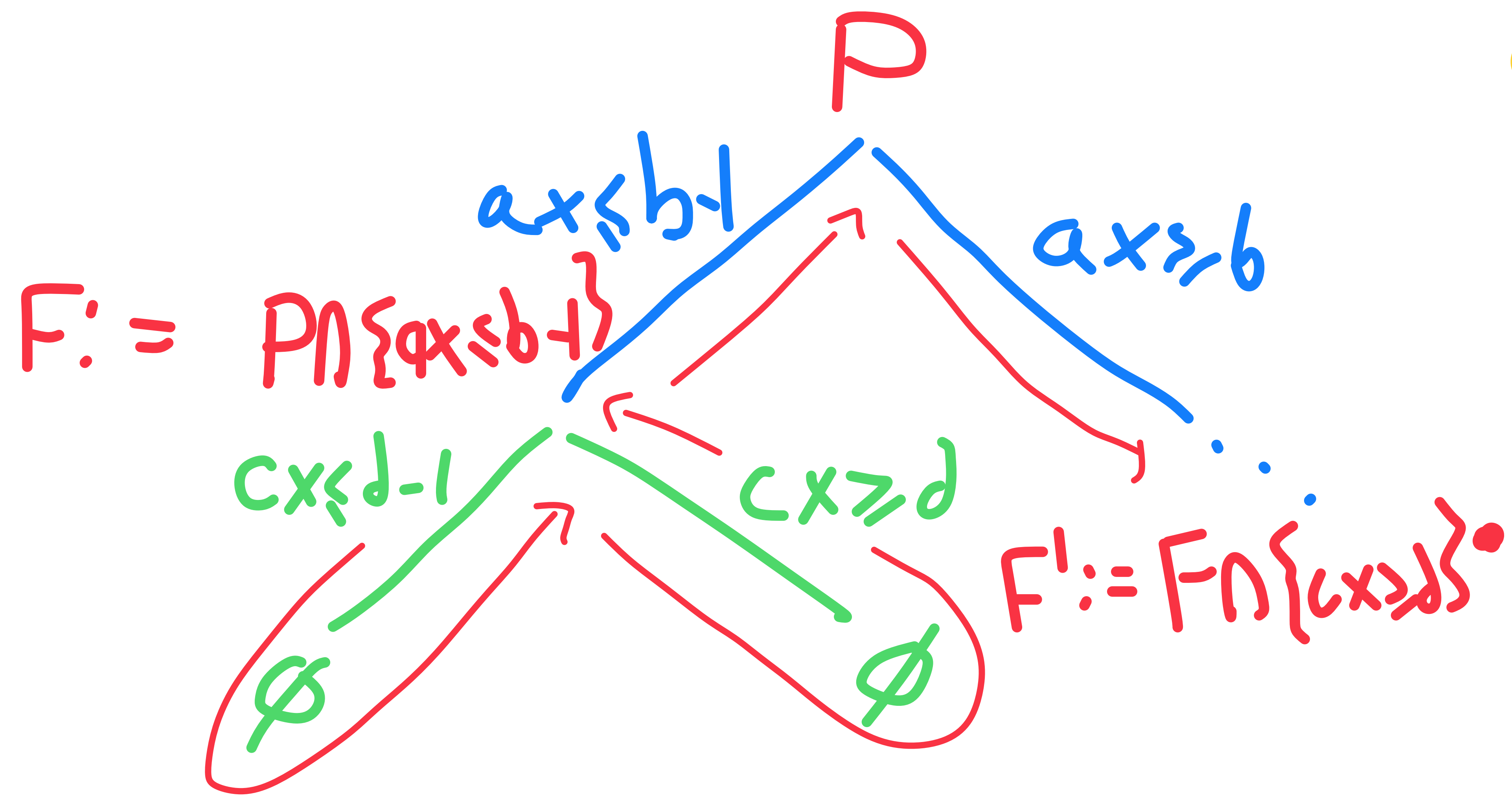
CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P'' s.t. $P \vdash P'$ and $P' \cap F \subseteq F'$

Proof (CP = Facelike SP): In-order traversal of the SP proof using Shrijver

- e.g.



- By Schrijver there is P' s.t. $P \vdash P'$ and $P' \cap F \subseteq F \cap \{cx \ge d\} = \emptyset$
- $\therefore$ there is $\varepsilon > 0$ st $ax \ge b-1+\varepsilon$ is valid for P'

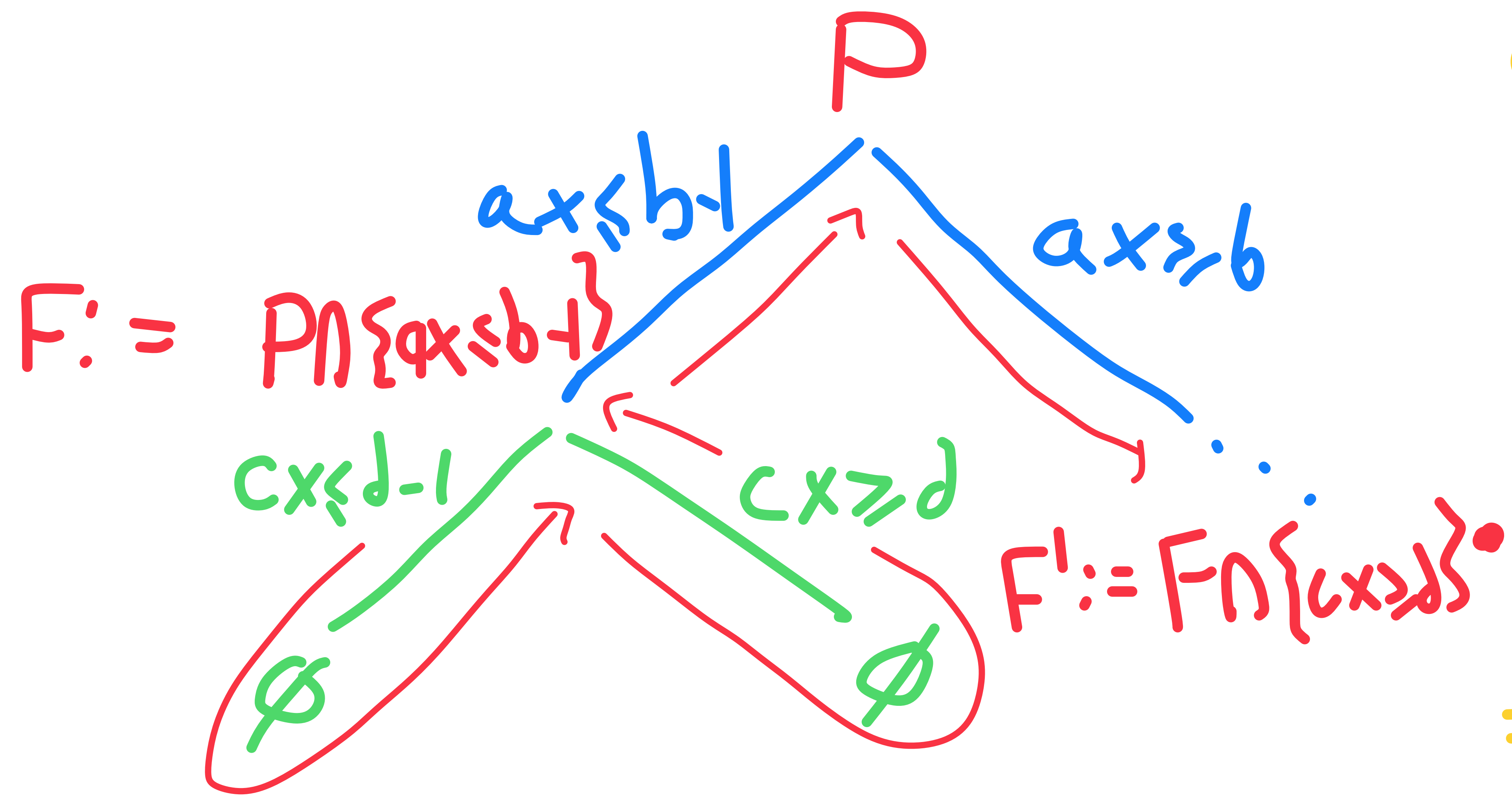
CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P' s.t. $P \vdash P'$ and $P' \cap F \subseteq F'$

Proof (CP = Facelike SP): In-order traversal of the SP proof using Shrijver

- e.g.



- By Schrijver there is P' s.t. $P \vdash P'$ and $P' \cap F \subseteq F \cap \{cx \geq d\} = \emptyset$
- $\therefore$ there is $\varepsilon > 0$ st $ax \geq b-1+\varepsilon$ is valid for P'
- $\Rightarrow$ $(ax \leq b-1,\ ax \geq b)$ is pathlike for P'

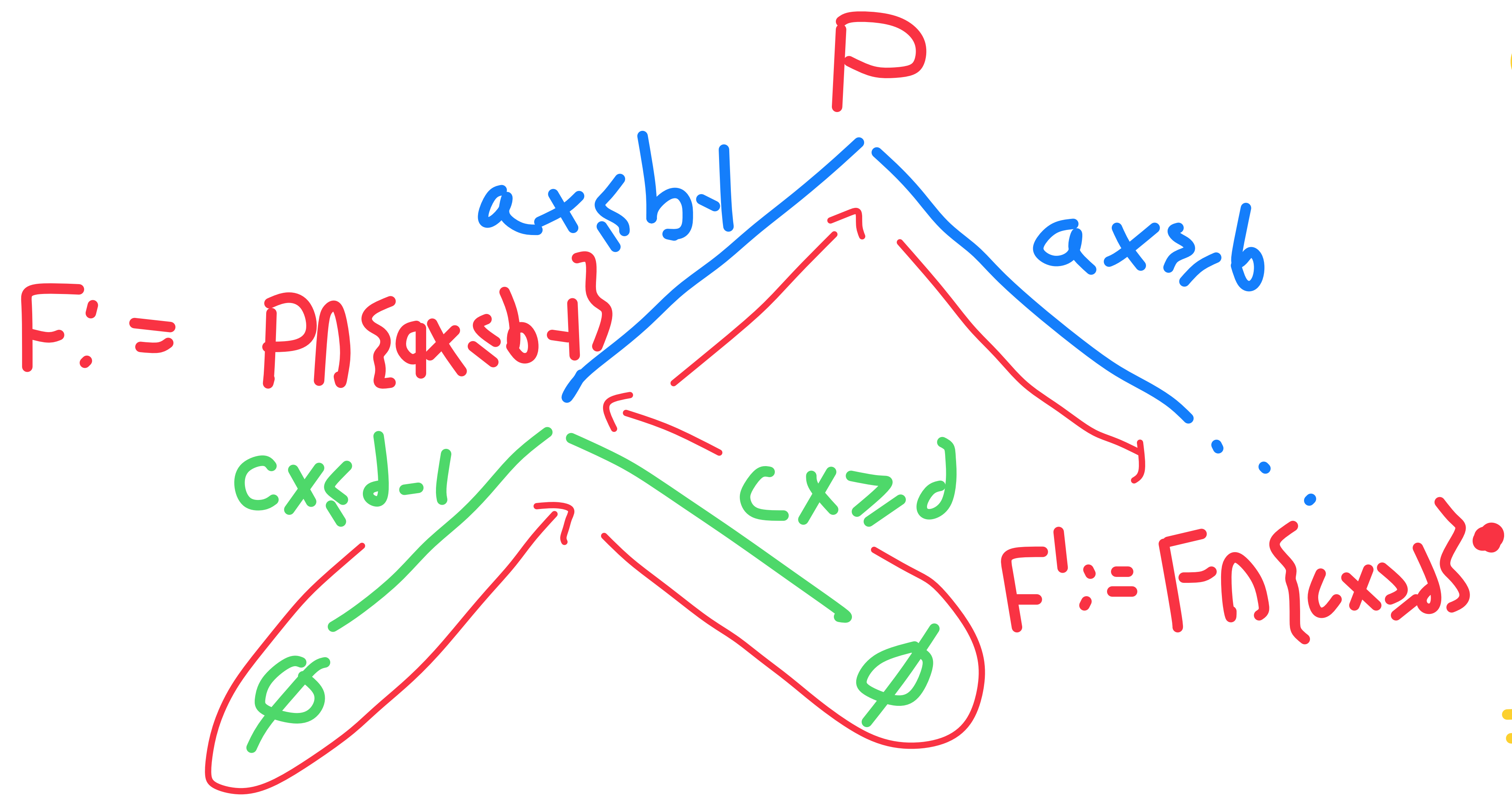
CP = Facelike SP

Notation. Say $P \vdash P'$ if P' is obtained from P by a pathlike query

Schrijver Lemma: Let F be a face of P if $F \vdash F'$ then there is P' s.t. $P \vdash P'$ and $P' \cap F \subseteq F'$

Proof (CP = Facelike SP): In-order traversal of the SP proof using Shrijver

- e.g.



- By Schrijver there is P' s.t. $P \vdash P'$ and $P' \cap F \subseteq F \cap \{cx \geq d\} = \emptyset$
- $\therefore$ there is $\varepsilon > 0$ st $ax \geq b-1+\varepsilon$ is valid for P'

$\Longrightarrow$ $(ax \leq b-1,\ ax \geq b)$ is pathlike for P'

$\Longrightarrow$ $P' \vdash P \cap \{ax \geq b\}$

Quasi-poly Simulation of SP*

Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

Quasi-poly Simulation of SP*

Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

→ focus on polytopes in $[0,1]^n$, but holds for others.

Quasi-poly Simulation of SP^*

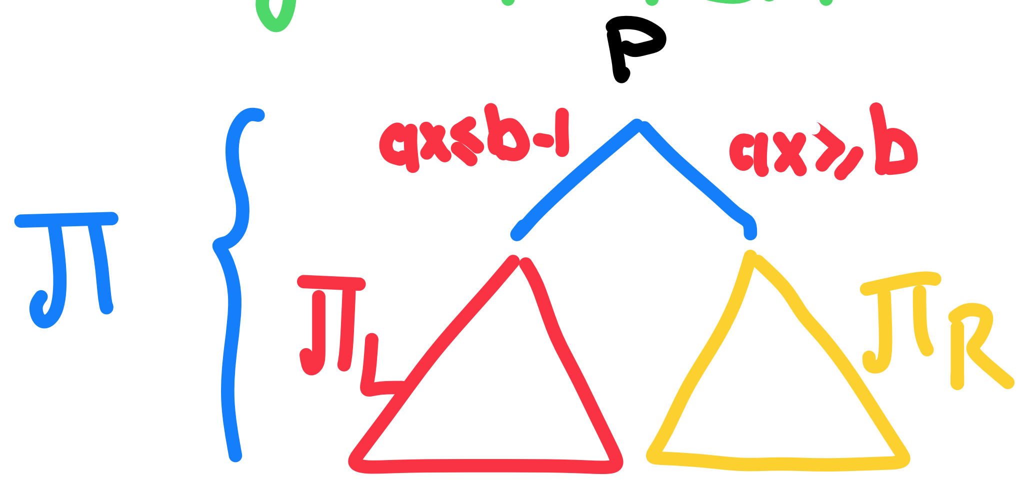
Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

Proof Sketch: Fix an SP^* proof π.

Quasi-poly Simulation of SP*

Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

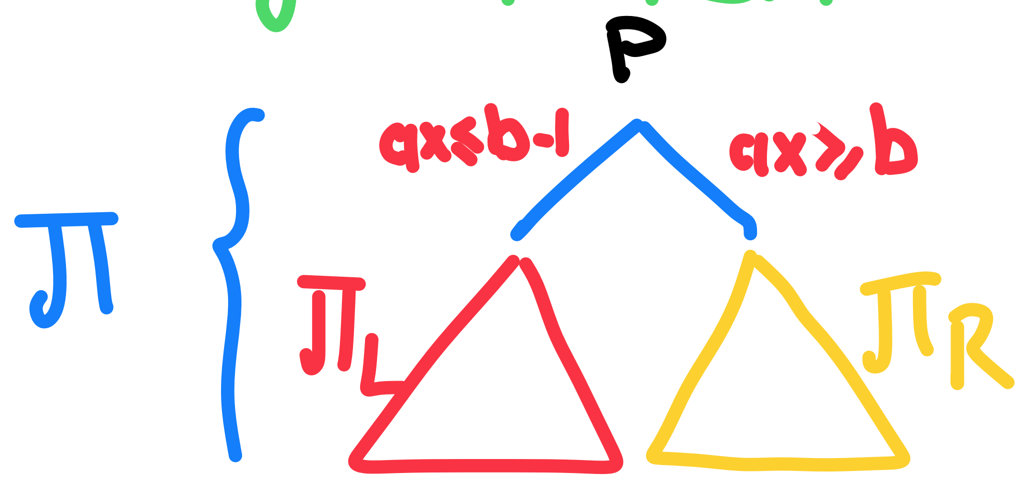
Proof Sketch: Fix an SP* proof π.

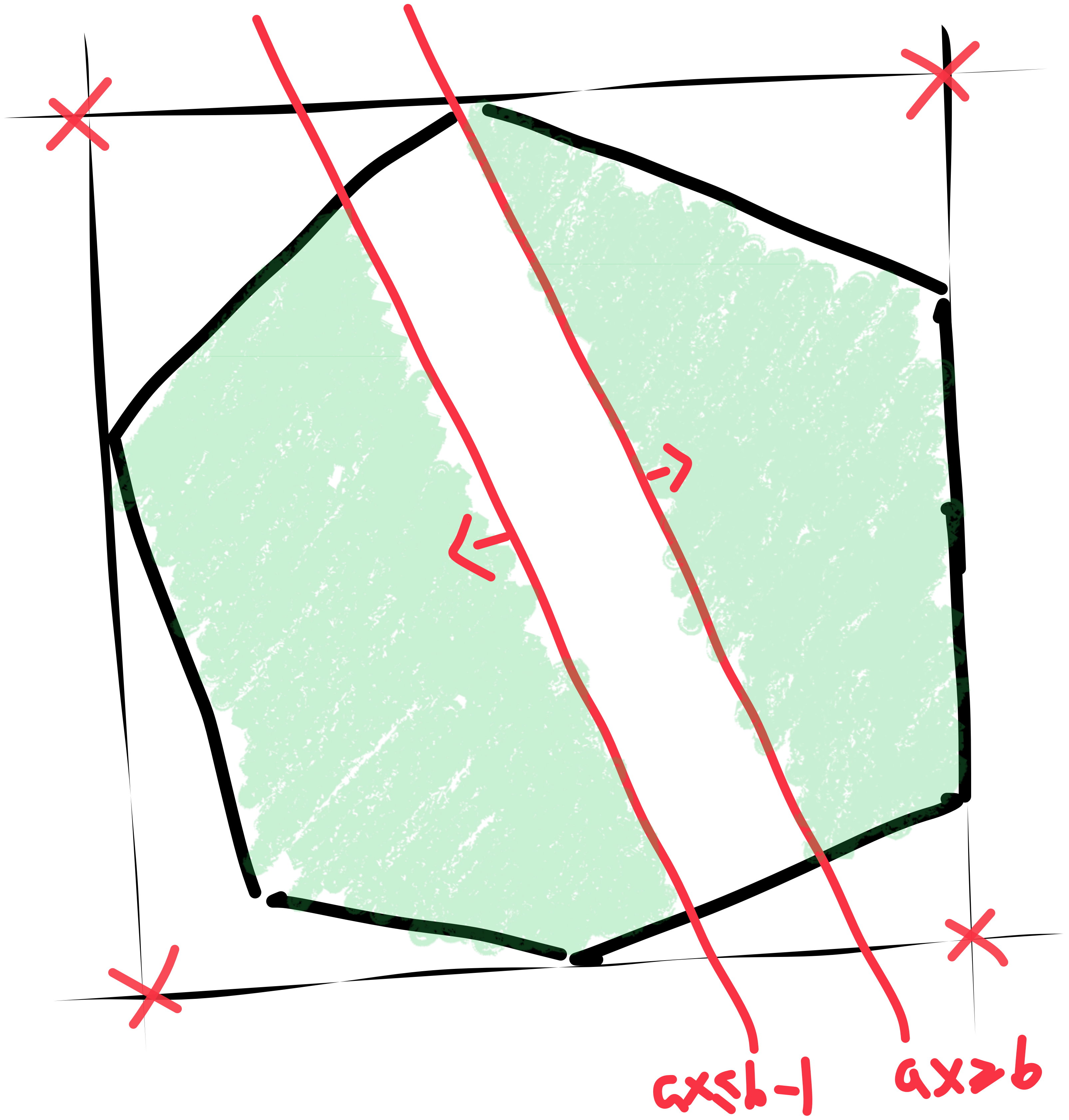


Quasi-poly Simulation of SP*

Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

Proof Sketch: Fix an SP* proof π.

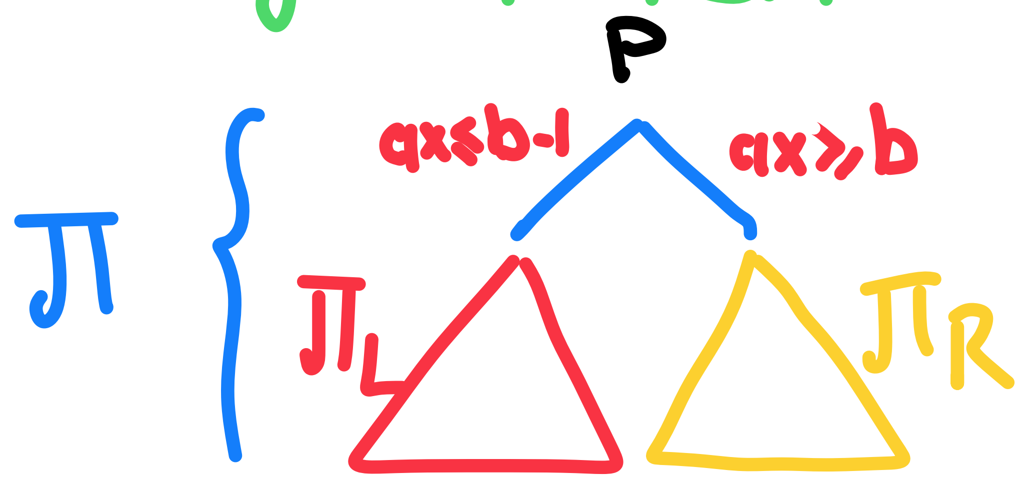




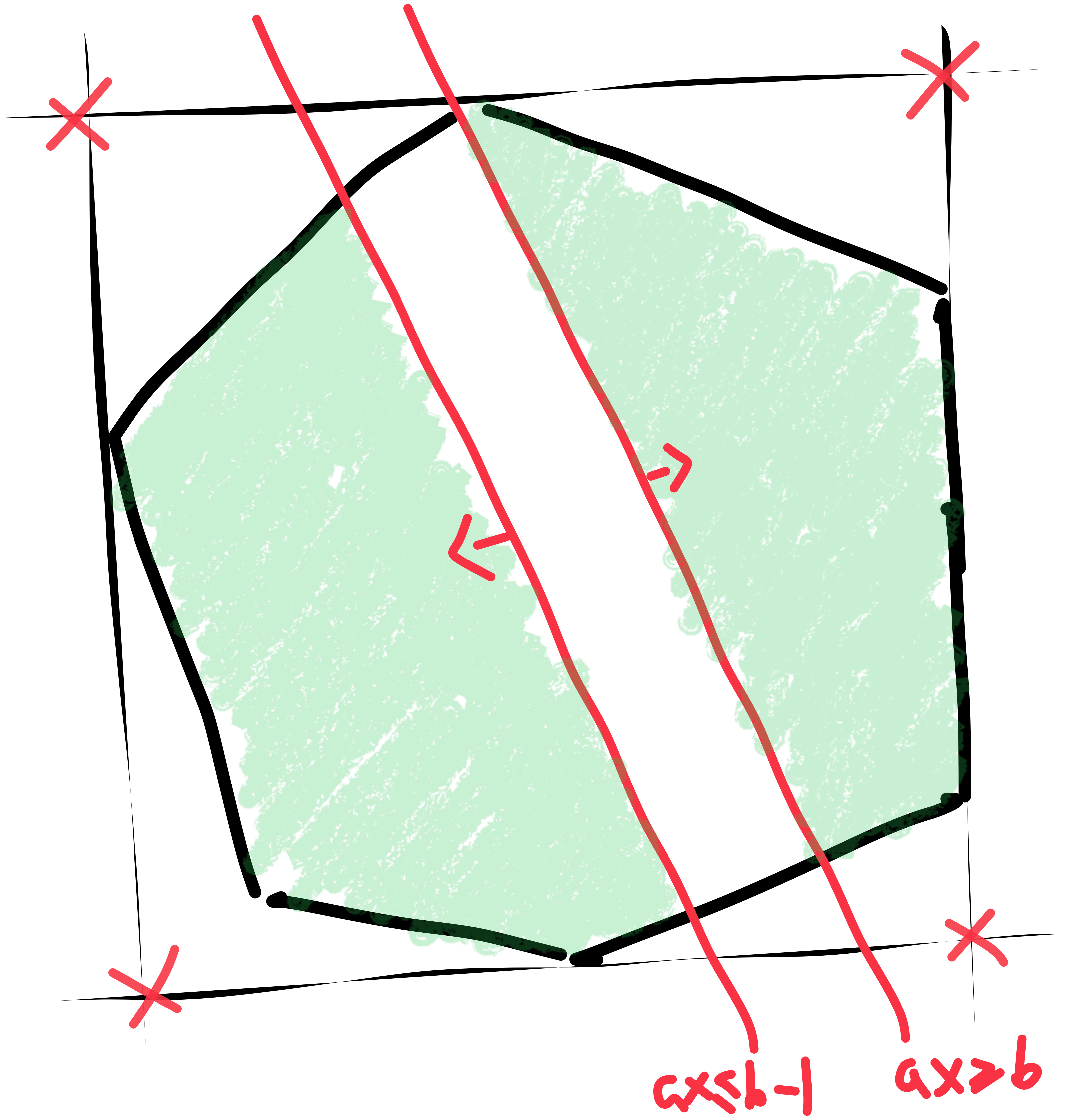
Quasi-poly Simulation of SP*

Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

Proof Sketch: Fix an SP* proof π.



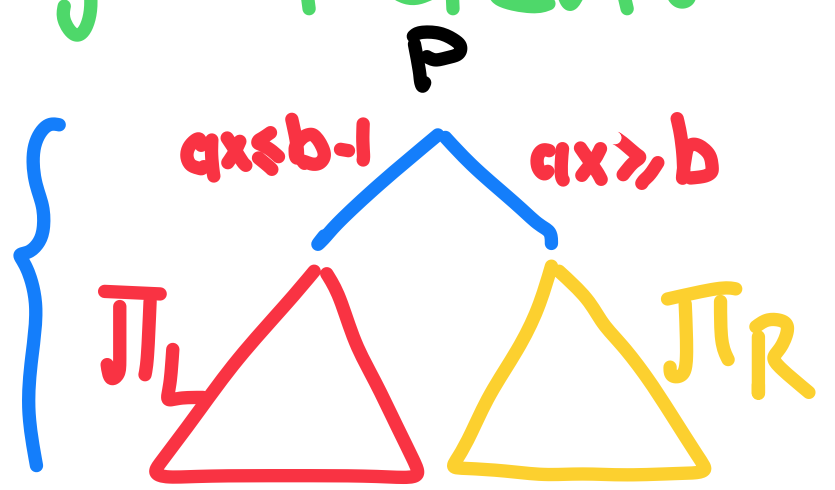
Suppose $|\pi_L| \leq |\pi_R|$



Quasi-poly Simulation of SP*

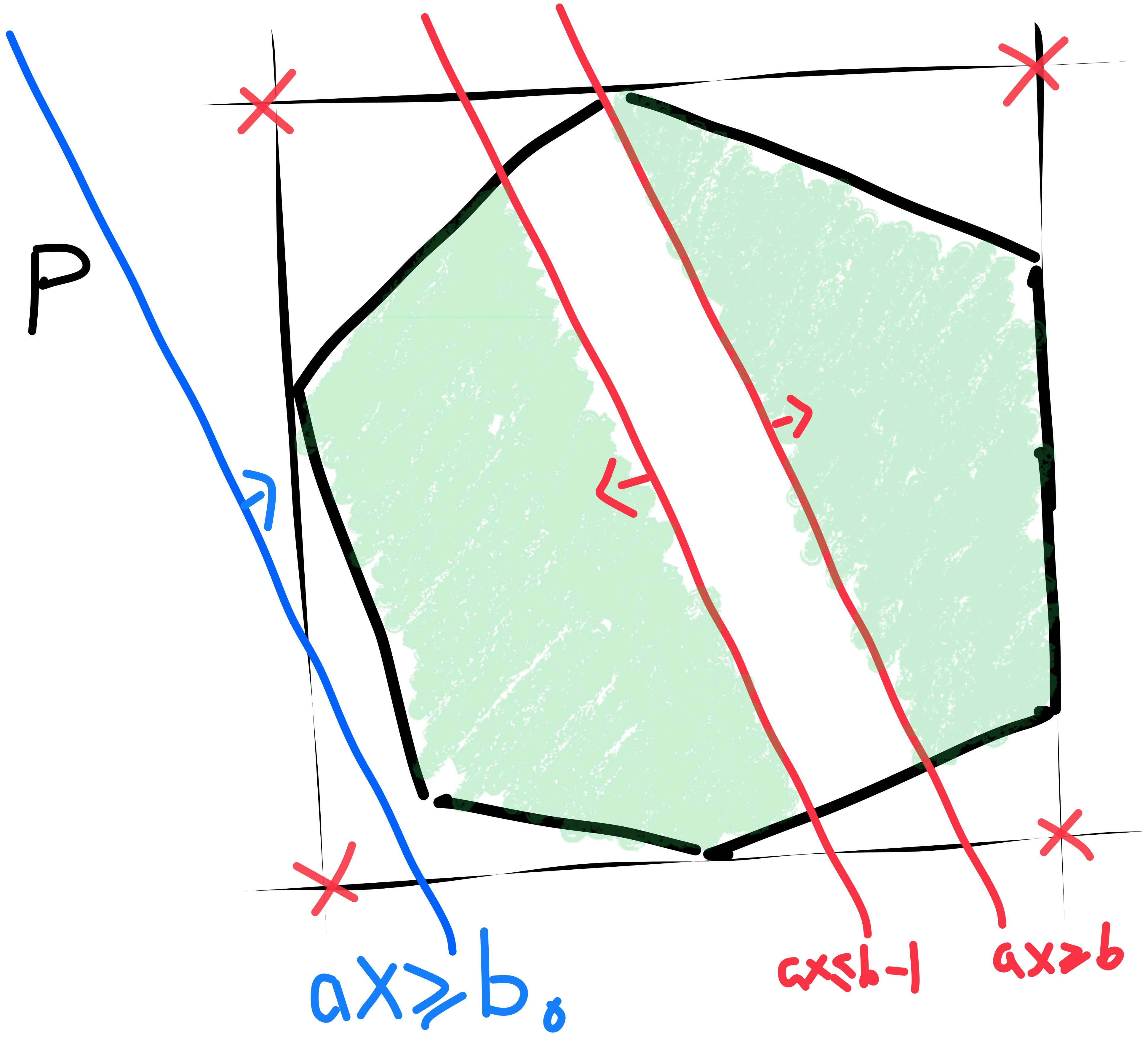
Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

Proof Sketch: Fix an SP* proof π.

π 

Suppose $|\pi_L| \leq |\pi_R|$

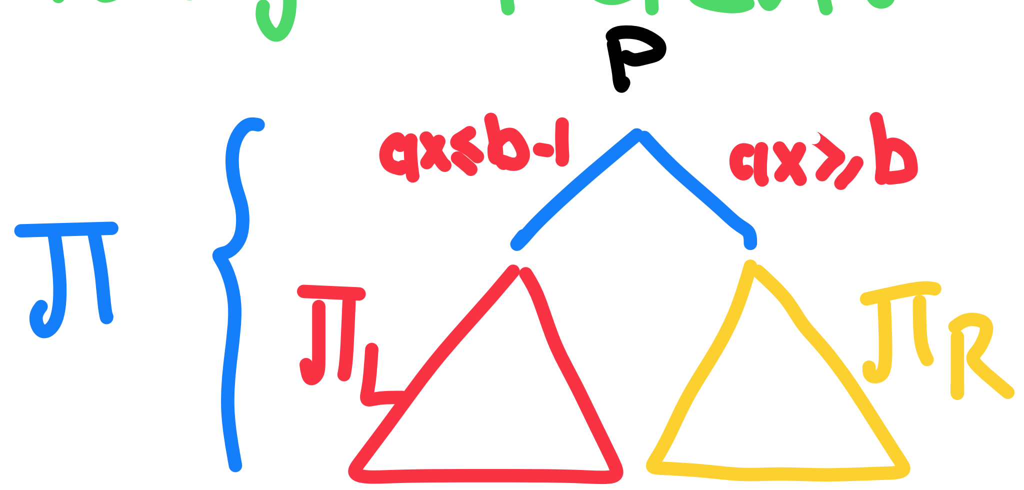
- Let b_0 be largest integer s.t. $ax \geq b_0$ valid for P



Quasi-poly Simulation of SP^*

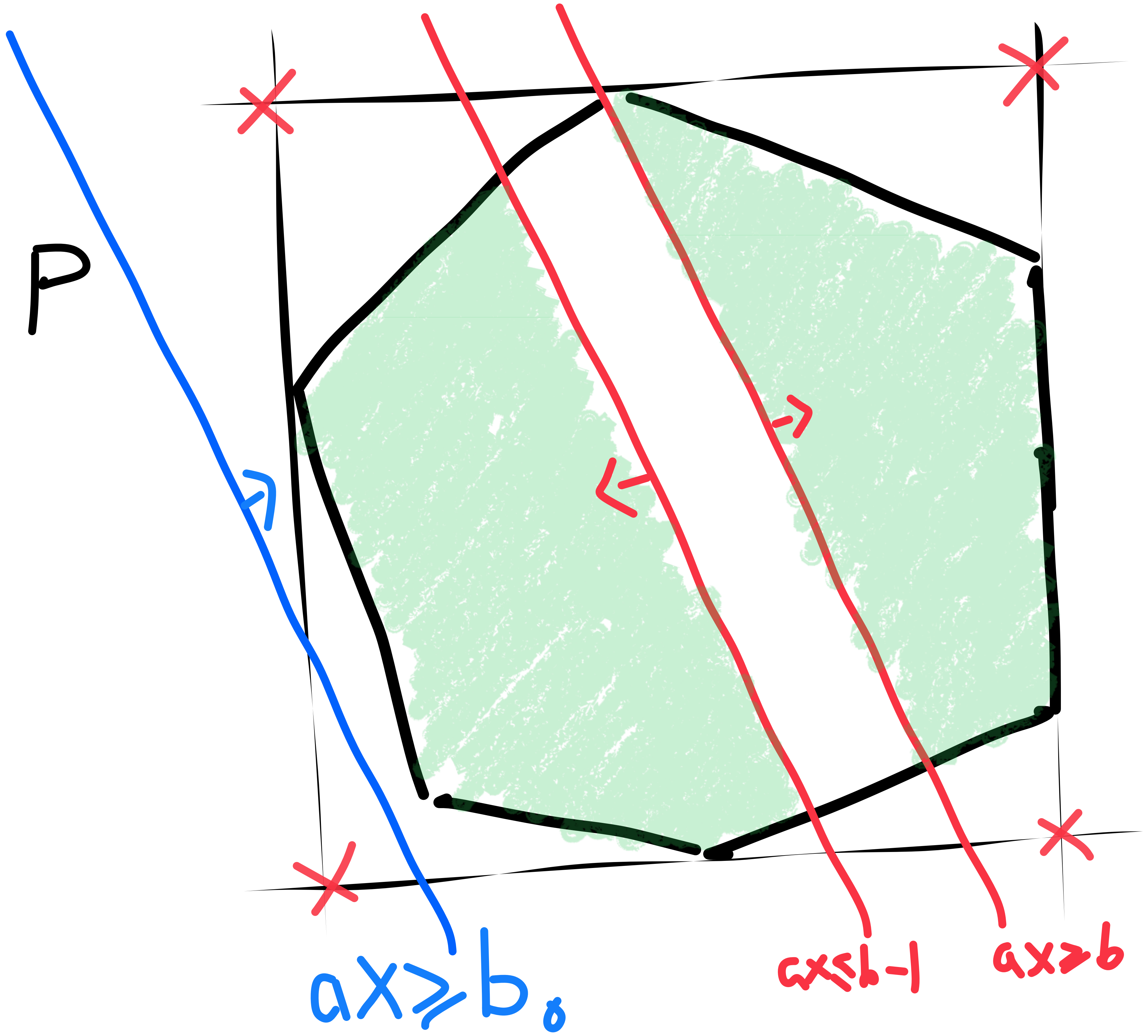
Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

Proof Sketch: Fix an SP^* proof π.



Suppose $|\pi_L| \le |\pi_R|$

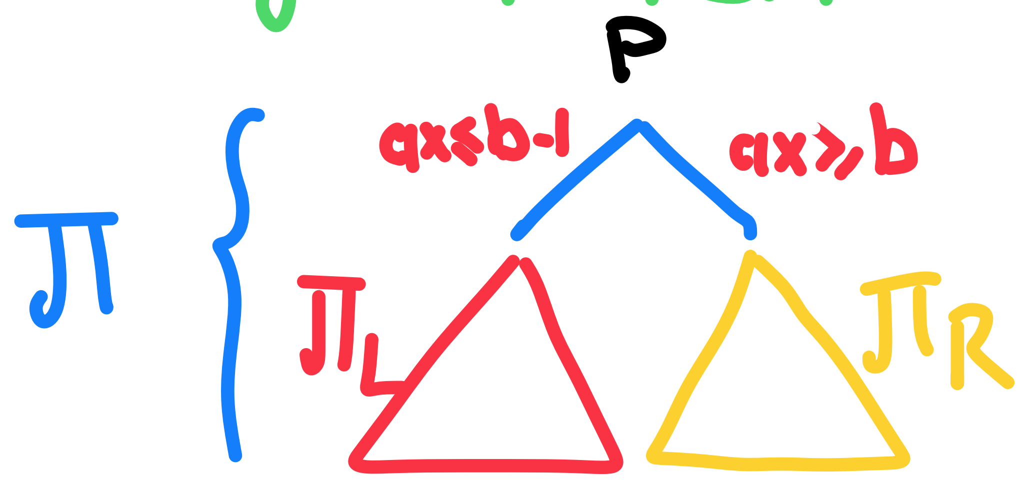
- Let b_0 be largest integer s.t. $ax \ge b_0$ valid for P
- Query $(ax \le b_0 - 1 + i,\ ax \ge b_0 + i)$ for $i = 0, \dots, b - b_0$



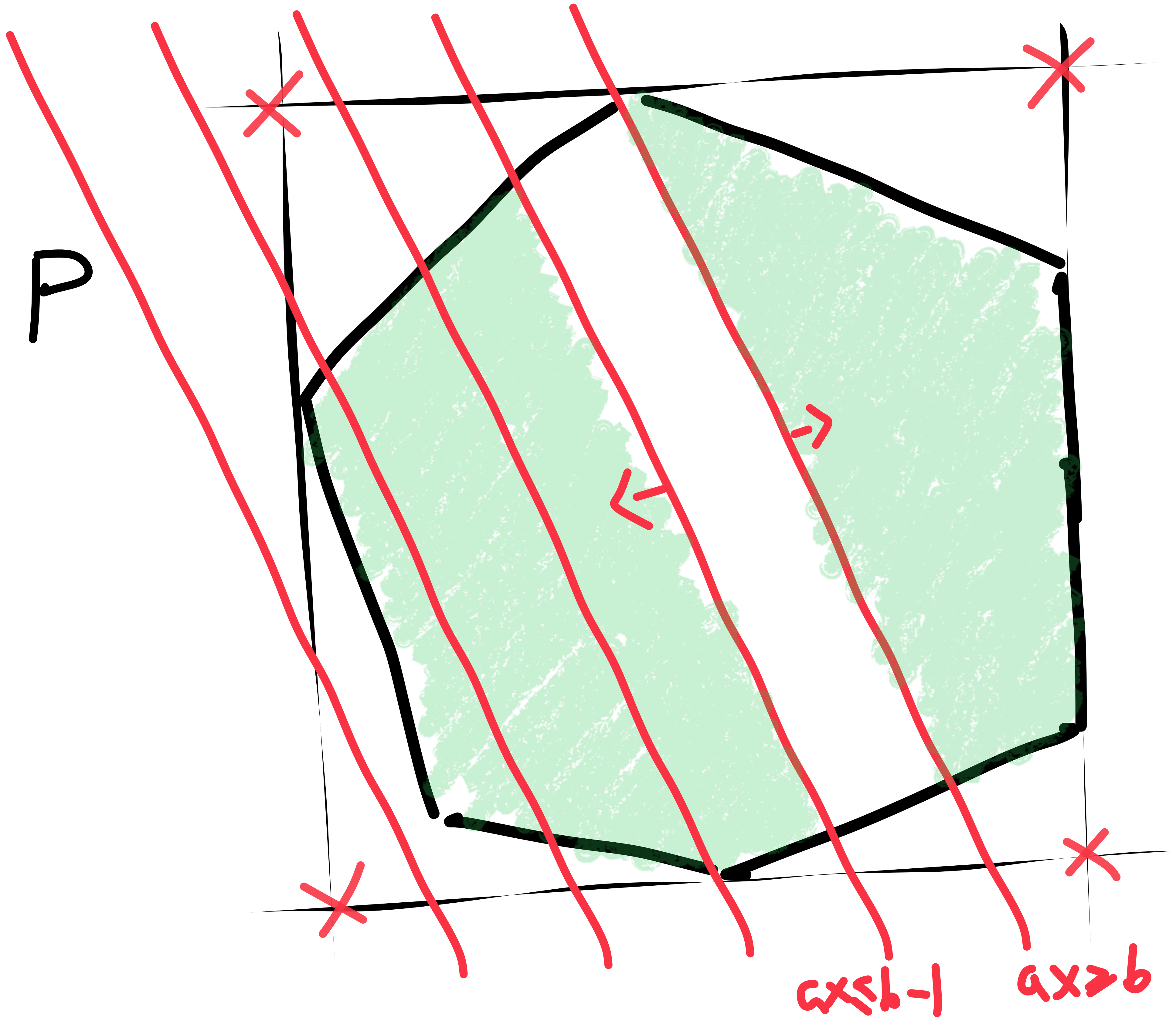
Quasi-poly Simulation of SP*

Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

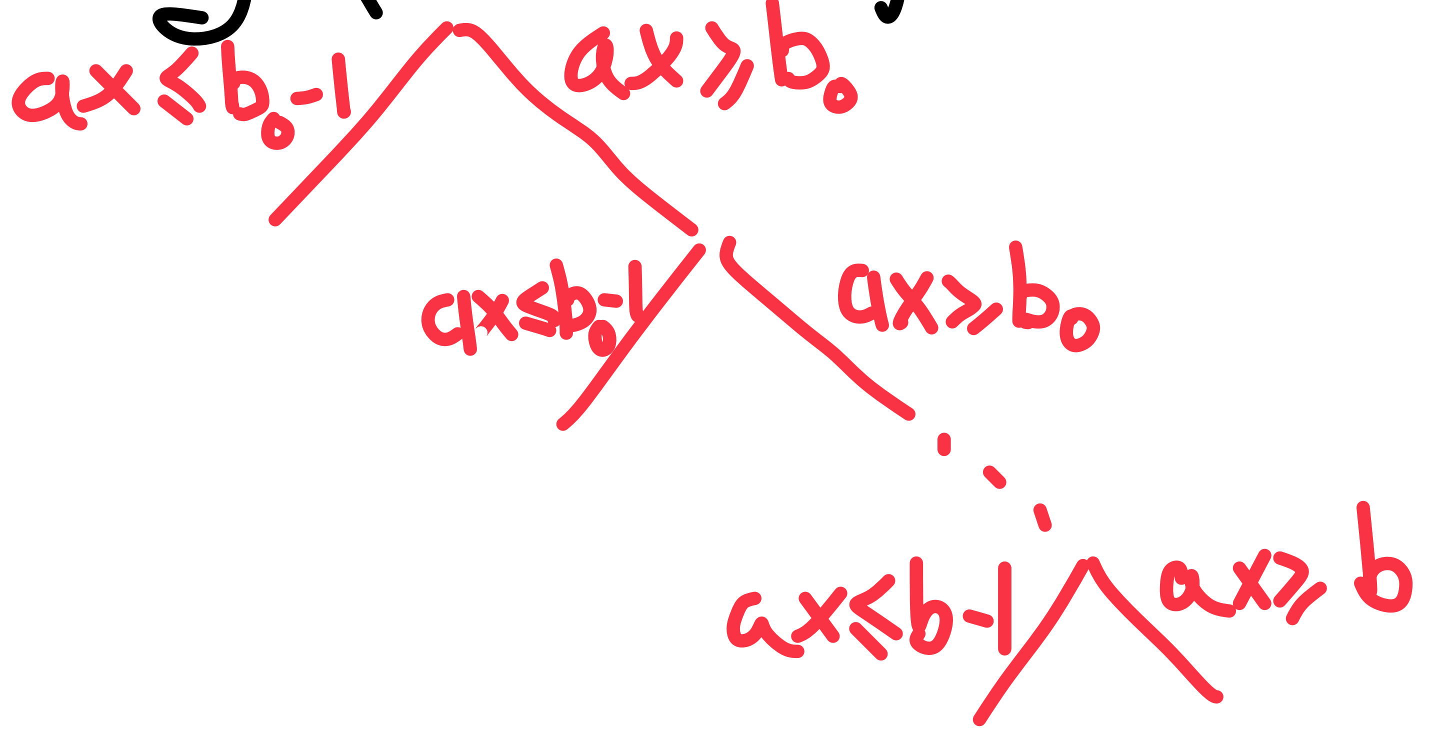
Proof Sketch: Fix an SP* proof π.



Suppose $|\pi_L| \leq |\pi_R|$



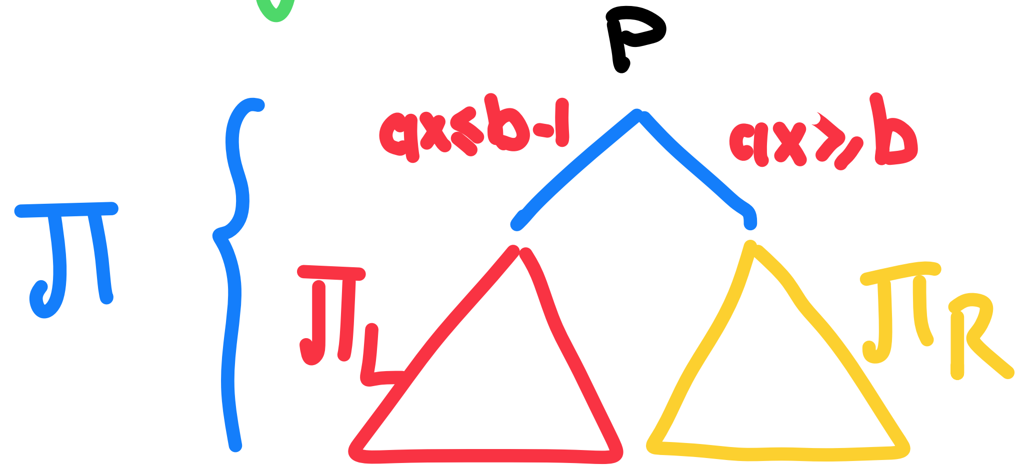
- Let b_0 be largest integer s.t. $ax \geq b_0$ valid for P
- Query $(ax \leq b_0 - 1 + i,\ ax \geq b_0 + i)$ for $i = 0, \ldots, b - b_0$



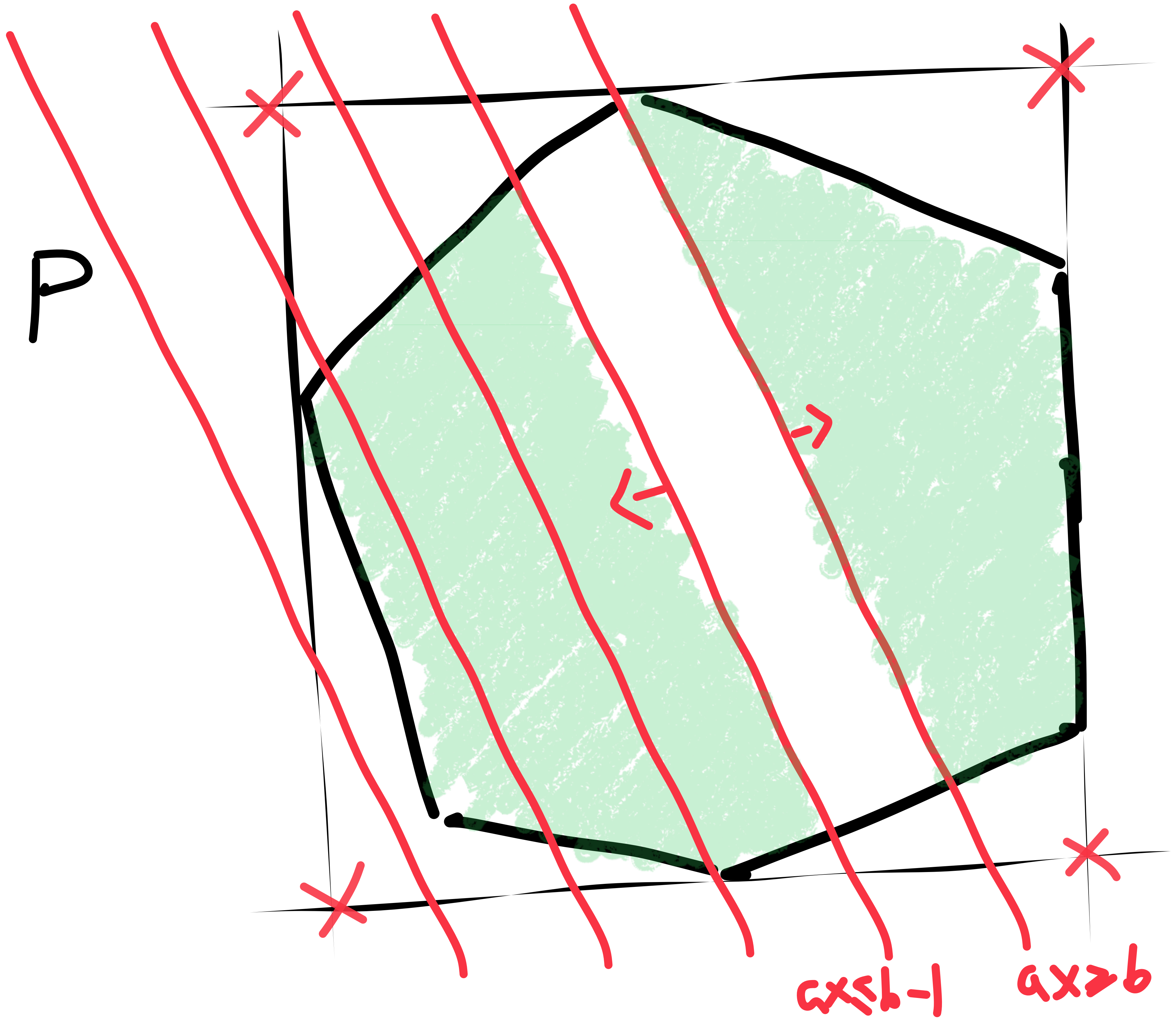
Quasi-poly Simulation of SP*

Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

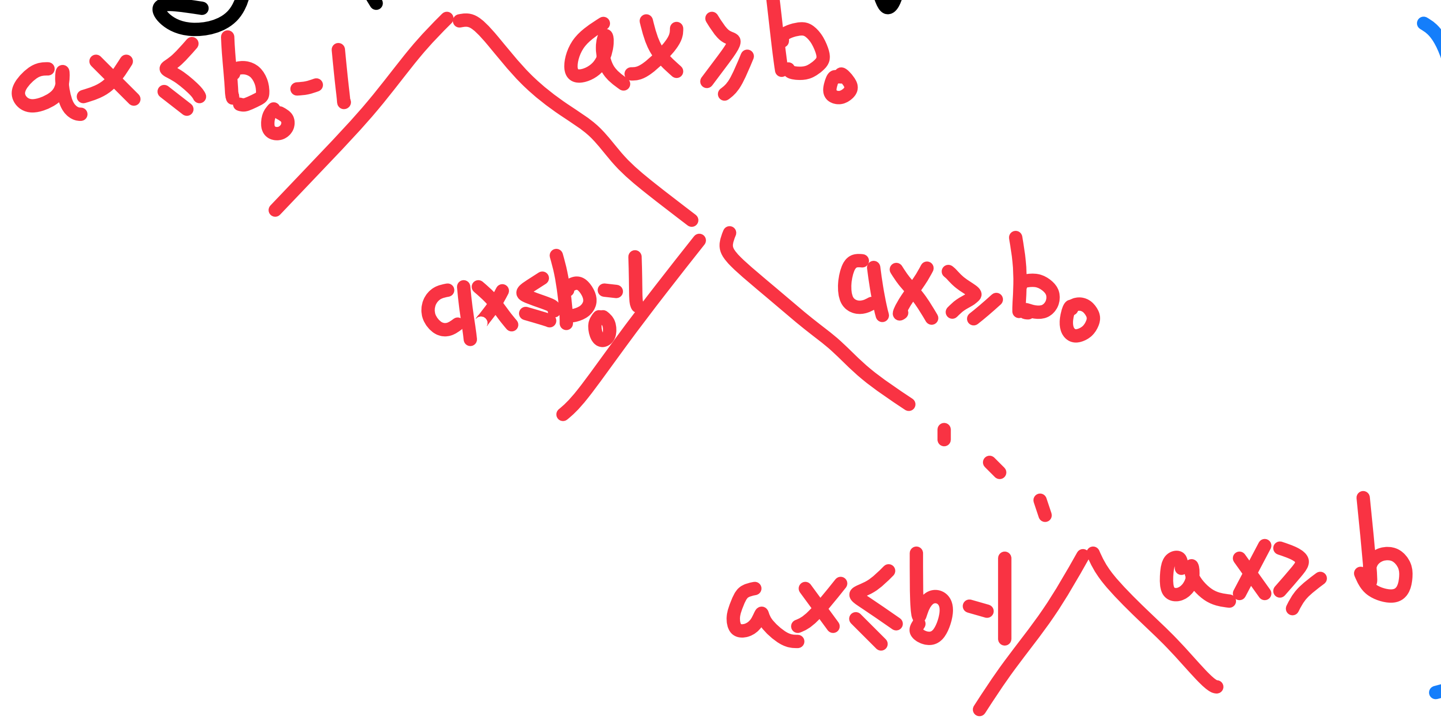
Proof Sketch: Fix an SP* proof π.



Suppose $|\pi_L| \le |\pi_R|$



- Let b_0 be largest integer s.t. $ax \ge b_0$ valid for P
- Query $(ax \le b_0 - 1 + i,\ ax \ge b_0 + i)$ for $i = 0, \ldots, b - b_0$

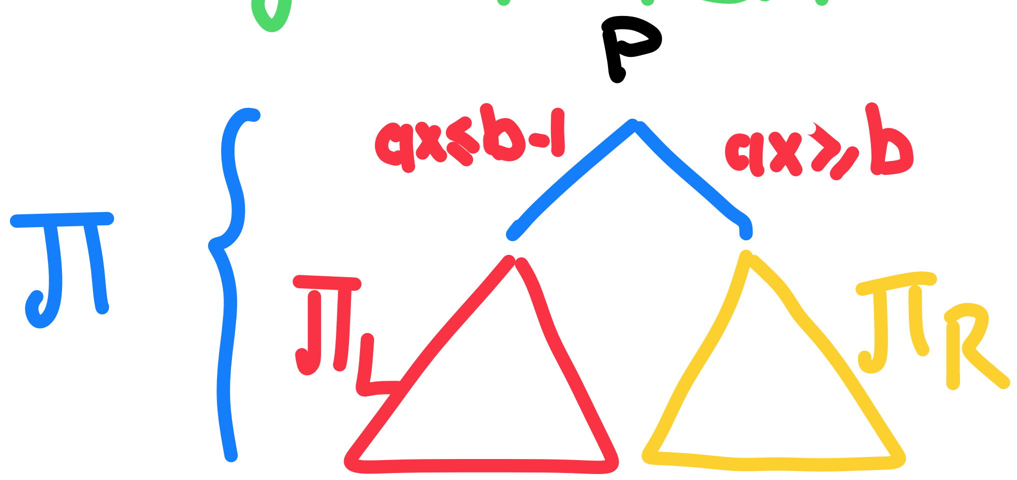


Facelike

Quasi-poly Simulation of SP*

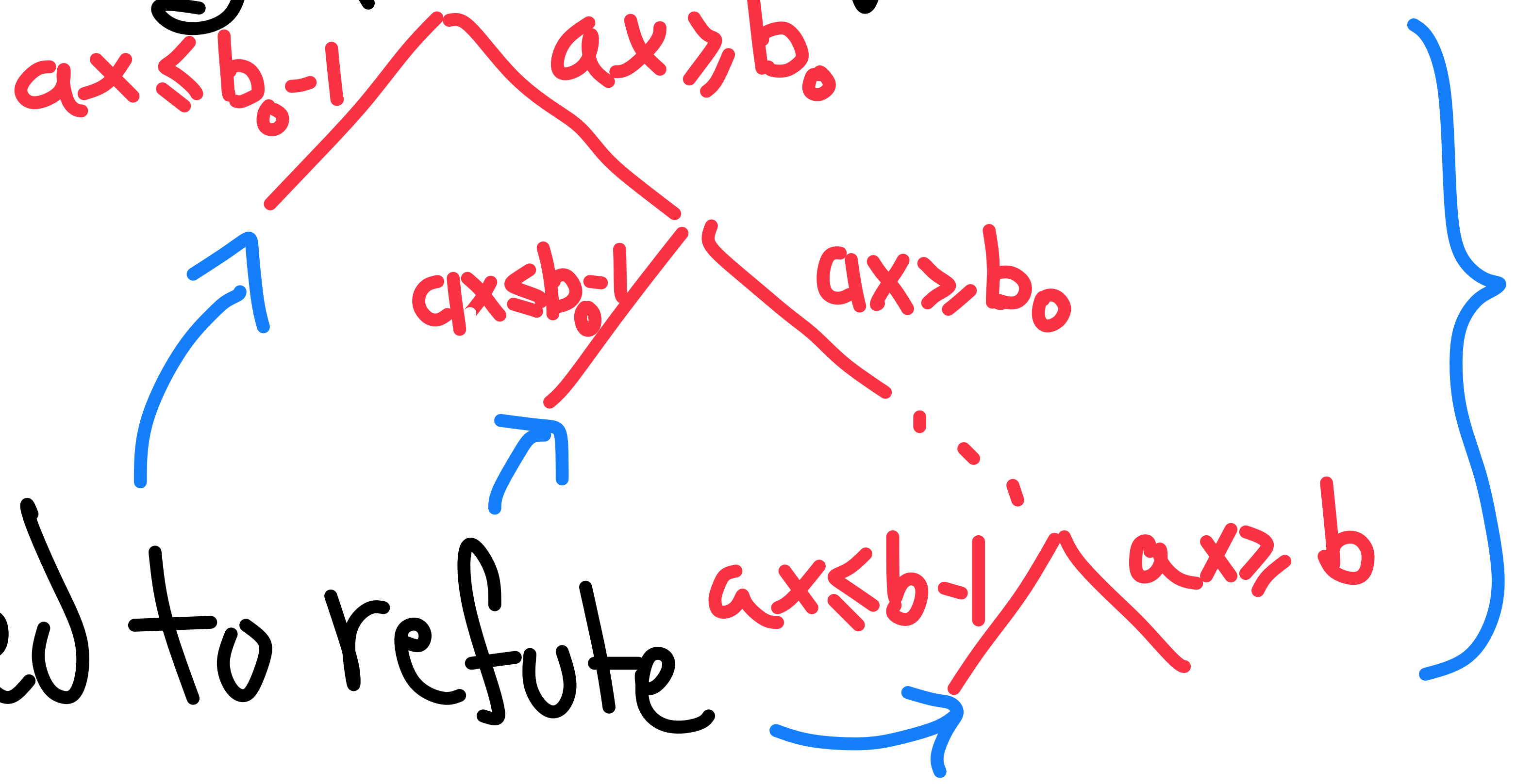
Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

Proof Sketch: Fix an SP* proof π.

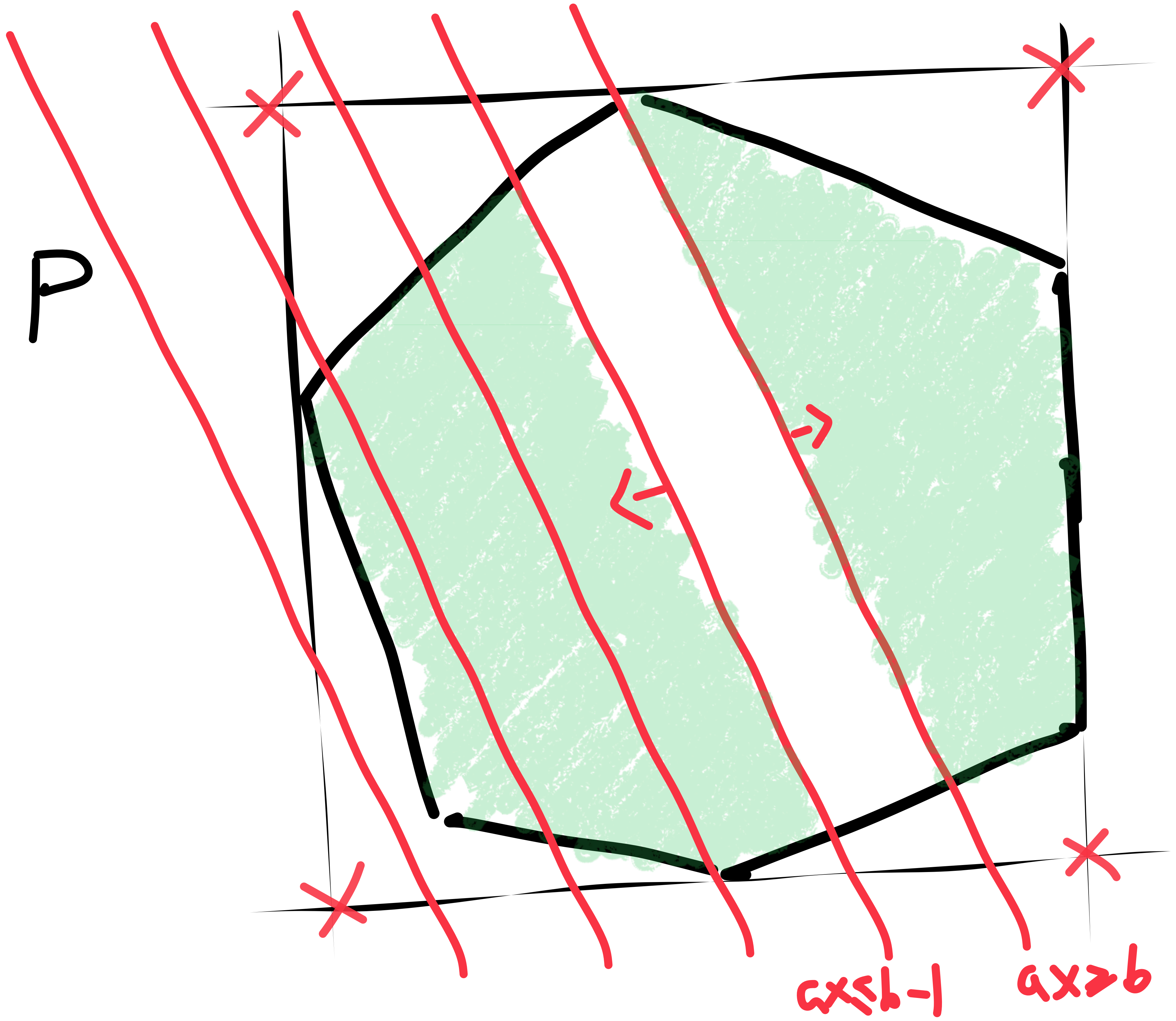


Suppose $|\pi_L| \leq |\pi_R|$

- Let b_0 be largest integer s.t. $ax \geq b_0$ valid for P
- Query $(ax \leq b_0 - 1 + i,\ ax \geq b_0 + i)$ for $i = 0, \ldots, b - b_0$



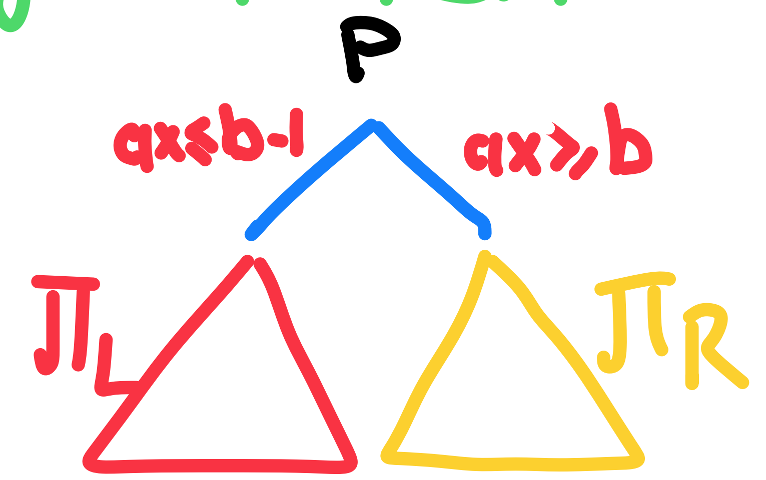
Need to refute



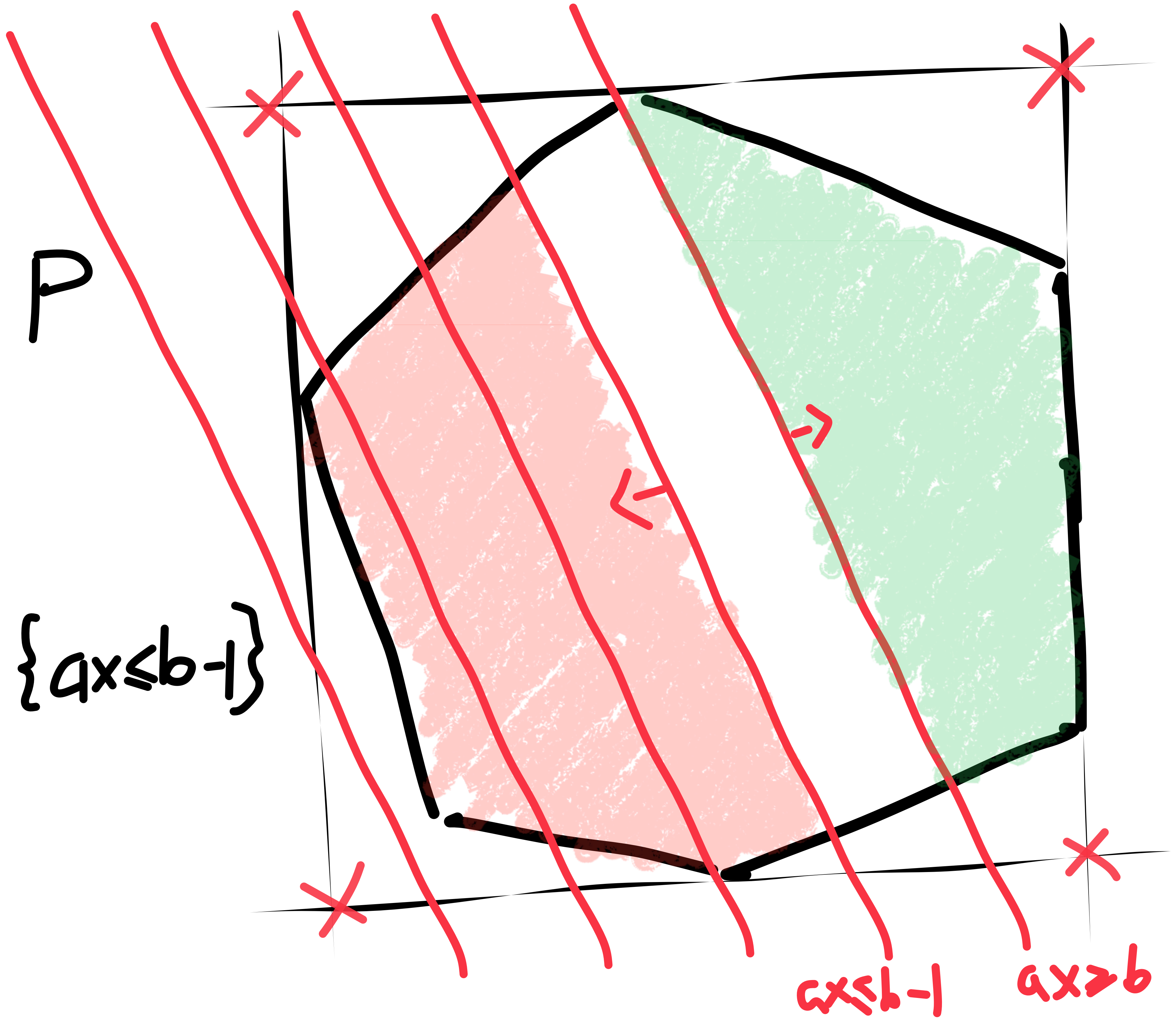
Quasi-poly Simulation of SP*

Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

Proof Sketch: Fix an SP* proof π.

π 

Suppose $|\pi_L| \leq |\pi_R|$



- Let b_0 be largest integer s.t. $ax \geq b_0$ valid for P
- Query $(ax \leq b_0 - 1 + i,\ ax \geq b_0 + i)$ for $i = 0, \ldots, b - b_0$

$ax \leq b_0 - 1$ $\quad$ $ax \geq b_0$

$ax \leq b_0 - 1$ $\quad$ $ax \geq b_0$

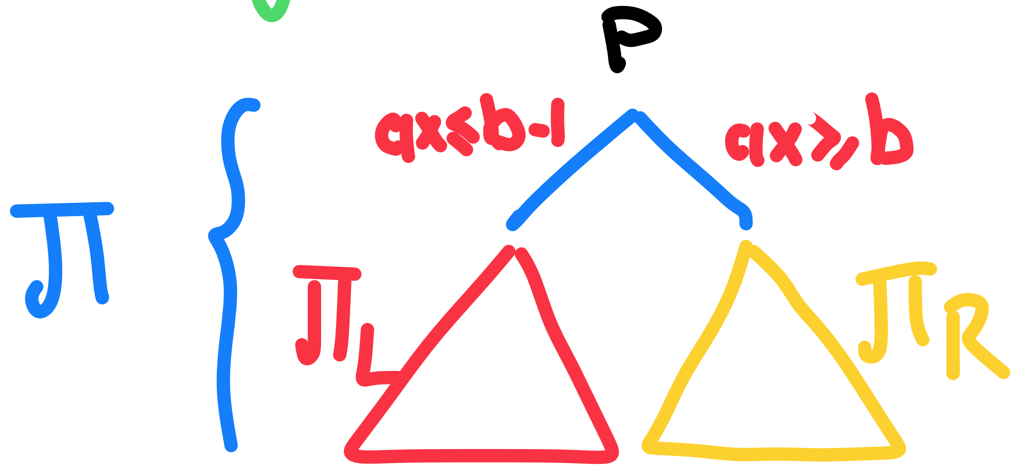
Need to refute $\quad ax \leq b - 1 \quad ax \geq b$

Observe: π_L refutes $P \cap \{ax \leq b-1\}$

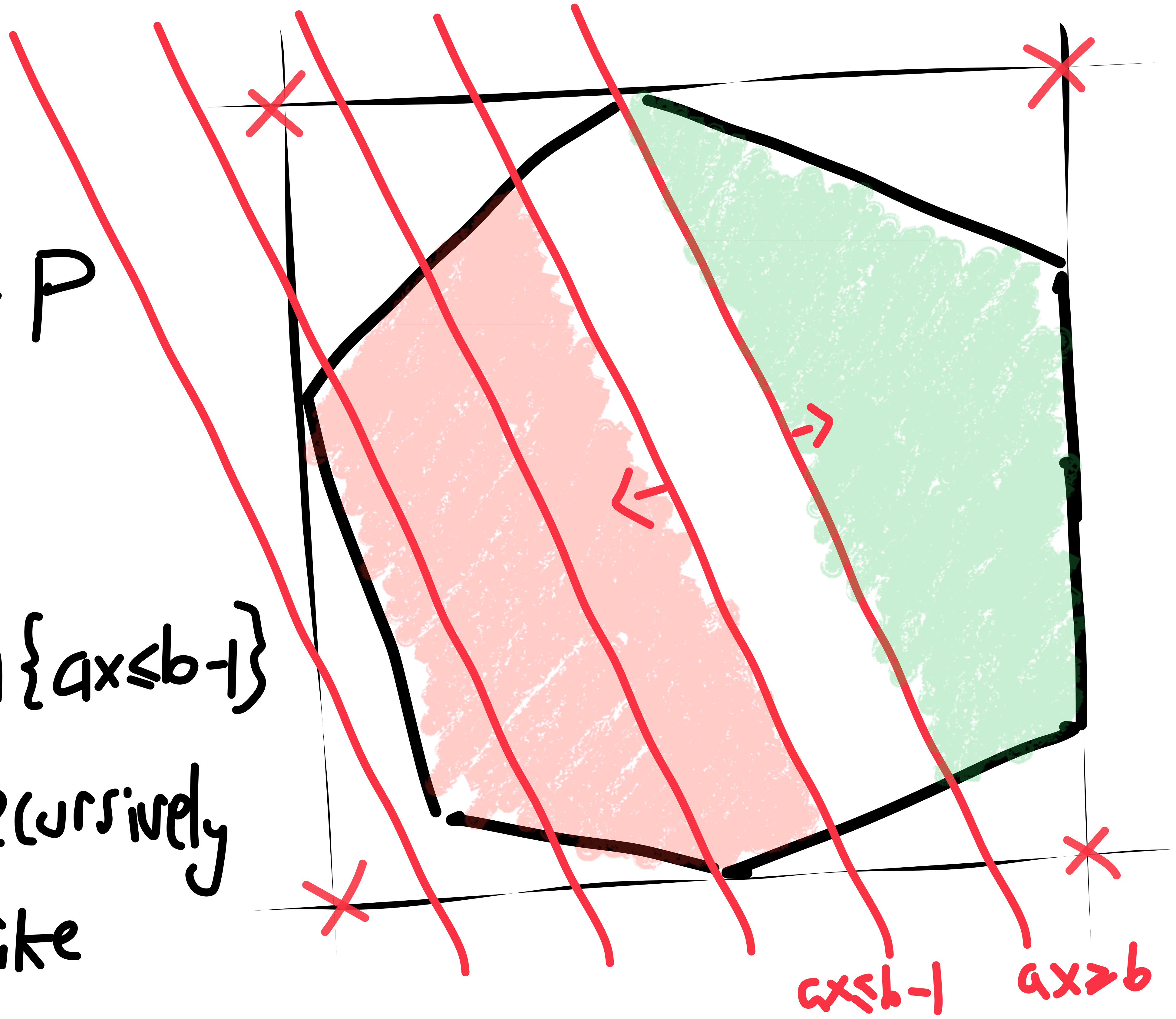
Quasi-poly Simulation of SP^*

Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

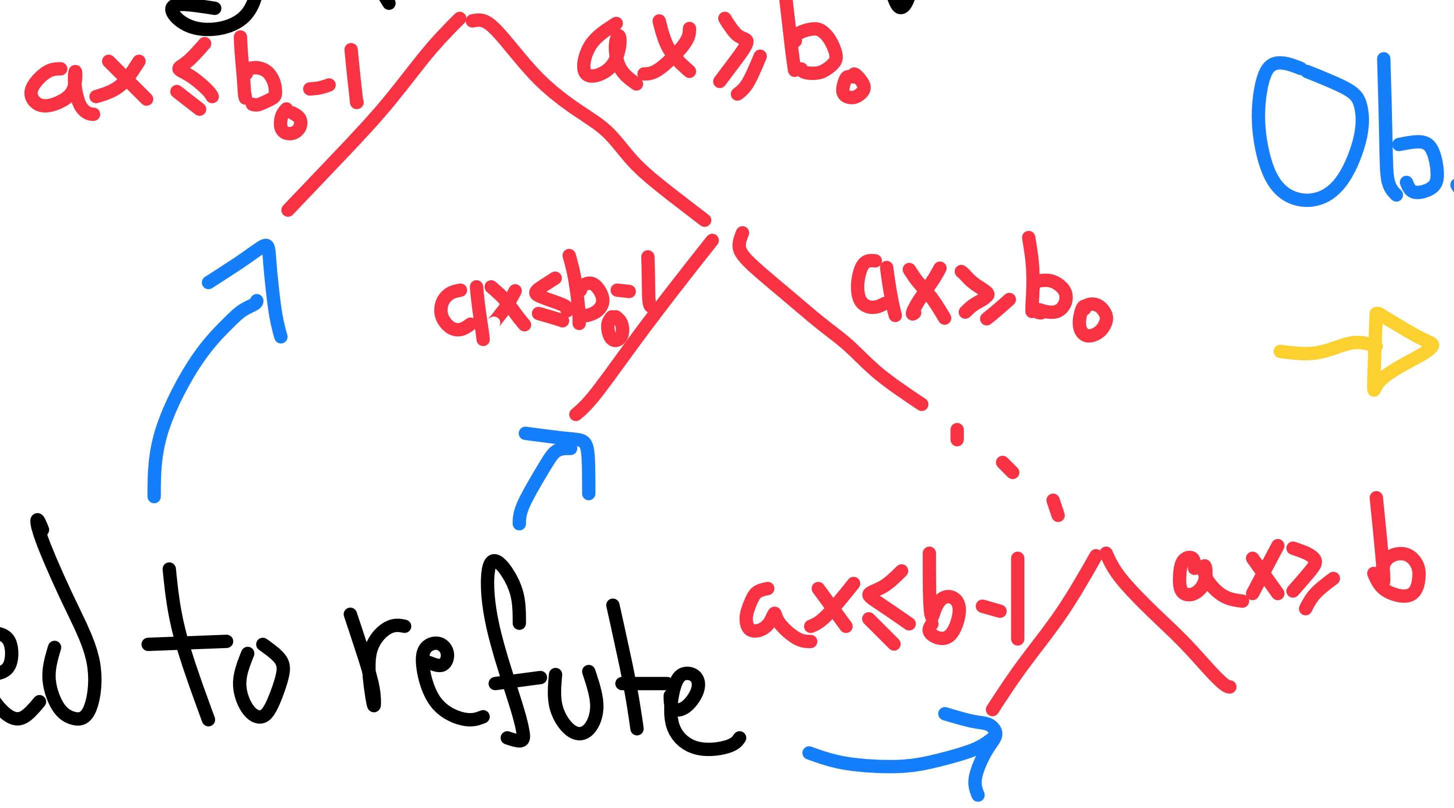
Proof Sketch: Fix an SP^* proof π.



Suppose $|\pi_L| \le |\pi_R|$



- Let b_0 be largest integer s.t. $ax \ge b_0$ valid for P
- Query $(ax \le b_0 - 1 + i,\ ax \ge b_0 + i)$ for $i = 0, \dots, b - b_0$



Need to refute

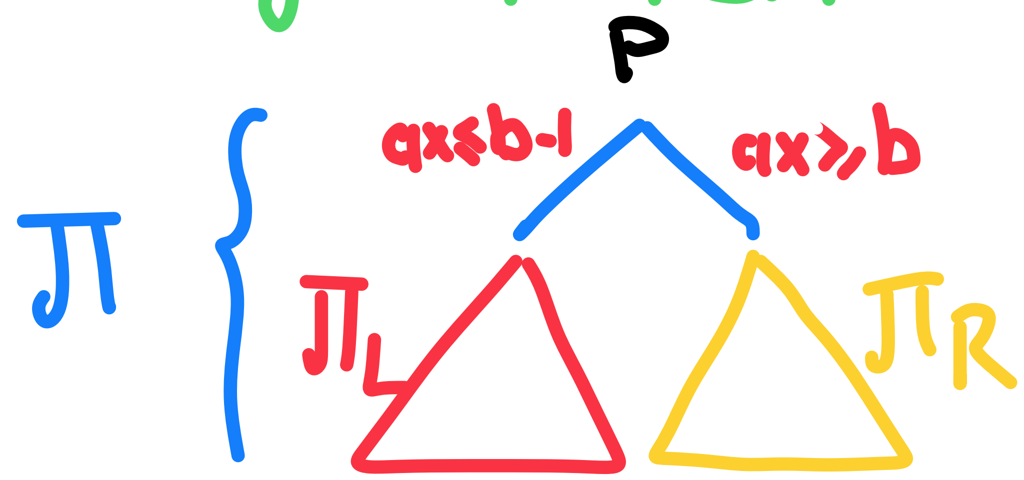
Observe: π_L refutes $P \cap \{ax \le b-1\}$

→ Let π_L' be π_L recursively converted a facelike proof

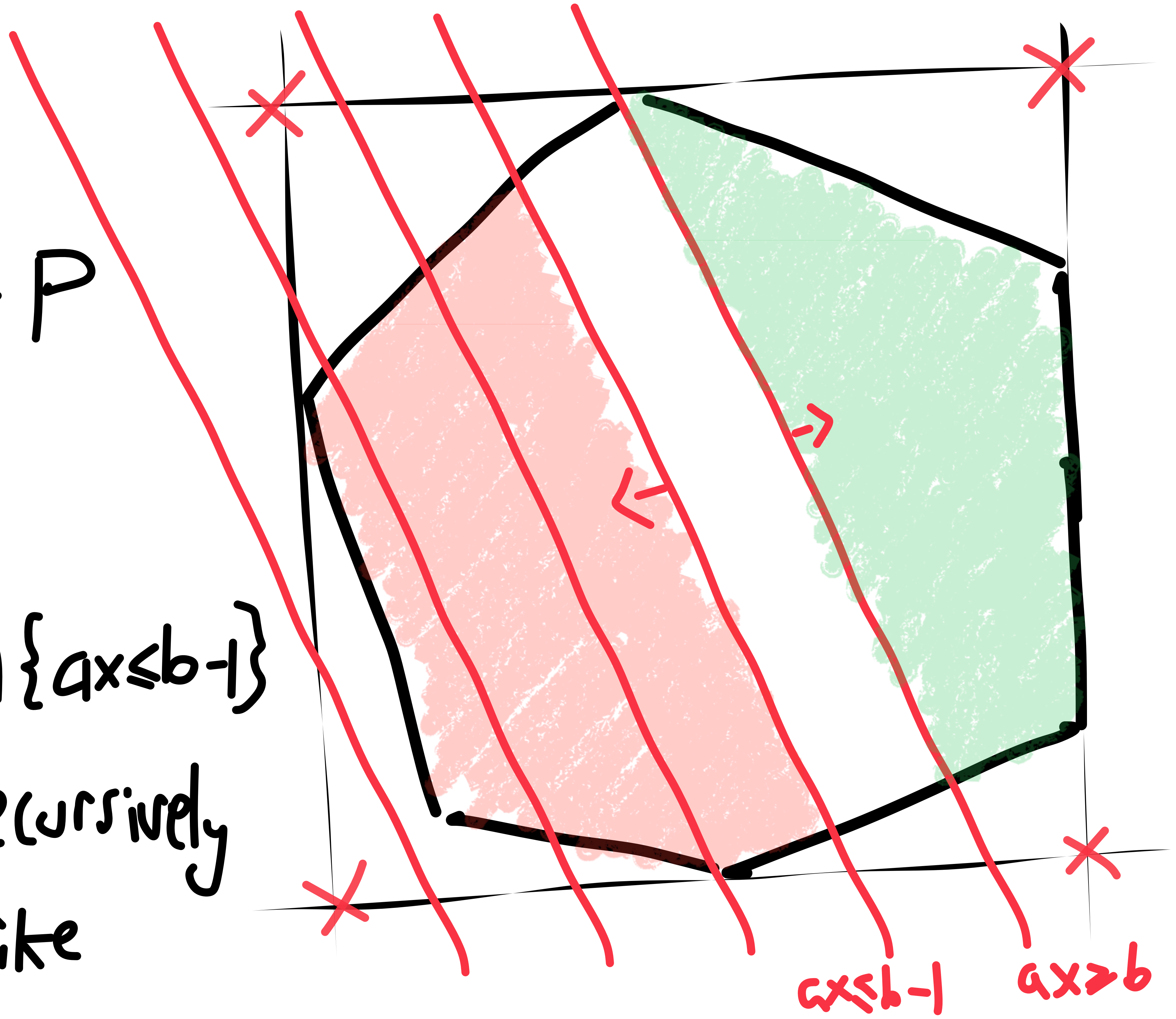
Quasi-poly Simulation of SP^*

Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

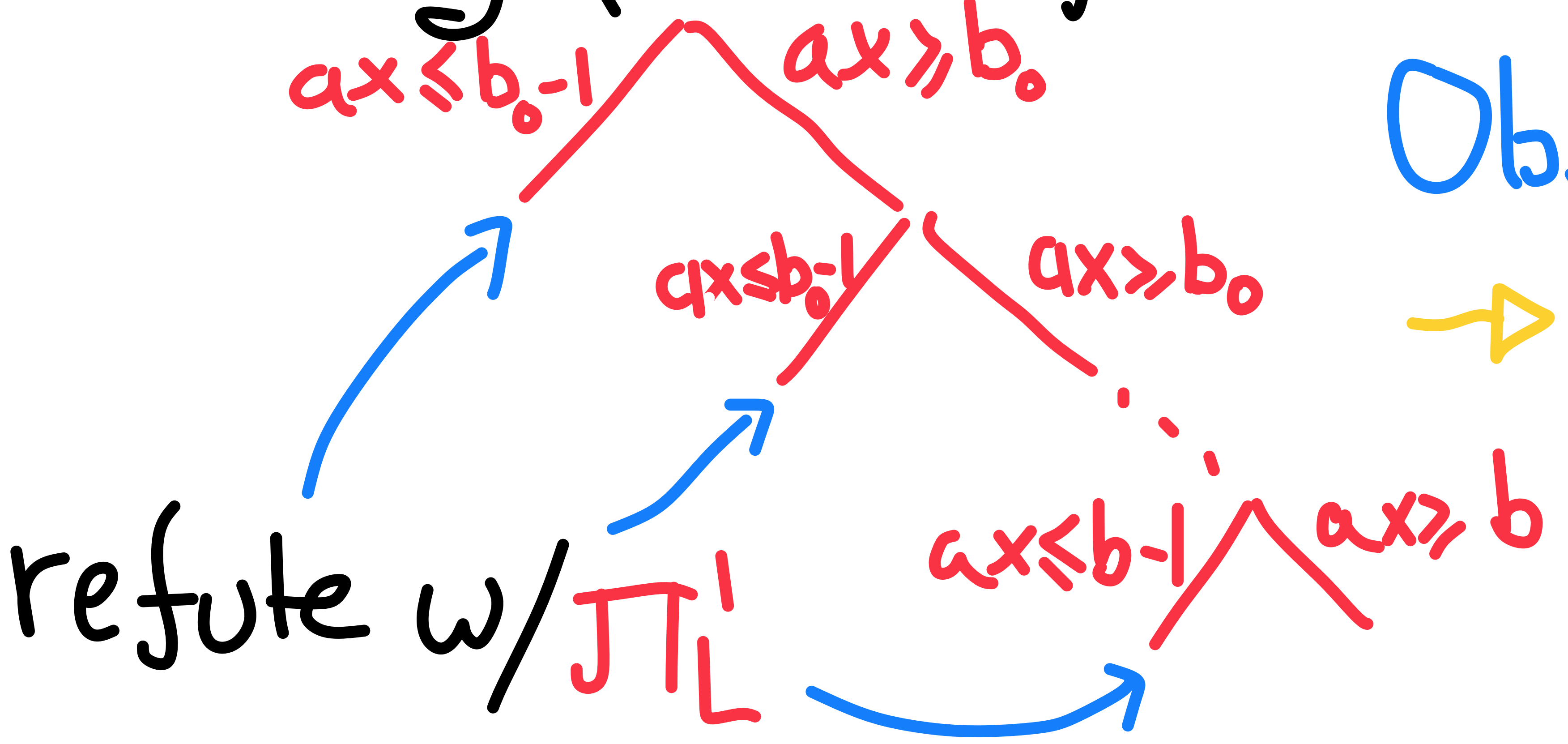
Proof Sketch: Fix an SP^* proof π.



Suppose $|\pi_L| \leq |\pi_R|$



- Let b_0 be largest integer s.t. $ax \geq b_0$ valid for P
- Query $(ax \leq b_0 - 1 + i,\ ax \geq b_0 + i)$ for $i = 0, \ldots, b - b_0$



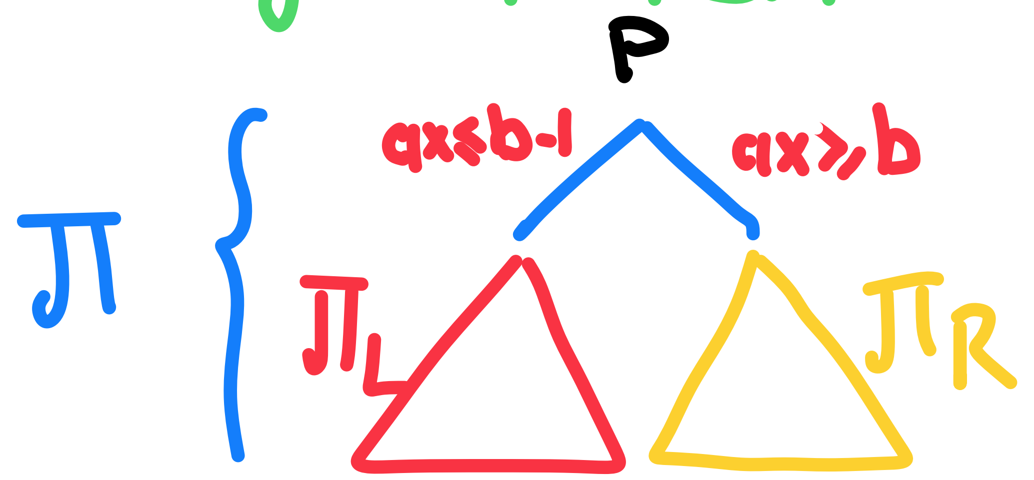
Observe: π_L refutes $P \cap \{ax \leq b-1\}$

→ Let π_L' be π_L recursively converted a facelike proof

Quasi-poly Simulation of SP*

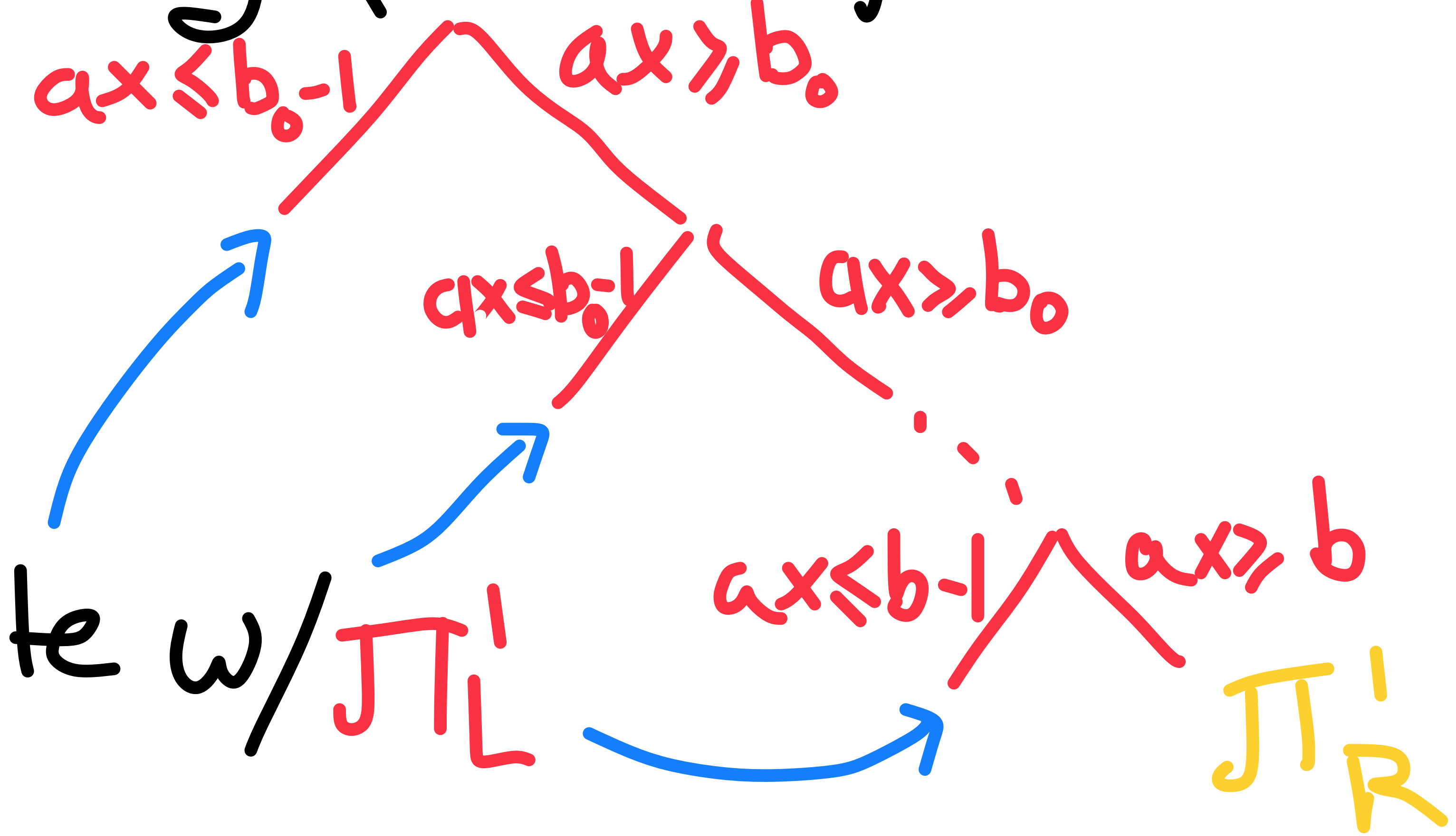
Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

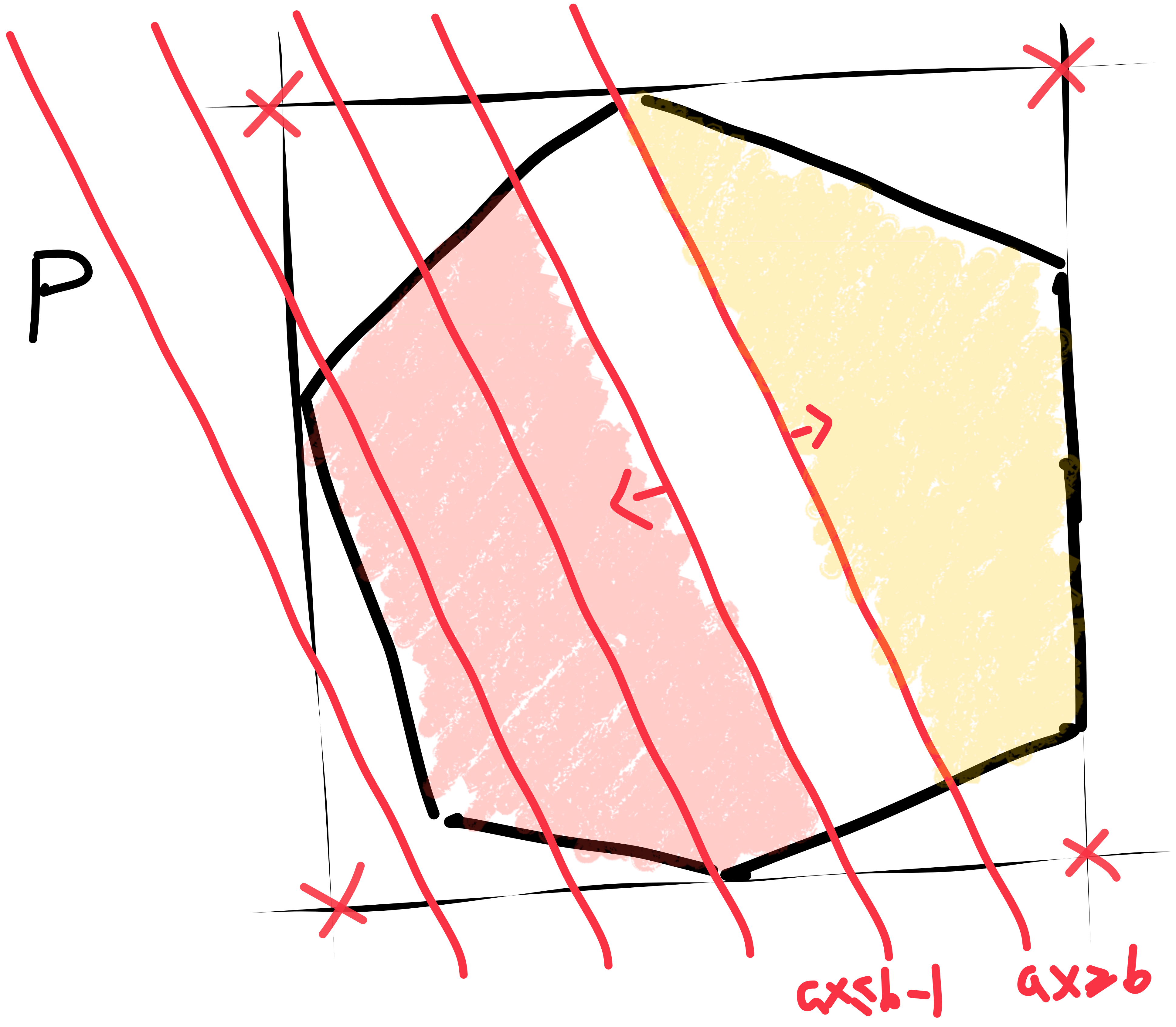
Proof Sketch: Fix an SP* proof π.



Suppose $|\pi_L| \leq |\pi_R|$

- Let b_0 be largest integer s.t. $ax \geq b_0$ valid for P
- Query $(ax \leq b_0 - 1 + i,\ ax \geq b_0 + i)$ for $i = 0, \ldots, b - b_0$

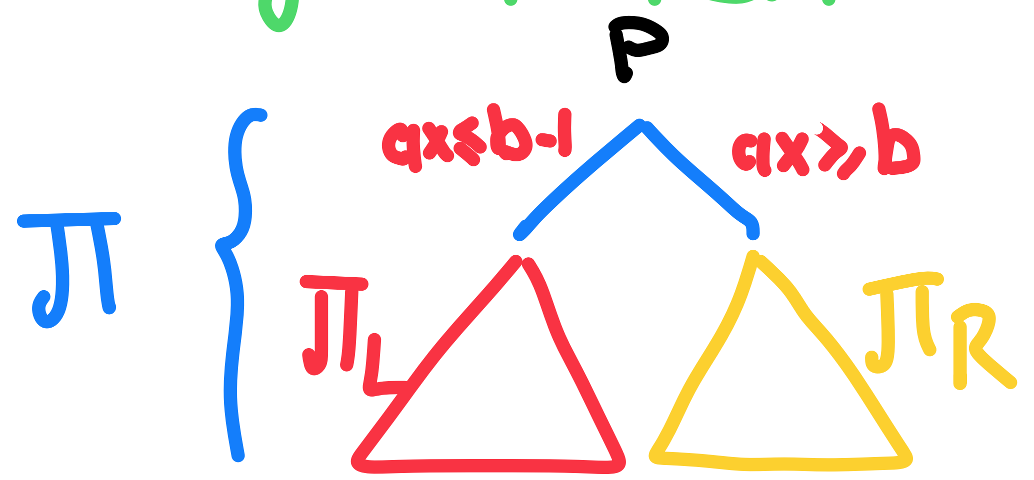




Quasi-poly Simulation of SP*

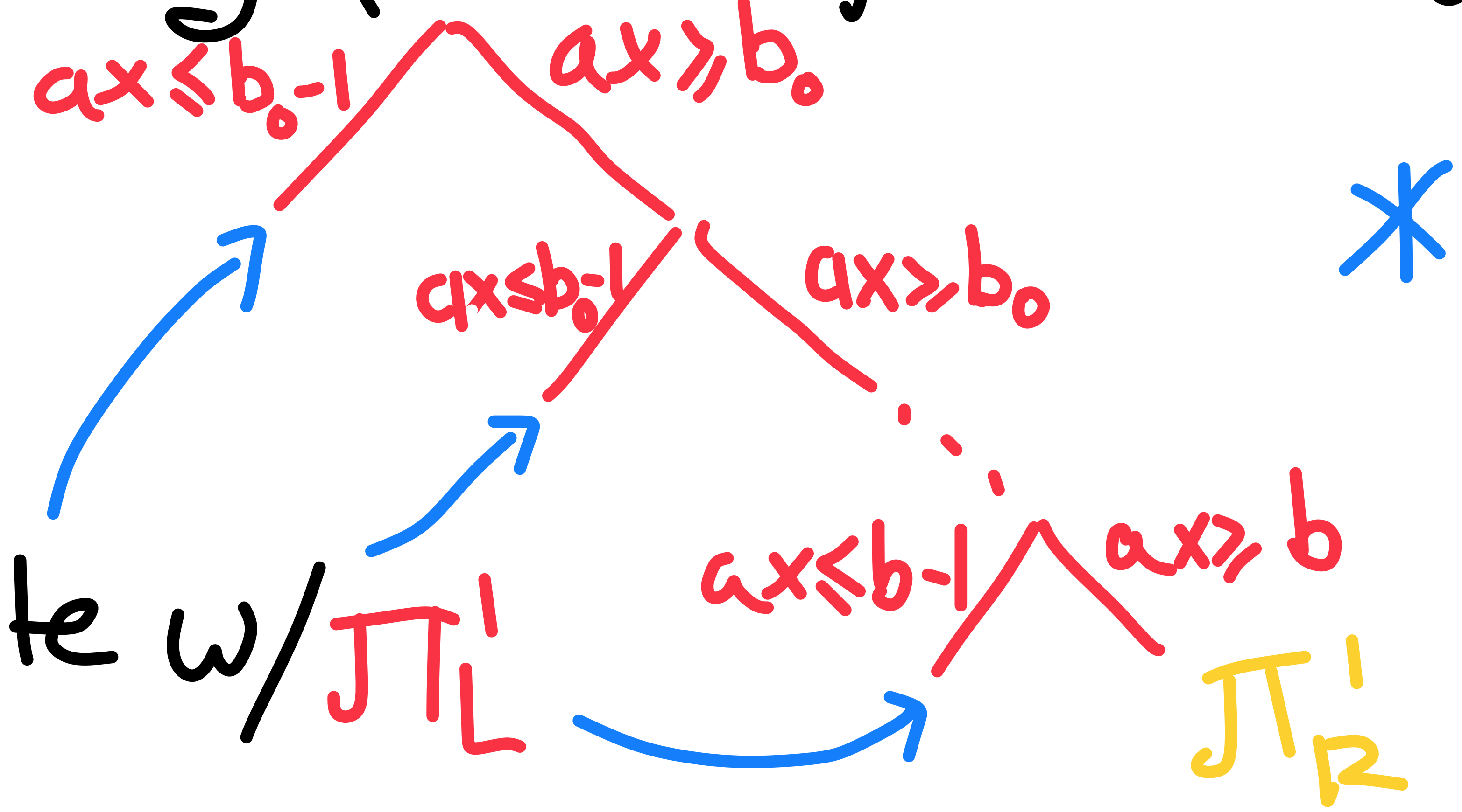
Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

Proof Sketch: Fix an SP* proof π.

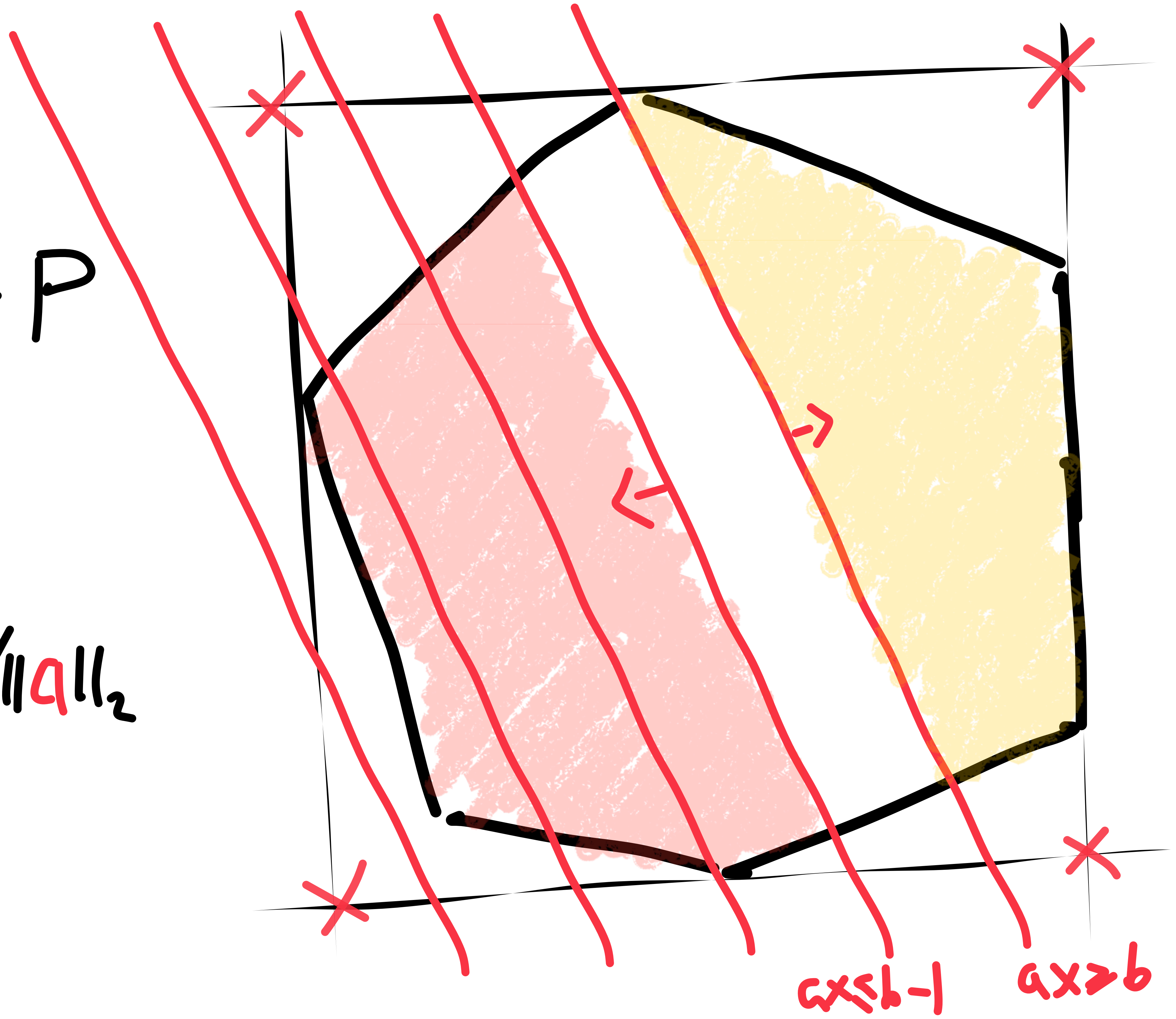


Suppose $|\pi_L| \leq |\pi_R|$

- Let b_0 be largest integer s.t. $ax \geq b_0$ valid for P
- Query $(ax \leq b_0 - 1 + i,\ ax \geq b_0 + i)$ for $i = 0, \dots, b - b_0$



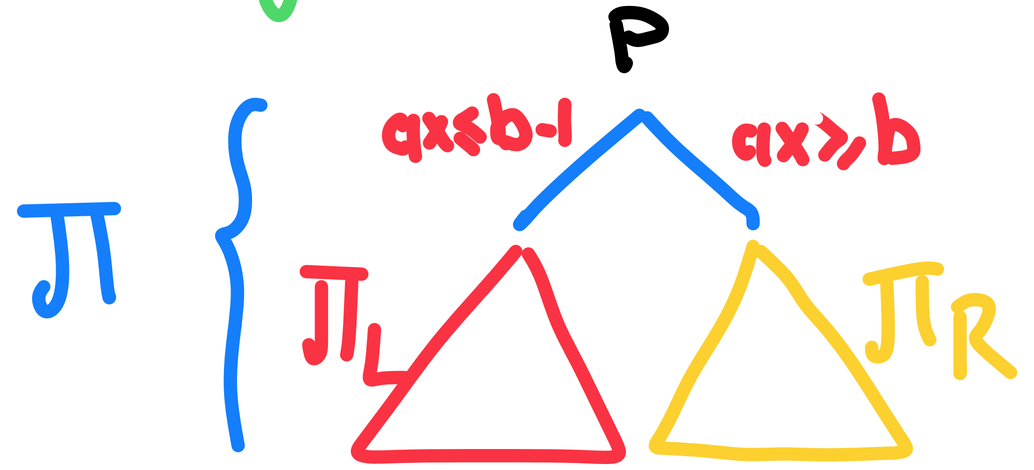
$*$ width of slab $= 1/\|a\|_2$



Quasi-poly Simulation of SP*

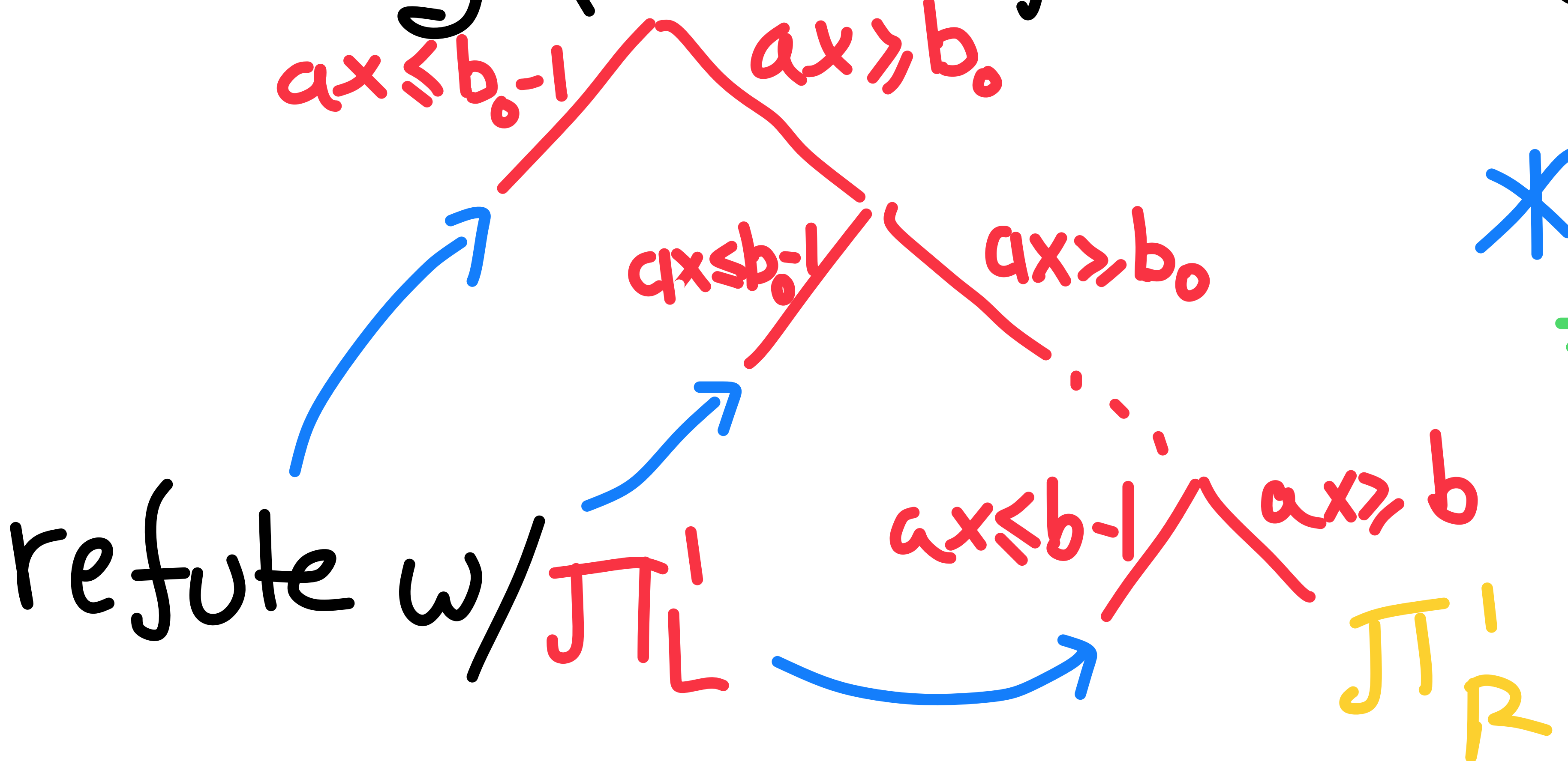
Thm: Any SP proof of size s implies a size $s(cn)^{\log s}$ Facelike proof

Proof Sketch: Fix an SP* proof π.



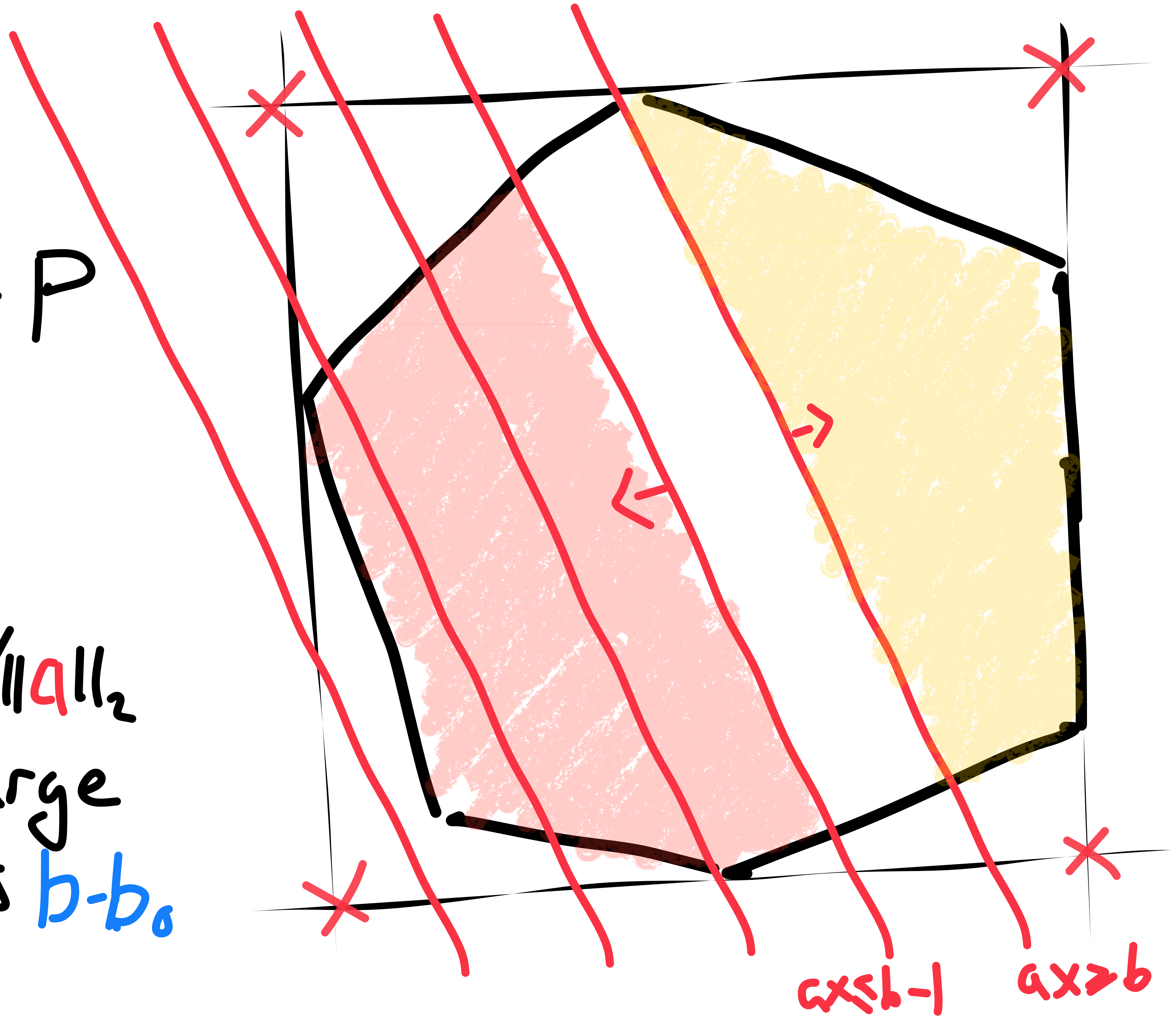
Suppose $|\pi_L| \leq |\pi_R|$

- Let b_0 be largest integer s.t. $ax \geq b_0$ valid for P
- Query $(ax \leq b_0 - 1 + i,\ ax \geq b_0 + i)$ for $i = 0, \ldots, b - b_0$



* width of slab $= 1/\|a\|_2$

$\Rightarrow$ if $\|a\|_2$ not large then neither is $b - b_0$



Stabbing Planes vs Cutting Planes

Thm: Every SP^* proof can be quasipolynomially translated into CP

Cor: $\exp(n^\varepsilon)$ lower bounds for SP^*

Stabbing Planes vs Cutting Planes

Thm: Every SP^* proof can be quasipolynomially translated into CP

Cor: $\exp(n^{\varepsilon})$ lower bounds for SP^* [Clique-Color [P97], Random CNFs [HP17, FPPR17], Lifted Formulas [GGKS18]]

Stabbing Planes vs Cutting Planes

Thm: Every SP^* proof can be quasipolynomially translated into CP

Cor: $\exp(n^{\varepsilon})$ lower bounds for SP^* [Clique-Color [P97], Random CNFs [HP17, FPPR17], Lifted Formulas [GGKS18]]

Cor: SP^* cannot be automated unless P=NP

Stabbing Planes vs Cutting Planes

Thm: Every SP^* proof can be quasipolynomially translated into CP

Cor: $\exp(n^{\varepsilon})$ lower bounds for SP^* [Clique-Color [P97], Random CNFs [HP17, FPPR17], Lifted Formulas [GGKS18]]

Cor: SP^* cannot be automated unless P=NP [by inspecting [GKMP20]]

Stabbing Planes vs Cutting Planes

Thm: Every SP^* proof can be quasipolynomially translated into CP

Cor: $\exp(n^{\varepsilon})$ lower bounds for SP^* [Clique-Color [P97], Random CNFs [HP17, FPPR17], Lifted Formulas [GGKS18]]

Cor: SP^* cannot be automated unless P=NP [by inspecting [GKMP20]]

Cor: CP has quasipolynomial size refutations of any unsat system of linear equations over a finite field

Refuting Systems of Equations in CP

Cor: CP has quasipolynomial size refutations of any unsat system of linear equations over a finite field

Step1. A general algorithm for refuting systems of equations

Refuting Systems of Equations in CP

Cor: CP has quasipolynomial size refutations of any unsat System of linear equations over a finite field

Step1. A general algorithm for refuting systems of equations
Step2. Implement in SP

Refuting Systems of Equations in CP

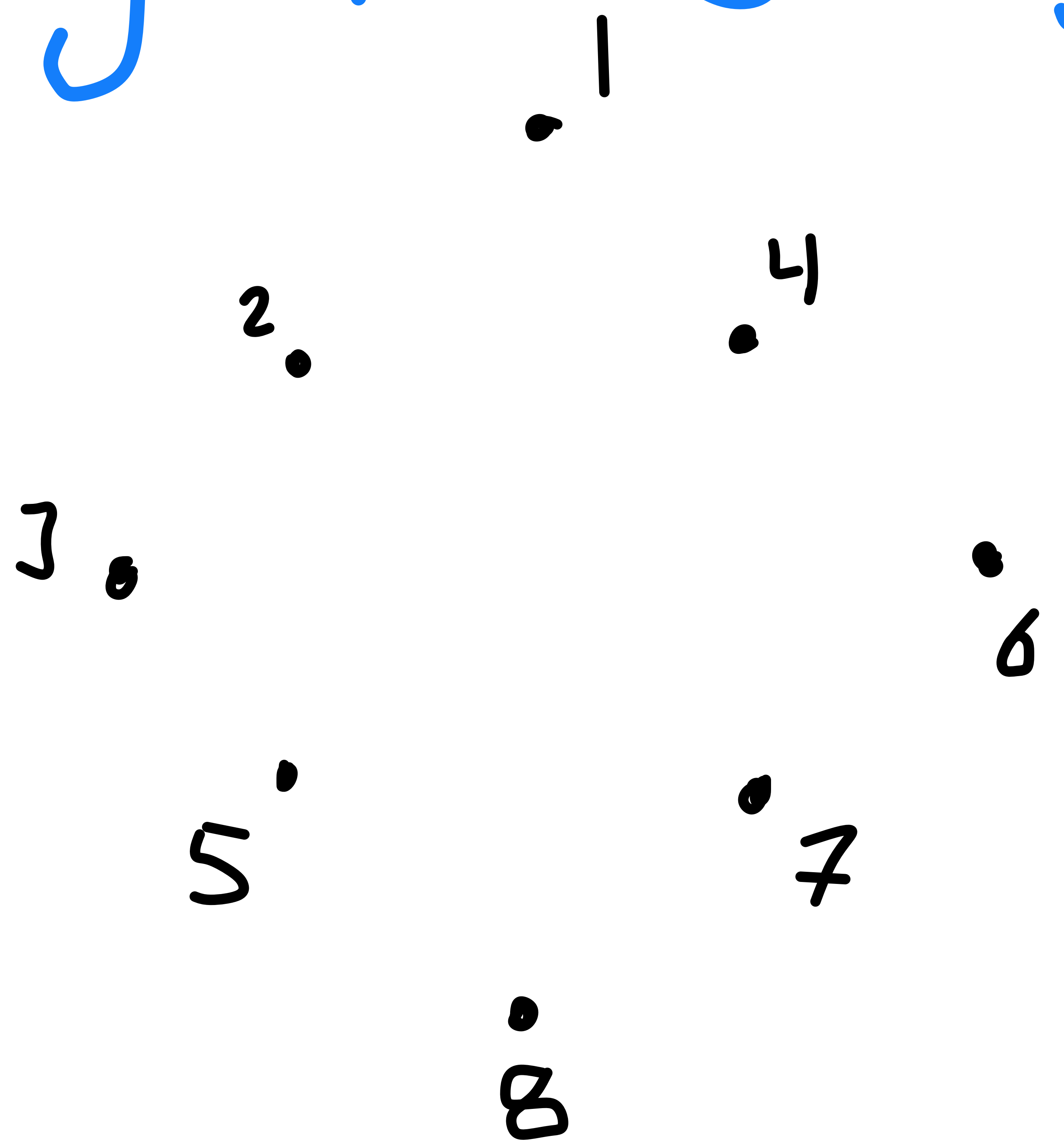
Cor: CP has quasipolynomial size refutations of any unsat system of linear equations over a finite field

Step 1. A general algorithm for refuting systems of equations
Step 2. Implement in SP

Ex $\mathbb{F}_2$ linear equations

Refuting Systems of Equations

$$x_1 + x_2 + x_4 = 1$$
$$x_1 + x_3 + x_5 = 1$$
$$x_2 + x_3 + x_7 = 1$$
$$x_4 + x_6 + x_7 = 0$$
$$x_5 + x_7 + x_8 = 0$$
$$x_6 + x_7 + x_8 = 0$$



Ex $\mathbb{F}_2$ linear equations

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

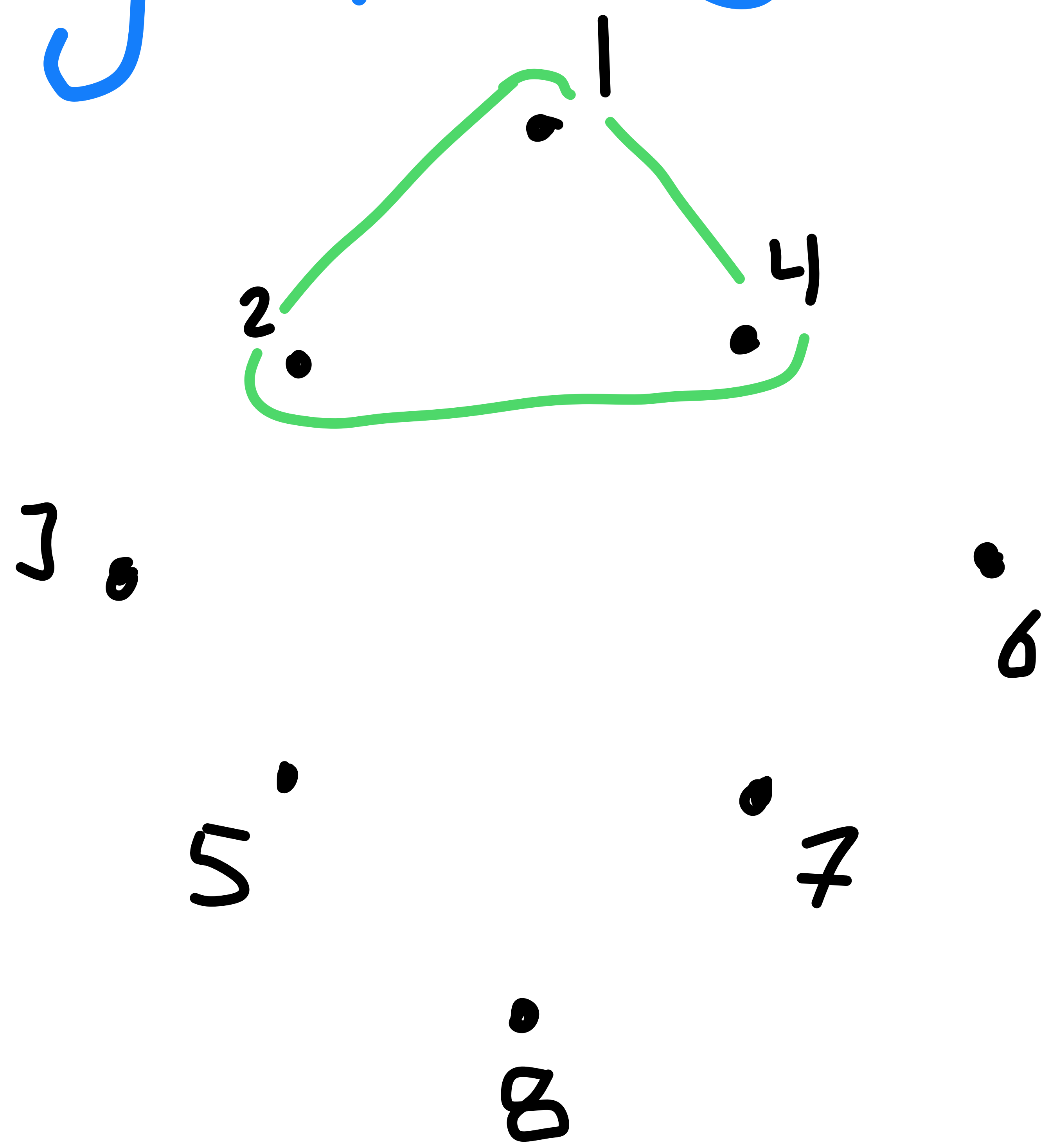
$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$

$x_4 + x_6 + x_7 = 0$

$x_5 + x_7 + x_8 = 0$

$x_6 + x_7 + x_8 = 0$



Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

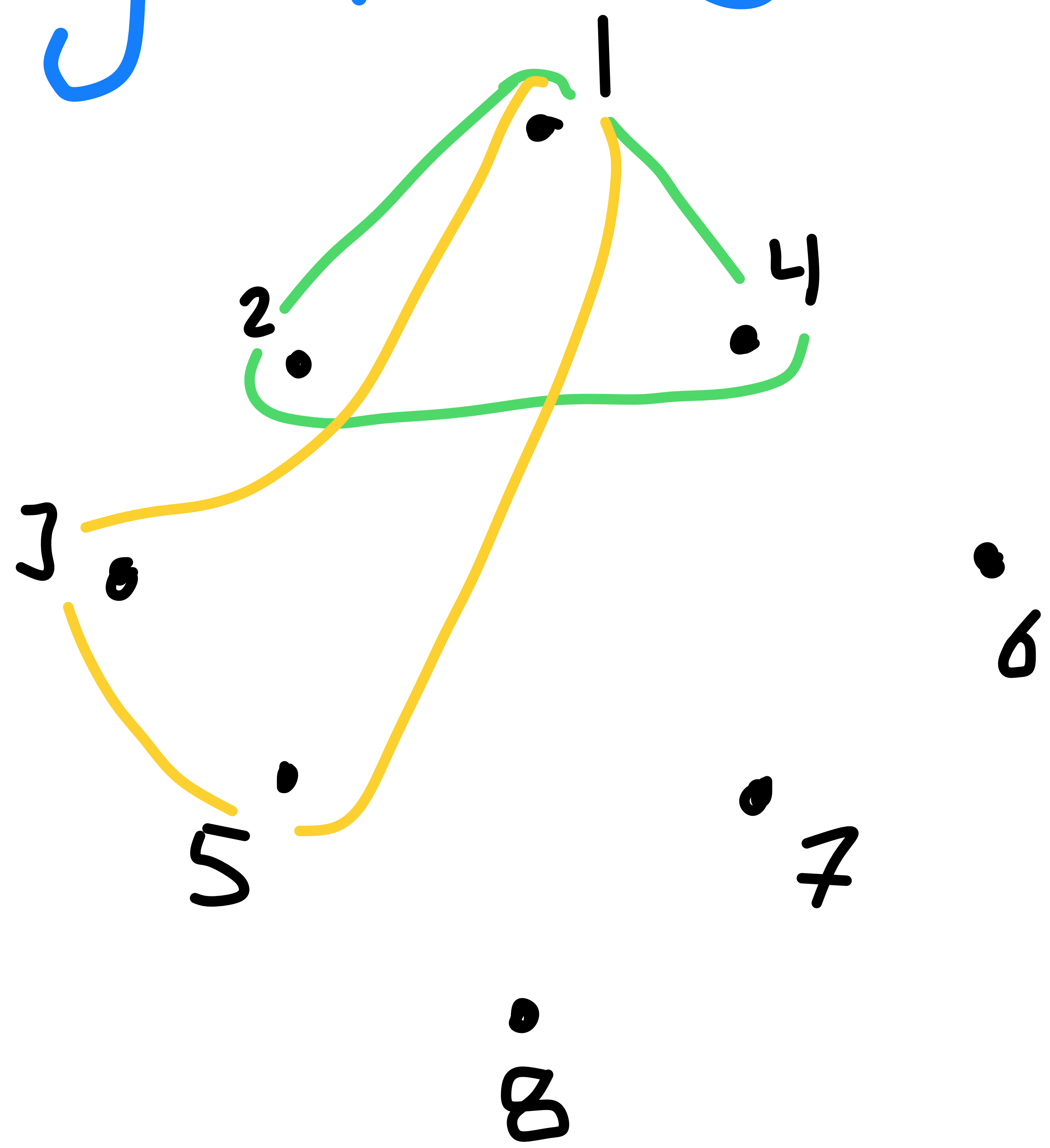
$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$

$x_4 + x_6 + x_7 = 0$

$x_5 + x_7 + x_8 = 0$

$x_6 + x_7 + x_8 = 0$



Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

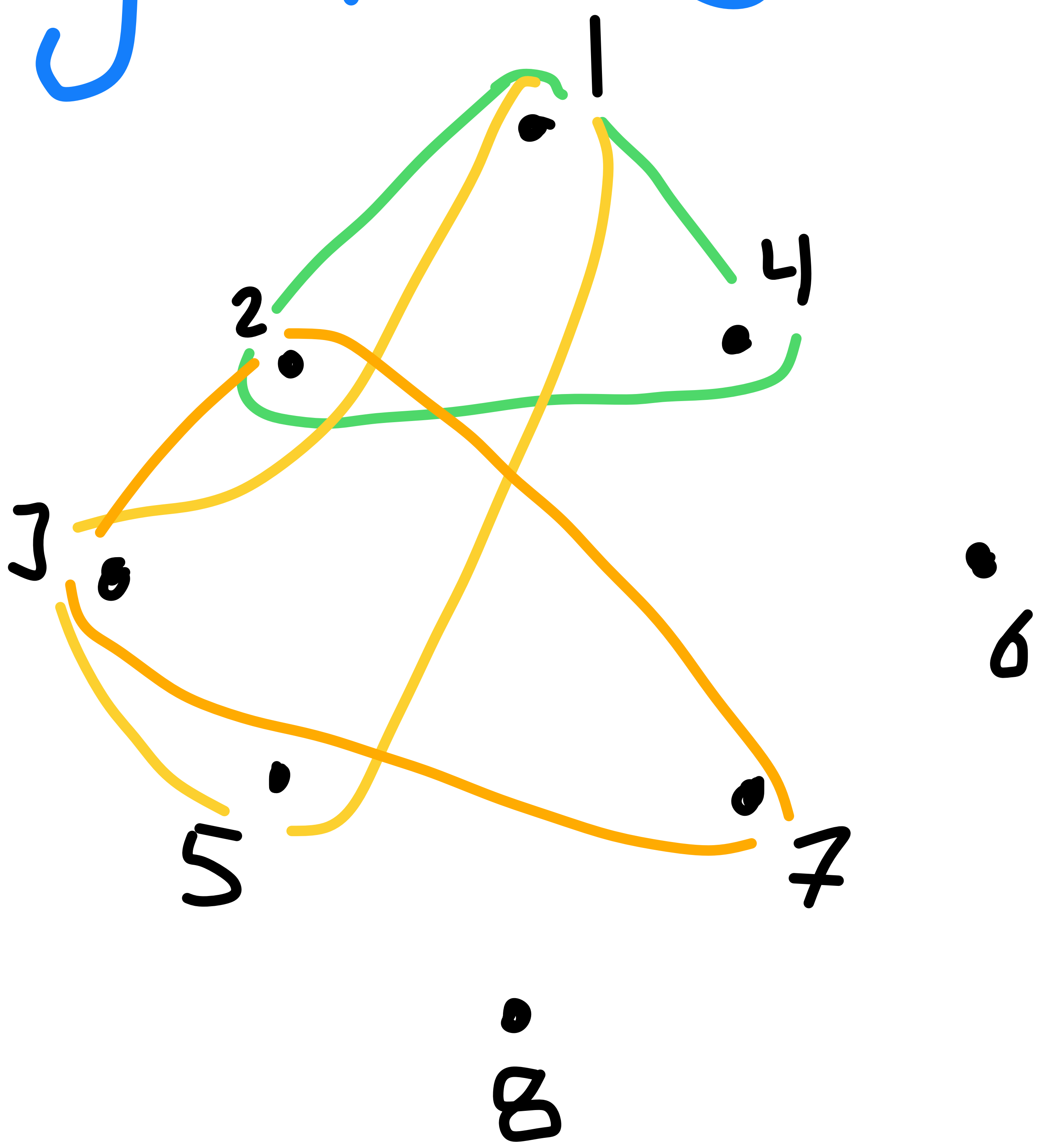
$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$

$x_4 + x_6 + x_7 = 0$

$x_5 + x_7 + x_8 = 0$

$x_6 + x_7 + x_8 = 0$



Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

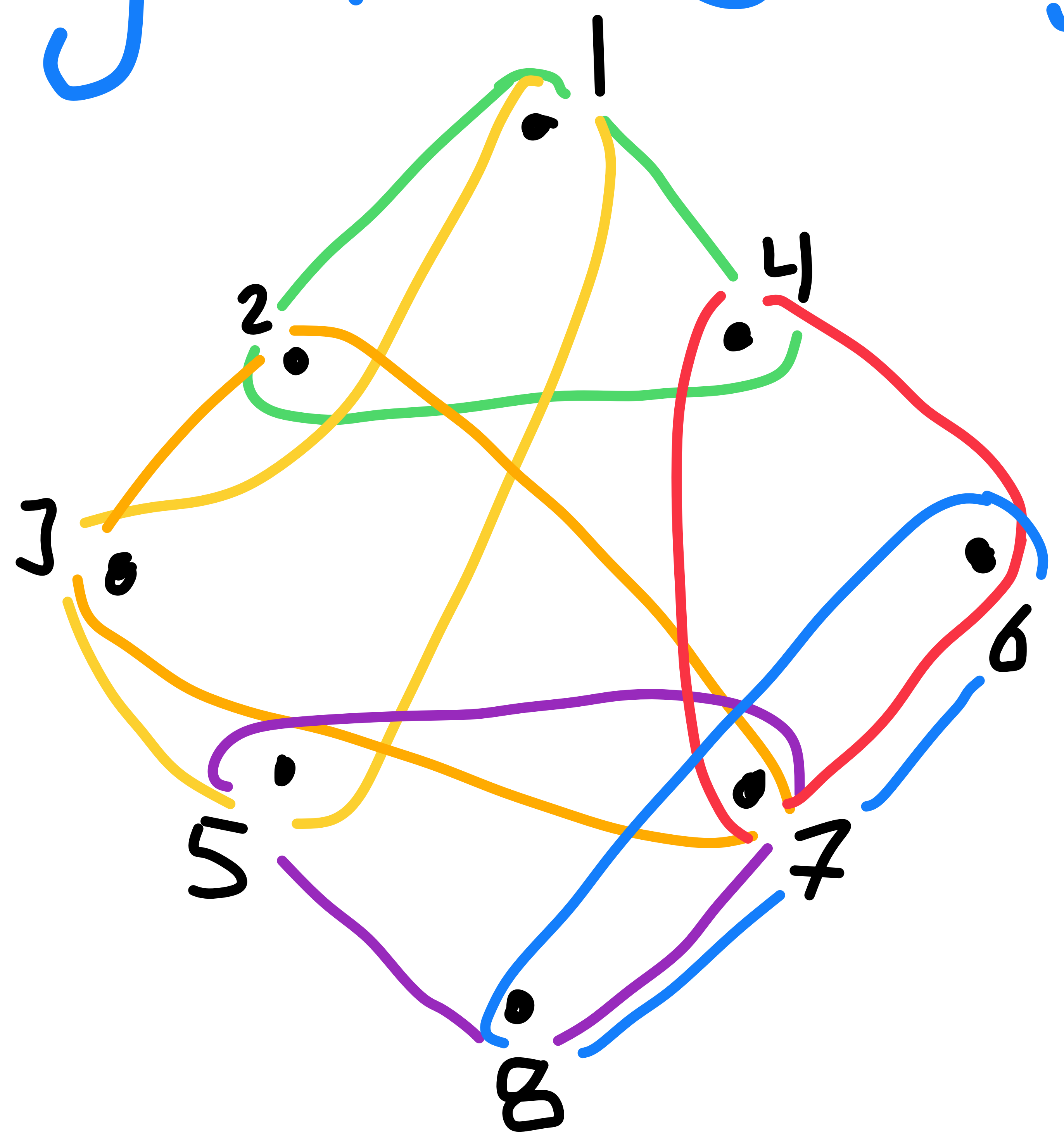
$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$

$x_4 + x_6 + x_7 = 0$

$x_5 + x_7 + x_8 = 0$

$x_6 + x_7 + x_8 = 0$



E = set of edges

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

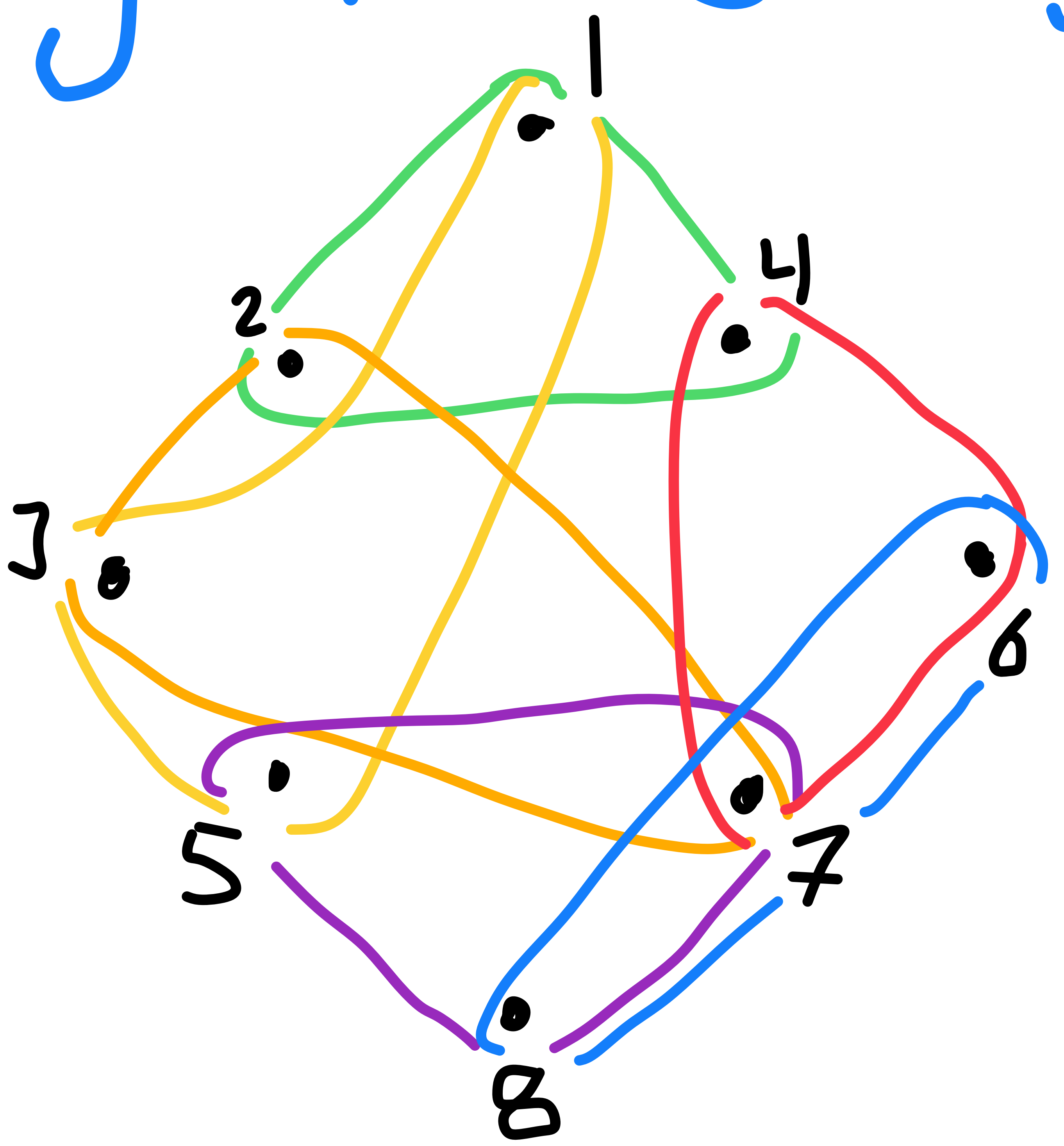
$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$

$x_4 + x_6 + x_7 = 0$

$x_5 + x_7 + x_8 = 0$

$x_6 + x_7 + x_8 = 0$

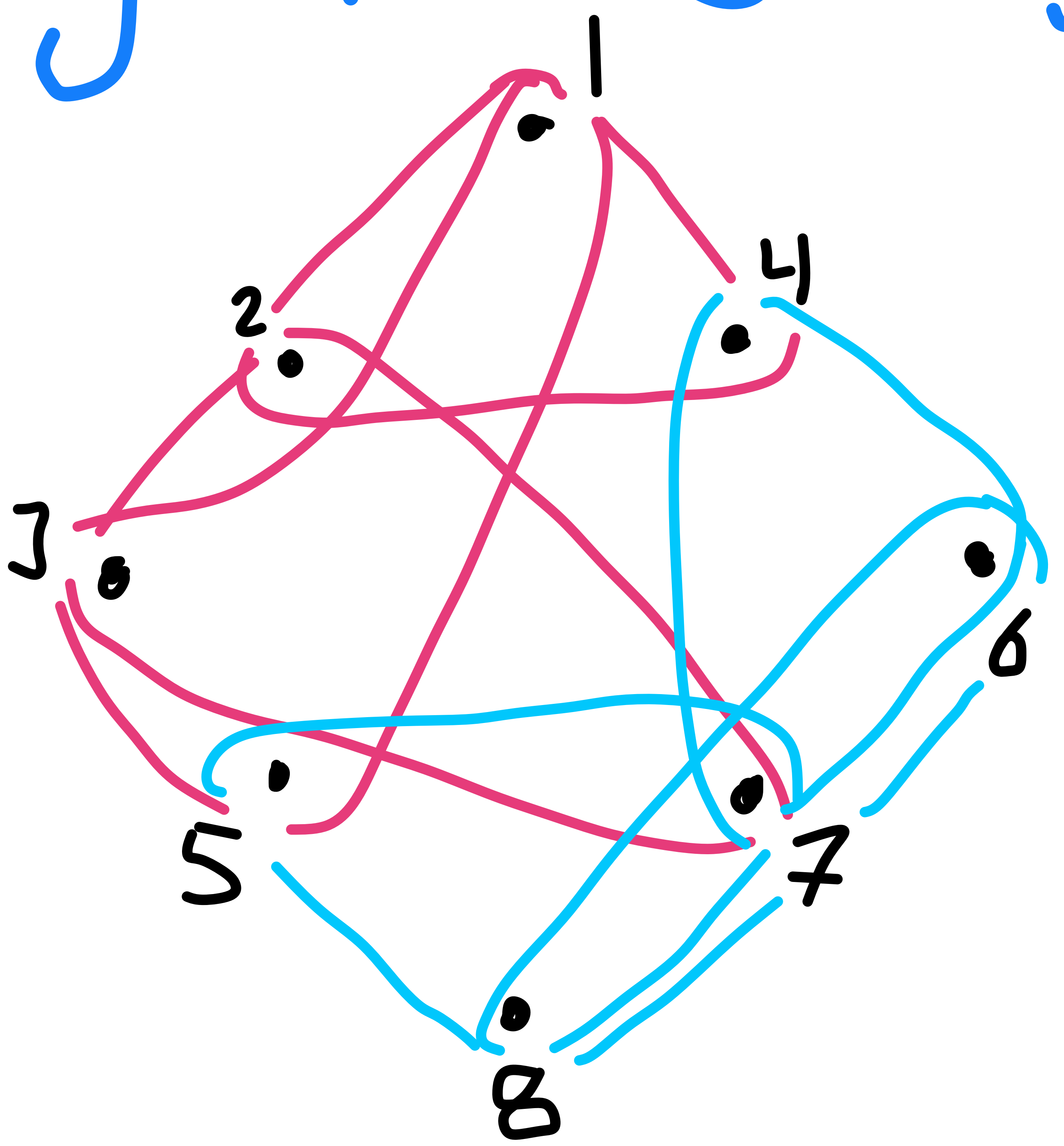


E = set of edges

1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$
$x_1 + x_3 + x_5 = 1$
$x_2 + x_3 + x_7 = 1$
$x_4 + x_6 + x_7 = 0$
$x_5 + x_7 + x_8 = 0$
$x_6 + x_7 + x_8 = 0$



1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

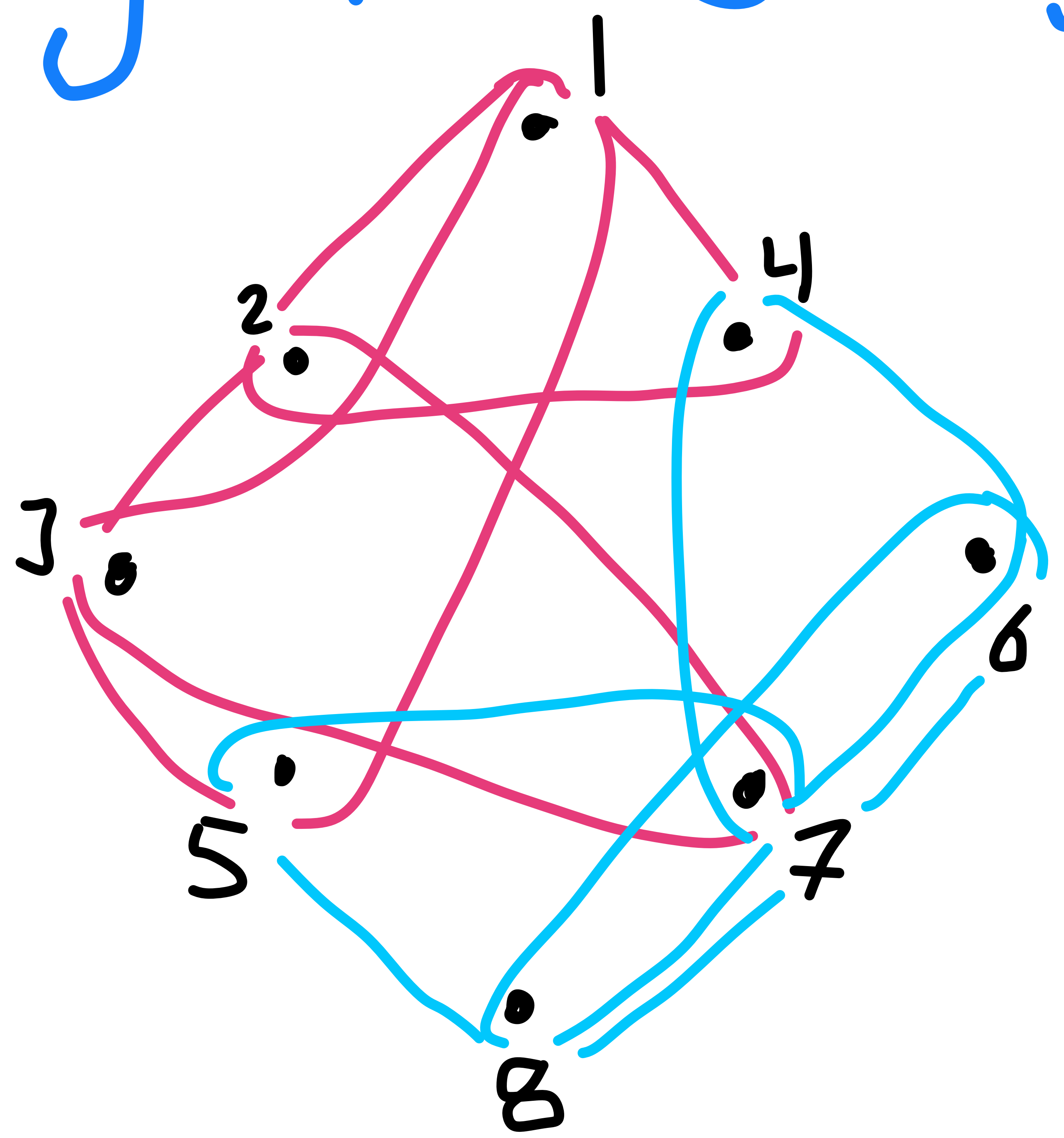
$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$

$x_4 + x_6 + x_7 = 0$

$x_5 + x_7 + x_8 = 0$

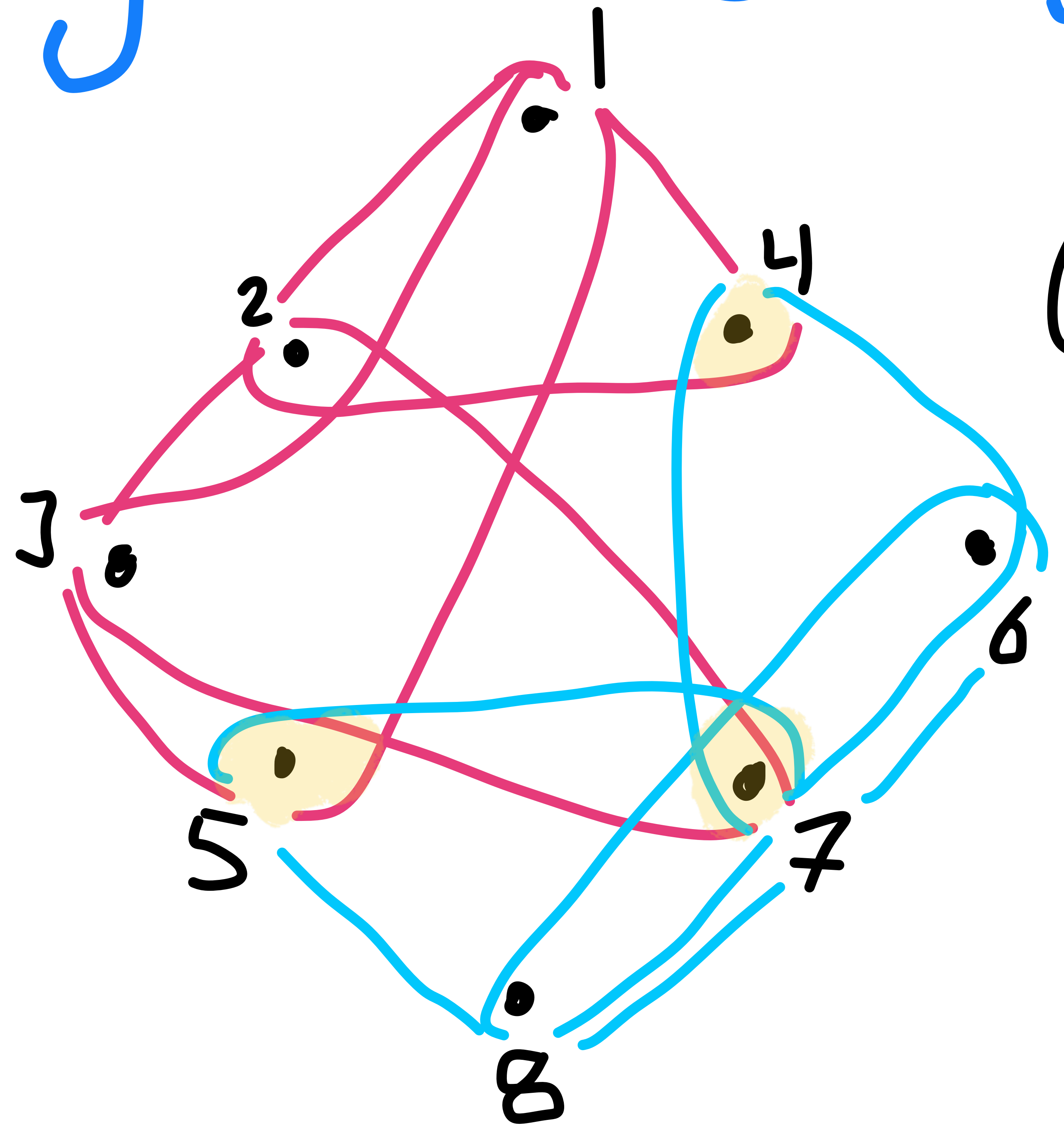
$x_6 + x_7 + x_8 = 0$



1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $\mathrm{Cut}_2(E_j) = \{x_i \mid x_i \text{ occurs an odd number of times in } E_j\}$

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$
$x_1 + x_3 + x_5 = 1$
$x_2 + x_3 + x_7 = 1$
$x_4 + x_6 + x_7 = 0$
$x_5 + x_7 + x_8 = 0$
$x_6 + x_7 + x_8 = 0$



$\mathrm{Cut}_2(E_1) = \mathrm{Cut}_2(E_2) = \{x_4, x_5, x_7\}$

1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $\mathrm{Cut}_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

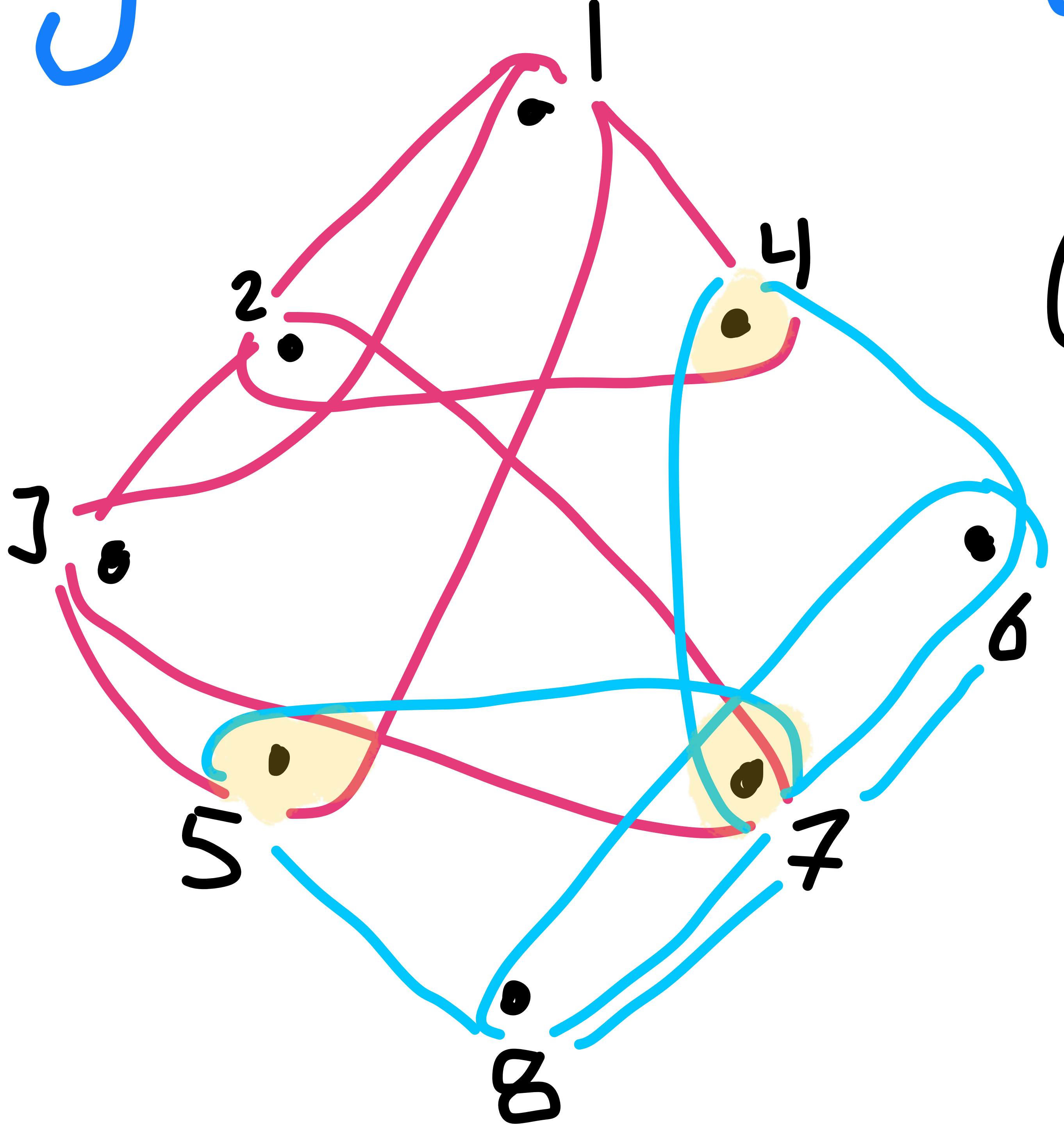
$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$

$x_4 + x_6 + x_7 = 0$

$x_5 + x_7 + x_8 = 0$

$x_6 + x_7 + x_8 = 0$



$\mathrm{Cut}_2(E_1) = \mathrm{Cut}_2(E_2) = \{x_4, x_5, x_7\}$

1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $\mathrm{Cut}_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$
3. Branch on the parity of the sum of $\mathrm{Cut}_2(E_1)$ and the parity of the sum of $\mathrm{Cut}_2(E_2)$

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

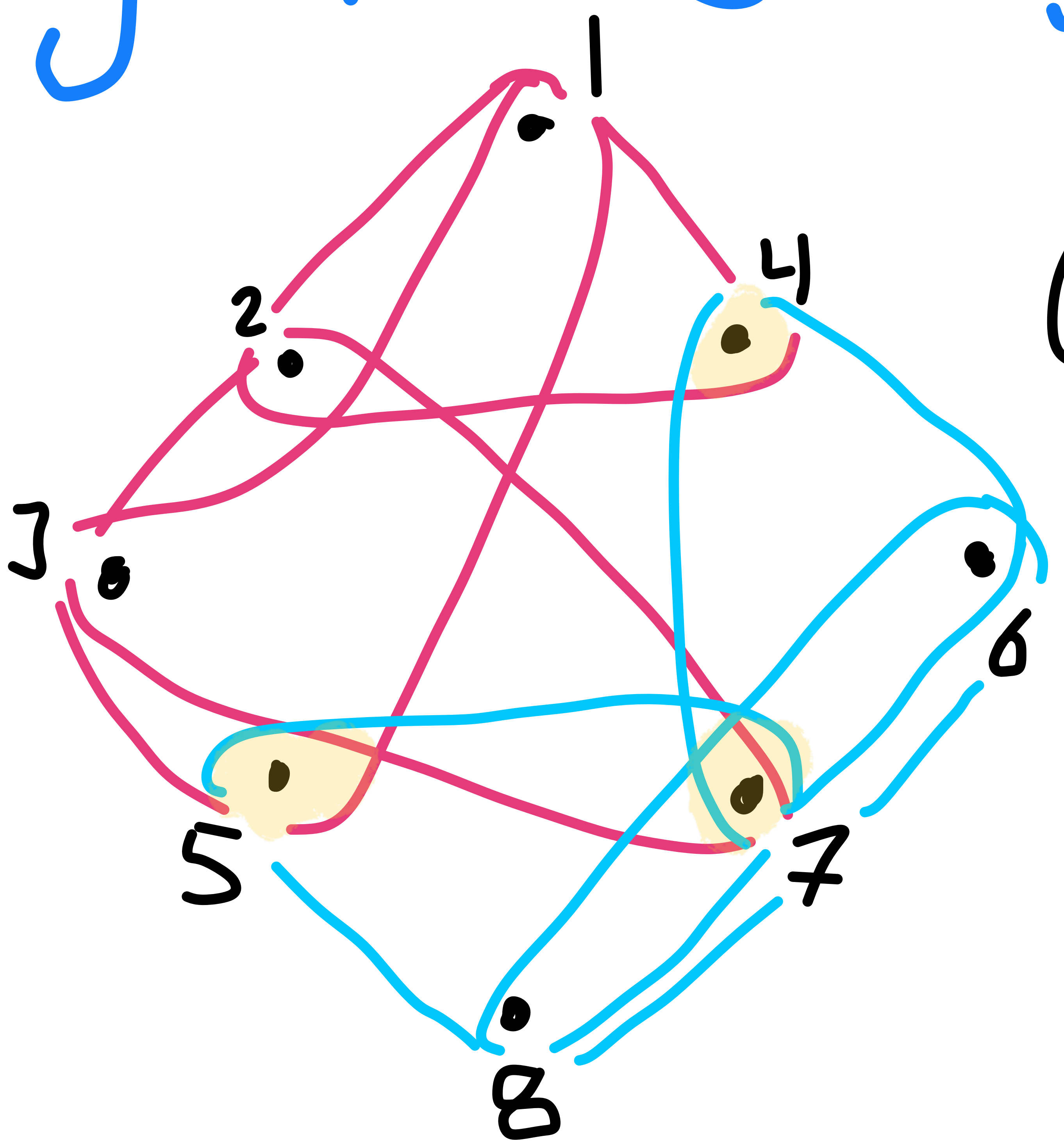
$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$

$x_4 + x_6 + x_7 = 0$

$x_5 + x_7 + x_8 = 0$

$x_6 + x_7 + x_8 = 0$



$Cut_2(E_1) = Cut_2(E_2) = \{x_4, x_5, x_7\}$

Suppose $x_4 + x_5 + x_7 \equiv 0 \mod 2$

1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $Cut_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$
3. Branch on the parity of the sum of $Cut_2(E_1)$ and the parity of the sum of $Cut_2(E_2)$

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

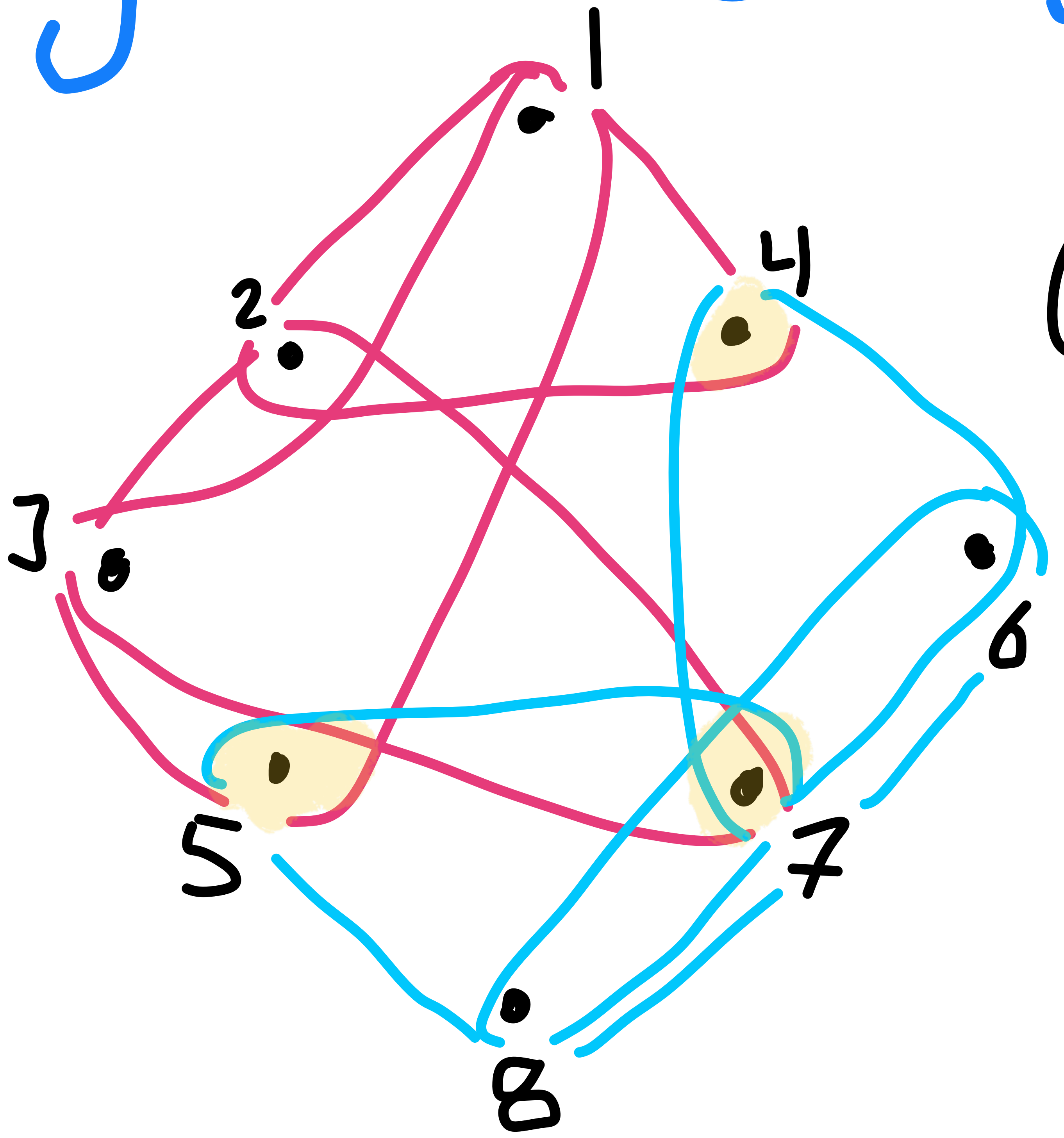
$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$

$x_4 + x_6 + x_7 = 0$

$x_5 + x_7 + x_8 = 0$

$x_6 + x_7 + x_8 = 0$



$Cut_2(E_1) = Cut_2(E_2) = \{x_4, x_5, x_7\}$

Suppose $x_4 + x_5 + x_7 \equiv 0 \bmod 2$

$\rightarrow E_1$ is unsat

1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $Cut_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$
3. Branch on the parity of the sum of $Cut_2(E_1)$ and the parity of the sum of $Cut_2(E_2)$

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

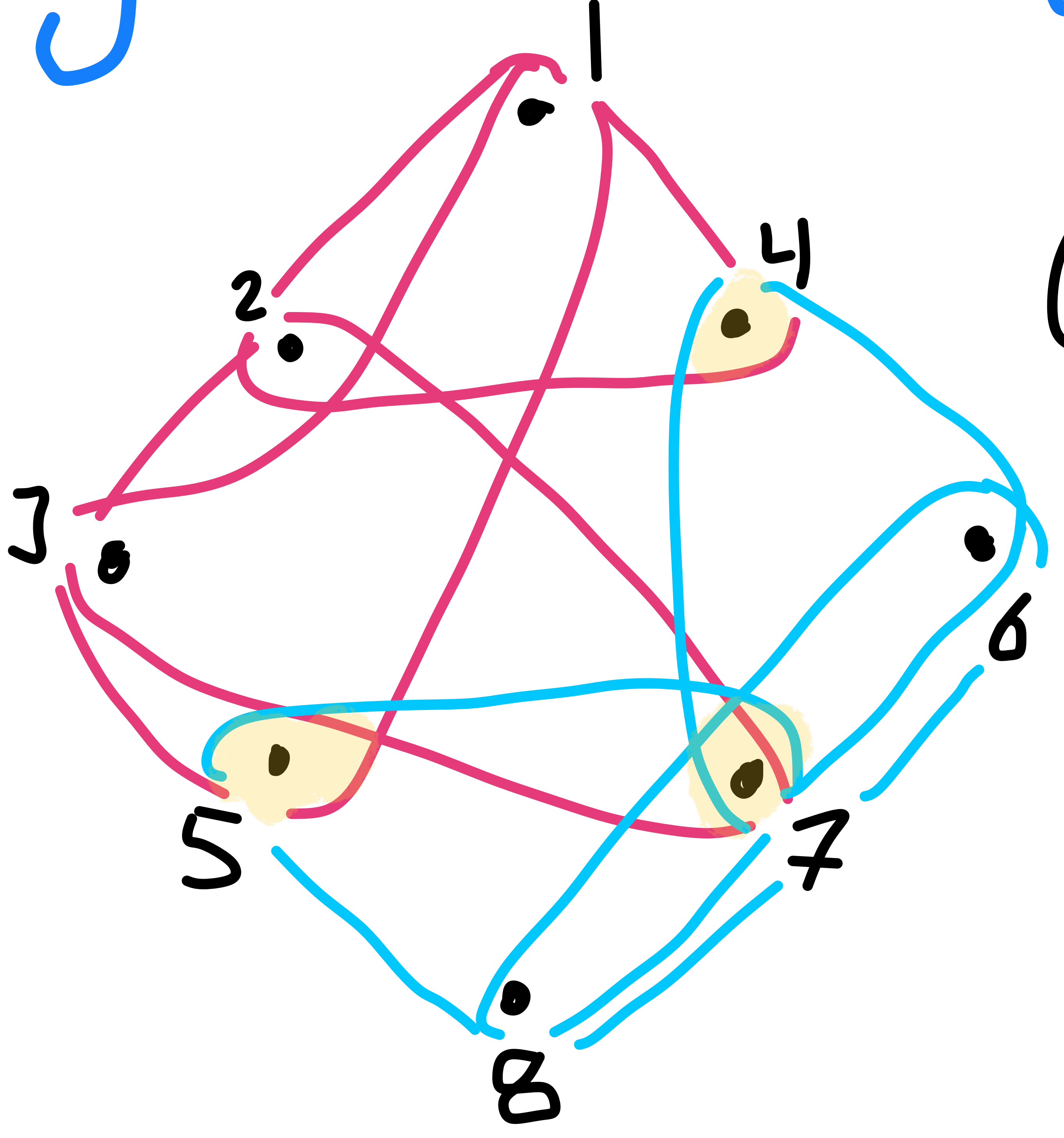
$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$

$x_4 + x_6 + x_7 = 0$

$x_5 + x_7 + x_8 = 0$

$x_6 + x_7 + x_8 = 0$



$\mathrm{Cut}_2(E_1) = \mathrm{Cut}_2(E_2) = \{x_4, x_5, x_7\}$

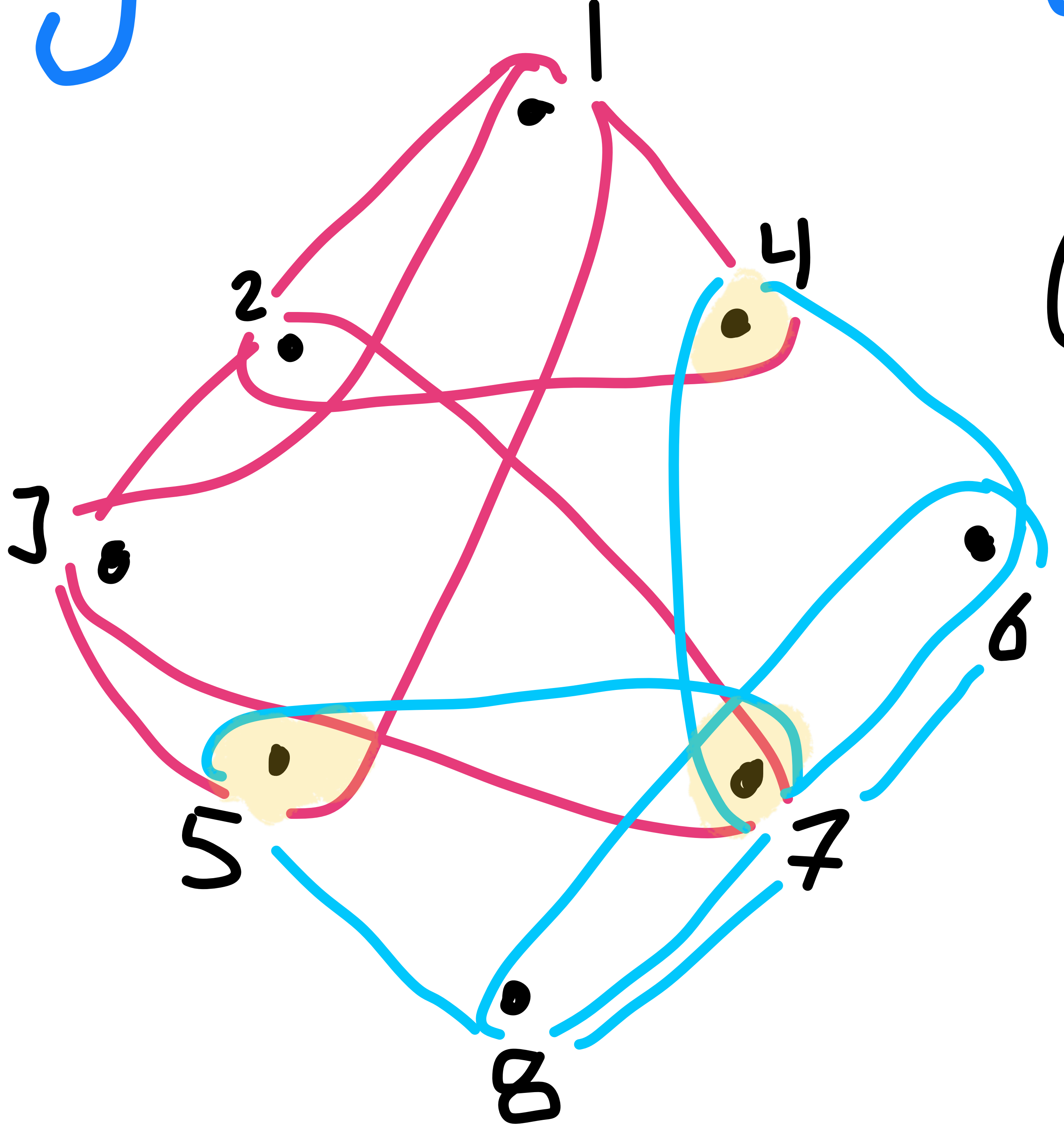
Suppose $x_4 + x_5 + x_7 \equiv 0 \bmod 2$

$\rightarrow E_1$ is unsat

1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $\mathrm{Cut}_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$
3. Branch on the parity of the sum of $\mathrm{Cut}_2(E_1)$ and the parity of the sum of $\mathrm{Cut}_2(E_2)$
4. Recurse on the unsat half.

Refuting Systems of Equations

$x_1+x_2+x_4=1$
$x_1+x_3+x_5=1$
$x_2+x_3+x_7=1$
$x_4+x_6+x_7=0$
$x_5+x_7+x_8=0$
$x_6+x_7+x_8=0$



$Cut_2(E_1) = Cut_2(E_2) = \{x_4, x_5, x_7\}$

Suppose $x_4+x_5+x_7 \equiv 0 \bmod 2$

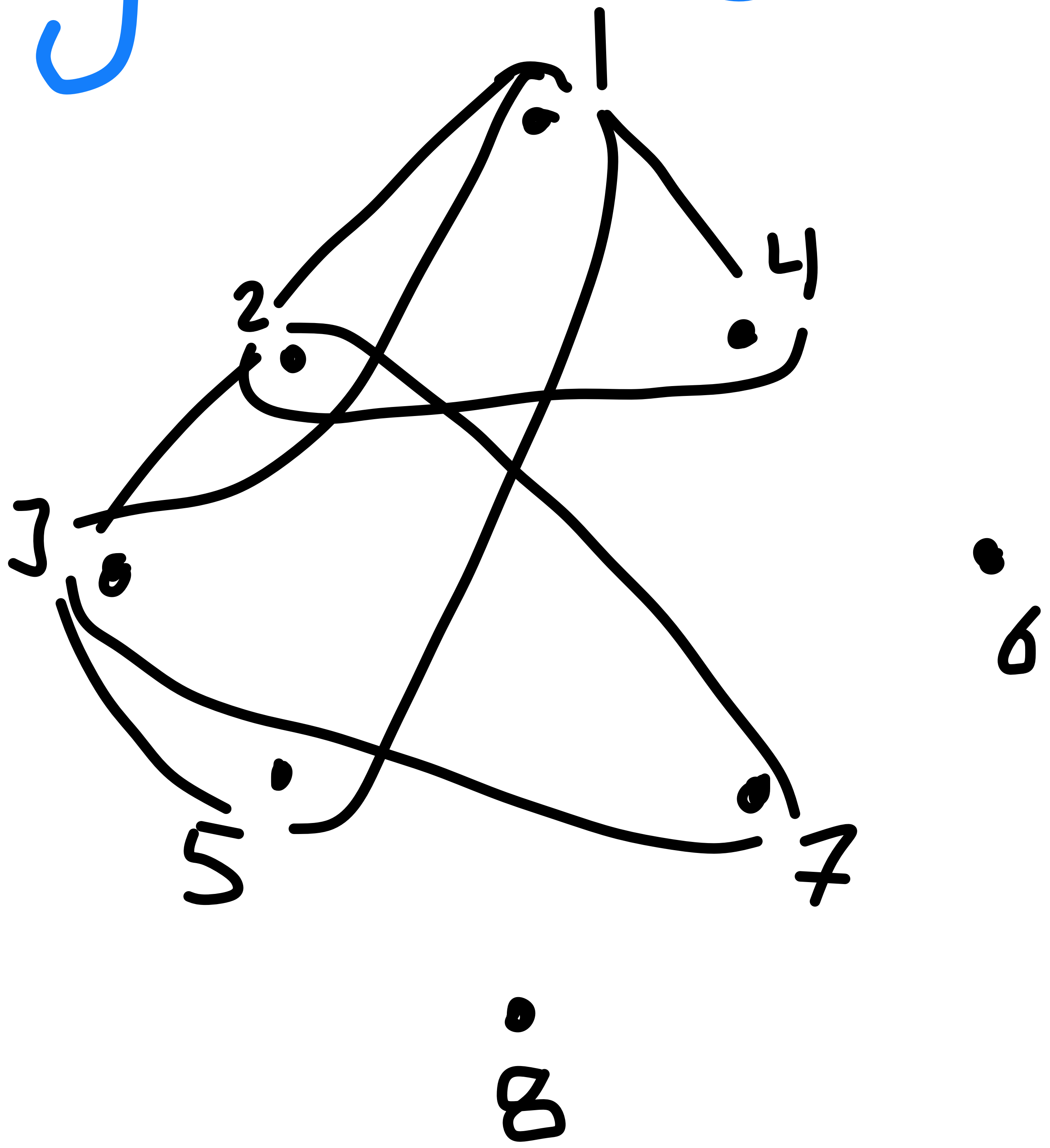
$\rightarrow E_1$ is unsat

Recurse on E_1

1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $Cut_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$
3. Branch on the parity of the sum of $Cut_2(E_1)$ and the parity of the sum of $Cut_2(E_2)$
4. Recurse on the unsat half.

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$
$x_1 + x_3 + x_5 = 1$
$x_2 + x_3 + x_7 = 1$



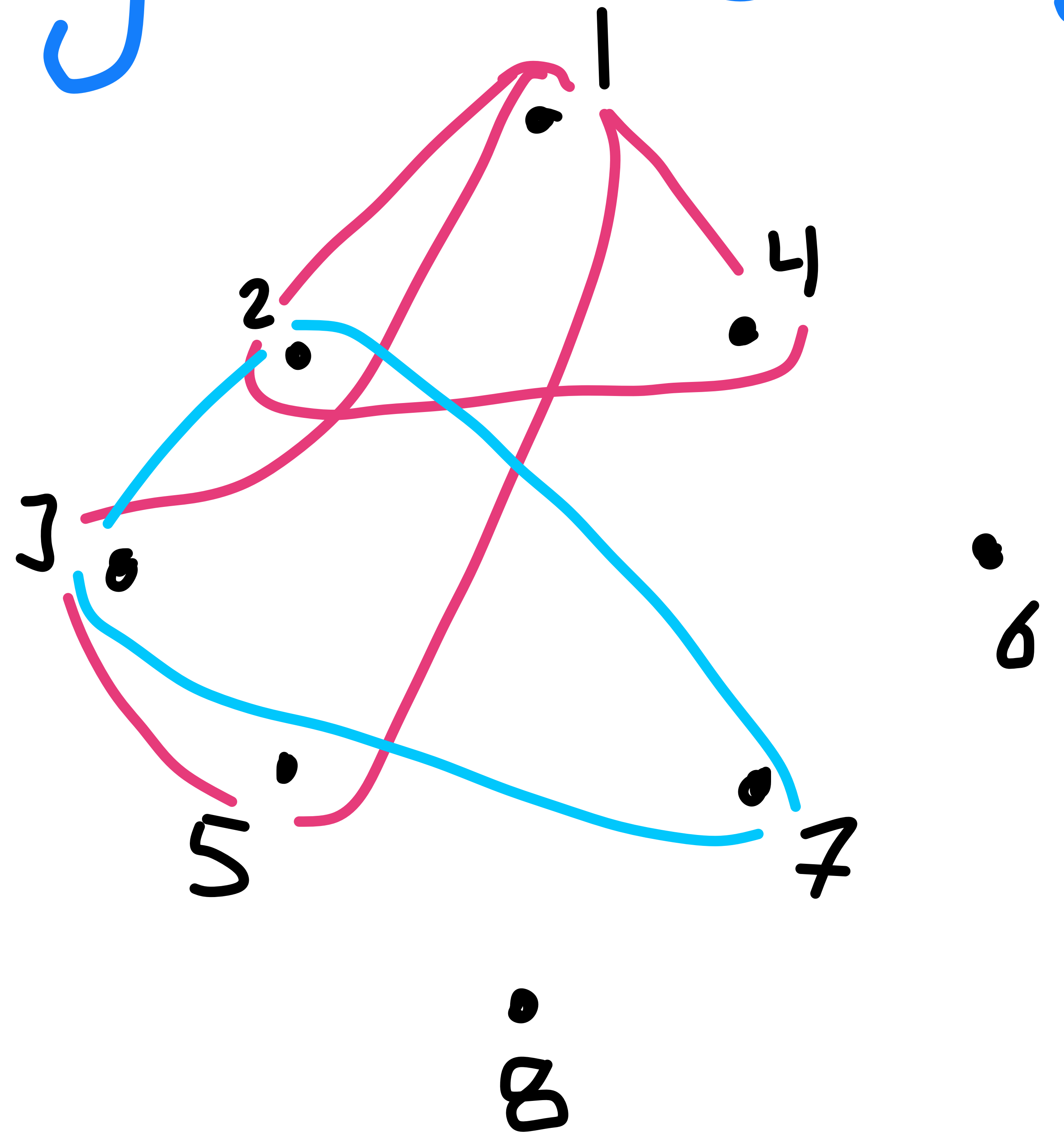
1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $Cut_2(E_j) = \{x_i \mid x_i \text{ occurs an odd number of times in } E_j\}$
3. Branch on the parity of the sum of $Cut_2(E_1)$ and the parity of the sum of $Cut_2(E_2)$
4. Recurse on the unsat half.

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$



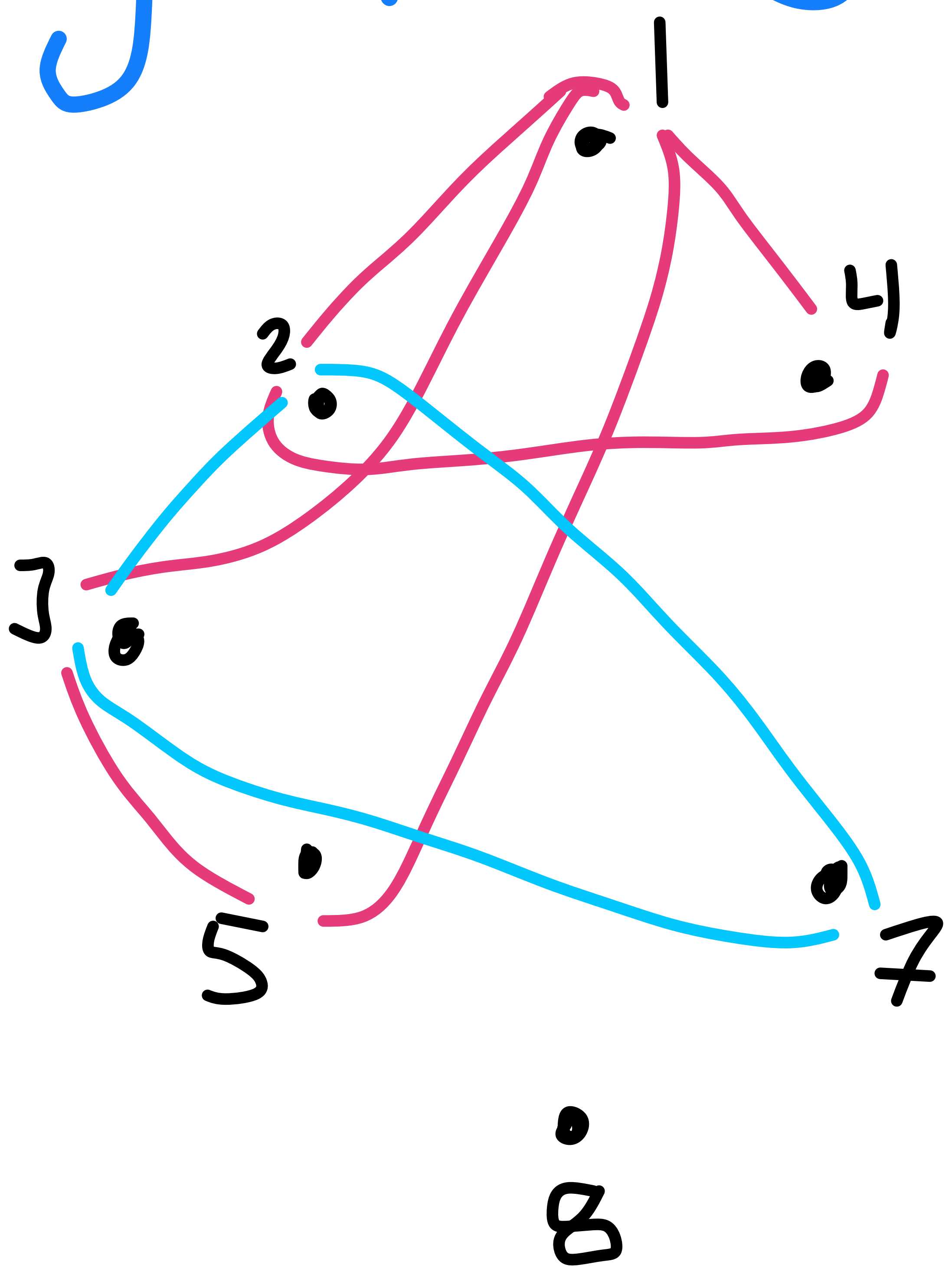
1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $Cut_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$
3. Branch on the parity of the sum of $Cut_2(E_1)$ and the parity of the sum of $Cut_2(E_2)$
4. Recurse on the unsat half.

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$



$Cut_2(E_1) = \{x_2, x_3, x_4, x_5\}$

6 $Cut_2(E_2) = \{x_2, x_3, x_7\}$

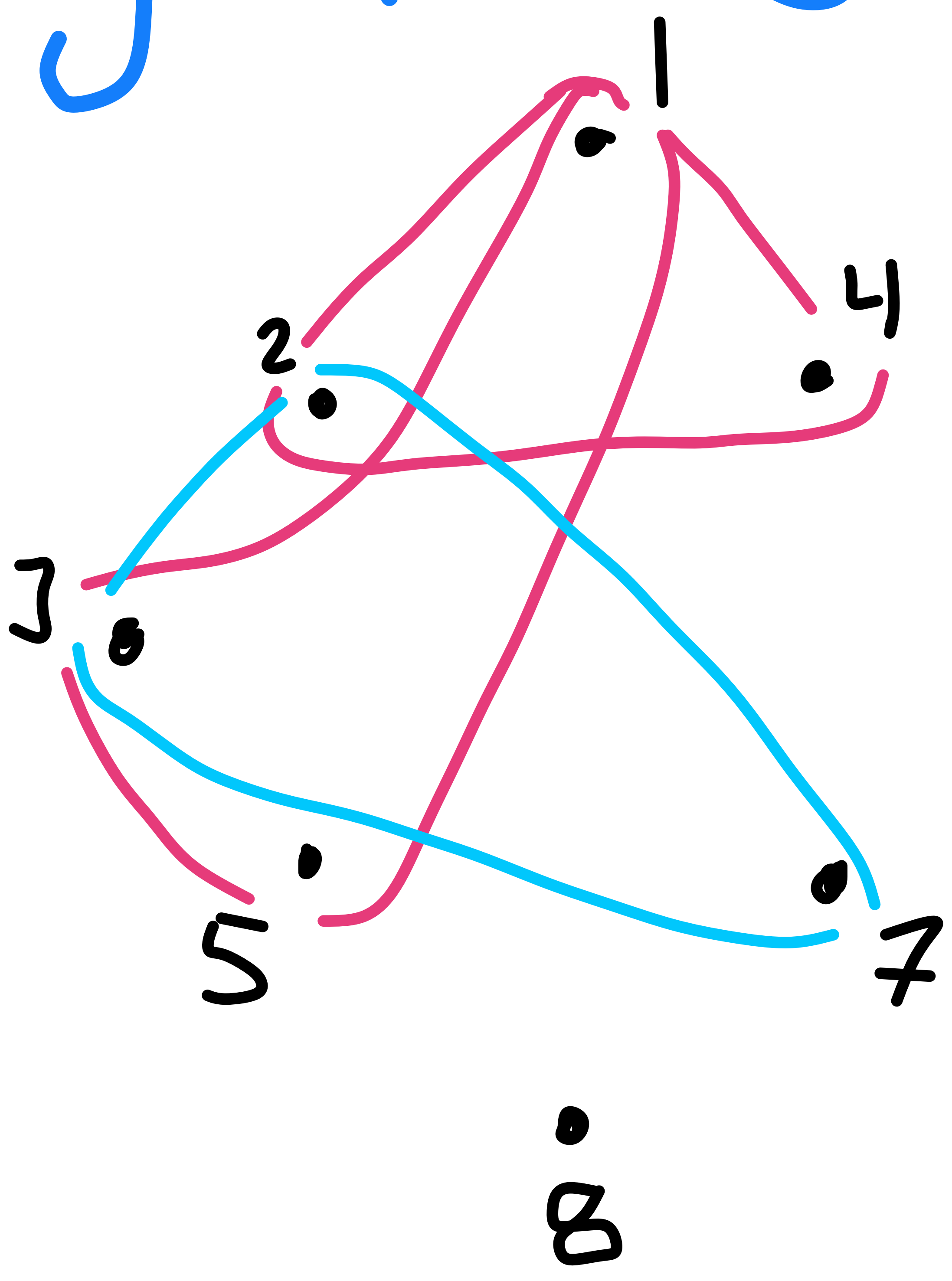
1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $Cut_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$
3. Branch on the parity of the sum of $Cut_2(E_1)$ and the parity of the sum of $Cut_2(E_2)$
4. Recurse on the unsat half.

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$



$Cut_2(E_1) = \{x_2, x_3, x_4, x_5\}$

Suppose $x_2 + x_3 + x_4 + x_5 \equiv 1 \bmod 2$

$\therefore$ $Cut_2(E_2) = \{x_2, x_3, x_7\}$

Suppose $x_2 + x_3 + x_7 \equiv 1 \bmod 2$

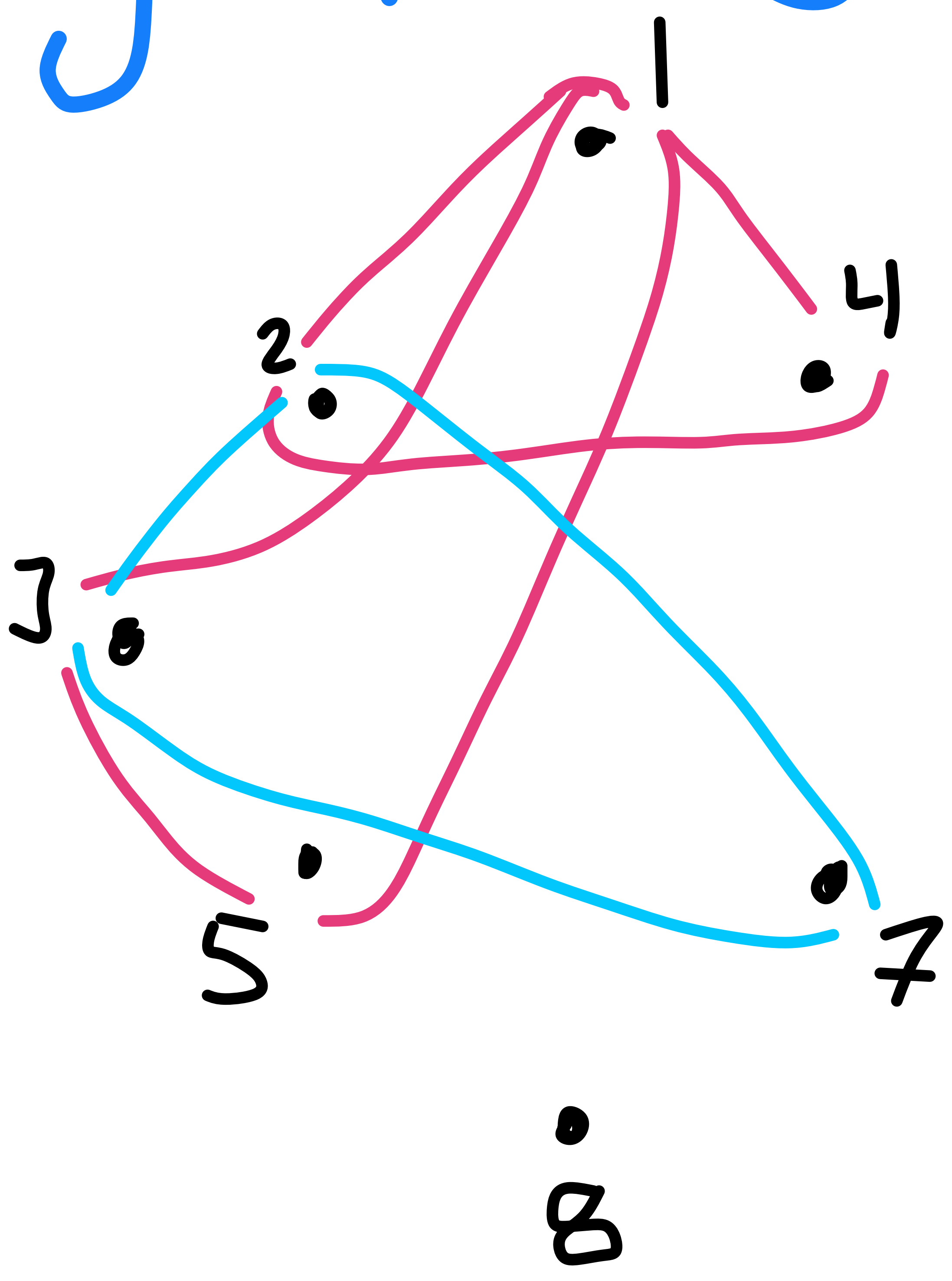
1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $Cut_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$
3. Branch on the parity of the sum of $Cut_2(E_1)$ and the parity of the sum of $Cut_2(E_2)$
4. Recurse on the unsat half.

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

$x_1 + x_3 + x_5 = 1$

$x_2 + x_3 + x_7 = 1$



$Cut_2(E_1) = \{x_2, x_3, x_4, x_5\}$

Suppose $x_2 + x_3 + x_4 + x_5 \equiv 1 \bmod 2$

$Cut_2(E_2) = \{x_2, x_3, x_7\}$

Suppose $x_2 + x_3 + x_7 \equiv 1 \bmod 2$

$\rightarrow E_1$ is unsat

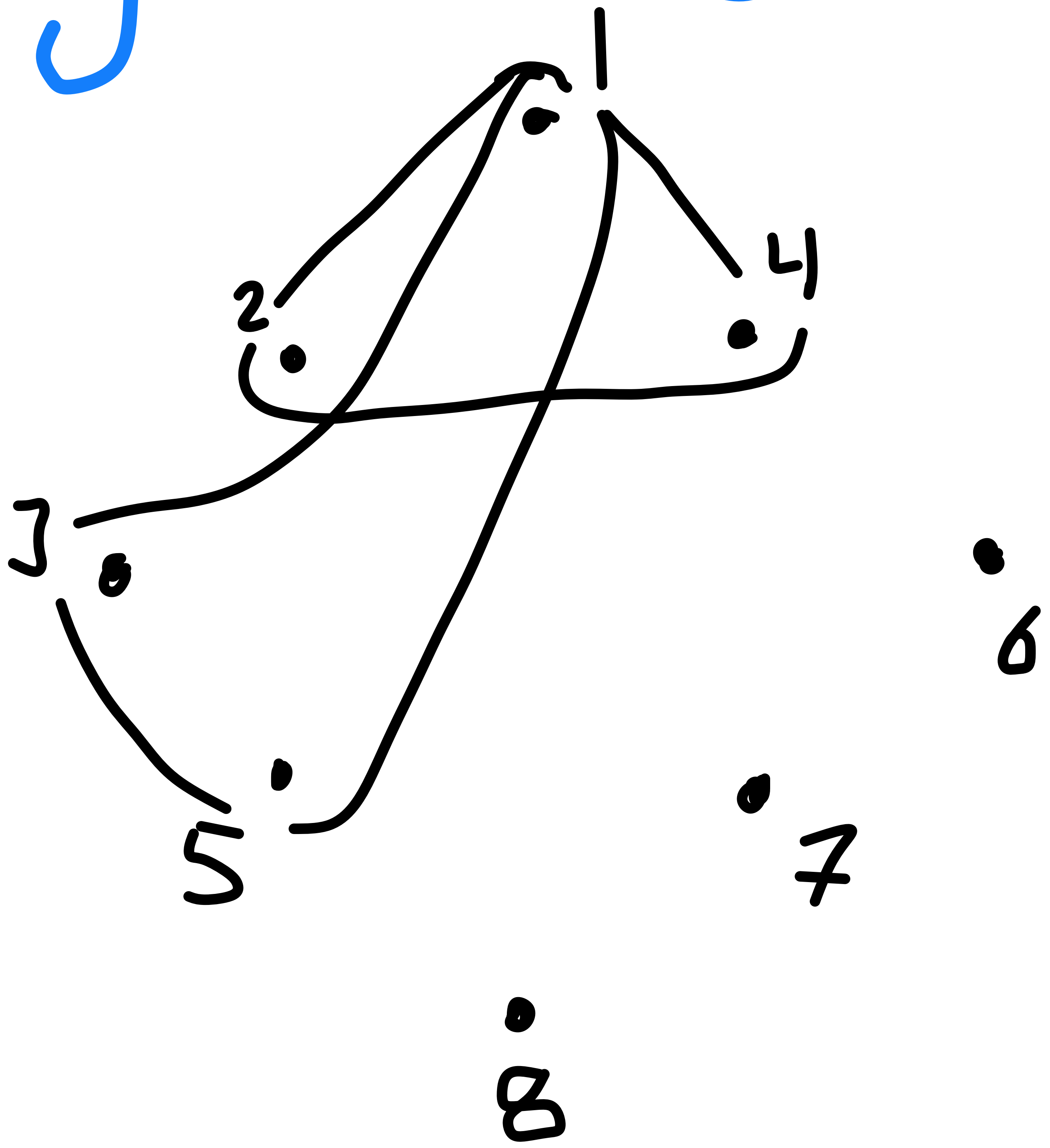
Recurse on E_1

1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $Cut_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$
3. Branch on the parity of the sum of $Cut_2(E_1)$ and the parity of the sum of $Cut_2(E_2)$
4. Recurse on the unsat half.

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

$x_1 + x_3 + x_5 = 1$

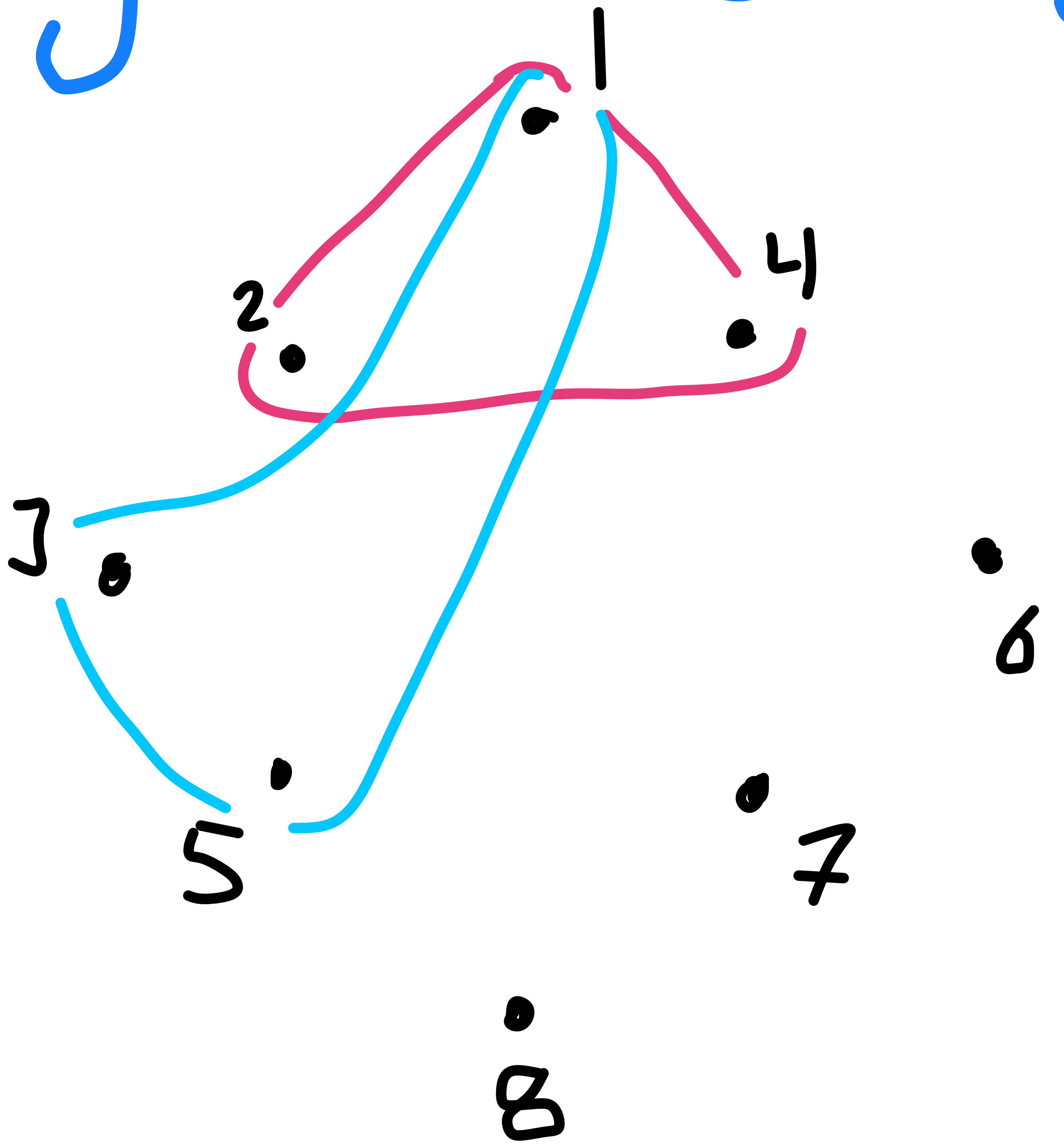


1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $Cut_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$
3. Branch on the parity of the sum of $Cut_2(E_1)$ and the parity of the sum of $Cut_2(E_2)$
4. Recurse on the unsat half.

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

$x_1 + x_3 + x_5 = 1$

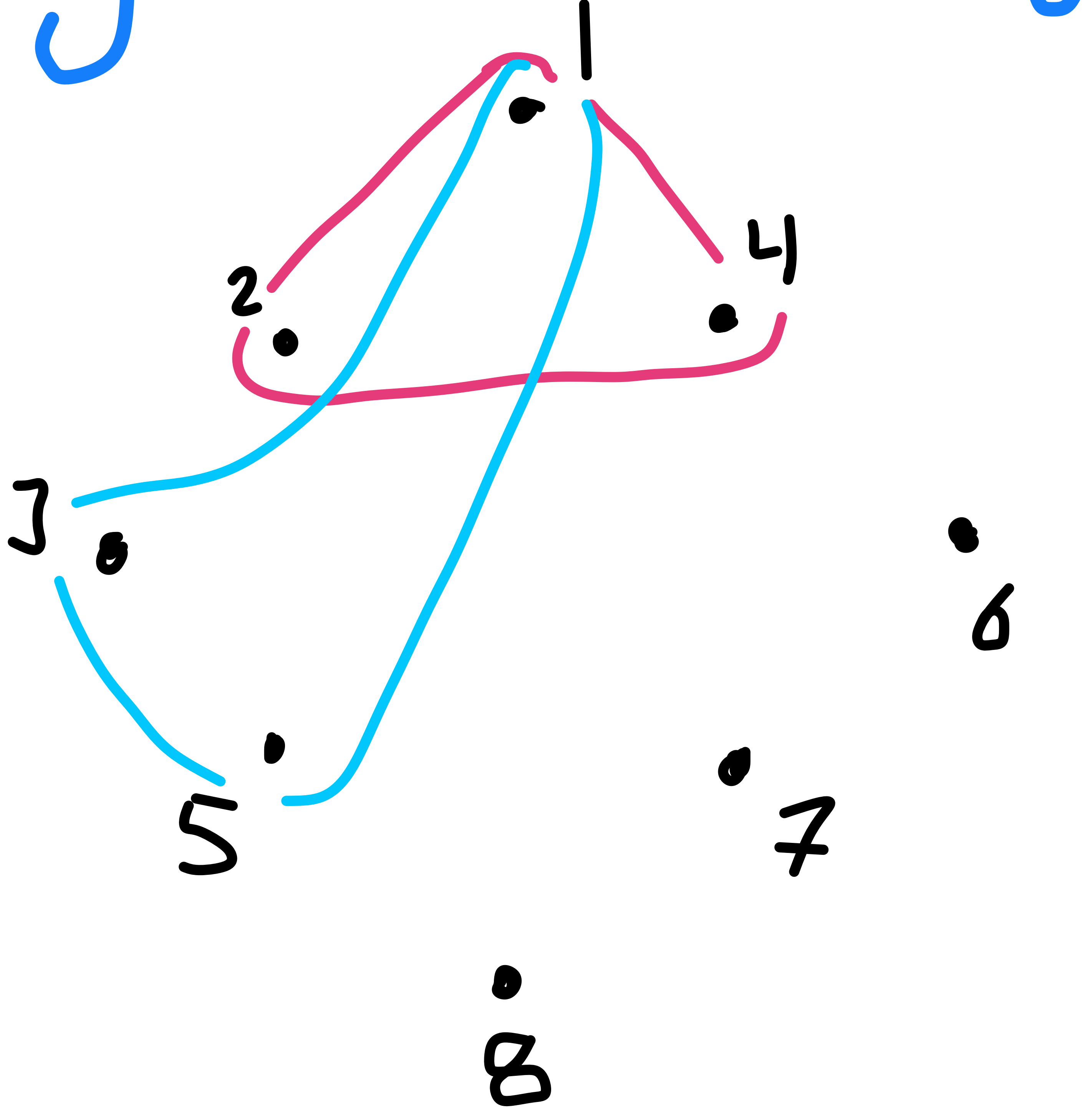


1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $\mathrm{Cut}_2(E_j) = \{x_i \mid x_i \text{ occurs an odd number of times in } E_j\}$
3. Branch on the parity of the sum of $\mathrm{Cut}_2(E_1)$ and the parity of the sum of $\mathrm{Cut}_2(E_2)$
4. Recurse on the unsat half.

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

$x_1 + x_3 + x_5 = 1$



$Cut_2(E_1) = \{x_1, x_2, x_4\}$

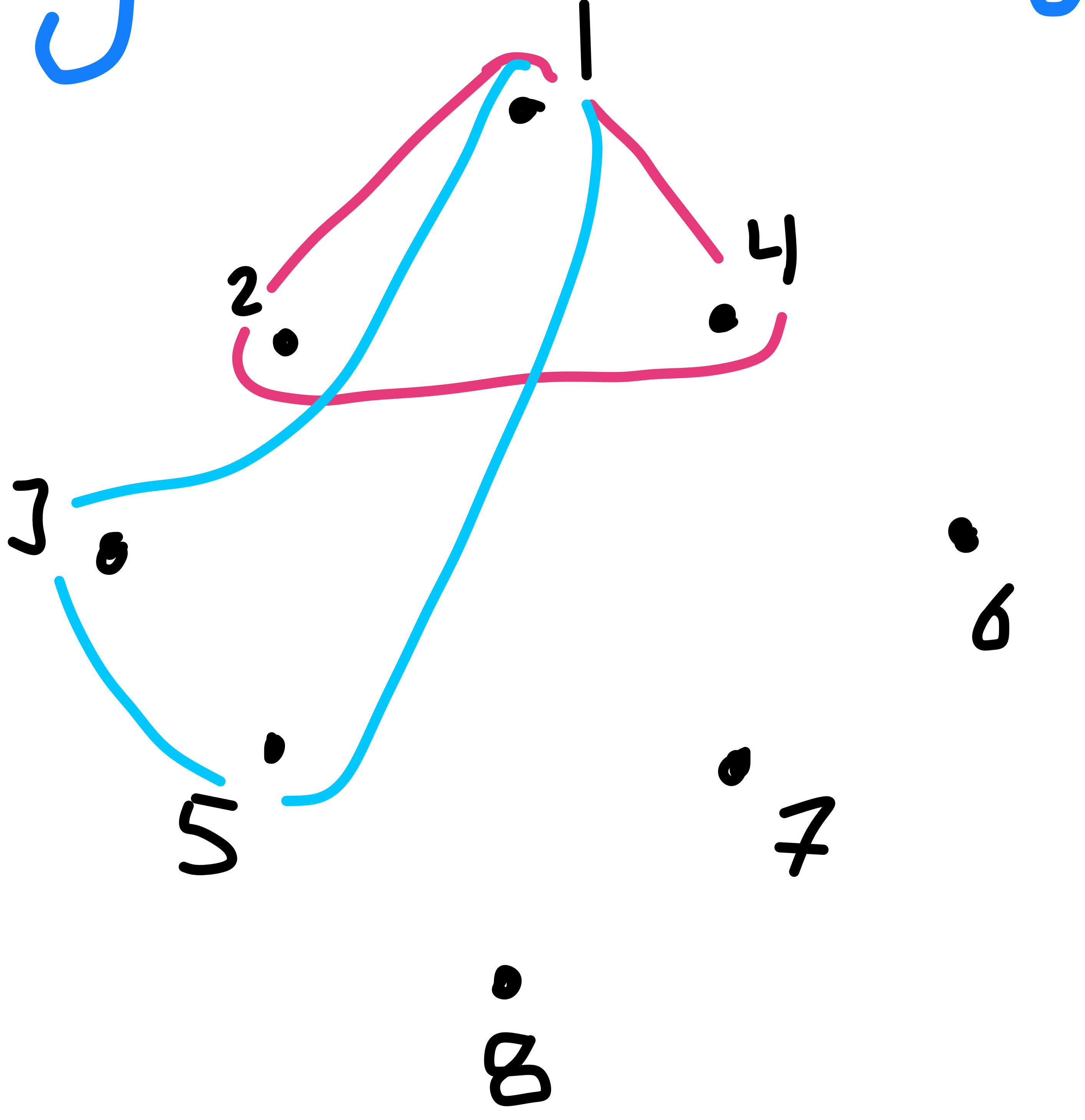
$Cut_2(E_2) = \{x_1, x_3, x_5\}$

1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $Cut_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$
3. Branch on the parity of the sum of $Cut_2(E_1)$ and the parity of the sum of $Cut_2(E_2)$
4. Recurse on the unsat half.

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

$x_1 + x_3 + x_5 = 1$



$Cut_2(E_1) = \{x_1, x_2, x_4\}$

Suppose $x_1+x_2+x_4 \equiv 1 \bmod 2$

$Cut_2(E_2) = \{x_1, x_3, x_5\}$

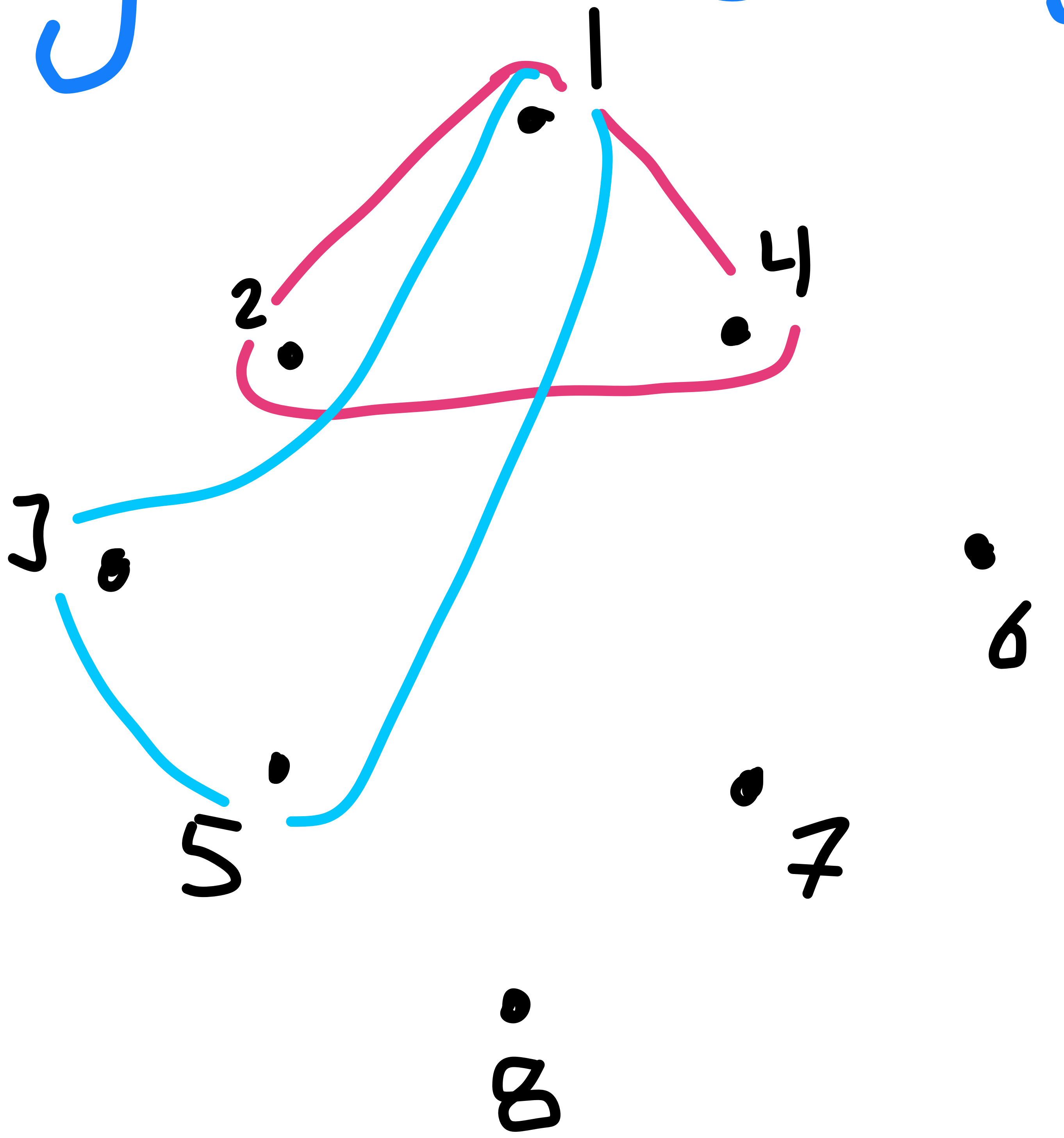
Suppose $x_1+x_3+x_5 \equiv 0 \bmod 2$

1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $Cut_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$
3. Branch on the parity of the sum of $Cut_2(E_1)$ and the parity of the sum of $Cut_2(E_2)$
4. Recurse on the unsat half.

Refuting Systems of Equations

$x_1 + x_2 + x_4 = 1$

$x_1 + x_3 + x_5 = 1$



$Cut_2(E_1) = \{x_1, x_2, x_4\}$

Suppose $x_1 + x_2 + x_4 \equiv 1 \bmod 2$

$Cut_2(E_2) = \{x_1, x_3, x_5\}$

Suppose $x_1 + x_3 + x_5 \equiv 0 \bmod 2$

Violates $x_1 + x_3 + x_5 \equiv 1$

1. Partition E into E_1, E_2 so that $|E_1| \approx |E_2|$
2. Let $Cut_2(E_j) = \{x_i \mid x_i$ occurs an odd number of times in $E_j\}$
3. Branch on the parity of the sum of $Cut_2(E_1)$ and the parity of the sum of $Cut_2(E_2)$
4. Recurse on the unsat half.

Refuting Systems of Equations In SP

To implement in SP:

- Must be able to determine the parity of $Cut_2(E)$

Refuting Systems of Equations In SP

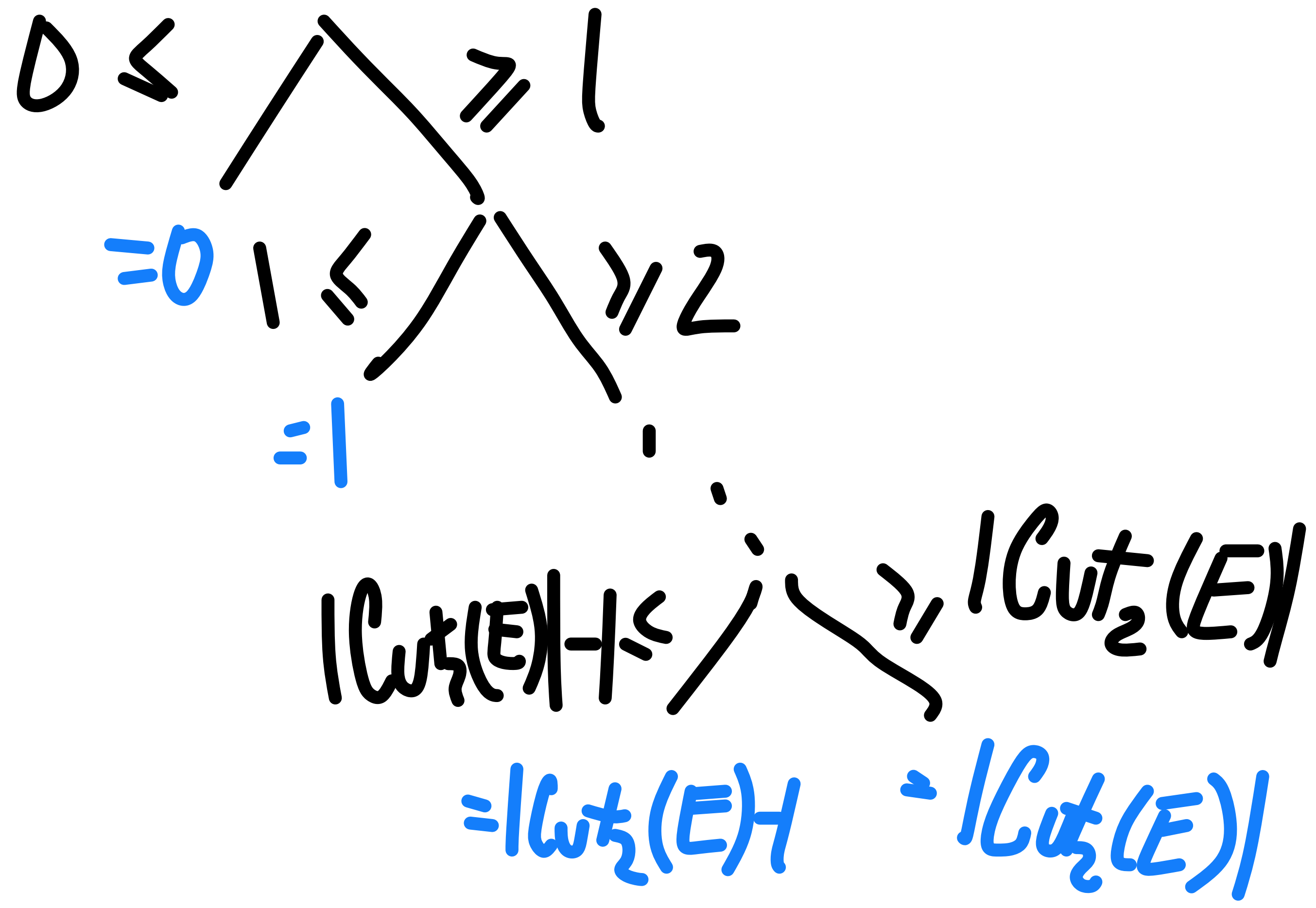
To implement in SP:

- Must be able to determine the parity of $Cut_2(E)$
- Query $\left(\sum_{i \in Cut_2(E)} x_i \leq t, \sum_{i \in Cut_2(E)} x_i \geq t+1\right)$ for $t = 0, \ldots, |Cut_2(E)|$

Refuting Systems of Equations In SP

To implement in SP:

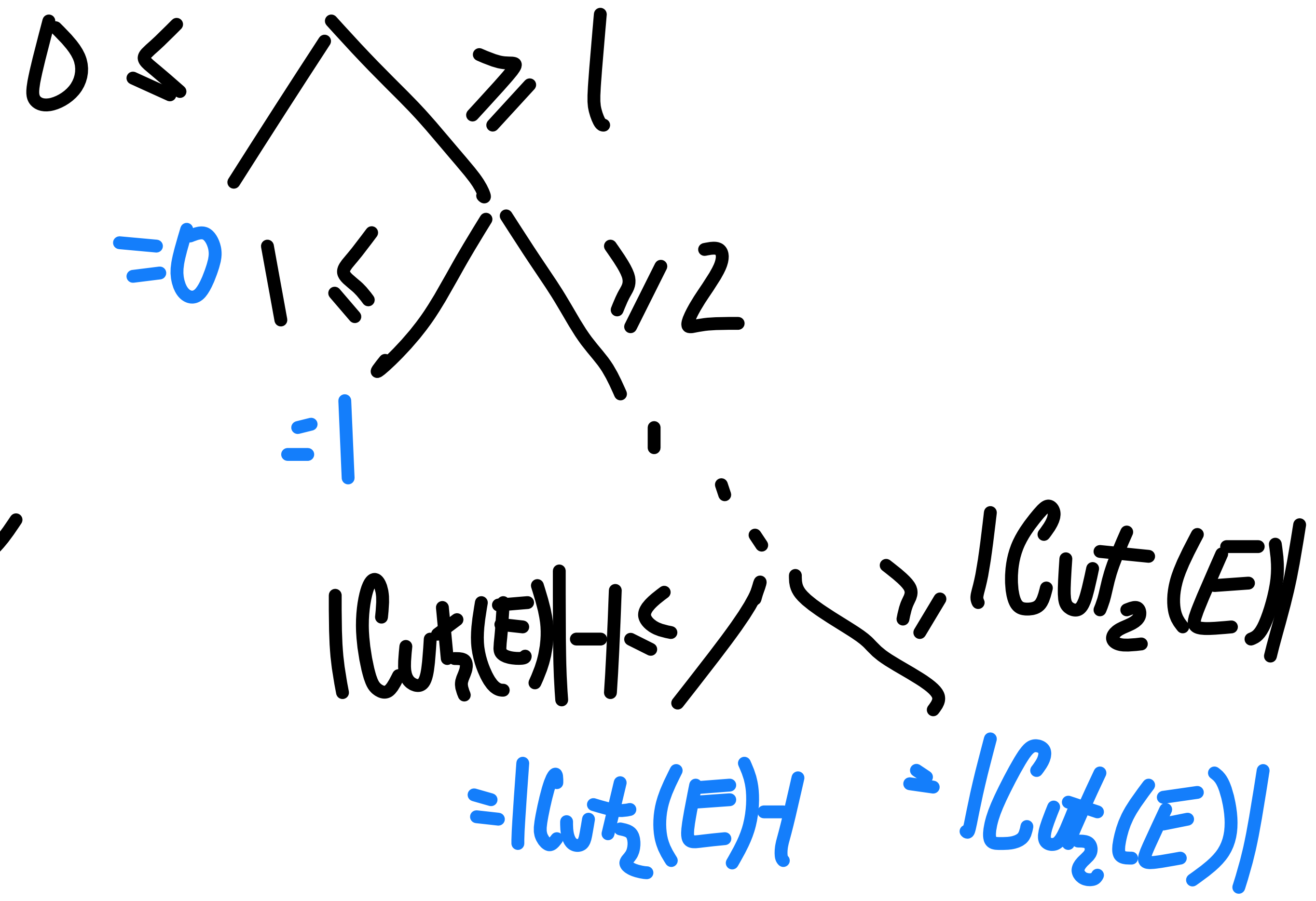
- Must be able to determine the parity of $Cut_2(E)$
- Query $\left(\sum_{i \in Cut_2(E)} x_i \leq t,\ \sum_{i \in Cut_2(E)} x_i \geq t+1\right)$ for $t = 0, \ldots, |Cut_2(E)|$



Refuting Systems of Equations In SP

To implement in SP:

- Must be able to determine the parity of $Cut_2(E)$
- Query $\left(\sum_{i \in Cut_2(E)} x_i \leq t, \sum_{i \in Cut_2(E)} x_i \geq t+1\right)$ for $t = 0, \ldots, |Cut_2(E)|$



- Once parity of $Cut_2(E_1)$ and of $Cut_2(E_2)$ determined, recurse on unsat side

Refuting Systems of Equations in SP

Complexity: $O(\log n)$ recursive rounds.

At most $|Cut_2(E_1)| \cdot |Cut_2(E_2)| \leq n^2$ queries per round

$\therefore$ $n^{O(\log n)}$ size

Depth of CP Proofs of Tseitin

- Both (ours and [DT20]) CP proofs of Tseitin are quasi-polynomially deep

Depth of CP Proofs of Tseitin

- Both (ours and [DT20]) CP proofs of Tseitin are quasi-polynomially deep
- Any CNF can be refuted in depth n

Depth of CP Proofs of Tseitin

- Both (ours and [DT20]) CP proofs of Tseitin are quasi-polynomially deep
- Any CNF can be refuted in depth n

Conjecture: Any quasi-polynomial size CP Proof of Tseitin requires superlinear depth

Towards the Conjecture

Conjecture: Any quasi-polynomial size CP Proof of Tseitin requires superlinear depth

Towards the Conjecture

Conjecture: Any quasi-polynomial size CP Proof of Tseitin requires superlinear depth

▷ Requires a better understanding of CP depth

Towards the Conjecture

Conjecture: Any quasi-polynomial size CP Proof of Tseitin requires superlinear depth

▷ Requires a better understanding of CP depth

Develop a new geometric technique for proving CP depth bounds

Towards the Conjecture

Conjecture: Any quasi-polynomial size CP Proof of Tseitin requires superlinear depth

▷ Requires a better understanding of CP depth

Develop a new geometric technique for proving CP depth bounds

Thm: Tseitin requires $\Omega(n)$ depth to refute in Semantic CP

Semantic CP

Semantic CP refutation of $P = \{Ax \geq b\} \subseteq [0,1]$:

$a_1 x \geq b_1$ $\quad$ $a_2 x \geq b_2$ $\quad \ldots \quad$ $a_m x \geq b_m$

Semantic CP

Semantic CP refutation of $P = \{Ax \geq b\} \subseteq [0,1]$:

Goal: Derive $0 \geq 1$

$0 \geq 1$

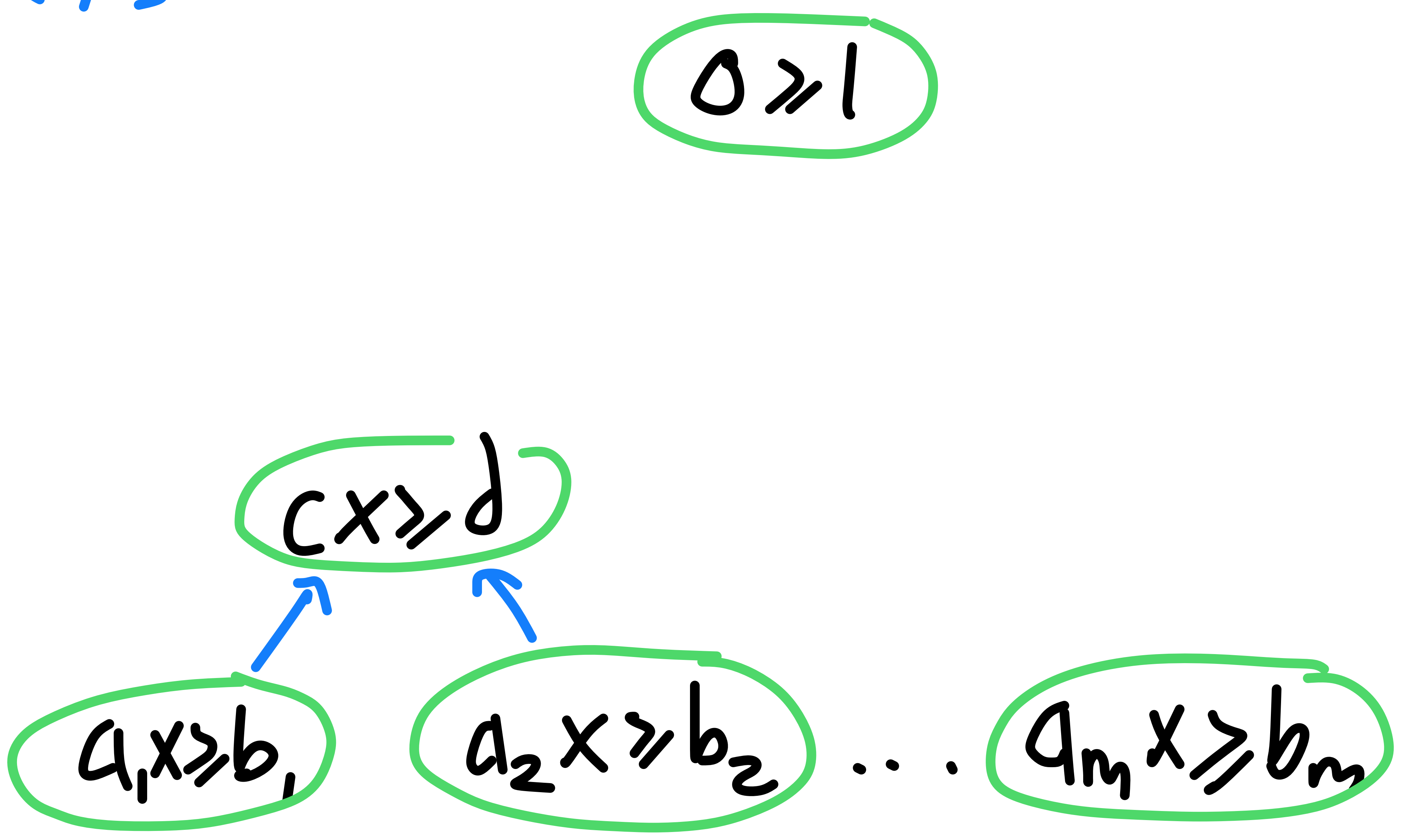
$a_1 x \geq b_1$ $\quad a_2 x \geq b_2$ $\quad \ldots \quad a_m x \geq b_m$

Semantic CP

Semantic CP refutation of $P = \{Ax \geq b\} \subseteq [0,1]$:

Goal: Derive $0 \geq 1$

By: Semantic Deduction: Derive $cx \geq d$ from previously derived $c_1 x \geq d_1$, $c_2 x \geq d_2$ if every $x \in \{0,1\}^n$ st $c_1 x \geq d_1$ and $c_2 x \geq d_2$ Satisfies $cx \geq d$

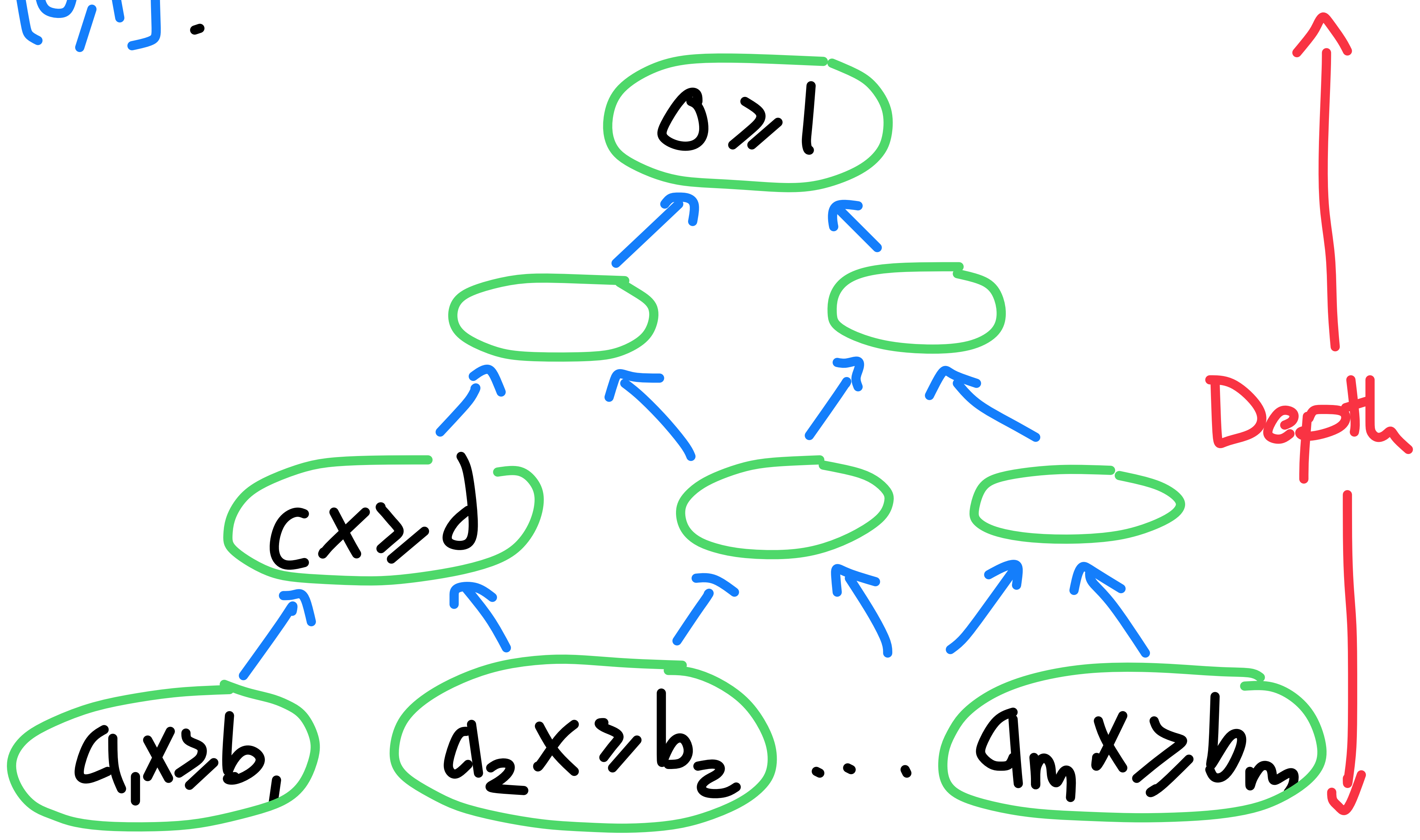


Semantic CP

Semantic CP refutation of $P = \{Ax \geq b\} \subseteq [0,1]$:

Goal: Derive $0 \geq 1$

By: Semantic Deduction: Derive $cx \geq d$ from previously derived $c_1 x \geq d_1$, $c_2 x \geq d_2$ if every $x \in \{0,1\}^n$ st $c_1 x \geq d_1$ and $c_2 x \geq d_2$ satisfies $cx \geq d$

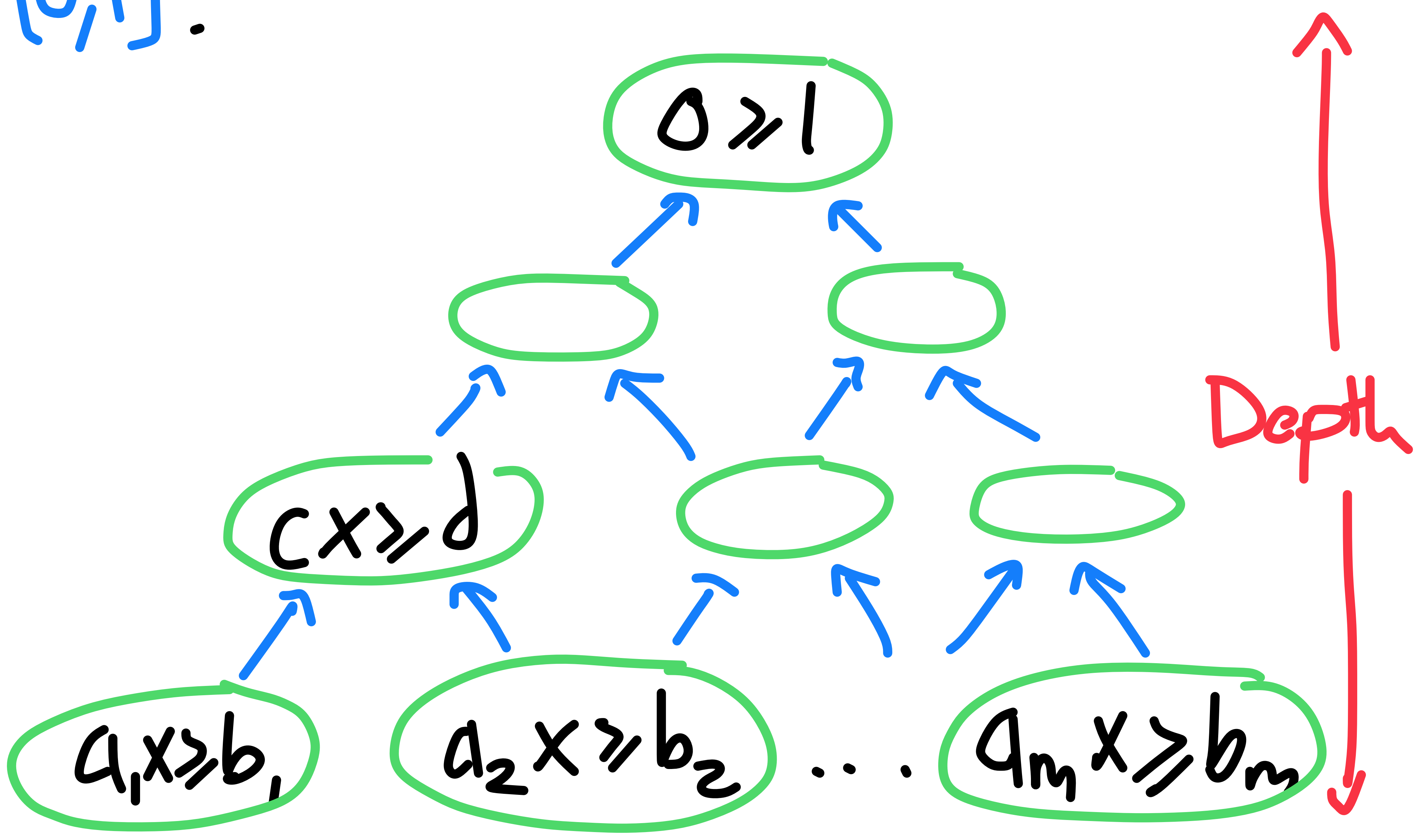


Semantic CP

Semantic CP refutation of $P = \{Ax \geq b\} \subseteq [0,1]$:

Goal: Derive $0 \geq 1$

By: Semantic Deduction: Derive $cx \geq d$ from previously derived $c_1 x \geq d_1$, $c_2 x \geq d_2$ if every $x \in \{0,1\}^n$ st $c_1 x \geq d_1$ and $c_2 x \geq d_2$ satisfies $cx \geq d$



Similar for CP, semantic deduction replaced by CG cut

Chvátal Closure

Previous depth lower bounds for CP were via reduction to Chvátal Rank or Communication Complexity lower bounds.

Chvátal Closure

Previous depth lower bounds for CP were via reduction to Chvátal Rank or Communication Complexity lower bounds.

- $O(\log n)$ upper bounds on Tseitin in CC

Chvátal Closure

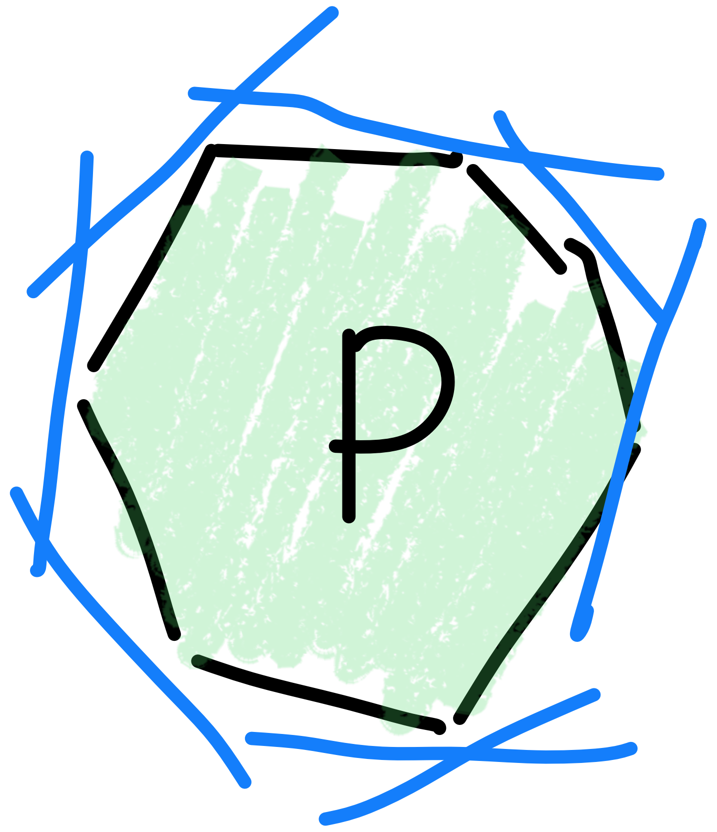
Previous depth lower bounds for CP were via reduction to Chvátal Rank or Communication Complexity lower bounds.

Chvátal Closure: of P is the polytope $P^{(1)}$ obtained by adding all CG cuts for P to P

Chvátal Closure

Previous depth lower bounds for CP were via reduction to Chvátal Rank or Communication Complexity lower bounds.

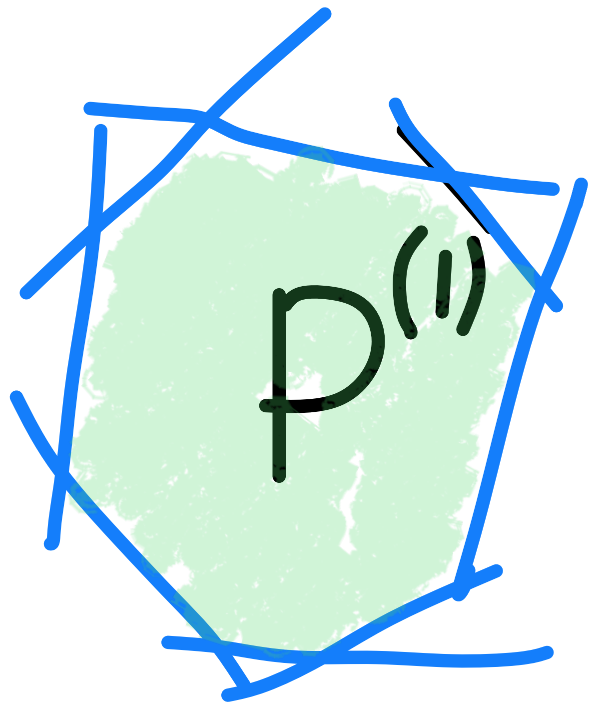
Chvátal Closure: of P is the polytope $P^{(1)}$ obtained by adding all CG cuts for P to P



Chvátal Closure

Previous depth lower bounds for CP were via reduction to Chvátal Rank or Communication Complexity lower bounds.

Chvátal Closure: of P is the polytope $P^{(1)}$ obtained by adding all CG cuts for P to P

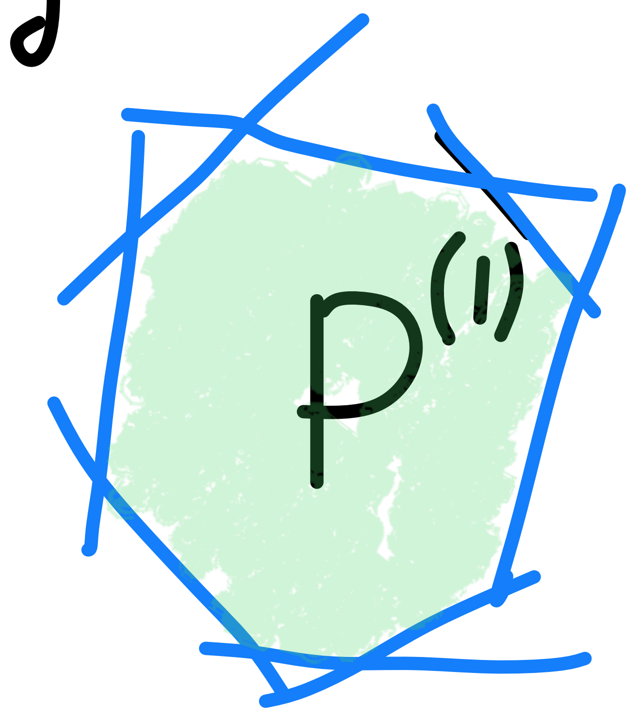


Chvátal Closure

Previous depth lower bounds for CP were via reduction to Chvátal Rank or Communication Complexity lower bounds.

Chvátal Closure: of P is the polytope $P^{(1)}$ obtained by adding all CG cuts for P to P

- Let $P^{(i)}$ be obtained from P by i rounds of Chvátal Closure



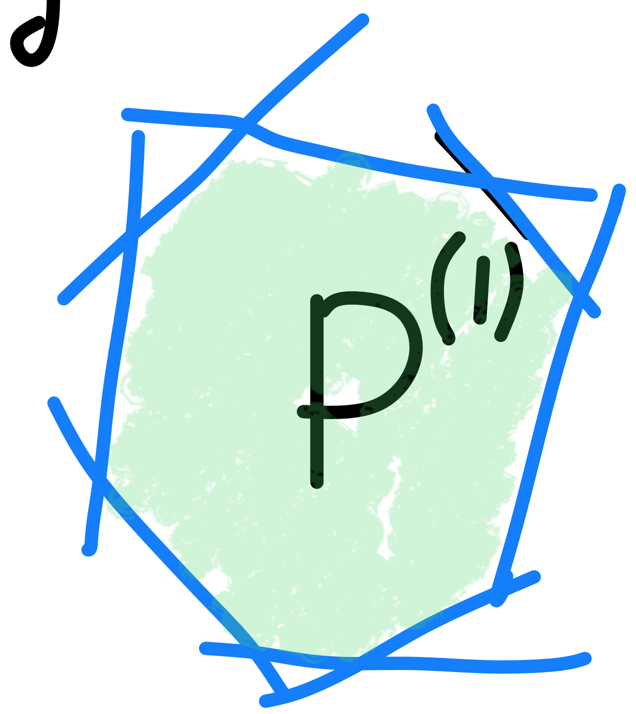
Chvátal Closure

Previous depth lower bounds for CP were via reduction to Chvátal Rank or Communication Complexity lower bounds.

Chvátal Closure: of P is the polytope $P^{(1)}$ obtained by adding all CG cuts for P to P

- Let $P^{(i)}$ be obtained from P by i rounds of Chvátal Closure

Chvátal Rank: Minimum t such that $P^{(t)} = \emptyset$



Chvátal Closure

Chvátal Closure: of P is the polytope $P^{(1)}$ obtained by adding all CG cuts for P to P

Chvátal Rank: Minimum t such that $P^{(t)} = \emptyset$

For a depth bound of d: Show $P^{(d)} \neq \emptyset$ ie find $p \in P^{(d)}$

Chvátal Closure

Chvátal Closure: of P is the polytope $P^{(1)}$ obtained by adding all CG cuts for P to P

Chvátal Rank: Minimum t such that $P^{(t)} = \emptyset$

For a depth bound of d: Show $P^{(d)} \neq \emptyset$ ie find $p \in P^{(d)}$

[BGHMP86]: $p \in [0,1]^n$, $p \in P^{(d)}$ if $\forall i$ with $p_i \in (0,1)$, $p \restriction (x_i = 0), p \restriction (x_i = 1) \in P^{(d-1)}$

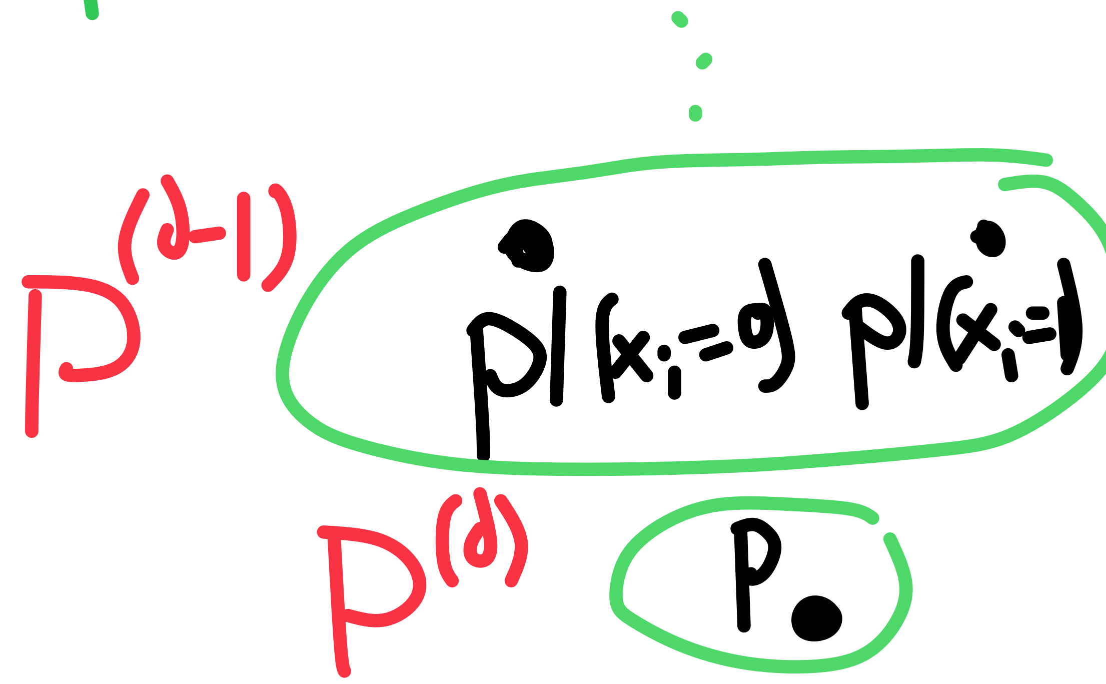
Chvátal Closure

Chvátal Closure: of P is the polytope $P^{(1)}$ obtained by adding all CG cuts for P to P

Chvátal Rank: Minimum t such that $P^{(t)} = \emptyset$

For a depth bound of d: Show $P^{(d)} \neq \emptyset$ ie find $p \in P^{(d)}$

[BGHMP86]: $p \in [0,1]^n$, $p \in P^{(d)}$ if $\forall i$ with $p_i \in (0,1)$, $p \upharpoonright (x_i = 0), p \upharpoonright (x_i = 1) \in P^{(d-1)}$



Chvátal Closure

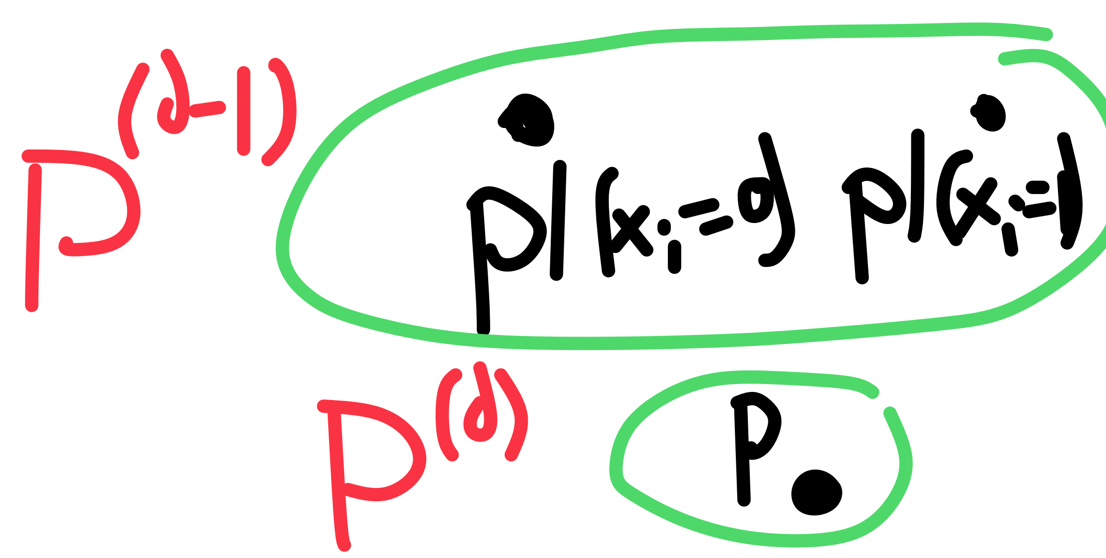
Chvátal Closure: of P is the polytope $P^{(1)}$ obtained by adding all CG cuts for P to P

Chvátal Rank: Minimum t such that $P^{(t)} = \emptyset$

For a depth bound of d: Show $P^{(d)} \neq \emptyset$ ie find $p \in P^{(d)}$

[BGHMP06]: $p \in [0,1]^n$, $p \in P^{(d)}$ if $\forall i$ with $p_i \in (0,1)$, $p\restriction(x_i=0), p\restriction(x_i=1) \in P^{(d-1)}$

- Proof relies on CG cuts



Chvátal Closure

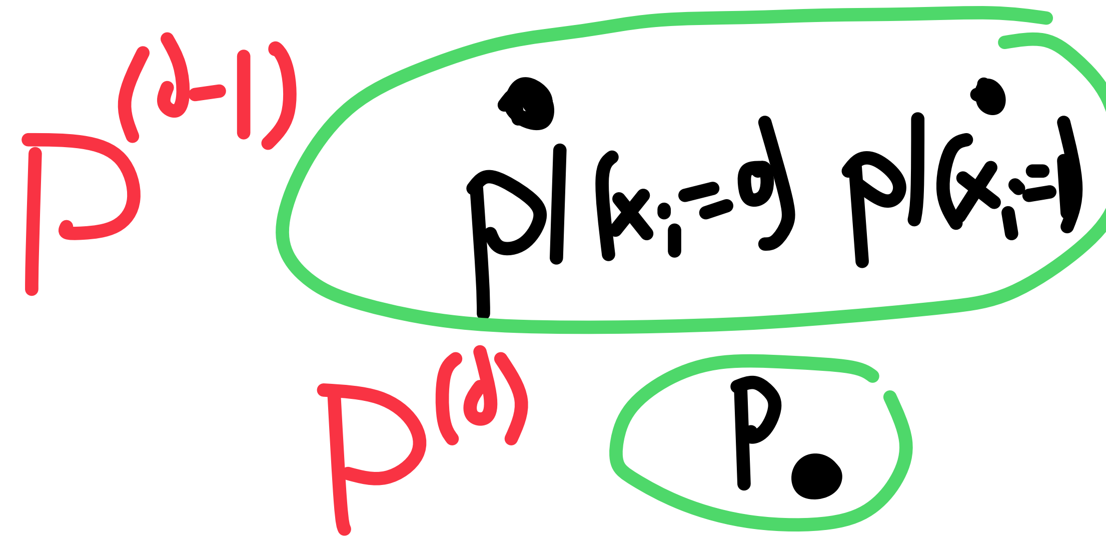
Chvátal Closure: of P is the polytope $P^{(1)}$ obtained by adding all CG cuts for P to P

Chvátal Rank: Minimum t such that $P^{(t)} = \emptyset$

For a depth bound of d: Show $P^{(d)} \neq \emptyset$ ie find $p \in P^{(d)}$

[BGHMP06]: $p \in [0,1]^n$, $p \in P^{(d)}$ if $\forall i$ with $p_i \in (0,1)$, $p\restriction(x_i=0),\ p\restriction(x_i=1) \in P^{(d-1)}$

- Proof relies on CG cuts
- No obvious way to account for proof size



Depth Lifting

We will show that DPLL depth can be lifted to semantic CP

Depth Lifting

We will show that DPLL depth can be lifted to semantic CP

XOR Lifting: From a CNF $F(z_1, \dots, z_n)$ obtain $F \circ XOR_4^n$ by replacing $z_i \leftarrow x_{i1} \oplus x_{i2} \oplus x_{i3} \oplus x_{i4}$

Depth Lifting

We will show that DPLL depth can be lifted to semantic CP

XOR Lifting: From a CNF $F(z_1, \ldots, z_n)$ obtain $F \circ XOR_4^n$ by replacing $z_i \leftarrow x_{i1} \oplus x_{i2} \oplus x_{i3} \oplus x_{i4}$

block 1 block 2 block n

$$\underbrace{x_{11}\ x_{12}\ x_{13}\ x_{14}}_{z_1}\ \underbrace{x_{21}\ x_{22}\ x_{23}\ x_{24}}_{z_2} \cdots \underbrace{x_{n1}\ x_{n2}\ x_{n3}\ x_{n4}}_{z_n}$$

Depth Lifting

We will show that DPLL depth can be lifted to semantic CP

XOR Lifting: From a CNF $F(z_1,\dots,z_n)$ obtain $F\circ XOR_4^n$ by replacing $z_i \leftarrow x_{i1}\oplus x_{i2}\oplus x_{i3}\oplus x_{i4}$

$$\underbrace{x_{11}\,x_{12}\,x_{13}\,x_{14}}_{z_1}^{\text{block 1}}\ \underbrace{x_{21}\,x_{22}\,x_{23}\,x_{24}}_{z_2}^{\text{block 2}}\cdots\underbrace{x_{n1}\,x_{n2}\,x_{n3}\,x_{n4}}_{z_n}^{\text{block } n}$$

Thm: For any F, $D_{sCP}(F\circ XOR_4)\geq D_{DPLL}(F)/2$

Depth Lifting

We will show that DPLL depth can be lifted to semantic CP

XOR Lifting: From a CNF $F(z_1,\dots,z_n)$ obtain $F \circ XOR_4^n$ by replacing $z_i \leftarrow x_{i1} \oplus x_{i2} \oplus x_{i3} \oplus x_{i4}$

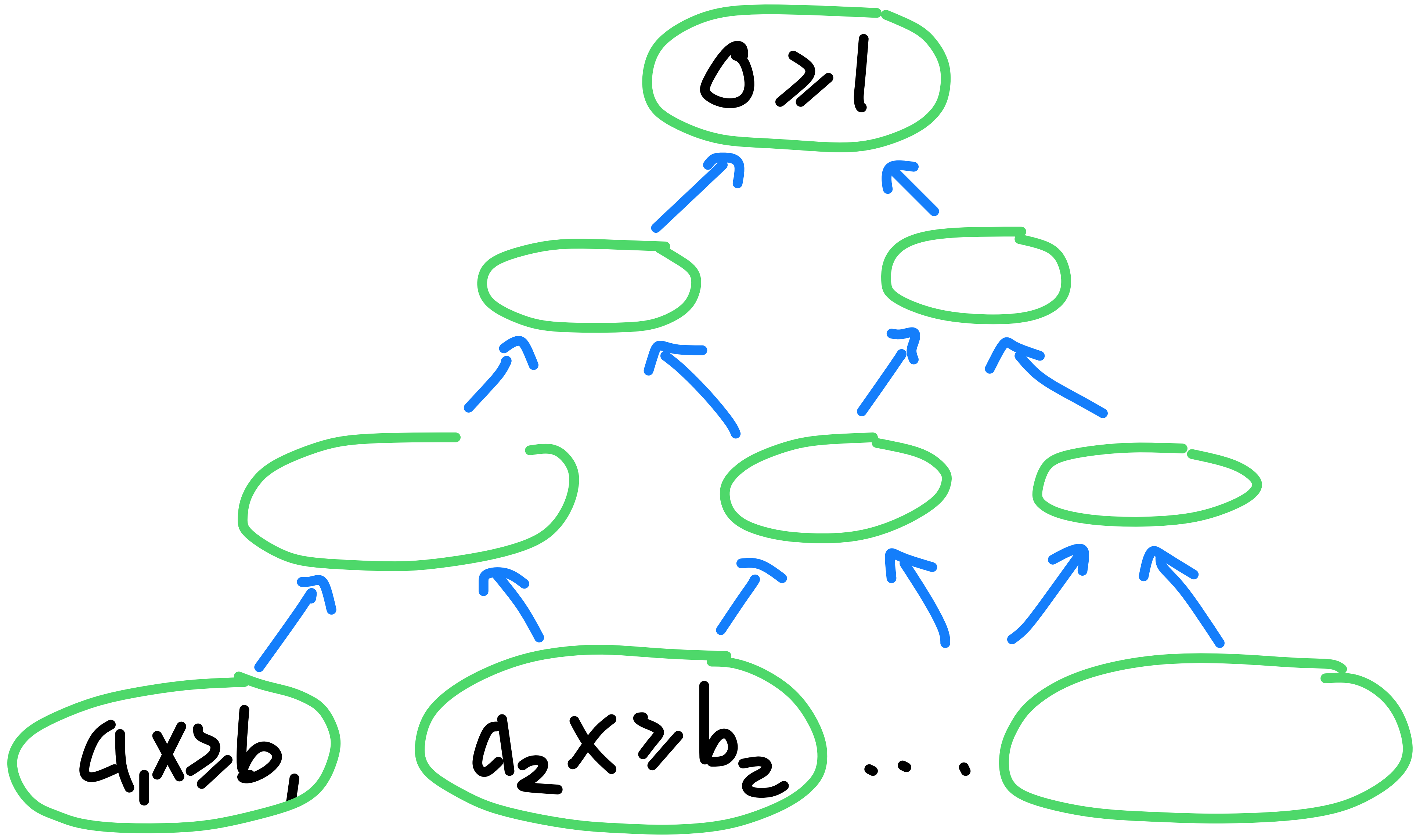
$$\overbrace{\underbrace{x_{11}\ x_{12}\ x_{13}\ x_{14}}_{z_1}}^{\text{block } 1}\ \overbrace{\underbrace{x_{21}\ x_{22}\ x_{23}\ x_{24}}_{z_2}}^{\text{block } 2} \dots \overbrace{\underbrace{x_{n1}\ x_{n2}\ x_{n3}\ x_{n4}}_{z_n}}^{\text{block } n}$$

Thm: For any F, $D_{sCP}(F \circ XOR_4) \geq D_{DPLL}(F)/2$

→ lower bound on Tseitin is proved directly by replacing lifting with expansion

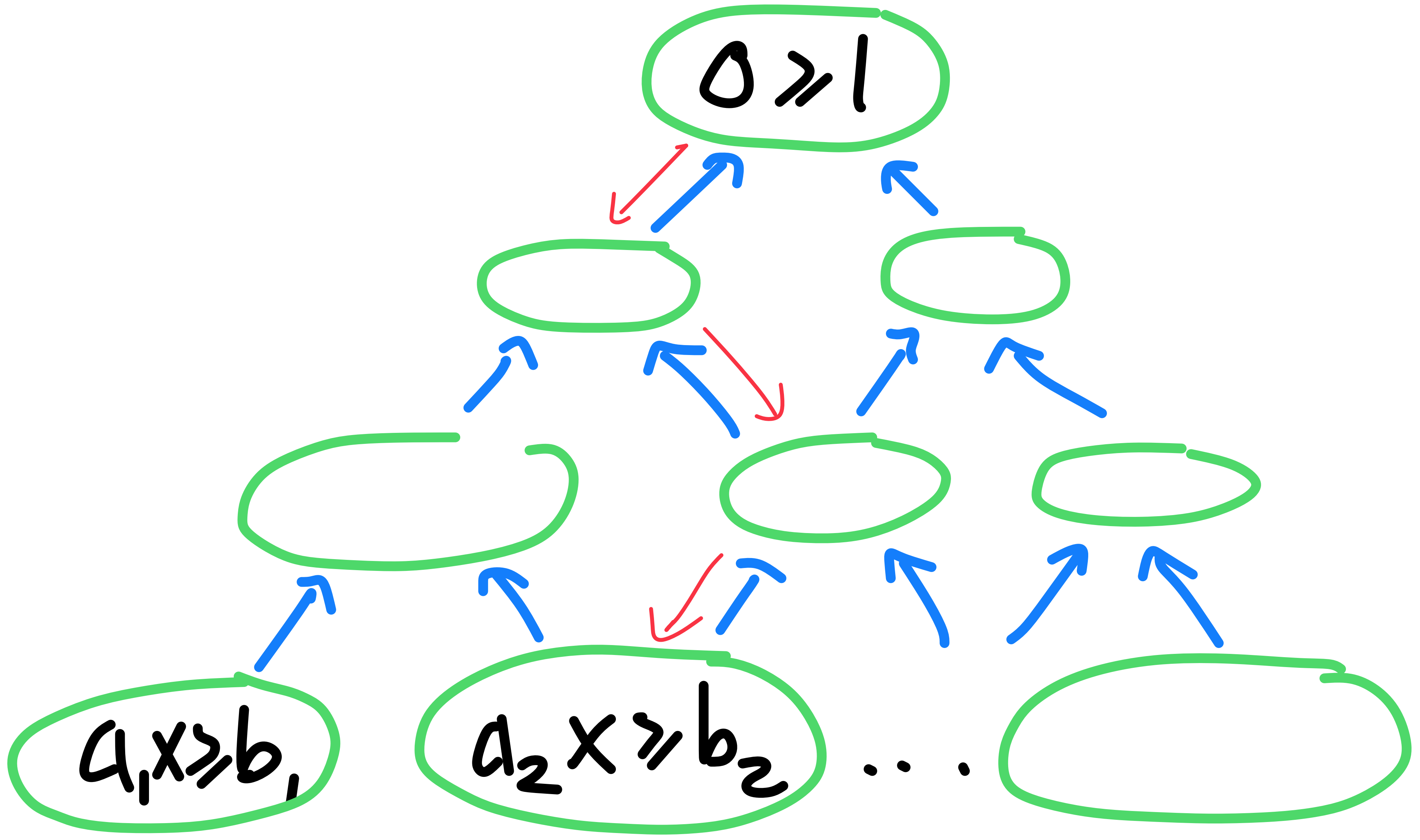
Depth Bounds for Semantic CP

Goal: Find a long root-to-leaf path in the proof



Depth Bounds for Semantic CP

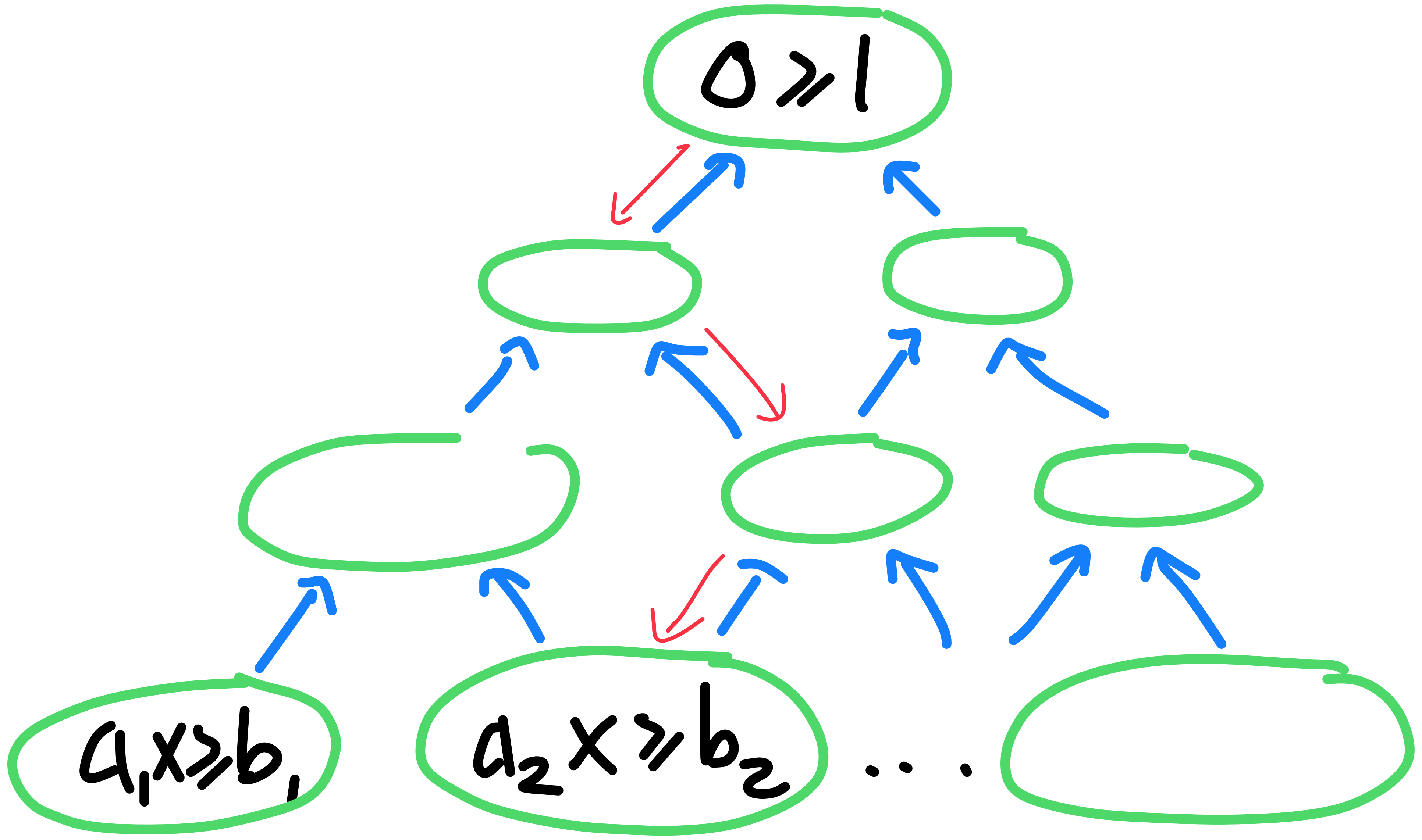
Goal: Find a long root-to-leaf path in the proof



Depth Bounds for Semantic CP

Goal: Find a long root-to-leaf path in the proof

Say halfspace H is good if $H(\frac{1}{2}^n) = 0$



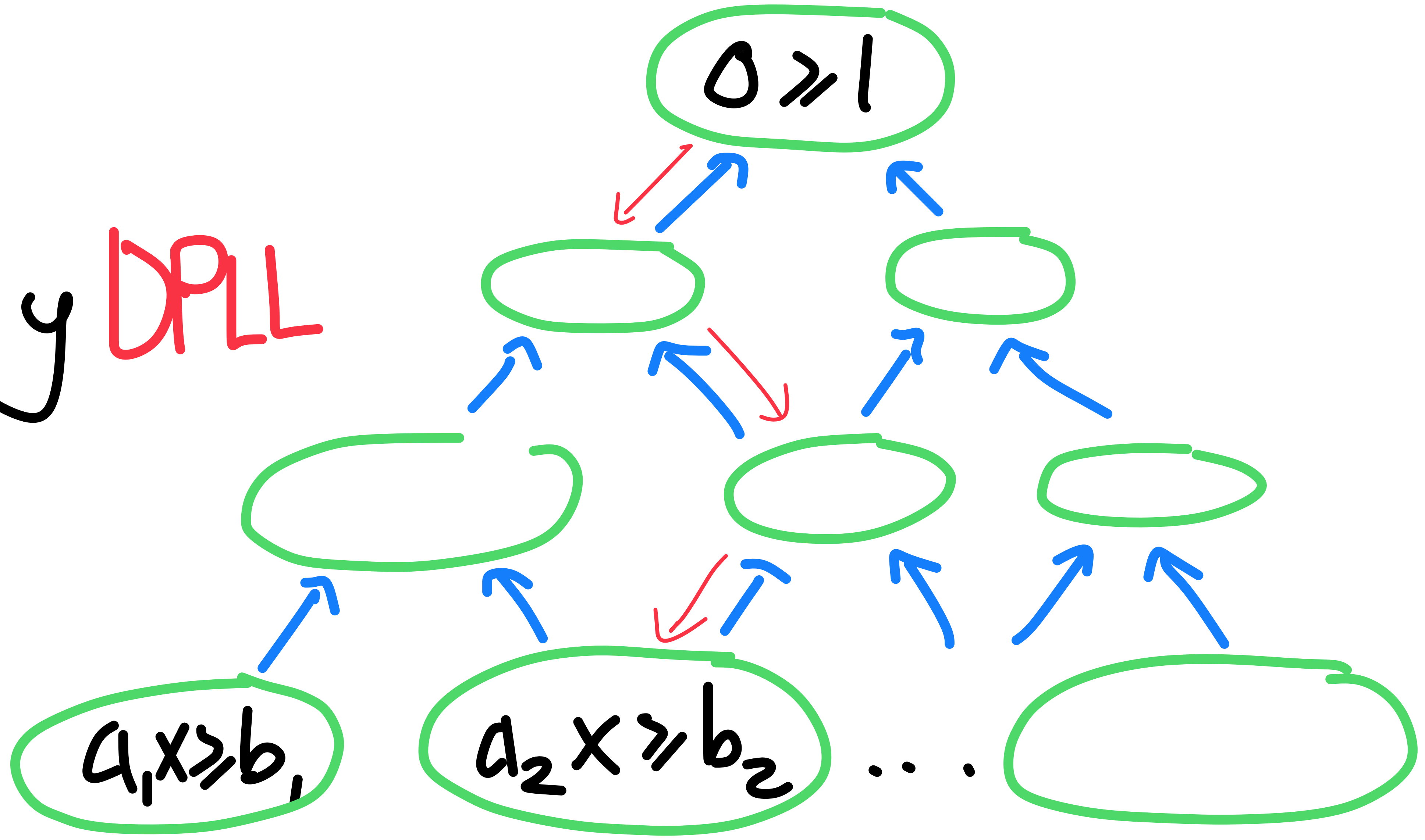
Depth Bounds for Semantic CP

Goal: Find a long root-to-leaf path in the proof

Say halfspace H is good if $H(\tfrac{1}{2}^{n}) = 0$

Idea: walk down the proof guided by DPLL

& a restriction $\rho \in \{0,1,*\}^{4n}$



Depth Bounds for Semantic CP

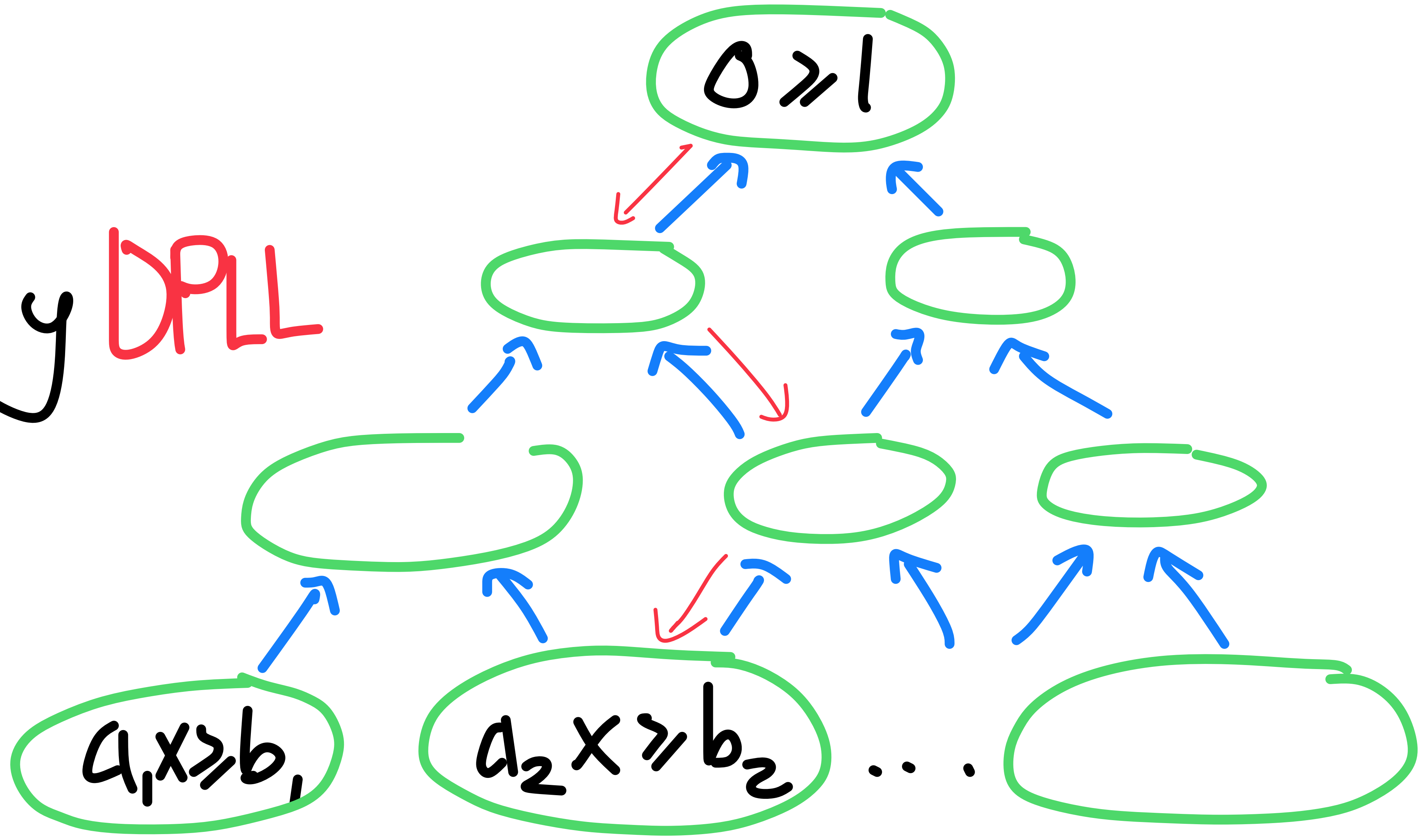
Goal: Find a long root-to-leaf path in the proof

Say halfspace H is good if $H(\frac{1}{2}^n) = 0$

Idea: walk down the proof guided by DPLL

& a restriction $\rho \in \{0,1,*\}^{4n}$ such that if

$H \restriction \rho$ is good then



Depth Bounds for Semantic CP

Goal: Find a long root-to-leaf path in the proof

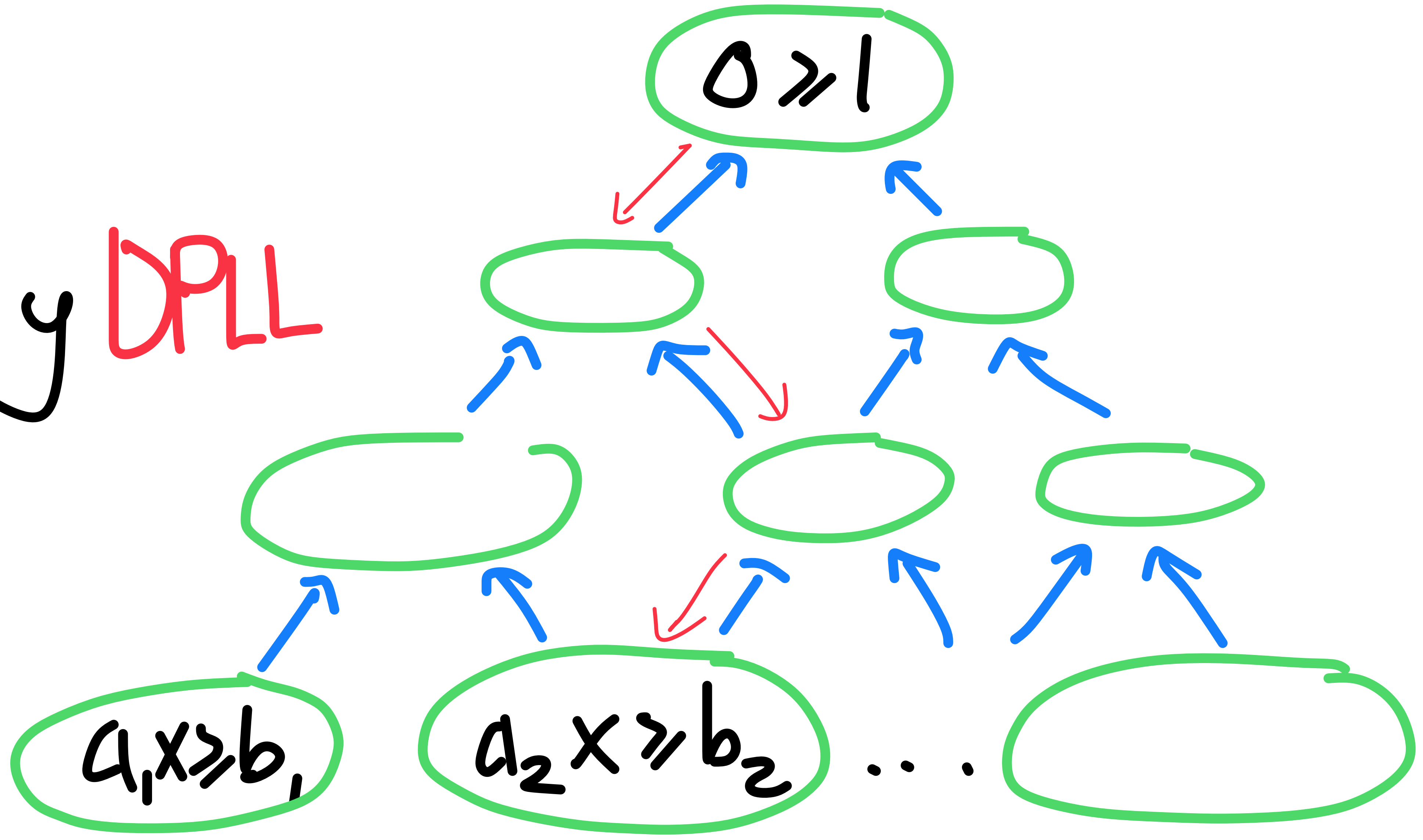
Say halfspace H is good if $H(\tfrac{1}{2}^n) = 0$

Idea: walk down the proof guided by DPLL

& a restriction $\rho \in \{0,1,*\}^{4n}$ such that if

$H \restriction \rho$ is good then

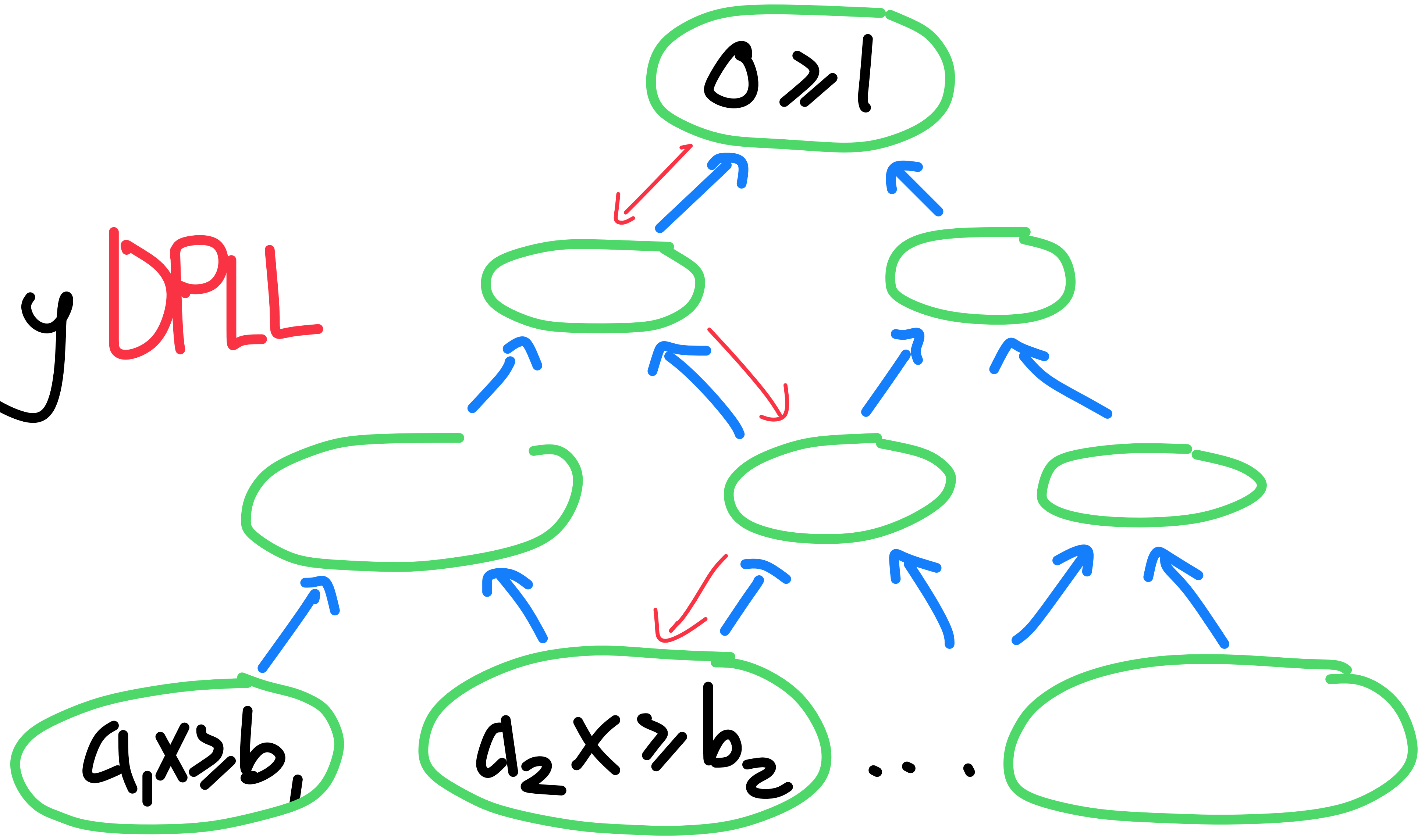
- H is not a leaf



Depth Bounds for Semantic CP

Goal: Find a long root-to-leaf path in the proof

Say halfspace H is good if $H(\frac{1}{2}^n) = 0$



Idea: walk down the proof guided by DPLL

& a restriction $\rho \in \{0,1,*\}^{4n}$ such that if

$H|\rho$ is good then

- H is not a leaf
- We can find a restriction ρ' such that one of the children H_1, H_2 of H is good under ρ'

Depth Bounds for Semantic CP

Say halfspace H is good if $H(\frac{1}{2}^n) = 0$

Technical Lemma: Let H_1, H_2 be the children of H and $H \restriction \rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1 \restriction \rho'$ or $H_2 \restriction \rho'$ is good

Depth Bounds for Semantic CP

Say halfspace H is good if $H(\frac{1}{2}^n) = 0$

Technical Lemma: Let H_1, H_2 be the children of H and $H \upharpoonright \rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1 \upharpoonright \rho'$ or $H_2 \upharpoonright \rho'$ is good

Thm: For any F, $D_{sCP}(F \circ XOR_4^n) \geq D_{DPLL}(F)/2$

Depth Bounds for Semantic CP

Say halfspace H is good if $H(\frac{1}{2}^n) = 0$

Technical Lemma: Let H_1, H_2 be the children of H and $H \restriction \rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1 \restriction \rho'$ or $H_2 \restriction \rho'$ is good

Thm: For any F, $D_{sCP}(F \circ XOR_4^n) \geq D_{DPLL}(F)/2$

Pf: Denote $D := D_{DPLL}(F)$

Depth Bounds for Semantic CP

Say halfspace H is good if $H(\frac{1}{2}^n) = 0$

Technical Lemma: Let H_1, H_2 be the children of H and $H \restriction \rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1 \restriction \rho'$ or $H_2 \restriction \rho'$ is good

Thm: For any F, $D_{sCP}(F \circ XOR_4^n) \geq D_{DPLL}(F)/2$

Pf: Denote $D := D_{DPLL}(F)$

Maintain the following invariants for H (initially $0 \geq 1$) and ρ (initially $*^n$)

Depth Bounds for Semantic CP

Say halfspace H is good if $H(1/2^n) = 0$

Technical Lemma: Let H_1, H_2 be the children of H and $H\restriction\rho$ be good.
Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1\restriction\rho'$ or $H_2\restriction\rho'$ is good

Thm: For any F, $D_{sCP}(F \circ XOR_4^n) \geq D_{DPLL}(F)/2$

Pf: Denote $D := D_{DPLL}(F)$
Maintain the following invariants for H (initially $0 \geq 1$) and ρ (initially $*^n$)
– Good: $H|\rho$ is good

Depth Bounds for Semantic CP

Say halfspace H is good if $H(\frac{1}{2}^n) = 0$

Technical Lemma: Let H_1, H_2 be the children of H and $H \upharpoonright \rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1 \upharpoonright \rho'$ or $H_2 \upharpoonright \rho'$ is good

Thm: For any F, $D_{sCP}(F \circ XOR_4^n) \geq D_{DPLL}(F)/2$

Pf: Denote $D := D_{DPLL}(F)$

Maintain the following invariants for H (initially $0 \geq 1$) and ρ (initially $*^n$)

- Good: $H \upharpoonright \rho$ is good
- Consistent: $XOR_4^n(\rho)$ doesn't falsify F

Depth Bounds for Semantic CP

Say halfspace H is good if $H(\frac{1}{2}^n) = 0$

Technical Lemma: Let H_1, H_2 be the children of H and $H\restriction\rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1\restriction\rho'$ or $H_2\restriction\rho'$ is good

Thm: For any F, $D_{sCP}(F \circ XOR_4^n) \geq D_{DPLL}(F)/2$

Pf: Denote $D := D_{DPLL}(F)$

Maintain the following invariants for H (initially $0 \geq 1$) and ρ (initially $*^n$)

- Good: $H\restriction\rho$ is good
- Consistent: $XOR_4^n(\rho)$ doesn't falsify F
- Block Closed: ρ sets all or none of the variables in every block

Depth Bounds for Semantic CP

Technical Lemma: Let H_1, H_2 be the children of H and $H \restriction \rho$ be good.
Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1 \restriction \rho'$ or $H_2 \restriction \rho'$ is good

Pf: Maintain the following invariants for H (initially $0 \geq 1$) and ρ (initially $*^n$)

- Good: $H|\rho$ is good
- Consistent: $XOR_4^n(\rho)$ doesn't falsify F
- Block Closed: ρ sets all or none of the variables in every block

Depth Bounds for Semantic CP

Technical Lemma: Let H_1, H_2 be the children of H and $H\restriction\rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1\restriction\rho'$ or $H_2\restriction\rho'$ is good

Pf: Maintain the following invariants for H (initially $0 \geq 1$) and ρ (initially $*^n$)

- Good: $H\restriction\rho$ is good
- Consistent: $XOR_4^n(\rho)$ doesn't falsify F
- Block Closed: ρ sets all or none of the variables in every block

Let (H, ρ) be current pair. Let (H_1, ρ') be given by technical lemma

Depth Bounds for Semantic CP

Technical Lemma: Let H_1, H_2 be the children of H and $H \restriction \rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1 \restriction \rho'$ or $H_2 \restriction \rho'$ is good

Pf: Maintain the following invariants for H (initially $0 \geq 1$) and ρ (initially $*^n$)

- Good: $H \restriction \rho$ is good
- Consistent: $XOR_{4}^{n}(\rho)$ doesn't falsify F
- Block Closed: ρ sets all or none of the variables in every block

Let (H, ρ) be current pair. Let (H_1, ρ') be given by technical lemma

Restore Invariants: Let $i \in [n]$ violate block closedness for ρ'

Depth Bounds for Semantic CP

Technical Lemma: Let H_1, H_2 be the children of H and $H \restriction \rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1 \restriction \rho'$ or $H_2 \restriction \rho'$ is good

Pf: Maintain the following invariants for H (initially $0 \geq 1$) and ρ (initially $*^n$)

- Good: $H|\rho$ is good
- Consistent: $XOR_4^n(\rho)$ doesn't falsify F
- Block Closed: ρ sets all or none of the variables in every block

Let (H, ρ) be current pair. Let (H_1, ρ') be given by technical lemma

Restore Invariants: Let $i \in [n]$ violate block closedness for ρ'. Set remaining variables in block i to restore the invariants.

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Restore Invariants: Let $i \in [n]$ violate block closedness for ρ'. Set remaining variables in block i to restore the invariants.

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

block 1, block 2, block n

$$\underbrace{x_{11}\ x_{12}\ x_{13}\ x_{14}}\ \underbrace{x_{21}\ x_{22}\ x_{23}\ x_{24}} \cdots \underbrace{x_{n1}\ x_{n2}\ x_{n3}\ x_{n4}}$$

Restore Invariants: Let $i \in [n]$ violate block closedness for ρ'. Set remaining variables in block i to restore the invariants.

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

$$\underbrace{x_{11}\ x_{12}\ x_{13}\ x_{14}}_{\text{block } 1},\ \underbrace{x_{21}\ x_{22}\ x_{23}\ x_{24}}_{\text{block } 2} \ldots \underbrace{x_{n1}\ x_{n2}\ x_{n3}\ x_{n4}}_{\text{block } n}$$

Restore Invariants: Let $i \in [n]$ violate block closedness for ρ'. Set remaining variables in block i to restore the invariants.

1. Query DPLL strategy on assignment $XOR_4^n(\rho')$ for $b \in \{0,1\}$ to set z_i to be consistent with a length D path

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

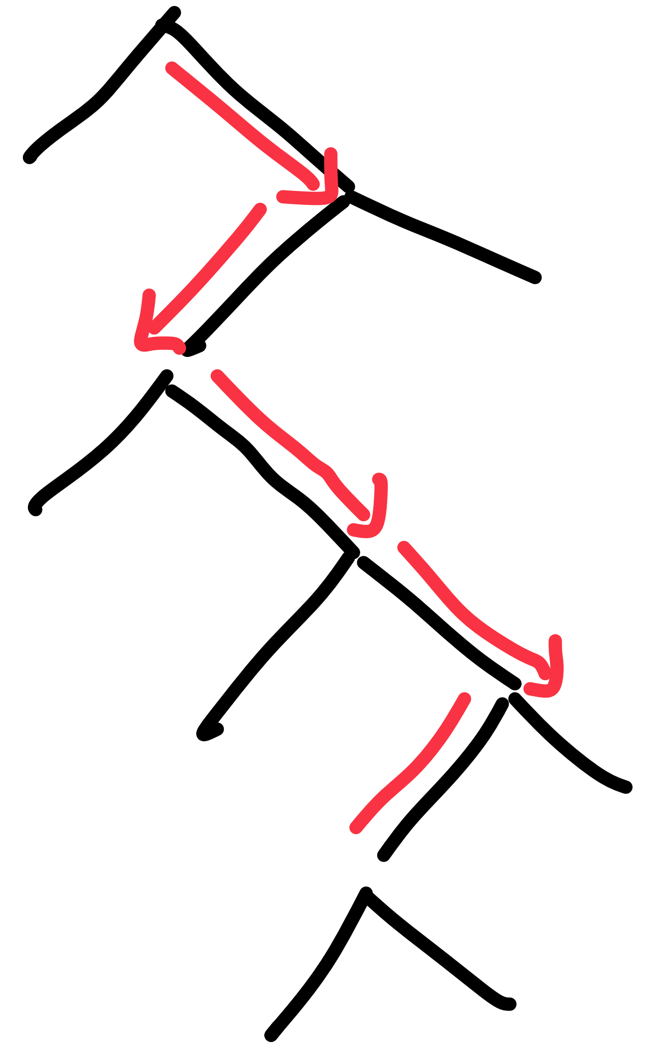
- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

block 1 block 2 block n

$\underbrace{x_{11}\, x_{12}\, x_{13}\, x_{14}}, \underbrace{x_{21}\, x_{22}\, x_{23}\, x_{24}} \ldots \underbrace{x_{n1}\, x_{n2}\, x_{n3}\, x_{n4}}$

Restore Invariants: Let $i \in [n]$ violate block closedness for ρ'. Set remaining variables in block i to restore the invariants.

1. Query DPLL strategy on assignment $XOR_4^n(\rho')$ for $b \in \{0,1\}$ to set z_i to be consistent with a length D path



Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

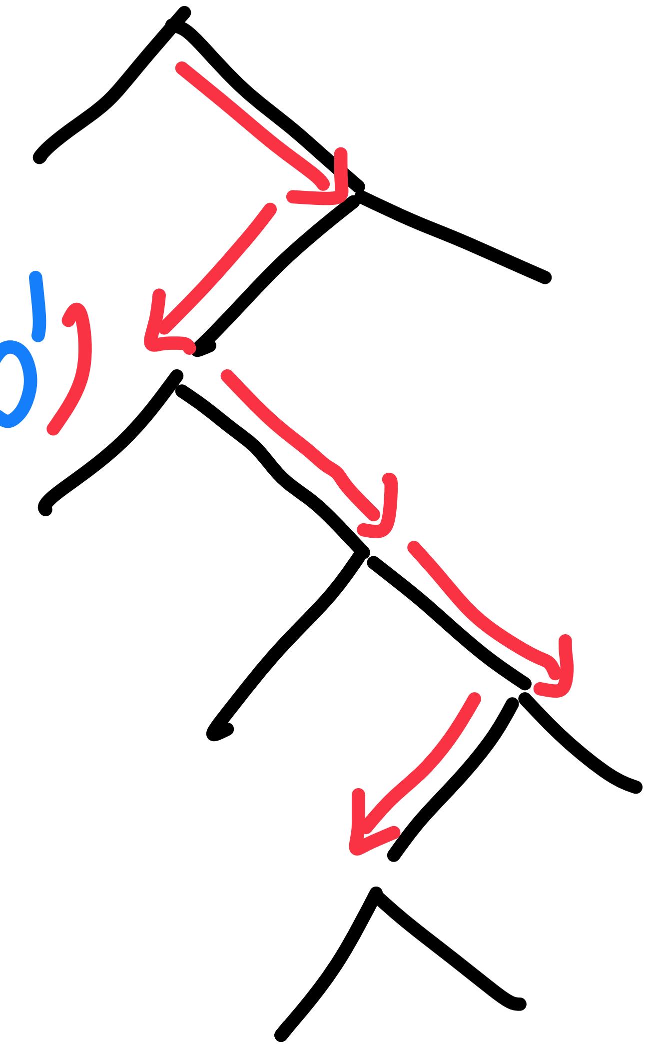
- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

$$\underbrace{x_{11}\ x_{12}\ x_{13}\ x_{14}}_{\text{block 1}},\ \underbrace{x_{21}\ x_{22}\ x_{23}\ x_{24}}_{\text{block 2}} \dots \underbrace{x_{n1}\ x_{n2}\ x_{n3}\ x_{n4}}_{\text{block } n}$$

Restore Invariants: Let $i \in [n]$ violate block closedness for ρ'. Set remaining variables in block i to restore the invariants.

1. Query DPLL strategy on assignment $XOR_4^n(\rho')$ for $b \in \{0,1\}$ to set z_i to be consistent with a length D path

$XOR_4^n(\rho')$



Depth Bounds for Semantic CP

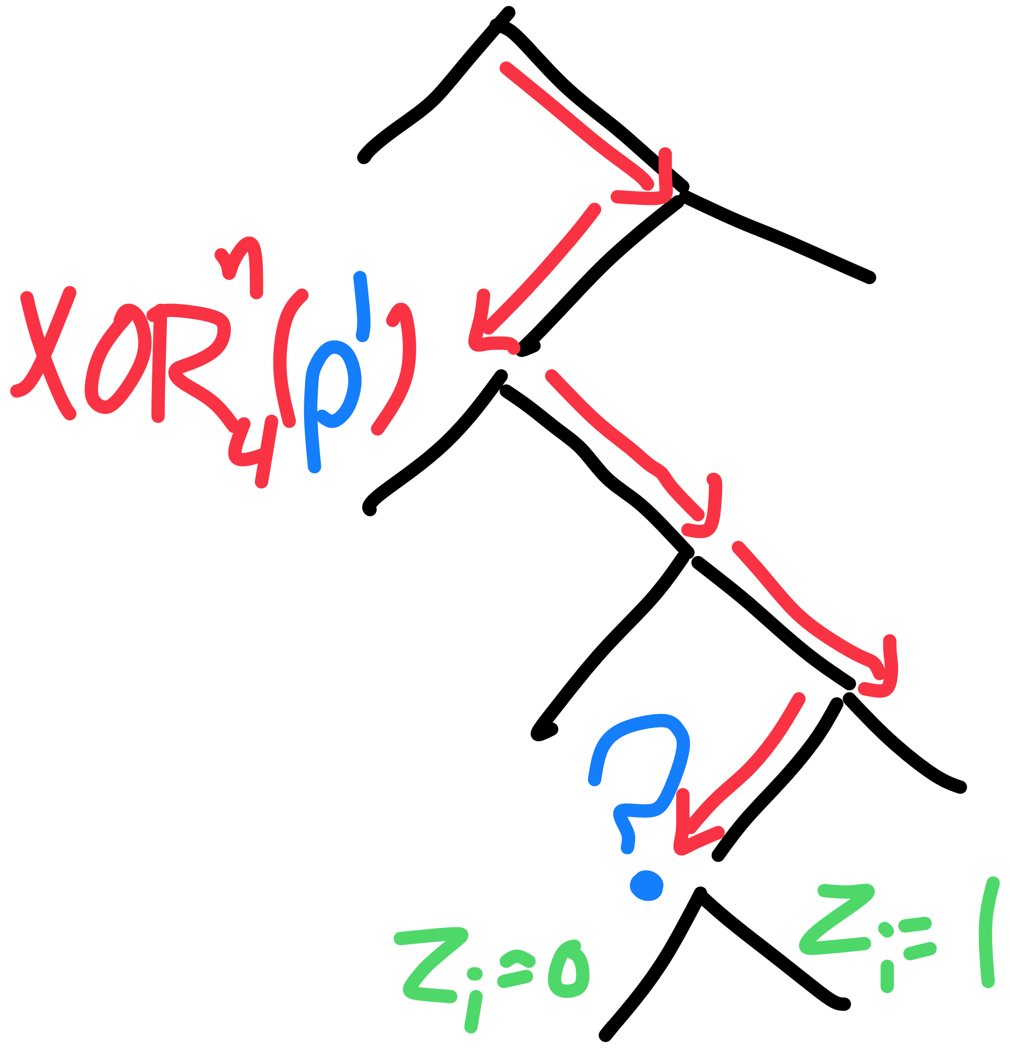
Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \upharpoonright \pi$ is good

$$\overset{\text{block 1}}{\underbrace{x_{11}\; x_{12}\; x_{13}\; x_{14}}}, \overset{\text{block 2}}{\underbrace{x_{21}\; x_{22}\; x_{23}\; x_{24}}} \dots \overset{\text{block } n}{\underbrace{x_{n1}\; x_{n2}\; x_{n3}\; x_{n4}}}$$

Restore Invariants: Let $i \in [n]$ violate block closedness for ρ'. Set remaining variables in block i to restore the invariants.

1. Query DPLL strategy on assignment $XOR_4^n(\rho')$ for $b \in \{0,1\}$ to set z_i to be consistent with a length D path



Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

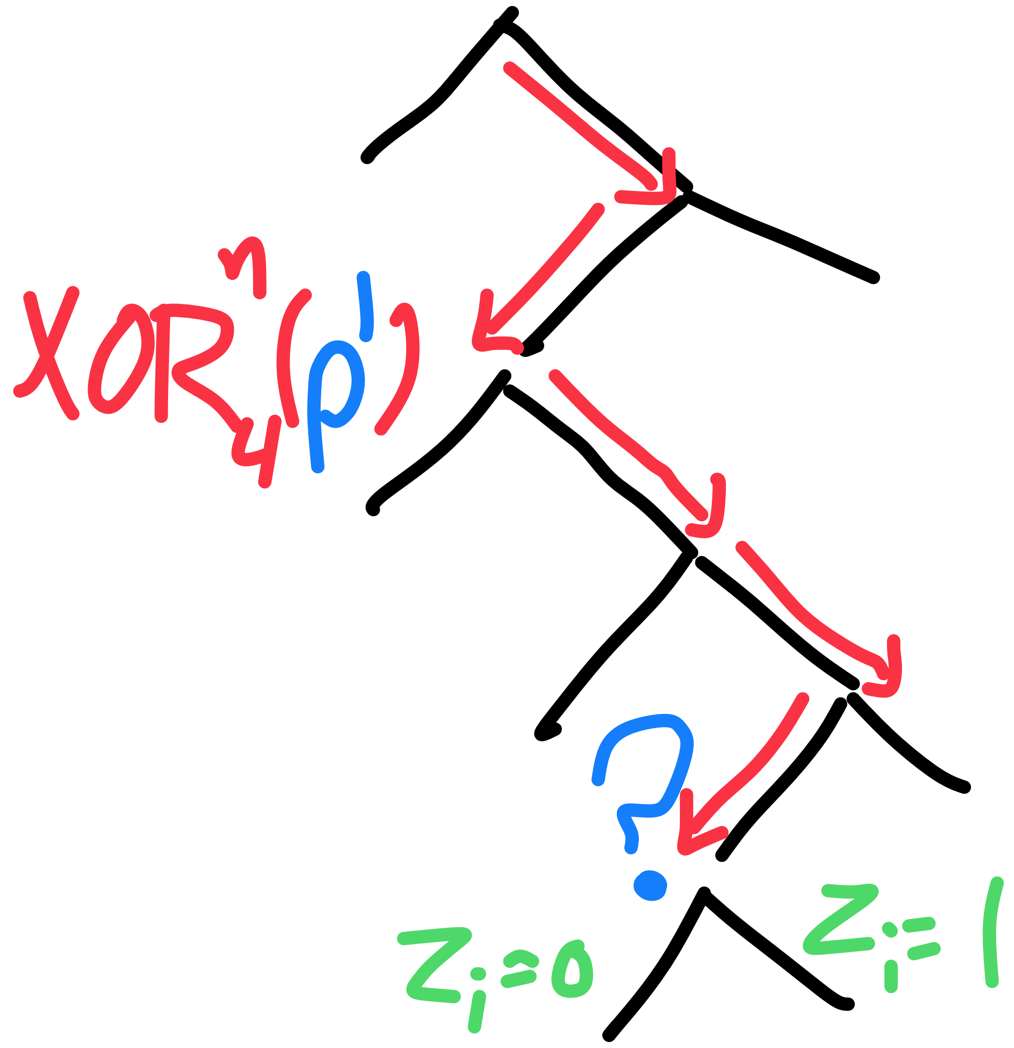
- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

block 1   block 2   block n

$\underbrace{x_{11}\, x_{12}\, x_{13}\, x_{14}}_{}, \underbrace{x_{21}\, x_{22}\, x_{23}\, x_{24}}_{} \ldots \underbrace{x_{n1}\, x_{n2}\, x_{n3}\, x_{n4}}_{}$

Restore Invariants: Let $i \in [n]$ violate block closedness for ρ'. Set remaining variables in block i to restore the invariants.

1. Query DPLL strategy on assignment $XOR_4^n(\rho')$ for $b \in \{0,1\}$ to set z_i to be consistent with a length D path
2. Apply Claim to $I = [\text{unset vars in block } i]$ to get π s.t.



Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

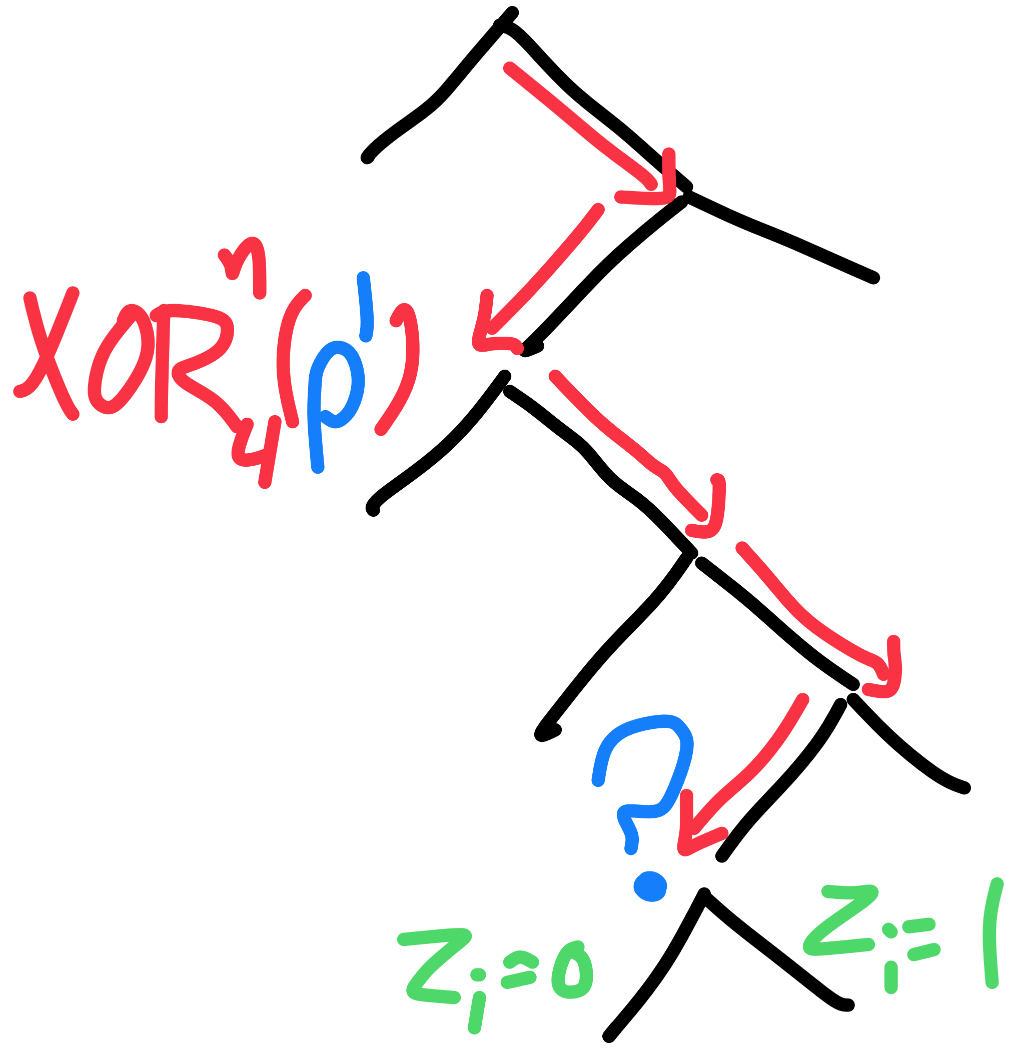
- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

block 1 block 2 block n

$\underbrace{x_{11}\ x_{12}\ x_{13}\ x_{14}}\ \underbrace{x_{21}\ x_{22}\ x_{23}\ x_{24}} \dots \underbrace{x_{n1}\ x_{n2}\ x_{n3}\ x_{n4}}$

Restore Invariants: Let $i \in [n]$ violate block closedness for ρ'. Set remaining variables in block i to restore the invariants.

1. Query DPLL strategy on assignment $XOR_4^n(\rho')$ for $b \in \{0,1\}$ to set z_i to be consistent with a length D path
2. Apply Claim to $I = [\text{unset vars in block } i]$ to get π s.t.
   - $\bigoplus_{j \in [4]} \rho'\pi(x_{ij}) = b$



Depth Bounds for Semantic CP

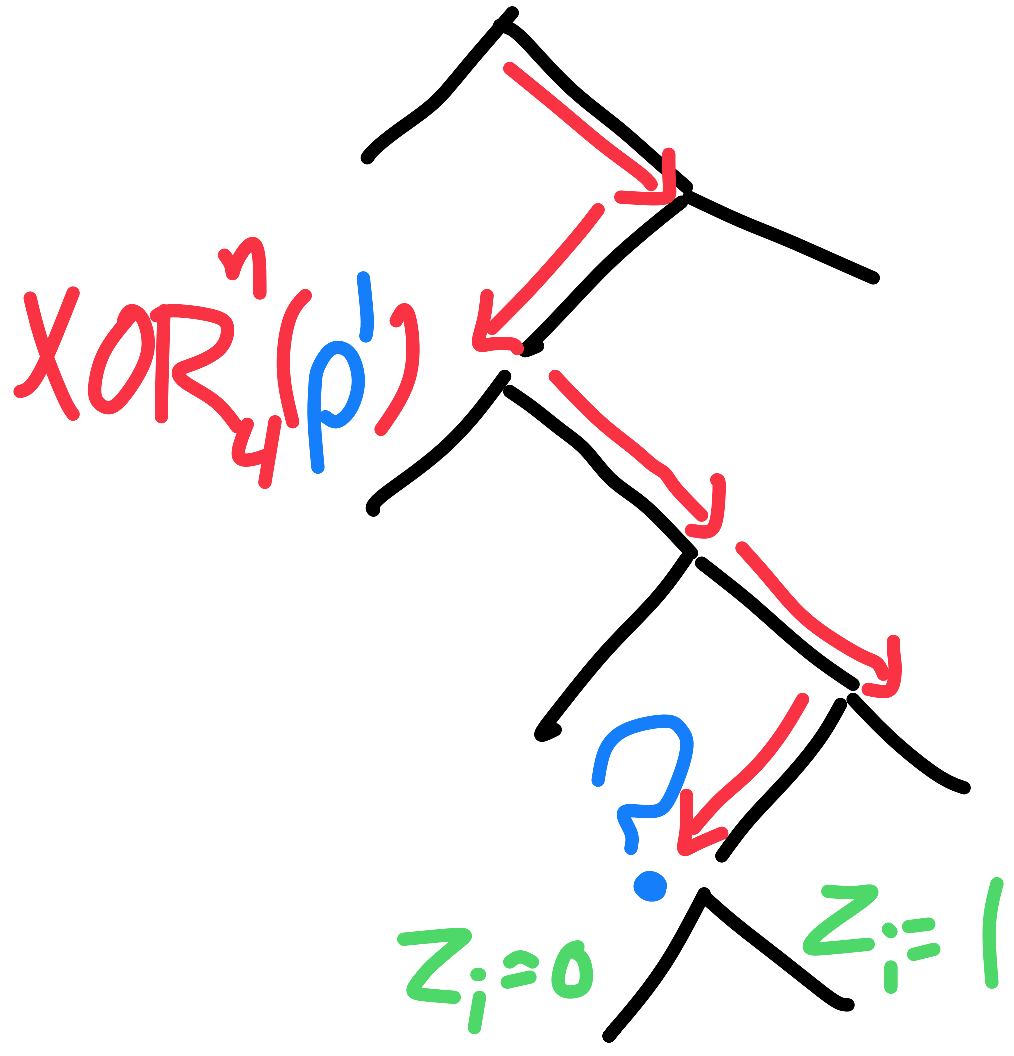
Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \upharpoonright \pi$ is good

$$\underbrace{x_{11}\ x_{12}\ x_{13}\ x_{14}}_{\text{block } 1}\ \underbrace{x_{21}\ x_{22}\ x_{23}\ x_{24}}_{\text{block } 2} \dots \underbrace{x_{n1}\ x_{n2}\ x_{n3}\ x_{n4}}_{\text{block } n}$$

Restore Invariants: Let $i \in [n]$ violate block closedness for ρ'. Set remaining variables in block i to restore the invariants.

1. Query DPLL strategy on assignment $XOR_4^n(\rho')$ for $b \in \{0,1\}$ to set z_i to be consistent with a length D path
2. Apply Claim to $I = [\text{unset vars in block } i]$ to get π s.t.
   - $\bigoplus_{j \in [4]} \rho'\pi(x_{ij}) = b$
   - $H \upharpoonright \rho'\pi$ is good



Depth Bounds for Semantic CP

Invariants: let $p \leftarrow p'\pi$ and $H \leftarrow H_1$

Depth Bounds for Semantic CP

Invariants: let $p \leftarrow p'\pi$ and $H \leftarrow H_1$

- Good: $H|p$ is good ✓

Depth Bounds for Semantic CP

Invariants: let $p \leftarrow p'\pi$ and $H \leftarrow H_1$

- Good: $H|p$ is good
- Consistent: $XOR_4^n(p)$ doesn't falsify F

Depth Bounds for Semantic CP

Invariants: let $p \leftarrow p'\pi$ and $H \leftarrow H_1$

- Good: $H|p$ is good ✓
- Consistent: $XOR_4^n(p)$ doesn't falsify F ✓

$XOR_4^n(p)$ is consistent with a path in the DPLL tree of depth D

Depth Bounds for Semantic CP

Invariants: let $\rho \leftarrow \rho'\pi$ and $H \leftarrow H_1$

- Good: $H|\rho$ is good ✓
- Consistent: $XOR_4^n(\rho)$ doesn't falsify F ✓
- Block Closed: ρ sets all or none of the variables in every block ✓

$XOR_4^n(\rho)$ is consistent with a Path in the DPLL tree of depth D

Depth Bounds for Semantic CP

Invariants: let $\rho \leftarrow \rho'\pi$ and $H \leftarrow H_1$

- Good: $H|\rho$ is good ✓
- Consistent: $XOR_4^n(\rho)$ doesn't falsify F ✓
- Block Closed: ρ sets all or none of the variables in every block ✓

$XOR_4^n(\rho)$ is consistent with a path in the DPLL tree of depth D

Thus we can proceed to the next round.

Depth Bounds for Semantic CP

Invariants: let $p \leftarrow p'\pi$ and $H \leftarrow H_1$

- Good: $H|p$ is good ✓
- Consistent: $XOR_4^n(p)$ doesn't falsify F ✓
- Block Closed: p sets all or none of the variables in every block ✓

$XOR_4^n(p)$ is consistent with a path in the DPLL tree of depth D

Thus we can proceed to the next round.

- Each round sets ≤ 2 blocks

Depth Bounds for Semantic CP

Invariants: let $\rho \leftarrow \rho'\pi$ and $H \leftarrow H_1$

- Good: $H|\rho$ is good ✓
- Consistent: $XOR_4^n(\rho)$ doesn't falsify F ✓
- Block Closed: ρ sets all or none of the variables in every block ✓

$XOR_4^n(\rho)$ is consistent with a path in the DPLL tree of depth D

Thus we can proceed to the next round.

- Each round sets ≤ 2 blocks $\Rightarrow$ we can proceed for $D_{DT}(F)/2$ rounds.

Depth Bounds for Semantic CP

Invariants: let $\rho \leftarrow \rho'\pi$ and $H \leftarrow H_1$

- Good: $H|\rho$ is good ✓
- Consistent: $XOR_4^n(\rho)$ doesn't falsify F ✓
- Block Closed: ρ sets all or none of the variables in every block ✓

$XOR_4^n(\rho)$ is consistent with a Path in the DPLL tree of depth D

Thus we can proceed to the next round.

- Each round sets ≤ 2 blocks $\Rightarrow$ we can proceed for $D_{DT}(F)/2$ rounds.

$$\therefore D_{sCP}(F \circ XOR_4^n) \geq D_{DPLL}(F)/2$$

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Main tool: Monotonicity of halfspaces

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Main tool: Monotonicity of halfspaces

Proof: Let $H := \sum a_i x_i \geq b$ and $I = \{1, \ldots, k\}$.

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \upharpoonright \pi$ is good

Main tool: Monotonicity of halfspaces

Proof: Let $H := \sum a_i x_i \geq b$ and $I = \{1, \dots, k\}$. Suppose wlog $|a_1| \geq |a_2| \geq \dots \geq |a_k|$.

Depth Bounds for Semantic CP

Claim: Let $H \in \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Main tool: Monotonicity of halfspaces

Proof: Let $H := \sum a_i x_i \geq b$ and $I = \{1, .., k\}$. Suppose wlog $|a_1| \geq |a_2| \geq \cdots \geq |a_k|$.

e.g. $b=0$, $3x_1 - 2x_2 - x_3 + x_4 \geq 2$

$I = \{1,2,3\}$

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Main tool: Monotonicity of halfspaces

Proof: Let $H := \sum a_i x_i \geq b$ and $I = \{1, \ldots, k\}$. Suppose wlog $|a_1| \geq |a_2| \geq \cdots \geq |a_k|$. Construct π as follows:

e.g. $b=0$, $3x_1 - 2x_2 - x_3 + x_4 \geq 2$

$I = \{1,2,3\}$

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Main tool: Monotonicity of halfspaces

Proof: Let $H := \sum a_i x_i \geq b$ and $I = \{1, \ldots, k\}$. Suppose wlog $|a_1| \geq |a_2| \geq \cdots \geq |a_k|$. Construct π as follows:

e.g. $b=0$, $3x_1 - 2x_2 - x_3 + x_4 \geq 2$, $I = \{1,2,3\}$

- For $i \in \{1, \ldots, k-1\}$ set $x_i \leftarrow 0$ if $a_i \geq 0$, $x_i \leftarrow 1$ o.w.

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Main tool: Monotonicity of halfspaces

Proof: Let $H := \sum a_i x_i \geq b$ and $I = \{1, .., k\}$. Suppose wlog $|a_1| \geq |a_2| \geq \cdots \geq |a_k|$. Construct π as follows:

- For $i \in \{1, \dots, k-1\}$ set $x_i \leftarrow 0$ if $a_i \geq 0$, $x_i \leftarrow 1$ o.w.

e.g. $b=0$, $3x_1 - 2x_2 - x_3 + x_4 \geq 2$

$I = \{1,2,3\}$

$$\begin{array}{cccc} 3x_1 & -2x_2 & -x_3 & +x_4 \geq 2 \\ \uparrow & \uparrow & & \\ 0 & \underline{1} & & \end{array}$$

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Main tool: Monotonicity of halfspaces

Proof: Let $H := \sum a_i x_i \geq b$ and $I = \{1, \ldots, k\}$. Suppose wlog $|a_1| \geq |a_2| \geq \cdots \geq |a_k|$. Construct π as follows:

- For $i \in \{1, \ldots, k-1\}$ set $x_i \leftarrow 0$ if $a_i \geq 0$, $x_i \leftarrow 1$ o.w.
- Set x_k so $\bigoplus_{i \in I} \pi(x_i) = b$

e.g. $b=0$, $3x_1 - 2x_2 - x_3 + x_4 \geq 2$

$I = \{1,2,3\}$

$$\begin{array}{cc} 3x_1 & -2x_2 \\ \uparrow & \uparrow \\ 0 & 1 \end{array}$$

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Main tool: Monotonicity of halfspaces

Proof: Let $H := \sum a_i x_i \geq b$ and $I = \{1, \ldots, k\}$. Suppose wlog $|a_1| \geq |a_2| \geq \cdots \geq |a_k|$. Construct π as follows:

- For $i \in \{1, \ldots, k-1\}$ set $x_i \leftarrow 0$ if $a_i \geq 0$, $x_i \leftarrow 1$ o.w.
- Set x_k so $\bigoplus_{i \in I} \pi(x_i) = b$

e.g. $b = 0$, $I = \{1,2,3\}$

$$\begin{array}{cccc} 3x_1 & -2x_2 & -x_3 & +x_4 \geq 2 \\ \uparrow & \uparrow & \uparrow & \\ 0 & \underline{1} & \underline{1} & \end{array}$$

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Main tool: Monotonicity of halfspaces

Proof: Let $H := \sum a_i x_i \geq b$ and $I = \{1, \ldots, k\}$. Suppose wlog $|a_1| \geq |a_2| \geq \cdots \geq |a_k|$. Construct π as follows:

- For $i \in \{1, \ldots, k-1\}$ set $x_i \leftarrow 0$ if $a_i \geq 0$, $x_i \leftarrow 1$ o.w.
- Set x_k so $\bigoplus_{i \in I} \pi(x_i) = b$

e.g. $b = 0$, $3x_1 - 2x_2 - x_3 + x_4 \geq 2$

$I = \{1,2,3\}$

$$\begin{array}{ccc} 3x_1 & -2x_2 & -x_3 \\ \uparrow & \uparrow & \uparrow \\ 0 & 1 & 1 \end{array}$$

"Setting $a_1, \ldots, a_{k-1}$ to compensate for a_k"

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Main tool: Monotonicity of halfspaces

Proof: Let $H := \sum a_i x_i \geq b$ and $I = \{1, \ldots, k\}$. Suppose wlog $|a_1| \geq |a_2| \geq \cdots \geq |a_k|$. Construct π as follows:

- For $i \in \{1, \ldots, k-1\}$ set $x_i \leftarrow 0$ if $a_i \geq 0$, $x_i \leftarrow 1$ o.w.
- Set x_k so $\bigoplus_{i \in I} \pi(x_i) = b$

e.g. $b=0$, $I = \{1,2,3\}$

$$\begin{array}{cccc} 3x_1 & -2x_2 & -x_3 & +x_4 \geq 2 \\ \uparrow & \uparrow & \uparrow & \\ 0 & \underline{1} & \underline{1} & \end{array}$$

Wts $H \restriction \pi$ is good

"Setting $a_1, \ldots, a_{k-1}$ to compensate for a_k"

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Main tool: Monotonicity of halfspaces

Proof: Let $H := \sum a_i x_i \geq b$ and $I = \{1, .., k\}$. Suppose wlog $|a_1| \geq |a_2| \geq \cdots \geq |a_k|$. Construct π as follows:

- For $i \in \{1, \ldots, k-1\}$ set $x_i \leftarrow 0$ if $a_i \geq 0$, $x_i \leftarrow 1$ o.w.
- Set x_k so $\bigoplus_{i \in I} \pi(x_i) = b$

e.g. $b = 0$, $3x_1 - 2x_2 - x_3 + x_4 \geq 2$

$I = \{1,2,3\}$

$$\begin{array}{ccc} 3x_1 & -2x_2 & -x_3 \\ \uparrow & \uparrow & \uparrow \\ 0 & 1 & 1 \end{array}$$

WTS $H \restriction \pi$ is good

"Setting $a_1, \ldots, a_{k-1}$ to compensate for a_k"

$$(H \restriction \pi)(\tfrac{1}{2}^k) = 3(0) - 2(1) - (1) + (\tfrac{1}{2})$$

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Main tool: Monotonicity of halfspaces

Proof: Let $H := \sum a_i x_i \geq b$ and $I = \{1, .., k\}$. Suppose wlog $|a_1| \geq |a_2| \geq \cdots \geq |a_k|$. Construct π as follows:

- For $i \in \{1, \ldots, k-1\}$ set $x_i \leftarrow 0$ if $a_i \geq 0$, $x_i \leftarrow 1$ o.w.
- Set x_k so $\bigoplus_{i \in I} \pi(x_i) = b$

e.g. $b=0$, $I = \{1,2,3\}$: $\underset{0}{3x_1} - \underset{1}{2x_2} - \underset{1}{x_3} + x_4 \geq 2$

Wts $H \restriction \pi$ is good

"Setting $a_1, \ldots, a_{k-1}$ to compensate for a_k"

$$(H \restriction \pi)(\tfrac{1}{2}^k) = 3(0) - 2(1) - (1) + (\tfrac{1}{2}) \leq 3(\tfrac{1}{2}) - 2(\tfrac{1}{2}) - (1) + (\tfrac{1}{2})$$

Depth Bounds for Semantic CP

Claim: Let $H \subseteq \mathbb{R}^n$ be good. For any $I \subseteq [n]$, $b \in \{0,1\}$, there is an assignment π to the variables x_i, $i \in I$ such that

- $\bigoplus_{i \in I} \pi(x_i) = b$
- $H \restriction \pi$ is good

Main tool: Monotonicity of halfspaces

Proof: Let $H := \sum a_i x_i \geq b$ and $I = \{1, .., k\}$. Suppose wlog $|a_1| \geq |a_2| \geq \cdots \geq |a_k|$. Construct π as follows:

- For $i \in \{1, \ldots, k-1\}$ set $x_i \leftarrow 0$ if $a_i \geq 0$, $x_i \leftarrow 1$ o.w.
- Set x_k so $\bigoplus_{i \in I} \pi(x_i) = b$

e.g. $b=0$, $3x_1 - 2x_2 - x_3 + x_4 \geq 2$

$I = \{1,2,3\}$

$$x_1 \leftarrow 0 \quad x_2 \leftarrow 1 \quad x_3 \leftarrow 1$$

Wts $H \restriction \pi$ is good

"Setting $a_1, \ldots, a_{k-1}$ to compensate for a_k"

$$(H \restriction \pi)(\tfrac{1}{2}^k) = 3(0) - 2(1) - (1) + (\tfrac{1}{2}) \leq 3(\tfrac{1}{2}) - 2(\tfrac{1}{2}) - (1) + (\tfrac{1}{2}) < 2$$

Proof Idea of Technical Lemma

Technical Lemma: Let H_1, H_2 be the children of H and $H\lceil\rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1\lceil\rho'$ or $H_2\lceil\rho'$ is good

Proof Idea of Technical Lemma

Technical Lemma: Let H_1, H_2 be the children of H and $H \restriction \rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1 \restriction \rho'$ or $H_2 \restriction \rho'$ is good

Change of basis: work over $\{+1,-1\}^n$ by the bijection $v \to (1-v)/2$

Proof Idea of Technical Lemma

Technical Lemma: Let H_1, H_2 be the children of H and $H\restriction\rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1\restriction\rho'$ or $H_2\restriction\rho'$ is good

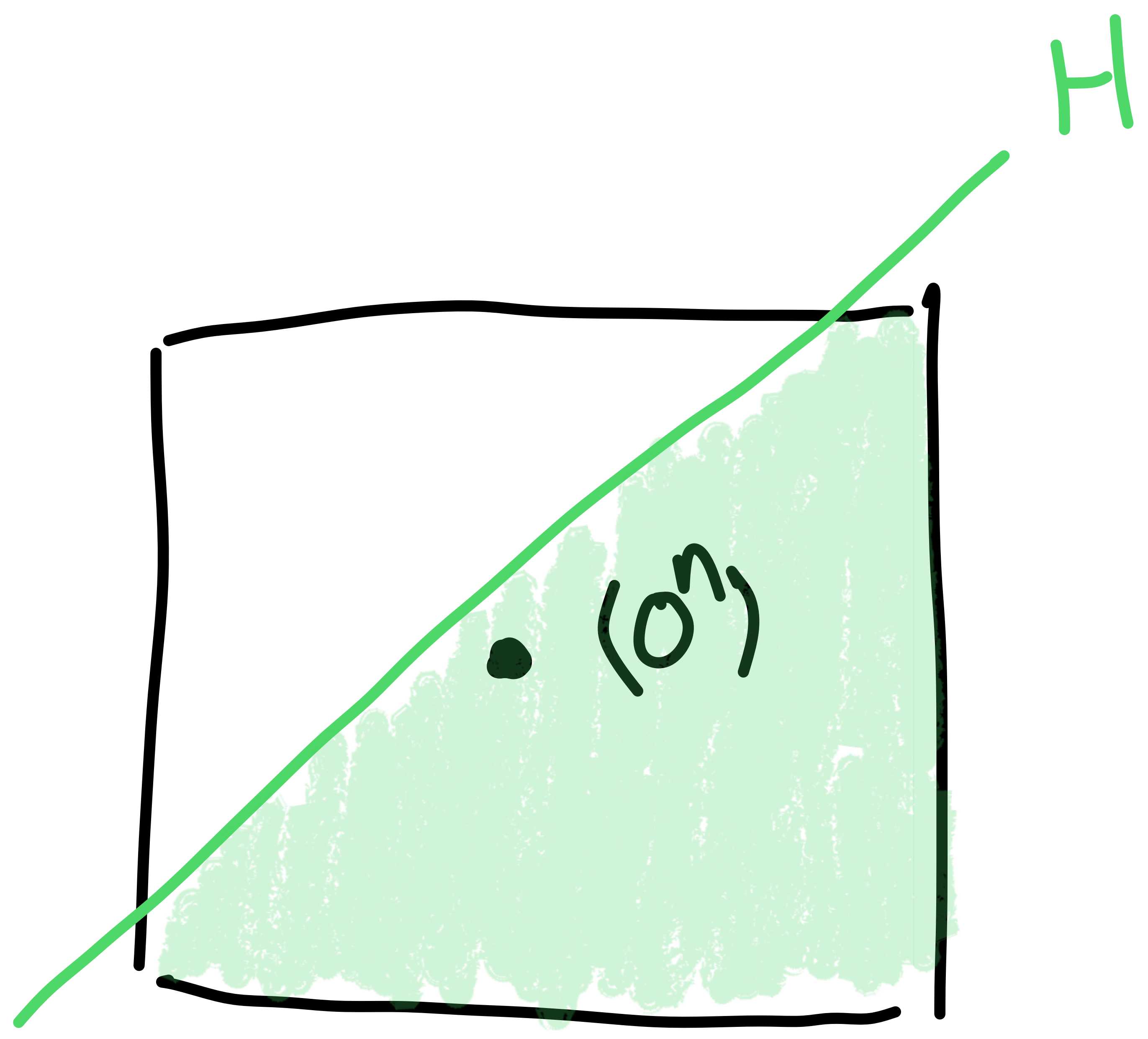
Change of basis: work over $\{+1,-1\}^n$ by the bijection $v \to (1-v)/2$

- H is good if $H(0^n)=0$

Proof Idea of Technical Lemma

Technical Lemma: Let H_1, H_2 be the children of H and $H \restriction \rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1 \restriction \rho'$ or $H_2 \restriction \rho'$ is good

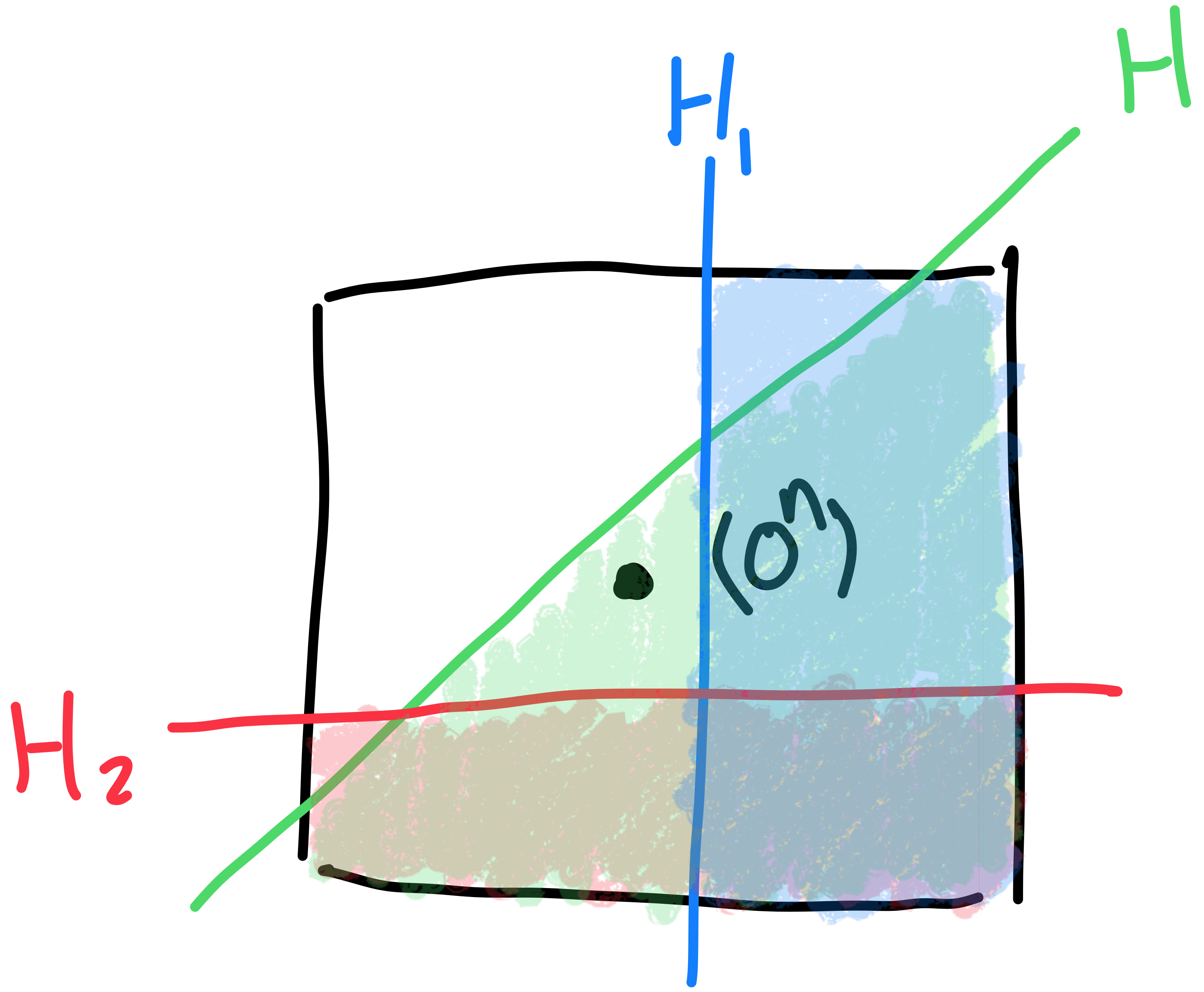
Suppose $\rho = *^n$



Proof Idea of Technical Lemma

Technical Lemma: Let H_1, H_2 be the children of H and $H\restriction\rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1\restriction\rho'$ or $H_2\restriction\rho'$ is good

Idea: Soundness means $H^{-1}(0) \cap \{-1,1\}^n \subseteq H_1^{-1}(0) \cup H_2^{-1}(0)$. Suppose $\rho = *^n$

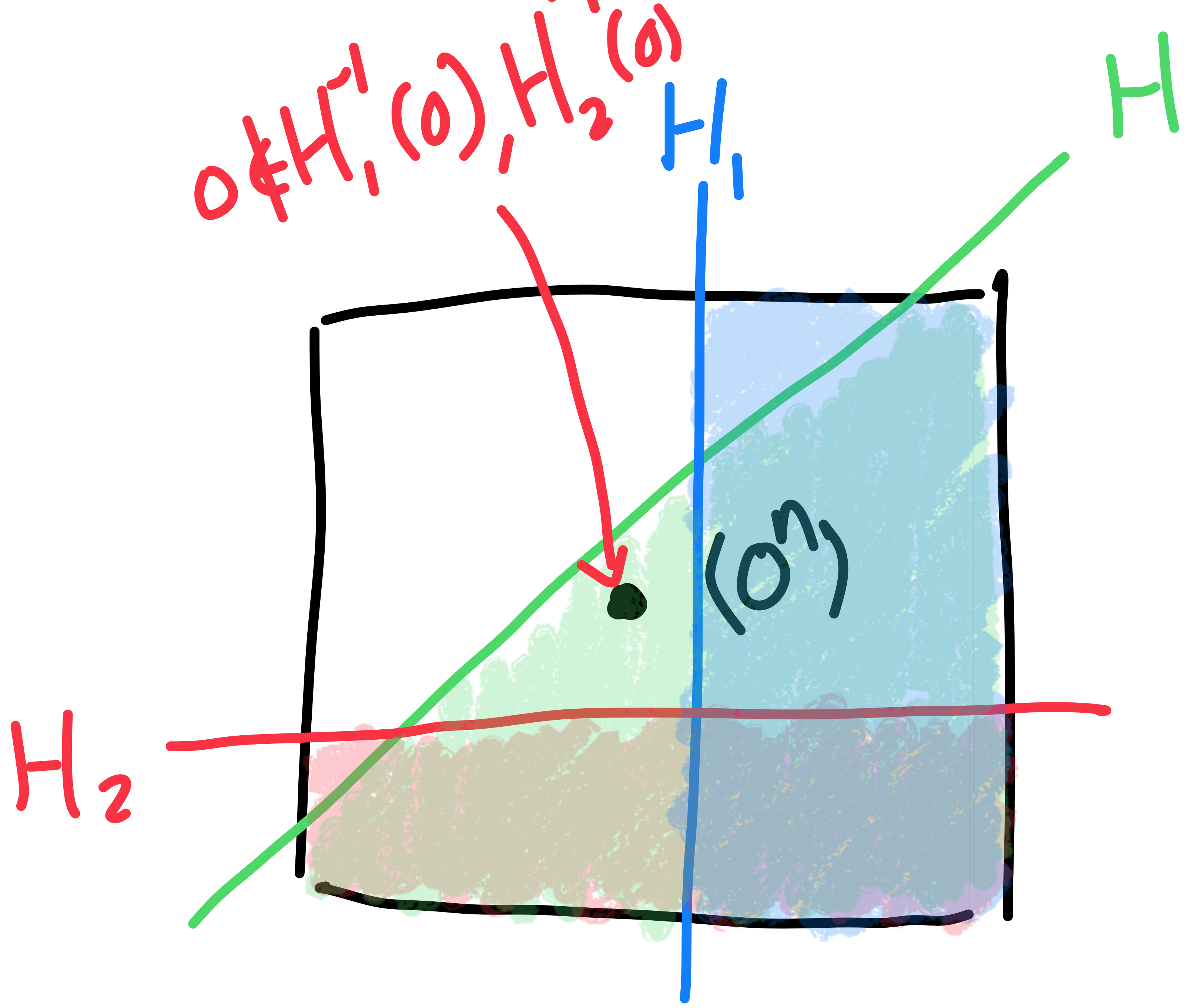


Proof Idea of Technical Lemma

Technical Lemma: Let H_1, H_2 be the children of H and $H \restriction \rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1 \restriction \rho'$ or $H_2 \restriction \rho'$ is good

Idea: Soundness means $H^{-1}(0) \cap \{-1,1\}^n \subseteq H_1^{-1}(0) \cup H_2^{-1}(0)$. Suppose $\rho = *^n$

However 0^n may not be in $H_1^{-1}(0), H_2^{-1}(0)$



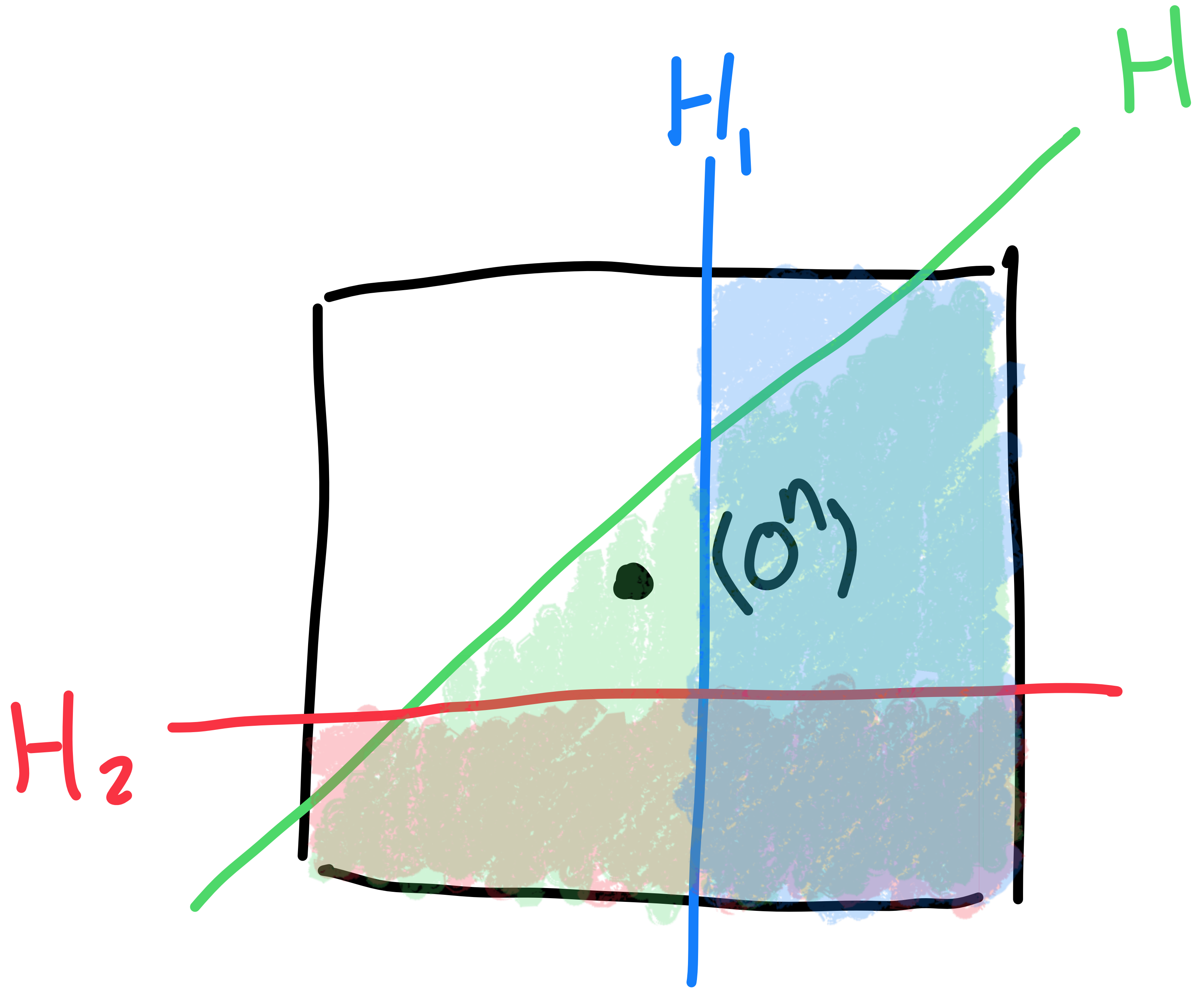
Proof Idea of Technical Lemma

Technical Lemma: Let H_1, H_2 be the children of H and $H\restriction\rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1\restriction\rho'$ or $H_2\restriction\rho'$ is good

Idea: Soundness means $H^{-1}(0)\cap\{-1,1\}^n \subseteq H_1^{-1}(0) \cup H_2^{-1}(0)$. Suppose $\rho = *^n$

However 0^n may not be in $H_1^{-1}(0), H_2^{-1}(0)$

- Let $H_1 = a_1x \geq b_1$, $H_2 = a_2x > b_2$



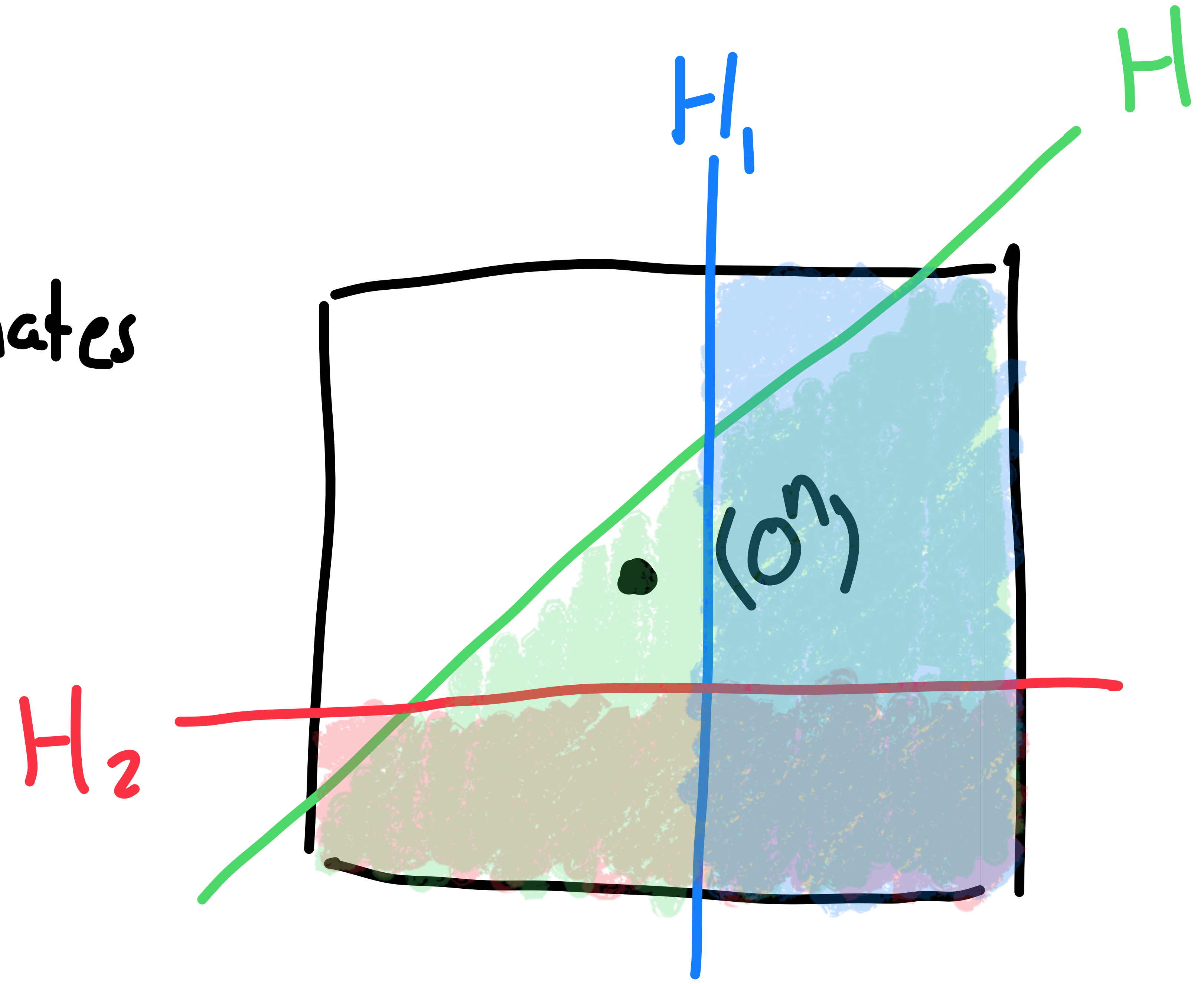
Proof Idea of Technical Lemma

Technical Lemma: Let H_1, H_2 be the children of H and $H \restriction \rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1 \restriction \rho'$ or $H_2 \restriction \rho'$ is good

Idea: Soundness means $H^{-1}(0) \cap \{-1,1\}^n \subseteq H_1^{-1}(0) \cup H_2^{-1}(0)$. Suppose $\rho = *^n$

However 0^n may not be in $H_1^{-1}(0), H_2^{-1}(0)$

- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1, 1$



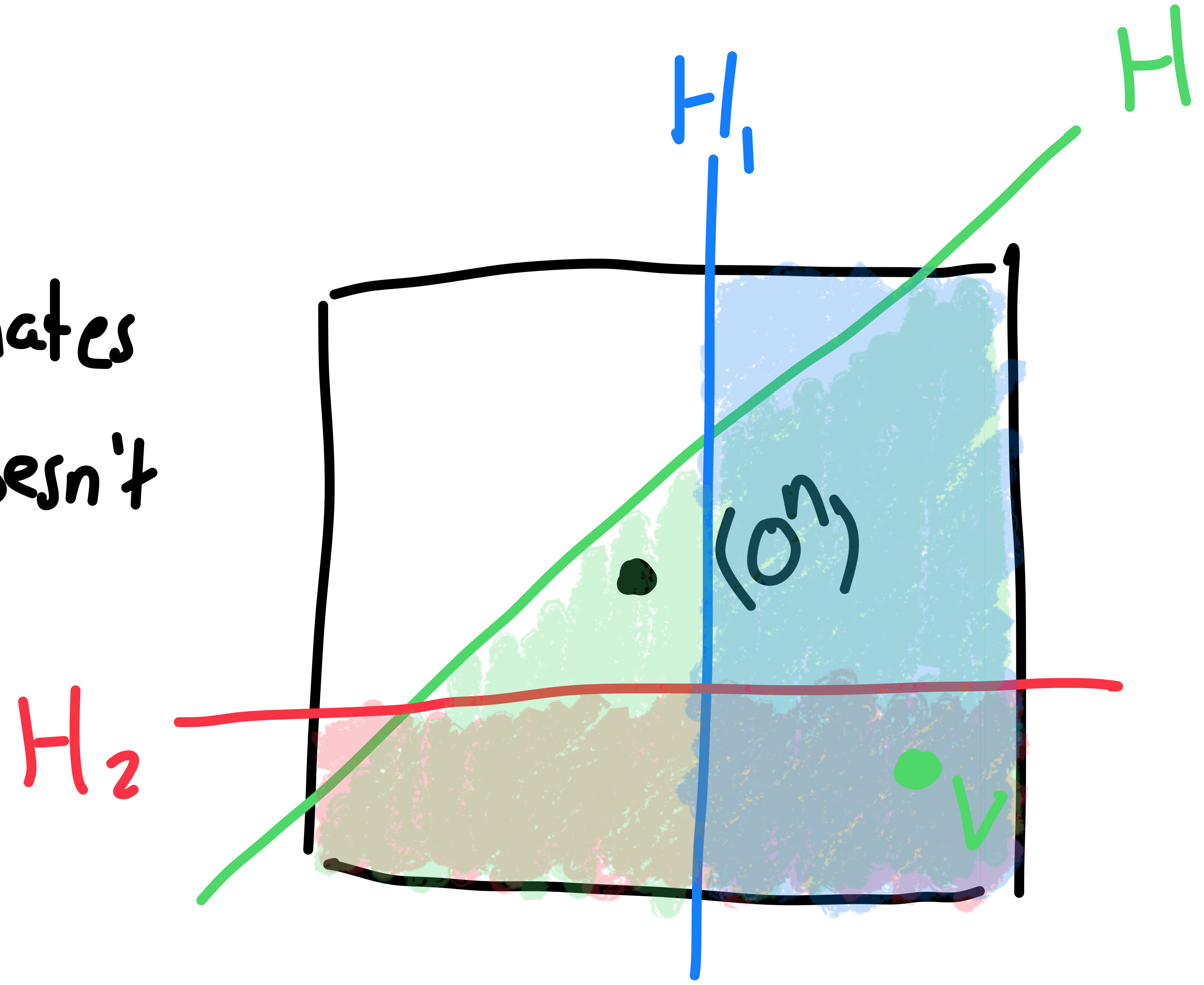
Proof Idea of Technical Lemma

Technical Lemma: Let H_1, H_2 be the children of H and $H \restriction \rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1 \restriction \rho'$ or $H_2 \restriction \rho'$ is good

Idea: Soundness means $H^{-1}(0) \cap \{-1,1\}^n \subseteq H_1^{-1}(0) \cup H_2^{-1}(0)$. Suppose $\rho = *^n$

However 0^n may not be in $H_1^{-1}(0), H_2^{-1}(0)$

- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1,1$ $\rightarrow$ moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$



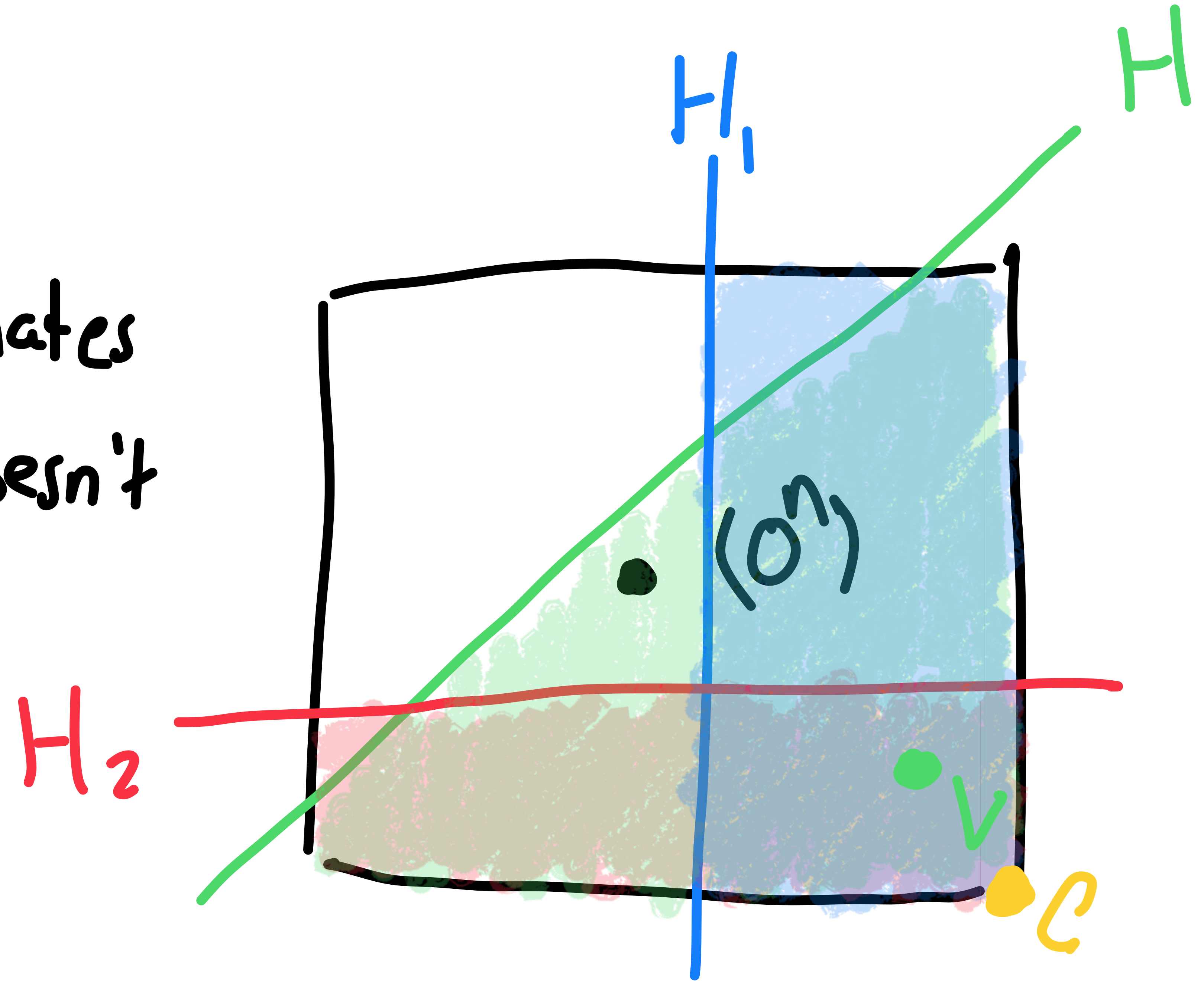
Proof Idea of Technical Lemma

Technical Lemma: Let H_1, H_2 be the children of H and $H\restriction\rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1\restriction\rho'$ or $H_2\restriction\rho'$ is good

Idea: Soundness means $H^{-1}(0) \cap \{-1,1\}^n \subseteq H_1^{-1}(0) \cup H_2^{-1}(0)$. Suppose $\rho = *^n$

However 0^n may not be in $H_1^{-1}(0), H_2^{-1}(0)$

- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1,1$ → moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$
- Let $c \in \{-1,1\}^n$ be closest to v



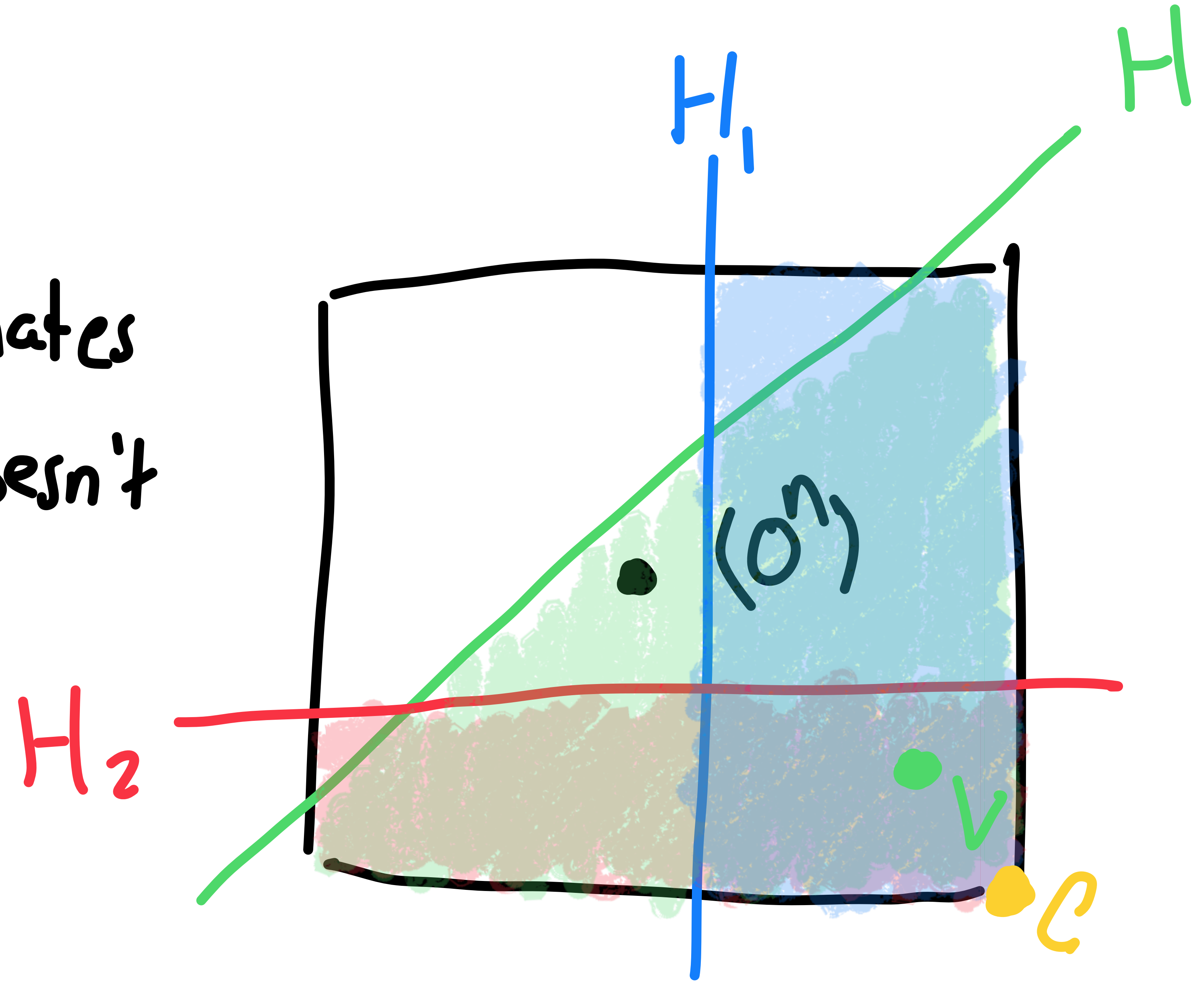
Proof Idea of Technical Lemma

Technical Lemma: Let H_1, H_2 be the children of H and $H\restriction\rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1\restriction\rho'$ or $H_2\restriction\rho'$ is good

Idea: Soundness means $H^{-1}(0) \cap \{-1,1\}^n \subseteq H_1^{-1}(0) \cup H_2^{-1}(0)$. Suppose $\rho = *^n$

However 0^n may not be in $H_1^{-1}(0), H_2^{-1}(0)$

- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1,1$ → moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v) = 0$



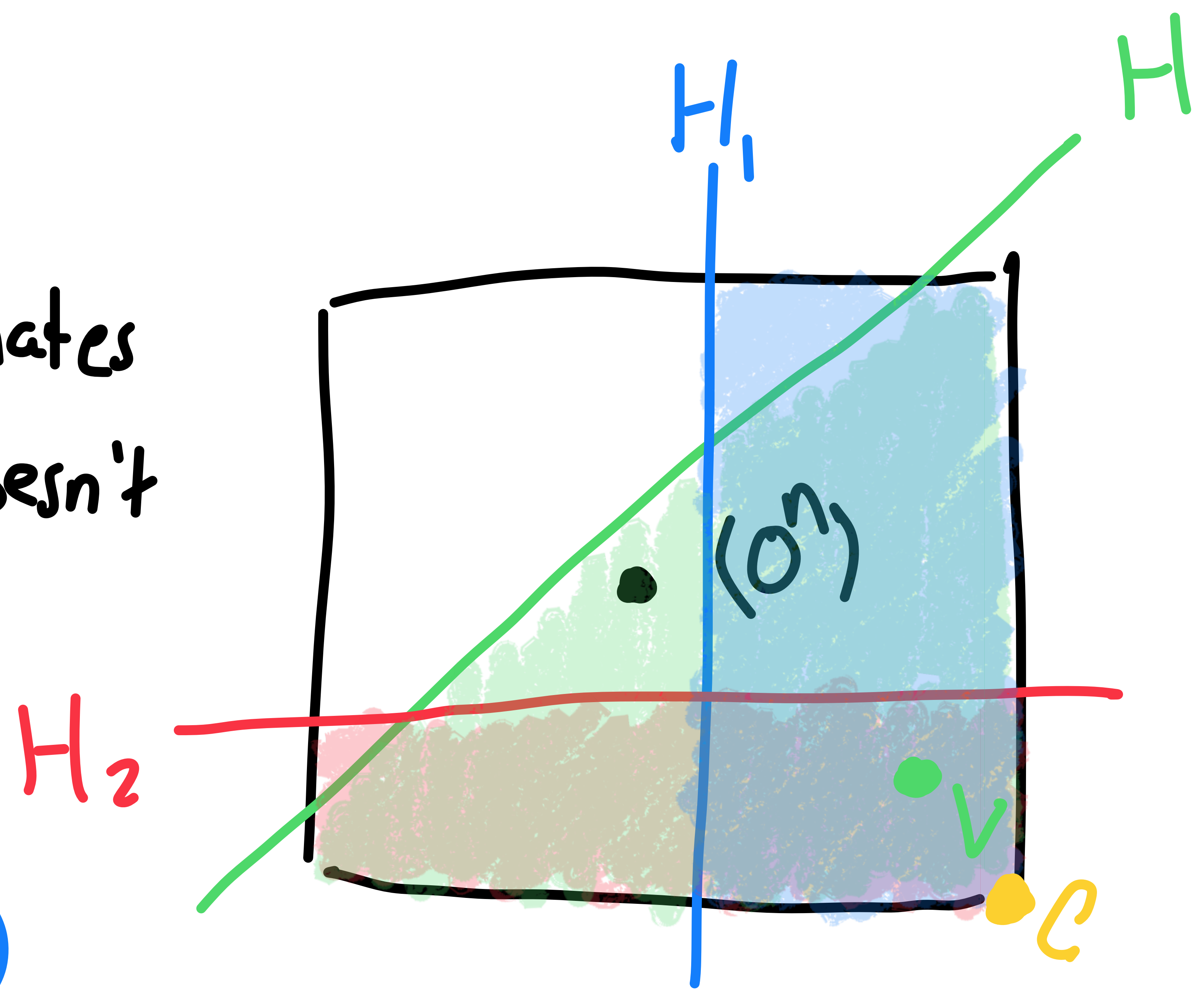
Proof Idea of Technical Lemma

Technical Lemma: Let H_1, H_2 be the children of H and $H\restriction\rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1\restriction\rho'$ or $H_2\restriction\rho'$ is good

Idea: Soundness means $H^{-1}(0) \cap \{-1,1\}^n \subseteq H_1^{-1}(0) \cup H_2^{-1}(0)$. Suppose $\rho = *^n$

However 0^n may not be in $H_1^{-1}(0), H_2^{-1}(0)$

- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1,1$ → moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v) = 0$ (otherwise set $v \leftarrow -v$, $c \leftarrow -c$)



Proof Idea of Technical Lemma

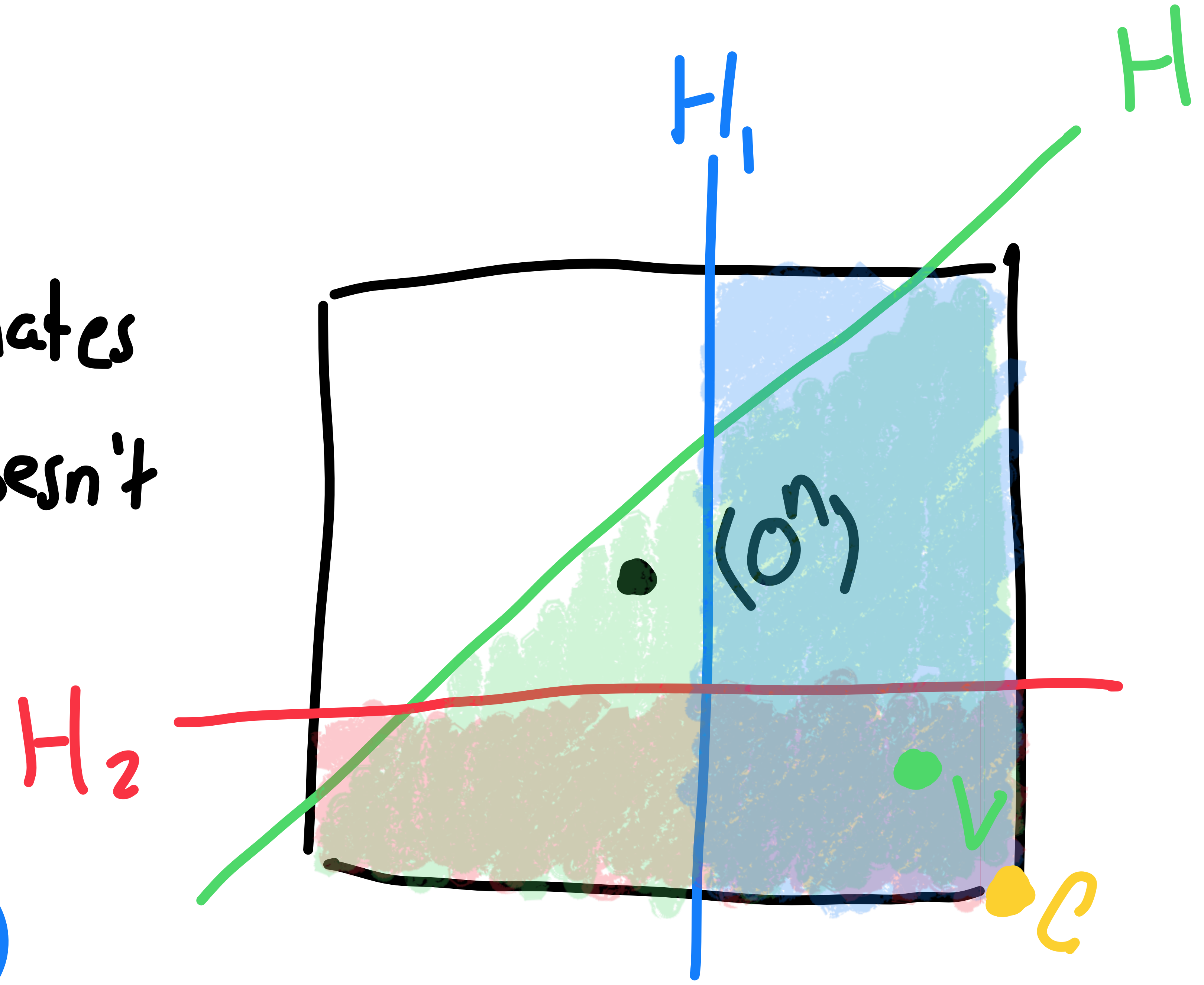
Technical Lemma: Let H_1, H_2 be the children of H and $H\restriction\rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1\restriction\rho'$ or $H_2\restriction\rho'$ is good

Idea: Soundness means $H^{-1}(0)\cap\{-1,1\}^n \subseteq H_1^{-1}(0) \cup H_2^{-1}(0)$. Suppose $\rho = *^n$

However 0^n may not be in $H_1^{-1}(0), H_2^{-1}(0)$

- Let $H_1 = a_1x \geq b_1$, $H_2 = a_2x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1,1$ → moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v)=0$ (otherwise set $v \leftarrow -v$, $c \leftarrow -c$)

→ $H_1(v)=0$ or $H_2(v)=0$



Proof Idea of Technical Lemma

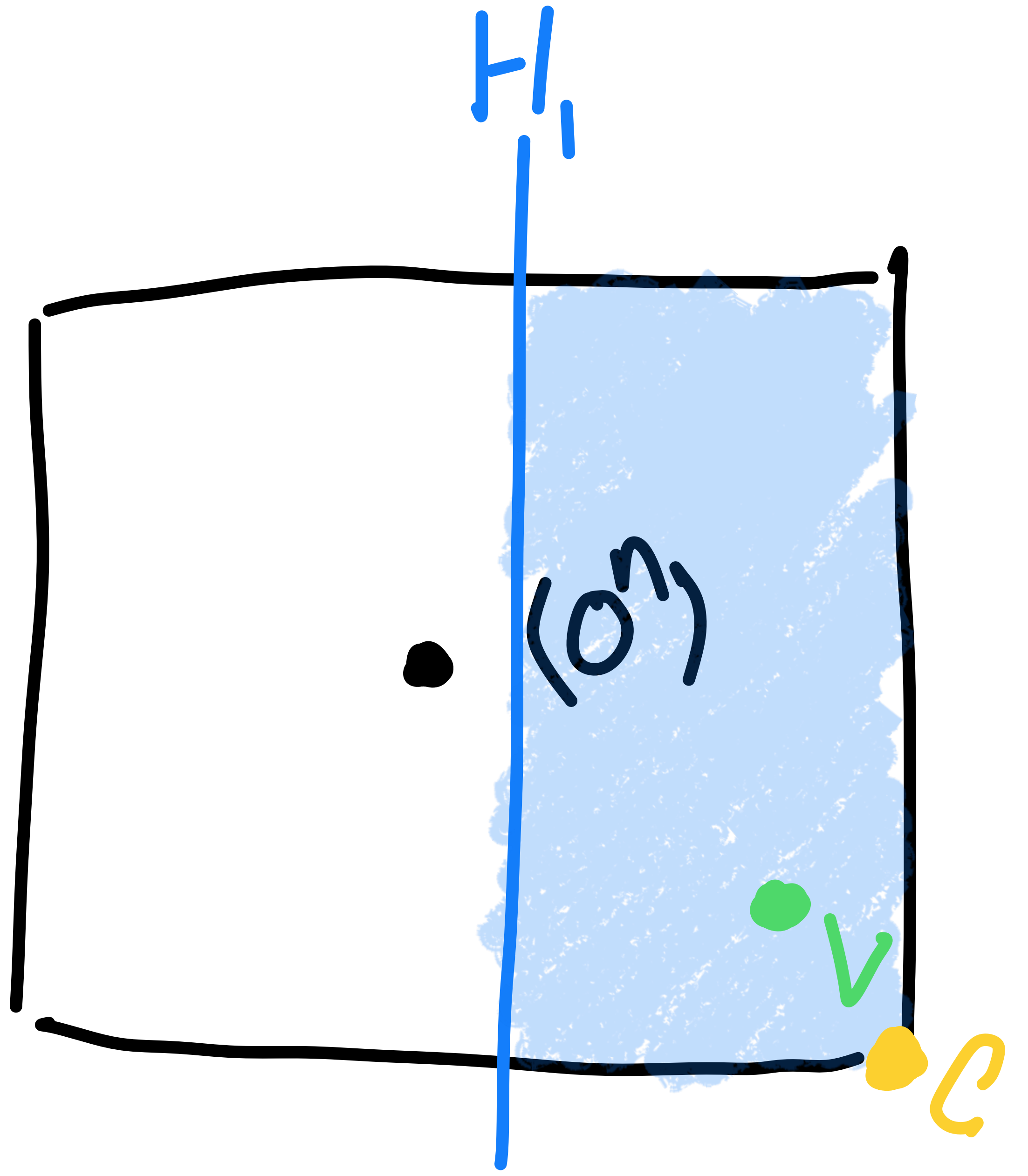
Technical Lemma: Let H_1, H_2 be the children of H and $H\lceil\rho$ be good. Then we can obtain ρ' by fixing 2 additional bits to $\{0,1\}$ s.t. $H_1\lceil\rho'$ or $H_2\lceil\rho'$ is good

Idea: Soundness means $H^{-1}(0)\cap\{-1,1\}^n \subseteq H_1^{-1}(0) \cup H_2^{-1}(0)$. Suppose $\rho = *^n$

However 0^n may not be in $H_1^{-1}(0), H_2^{-1}(0)$

- Let $H_1 = a_1x \geq b_1$, $H_2 = a_2x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1,1$ → moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v)=0$ (otherwise set $v \leftarrow -v$, $c \leftarrow -c$)

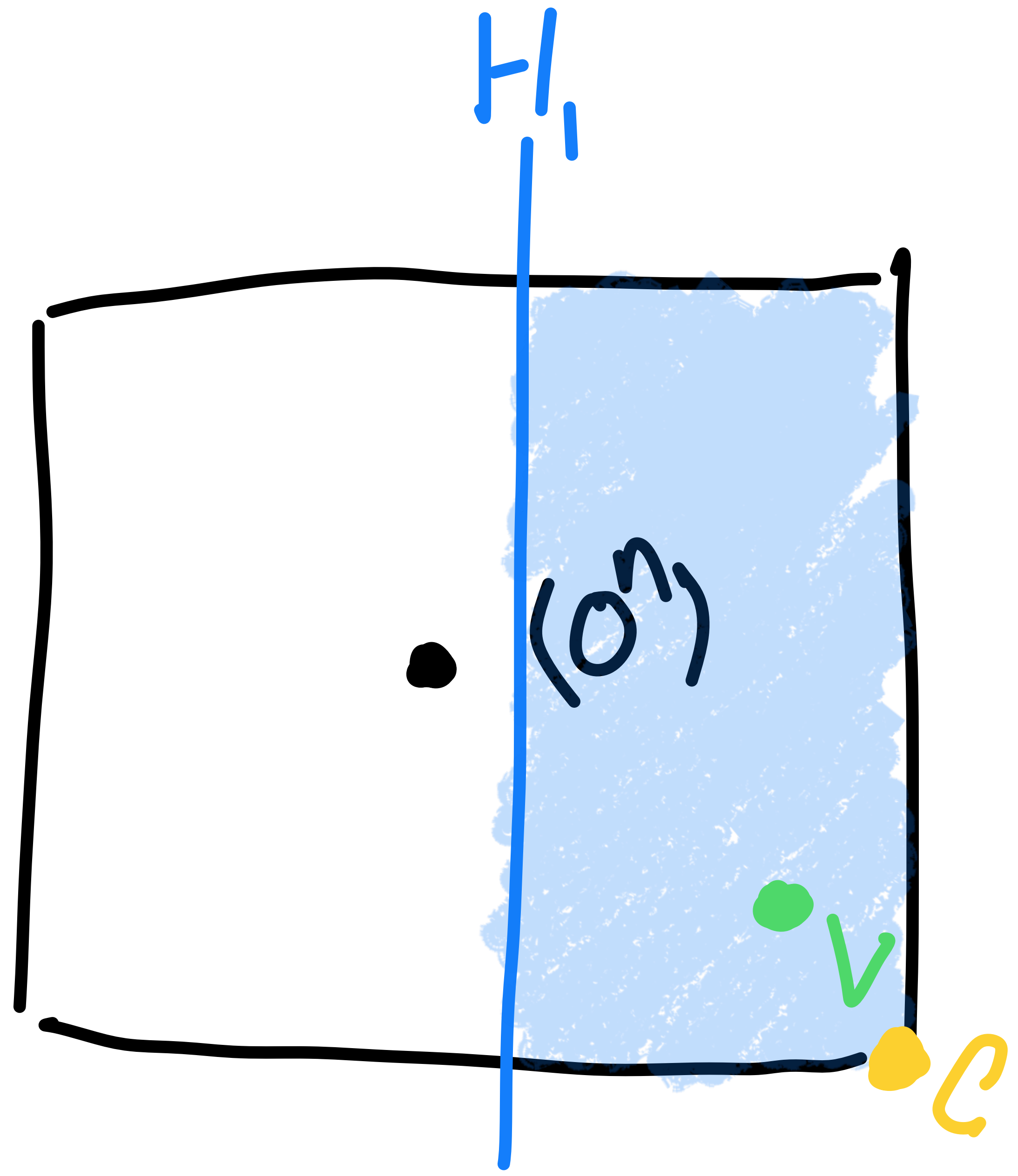
→ $H_1(v)=0$ or $H_2(v)=0$ (assume H_1)



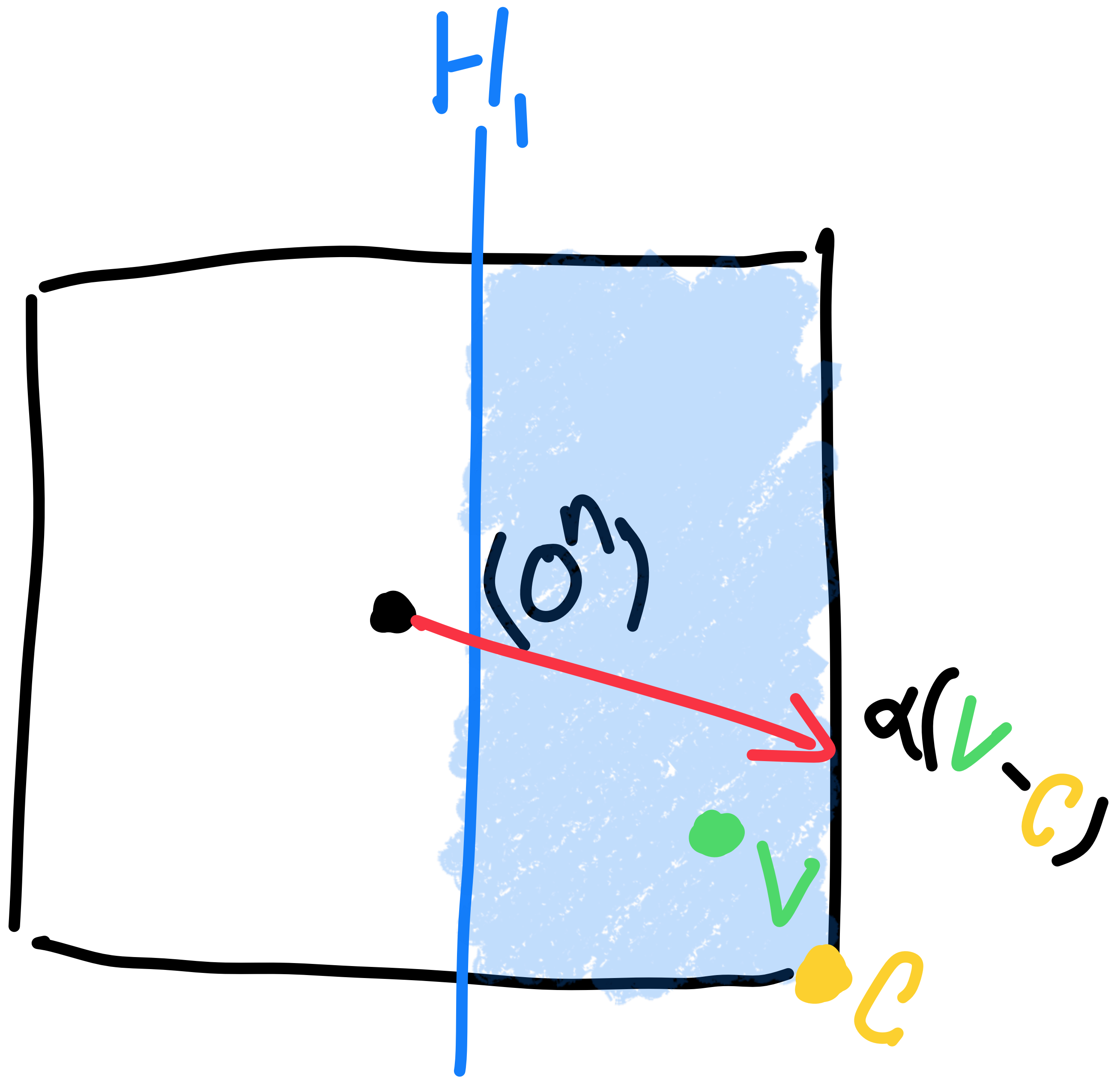
Proof Idea of Technical Lemma

- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1, 1$ → moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v) = 0$ (otherwise set $v \leftarrow -v$, $c \leftarrow -c$)

  → $H_1(v) = 0$ or $H_2(v) = 0$ (assume H_1)



Proof Idea of Technical Lemma

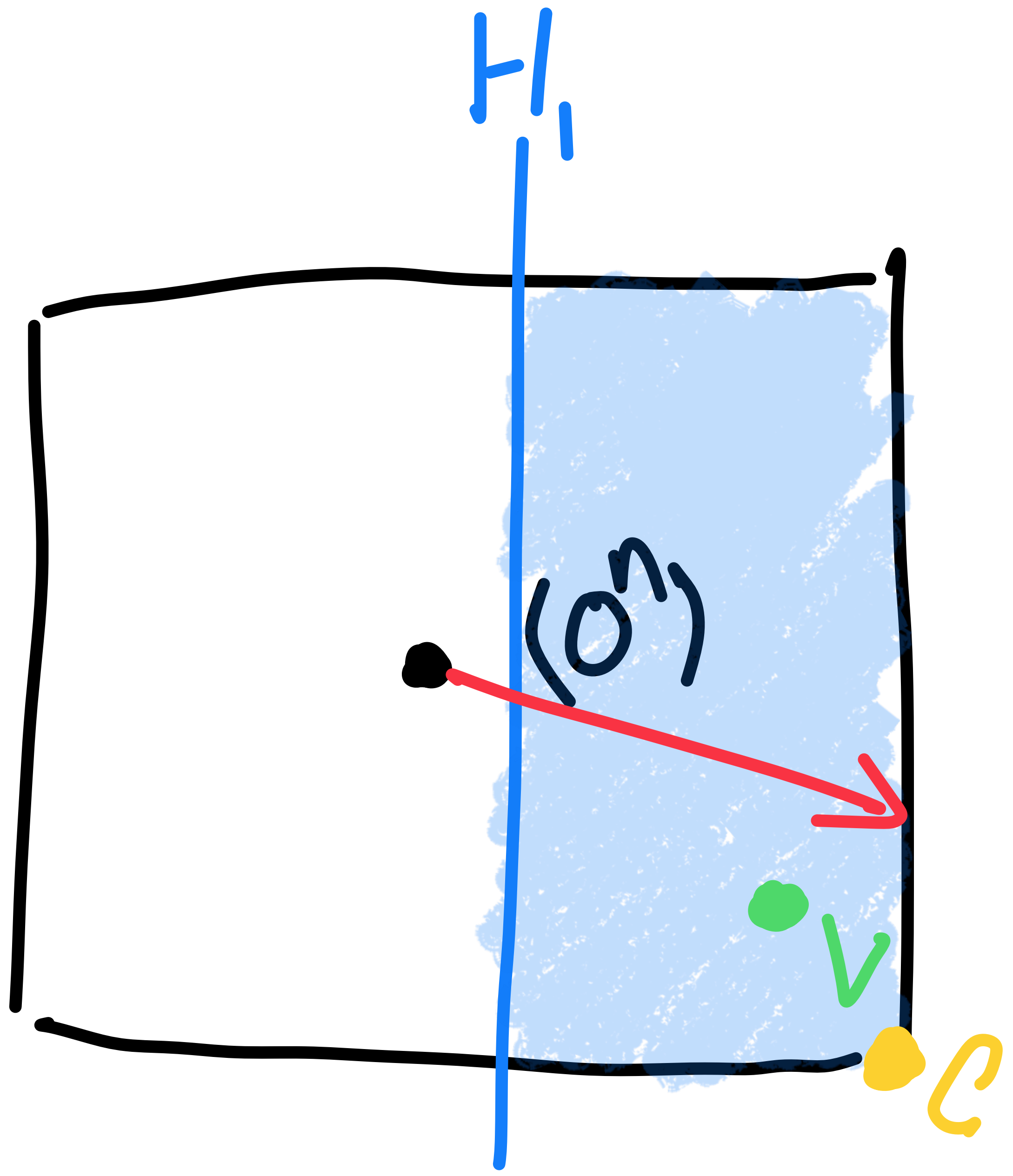


- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1, 1$ $\rightarrow$ moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v) = 0$ (otherwise set $v \leftarrow -v$, $c \leftarrow -c$)
  $\rightarrow$ $H_1(v) = 0$ or $H_2(v) = 0$ (assume H_1)
- Starting at $y = 0^n$ move in the direction of $c - v$ until one of the two non-zero coordinates becomes in $\{-1,1\}$. Let $\alpha(c-v)$ for $\alpha > 0$ be this point. Note: $\alpha > 1$ as c is the closest corner to v, so $\|c-v\|_\infty \leq 1$

  $$\rightarrow a_1(\alpha(c-v)) = \alpha a_1 c - \alpha a_1 v = \alpha a_1 c < b_1$$
- By monotonicity of H_1 we can round

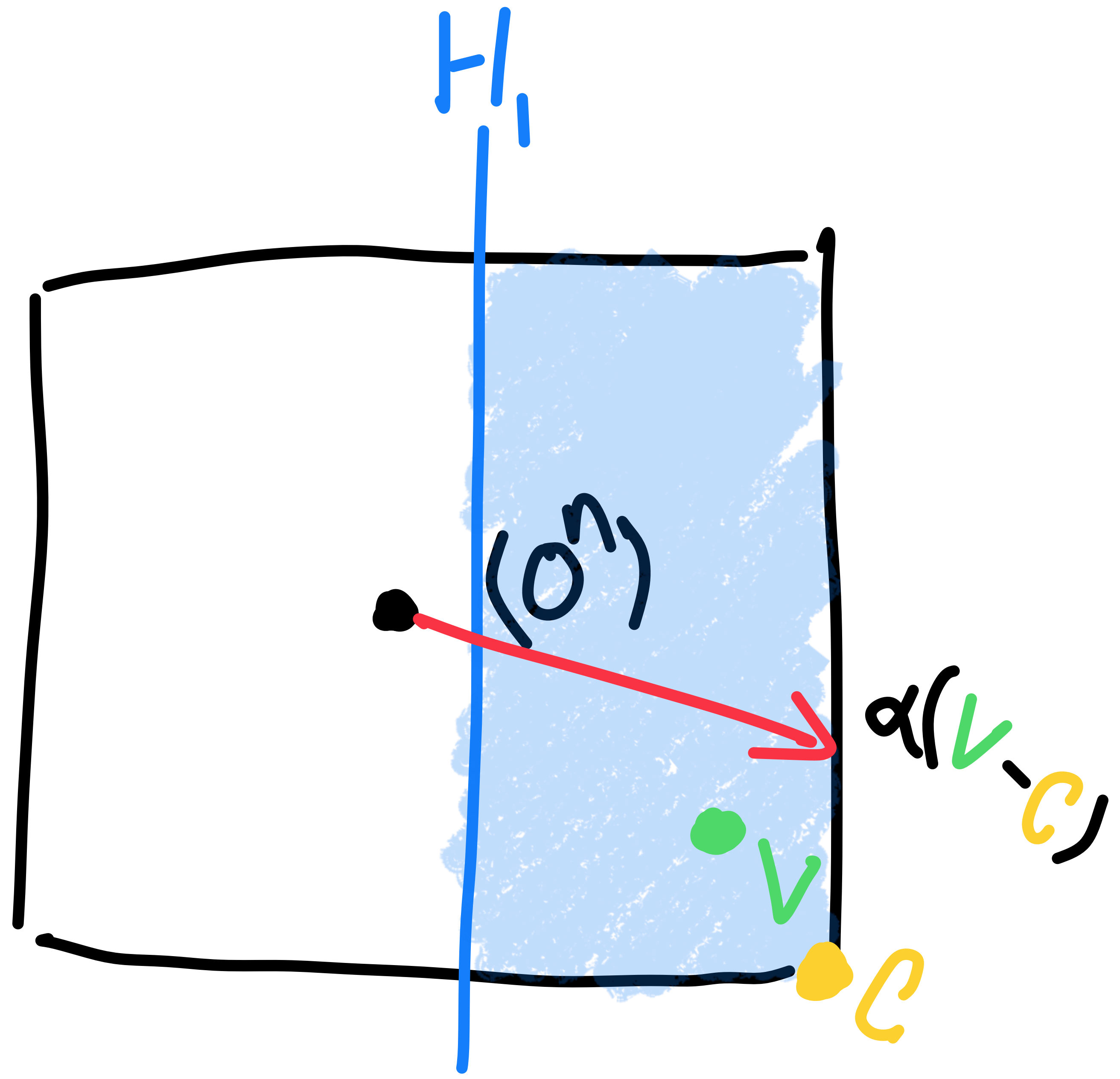
Proof Idea of Technical Lemma

- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1, 1$ → moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v)=0$ (otherwise set $v \leftarrow -v$, $c \leftarrow -c$)
  → $H_1(v)=0$ or $H_2(v)=0$ (assume H_1)
- Starting at $y = 0^n$ move in the direction of $c - v$ until one of the two non-zero coordinates becomes in $\{-1,1\}$.



Proof Idea of Technical Lemma

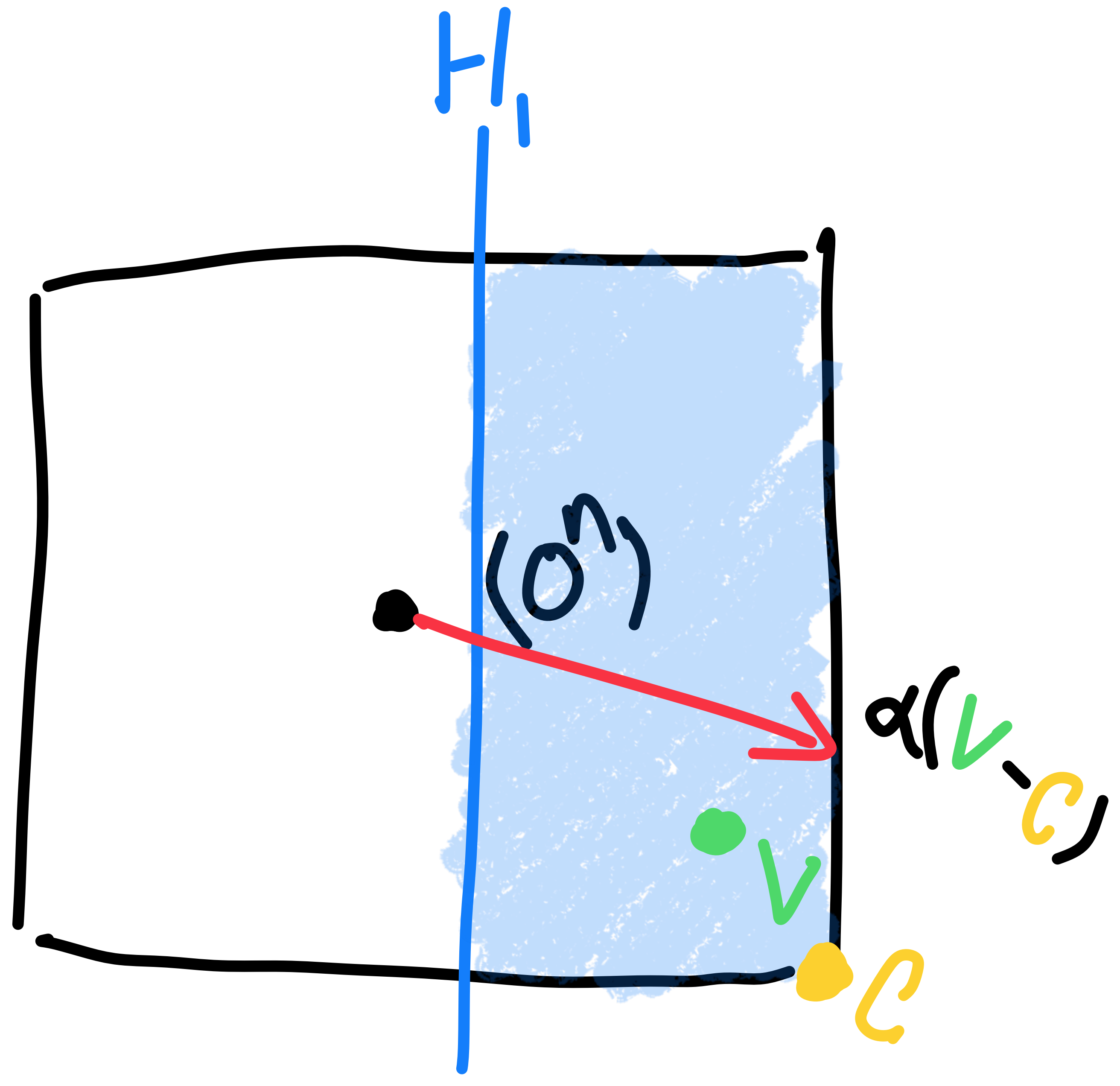
- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1, 1$ → moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v) = 0$ (otherwise set $v \leftarrow -v$, $c \leftarrow -c$)
  → $H_1(v) = 0$ or $H_2(v) = 0$ (assume H_1)
- Starting at $y = 0^n$ move in the direction of $c - v$ until one of the two non-zero coordinates becomes in $\{-1, 1\}$. Let $\alpha(c-v)$ for $\alpha > 0$ be this point.



Proof Idea of Technical Lemma

- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x \geq b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1, 1$ → moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v) = 0$ (otherwise set $v \leftarrow -v$, $c \leftarrow -c$)

  → $H_1(v) = 0$ or $H_2(v) = 0$ (assume H_1)



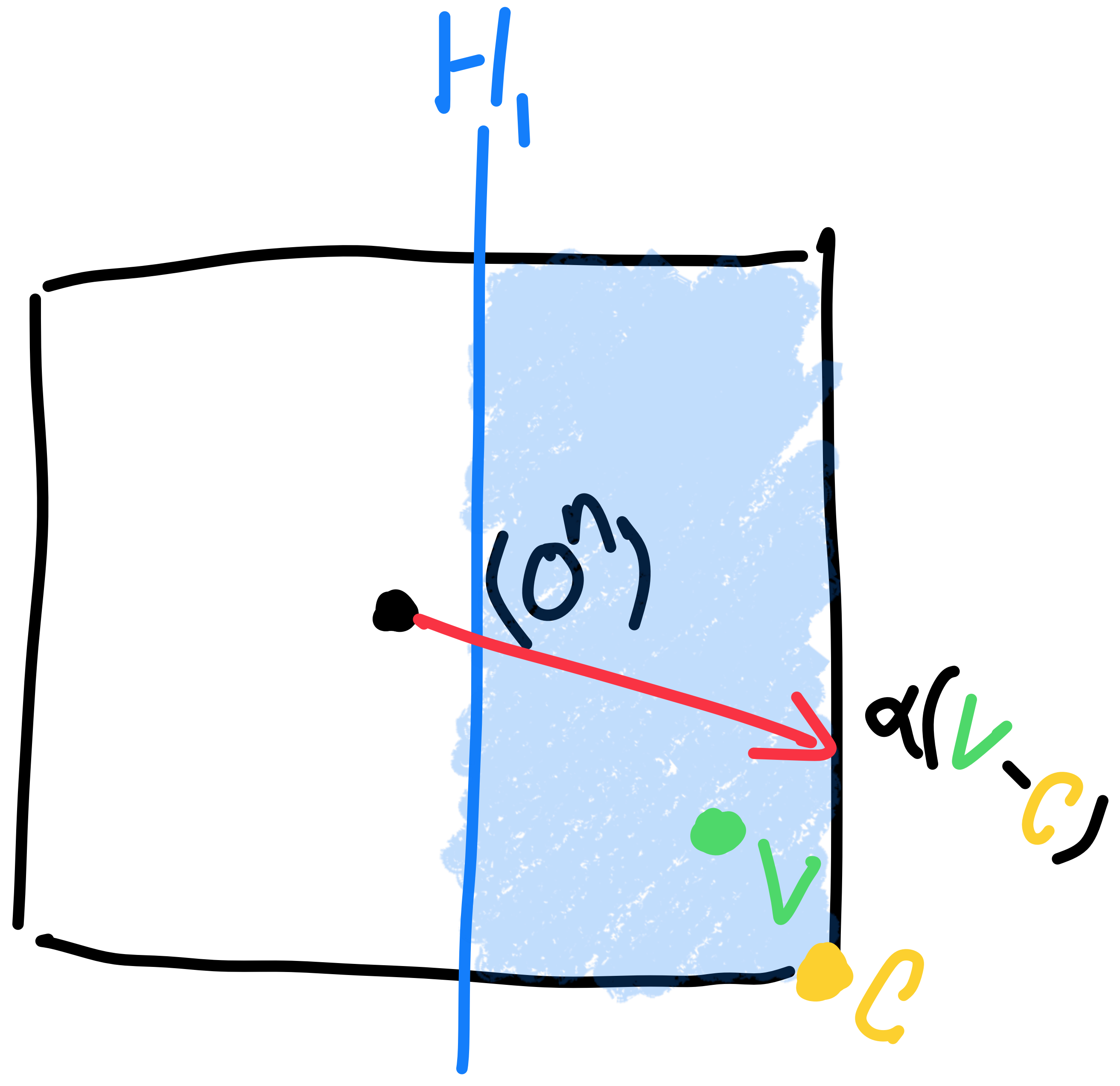
- Starting at $y = 0^n$ move in the direction of $c - v$ until one of the two non-zero coordinates becomes in $\{-1, 1\}$. Let $\alpha(c-v)$ for $\alpha > 0$ be this point. Note: $\alpha > 1$ as c is the closest corner to v, so $\|c - v\|_\infty \leq 1$

Proof Idea of Technical Lemma

- Let $H_1 = a_1x \geq b_1$, $H_2 = a_2x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1, 1$ → moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v) = 0$ (otherwise set $v \leftarrow -v$, $c \leftarrow -c$)

  → $H_1(v) = 0$ or $H_2(v) = 0$ (assume H_1)
- Starting at $y = 0^n$ move in the direction of $c - v$ until one of the two non-zero coordinates becomes in $\{-1, 1\}$. Let $\alpha(c-v)$ for $\alpha > 0$ be this point. Note: $\alpha > 1$ as c is the closest corner to v, so $\|c-v\|_\infty \leq 1$

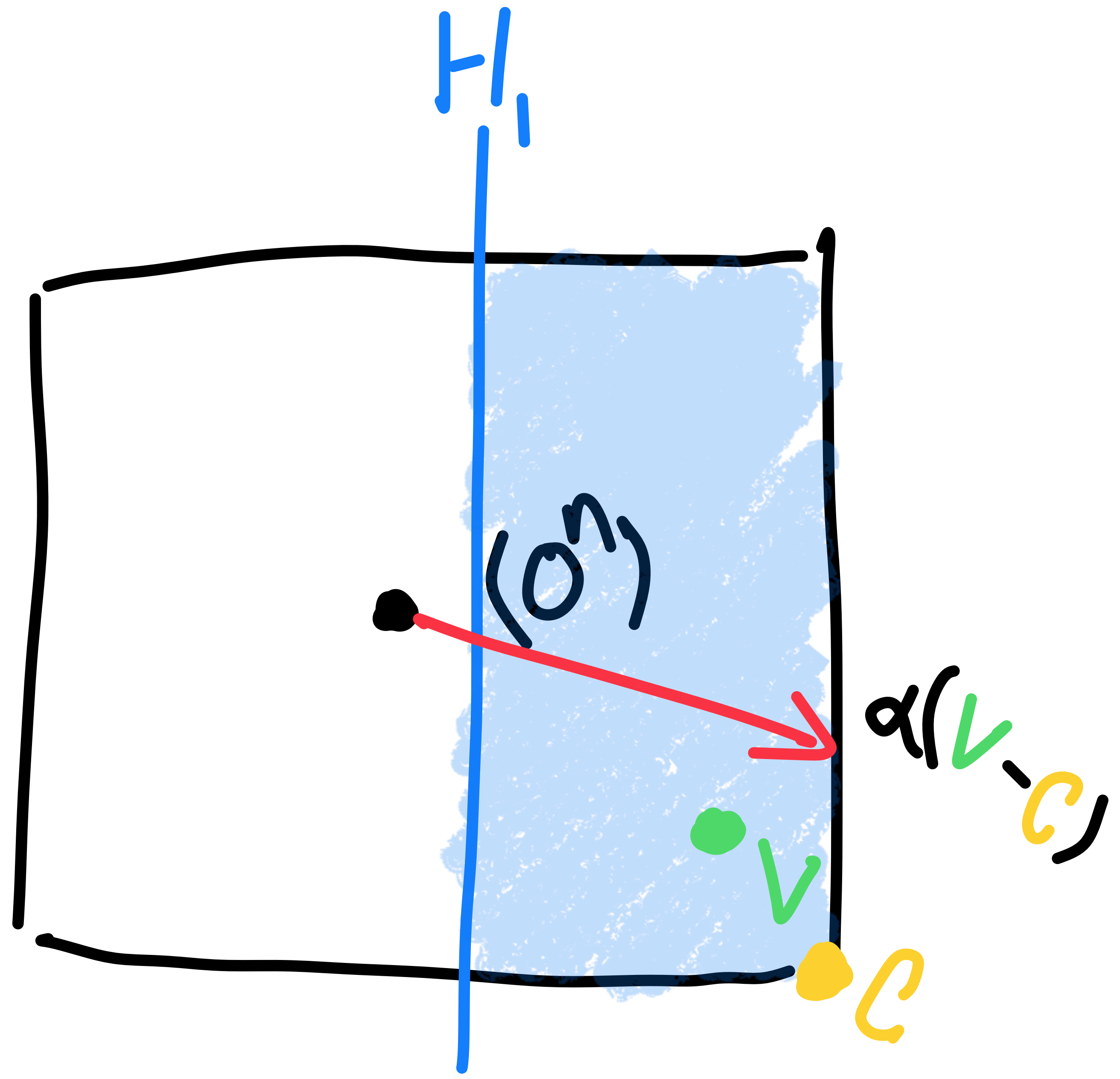
  → $a_1(\alpha(c-v)) =$



Proof Idea of Technical Lemma

- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1, 1$ → moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v) = 0$ (otherwise set $v \leftarrow -v$, $c \leftarrow -c$)

  → $H_1(v) = 0$ or $H_2(v) = 0$ (assume H_1)



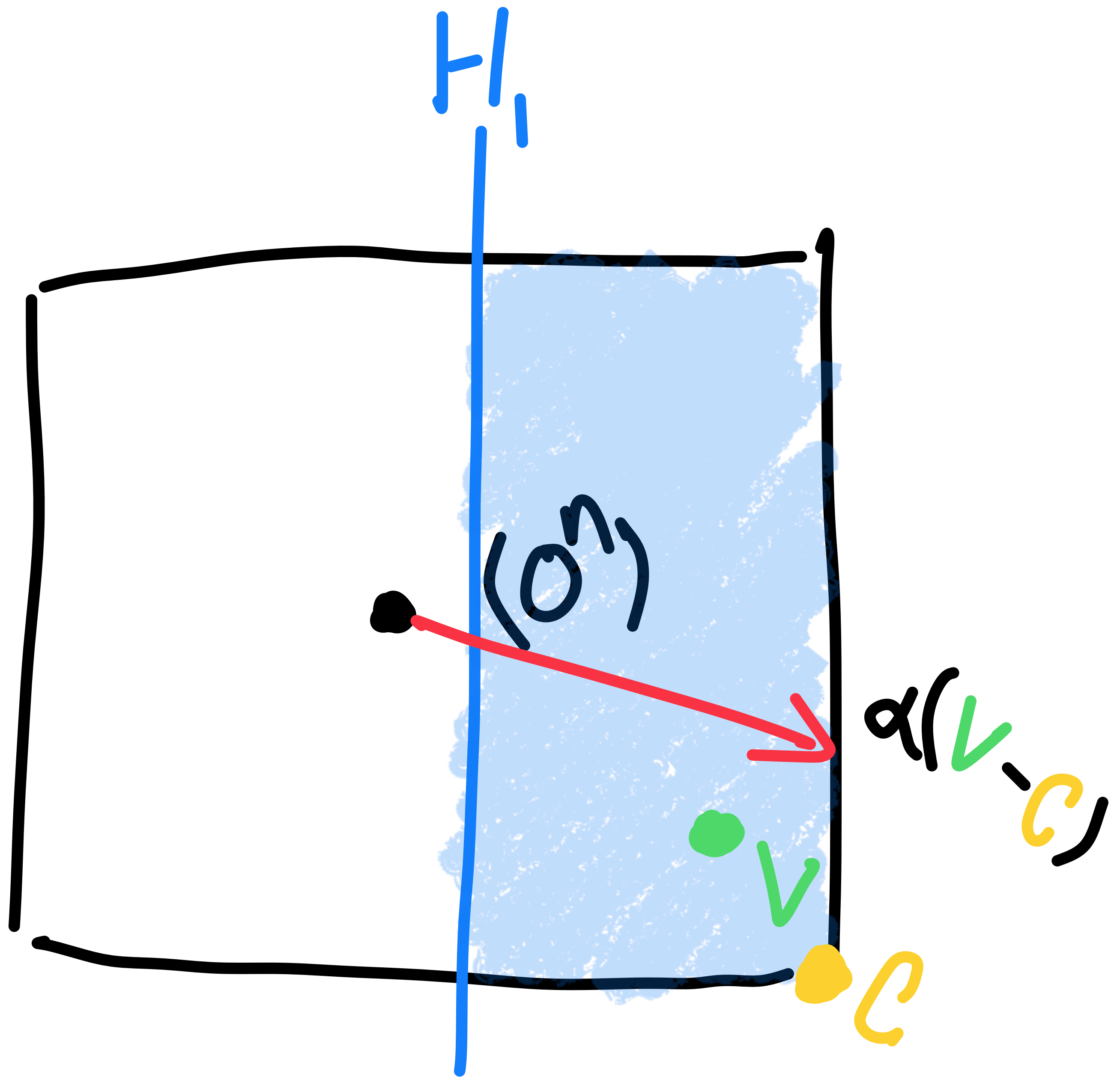
- Starting at $y = 0^n$ move in the direction of $c - v$ until one of the two non-zero coordinates becomes in $\{-1,1\}$. Let $\alpha(c-v)$ for $\alpha > 0$ be this point. Note: $\alpha > 1$ as c is the closest corner to v, so $\|c-v\|_\infty \leq 1$

  → $a_1(\alpha(c-v)) = \alpha a_1 c - \alpha a_1 v =$

Proof Idea of Technical Lemma

- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1,1$ $\rightarrow$ moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v)=0$ (otherwise set $v \leftarrow -v$, $c \leftarrow -c$)
  $\rightarrow$ $H_1(v)=0$ or $H_2(v)=0$ (assume H_1)
- Starting at $y=0^n$ move in the direction of $c-v$ until one of the two non-zero coordinates becomes in $\{-1,1\}$. Let $\alpha(c-v)$ for $\alpha>0$ be this point. Note: $\alpha>1$ as c is the closest corner to v, so $\|c-v\|_\infty \leq 1$

$$\rightarrow a_1(\alpha(c-v)) = \alpha a_1 c - \alpha a_1 v = \alpha a_1 c < b_1$$

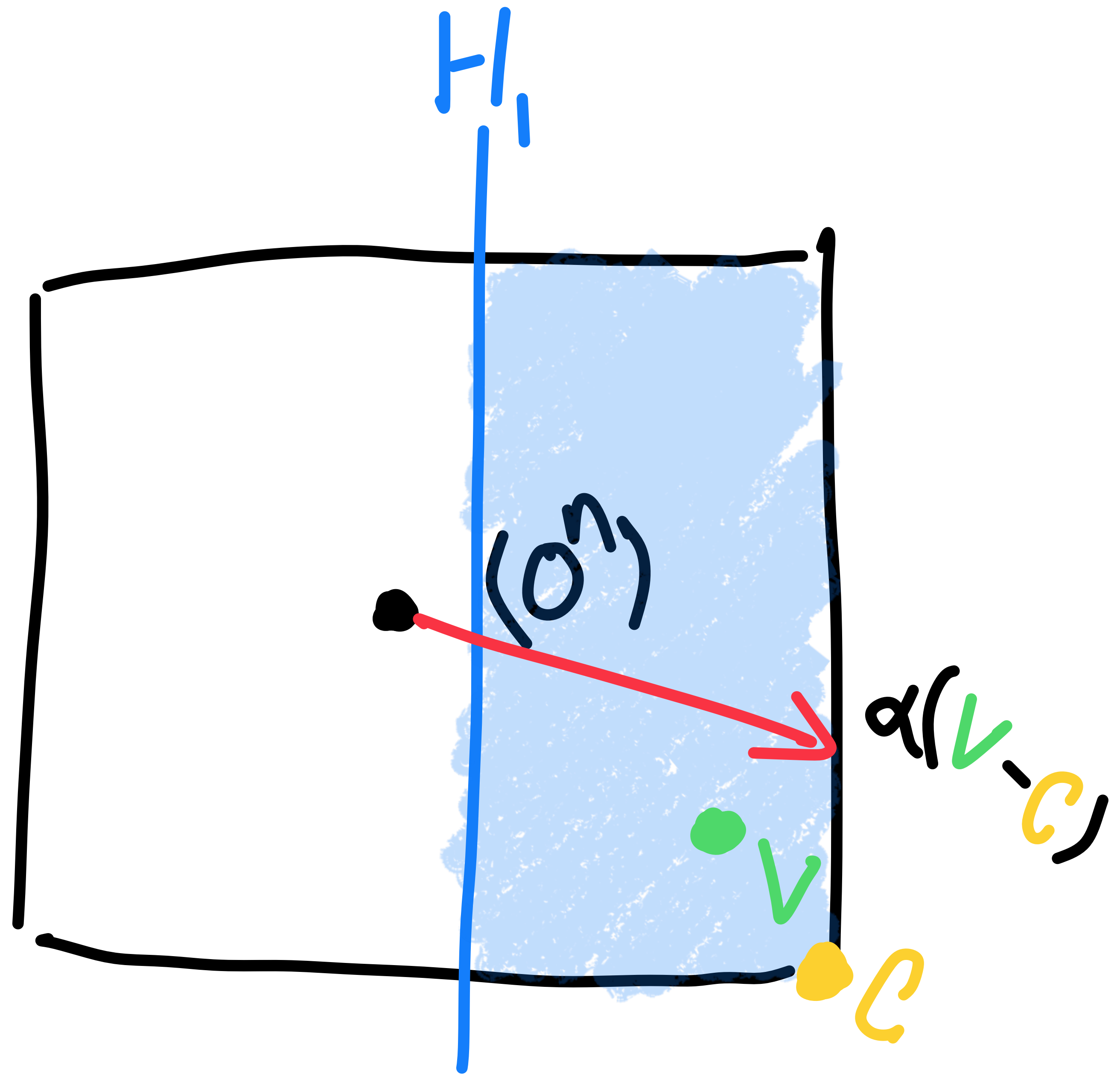


Proof Idea of Technical Lemma

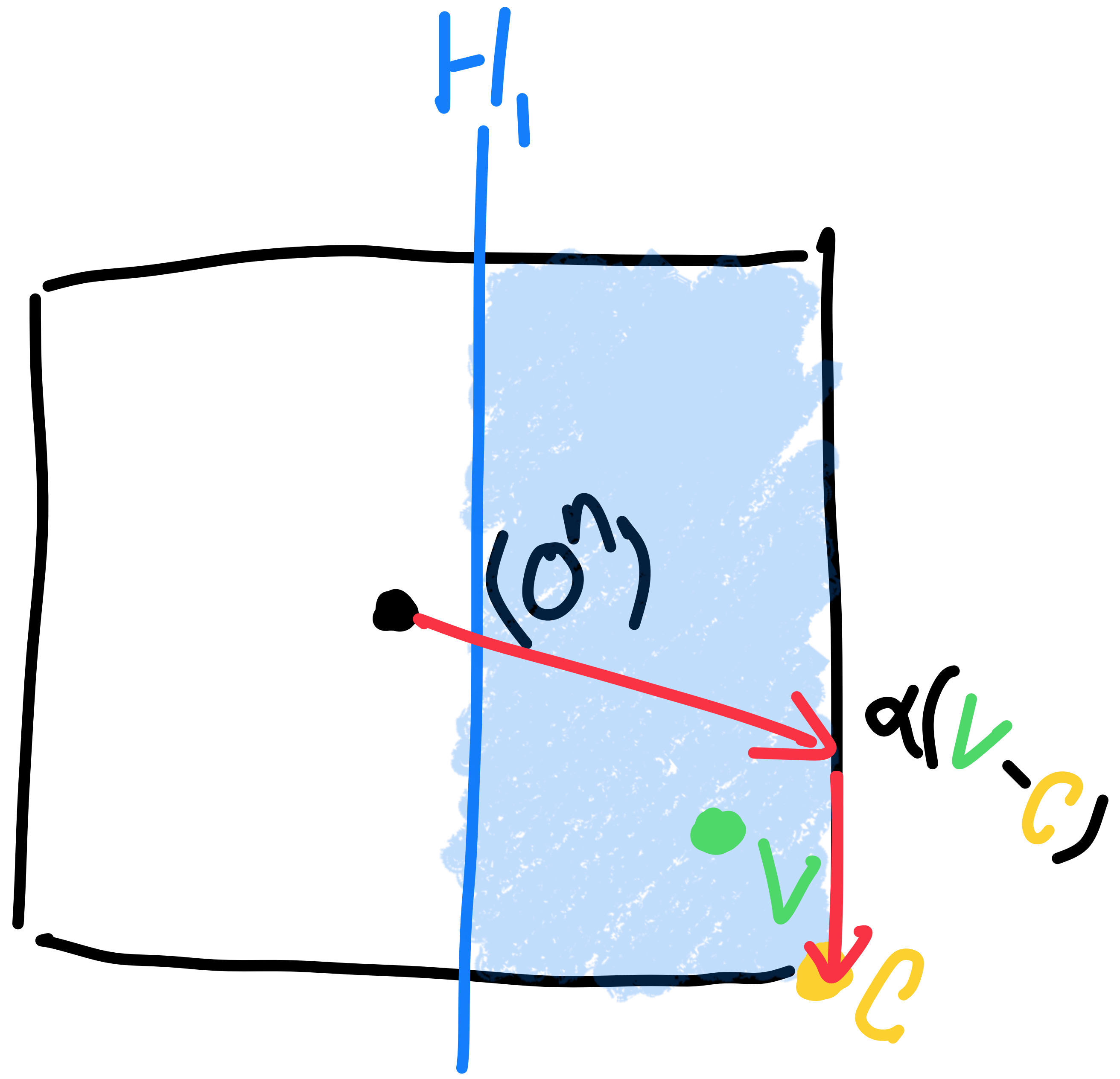
- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1, 1$ $\rightarrow$ moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v) = 0$ (otherwise set $v \leftarrow -v$, $c \leftarrow -c$)

  $\rightarrow$ $H_1(v) = 0$ or $H_2(v) = 0$ (assume H_1)
- Starting at $y = 0^n$ move in the direction of $c - v$ until one of the two non-zero coordinates becomes in $\{-1,1\}$. Let $\alpha(c-v)$ for $\alpha > 0$ be this point. Note: $\alpha > 1$ as c is the closest corner to v, so $\|c - v\|_\infty \leq 1$

  $\rightarrow$ $a_1(\alpha(c-v)) = \alpha a_1 c - \alpha a_1 v = \alpha a_1 c < b_1$
- By monotonicity of H_1 we can round the remaining non-$\{-1,0,1\}$ coordinate to $\{-1,1\}$



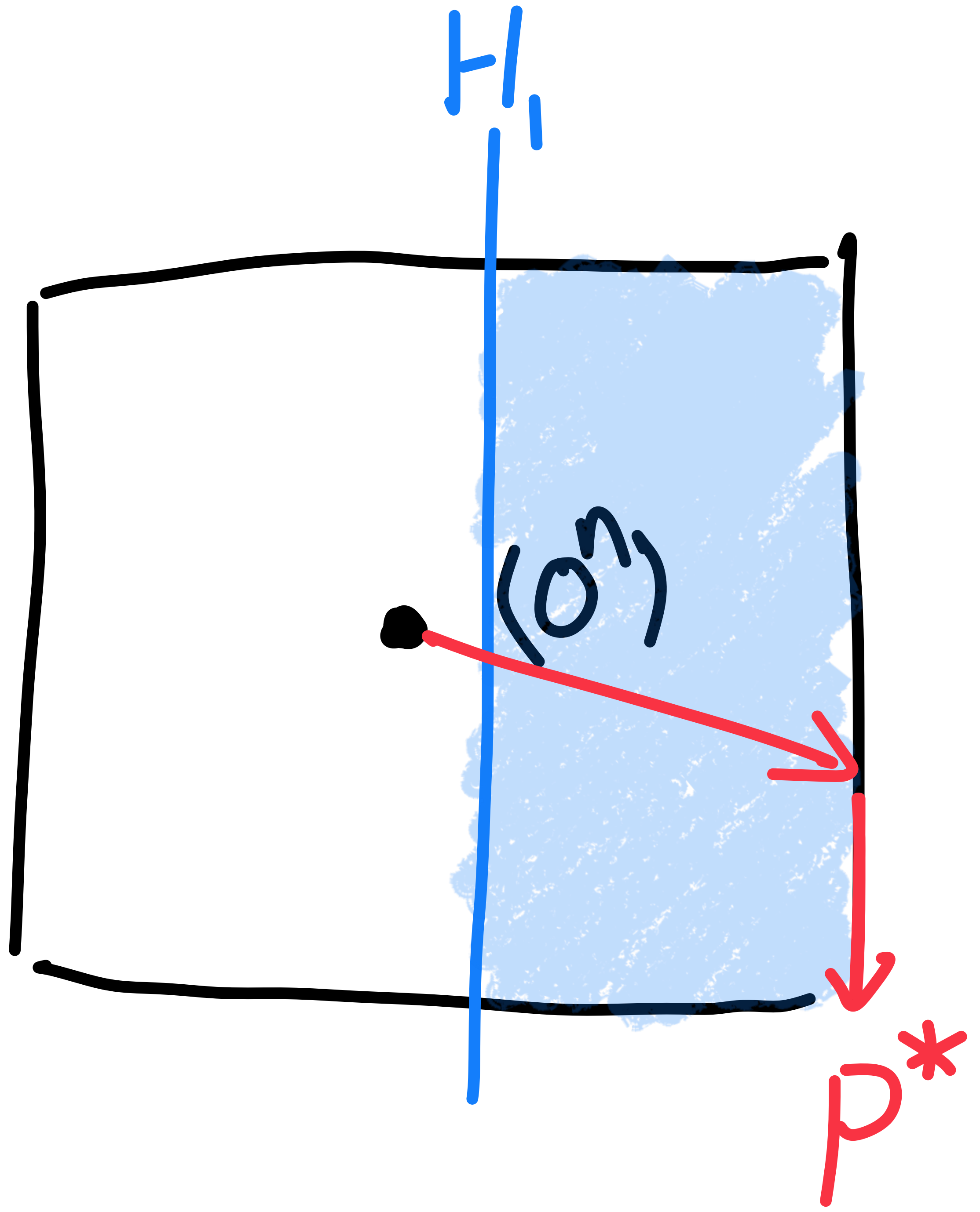
Proof Idea of Technical Lemma



- Let $H_1 = a_1 x \geq b_1$, $H_2 = a_2 x > b_2$
- Find $v \in [-1,1]^n$ orthogonal to a_1, a_2 s.t. at most two coordinates of v are not $-1, 1$ → moving in direction of v doesn't affect H_1, H_2 i.e. $a_1(x+v) = a_1 x$
- Let $c \in \{-1,1\}^n$ be closest to v
- Suppose $H(v) = 0$ (otherwise set $v \leftarrow -v$, $c \leftarrow -c$)
  → $H_1(v) = 0$ or $H_2(v) = 0$ (assume H_1)
- Starting at $y = 0^n$ move in the direction of $c - v$ until one of the two non-zero coordinates becomes in $\{-1,1\}$. Let $\alpha(c-v)$ for $\alpha > 0$ be this point. Note: $\alpha > 1$ as c is the closest corner to v, so $\|c - v\|_\infty \leq 1$
  → $a_1(\alpha(c-v)) = \alpha a_1 c - \alpha a_1 v = \alpha a_1 c < b_1$
- By monotonicity of H_1 we can round the remaining non-$\{-1,0,1\}$ coordinate to $\{-1,1\}$

Proof Idea of Technical Lemma

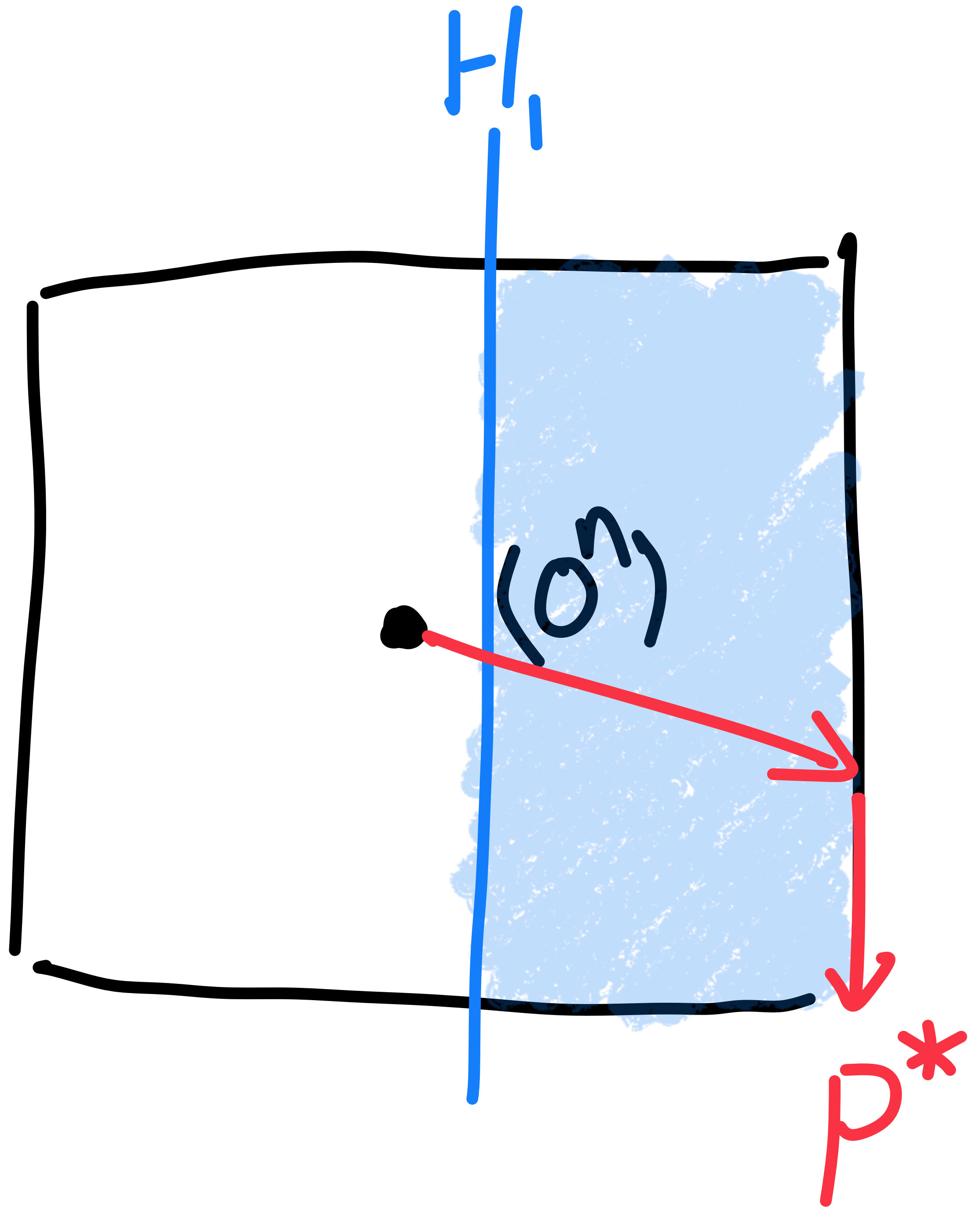
- ∴ We have found a point $p^* \in \{-1,0,1\}^n$ s.t.



Proof Idea of Technical Lemma

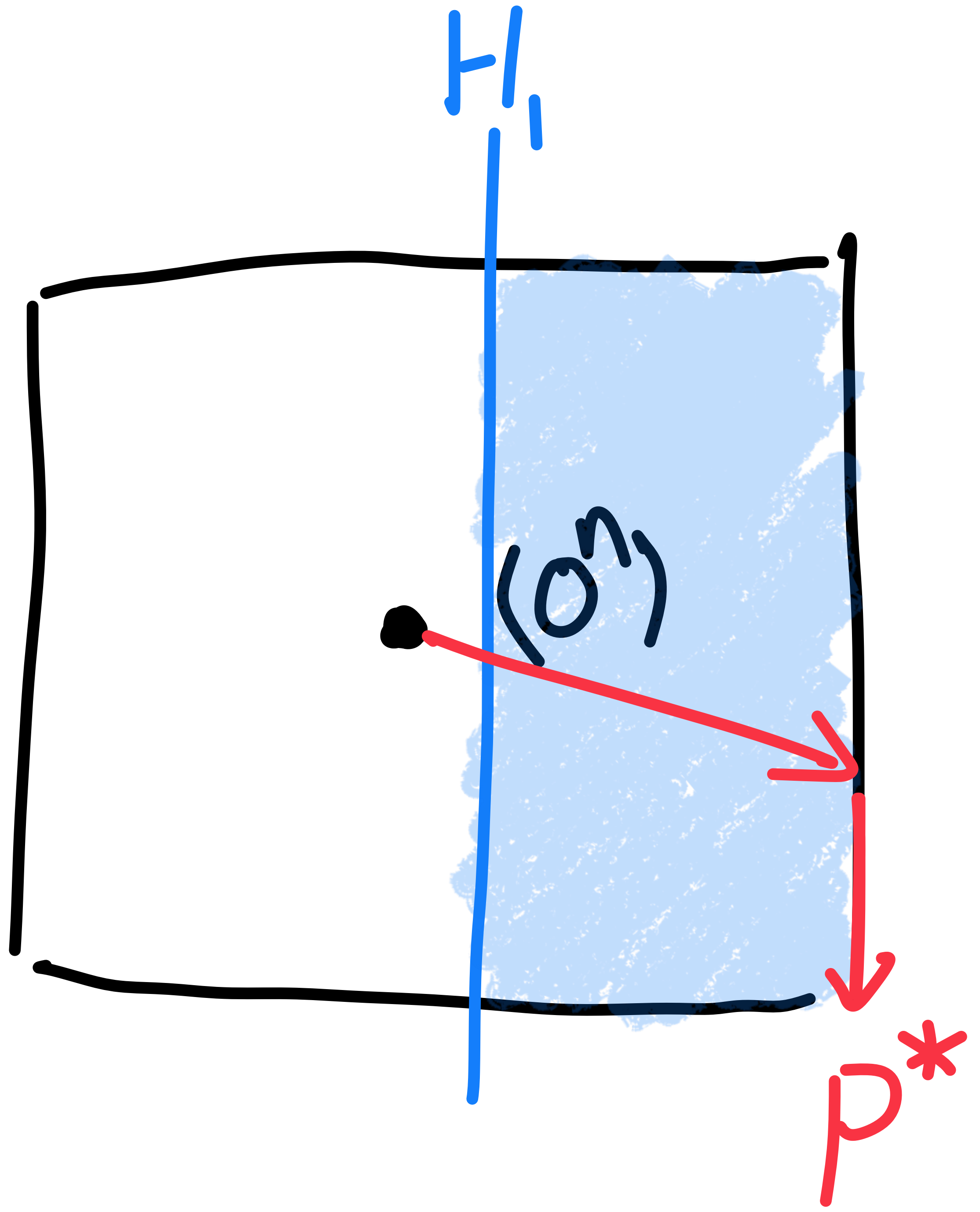
$\therefore$ We have found a point $p^* \in \{-1,0,1\}^n$ s.t.

- 2 coordinates are in $\{-1,1\}$



Proof Idea of Technical Lemma

- $\therefore$ We have found a point $p^* \in \{-1,0,1\}^n$ s.t.
  - 2 coordinates are in $\{-1,1\}$
  - $H_1(p^*) = 0$

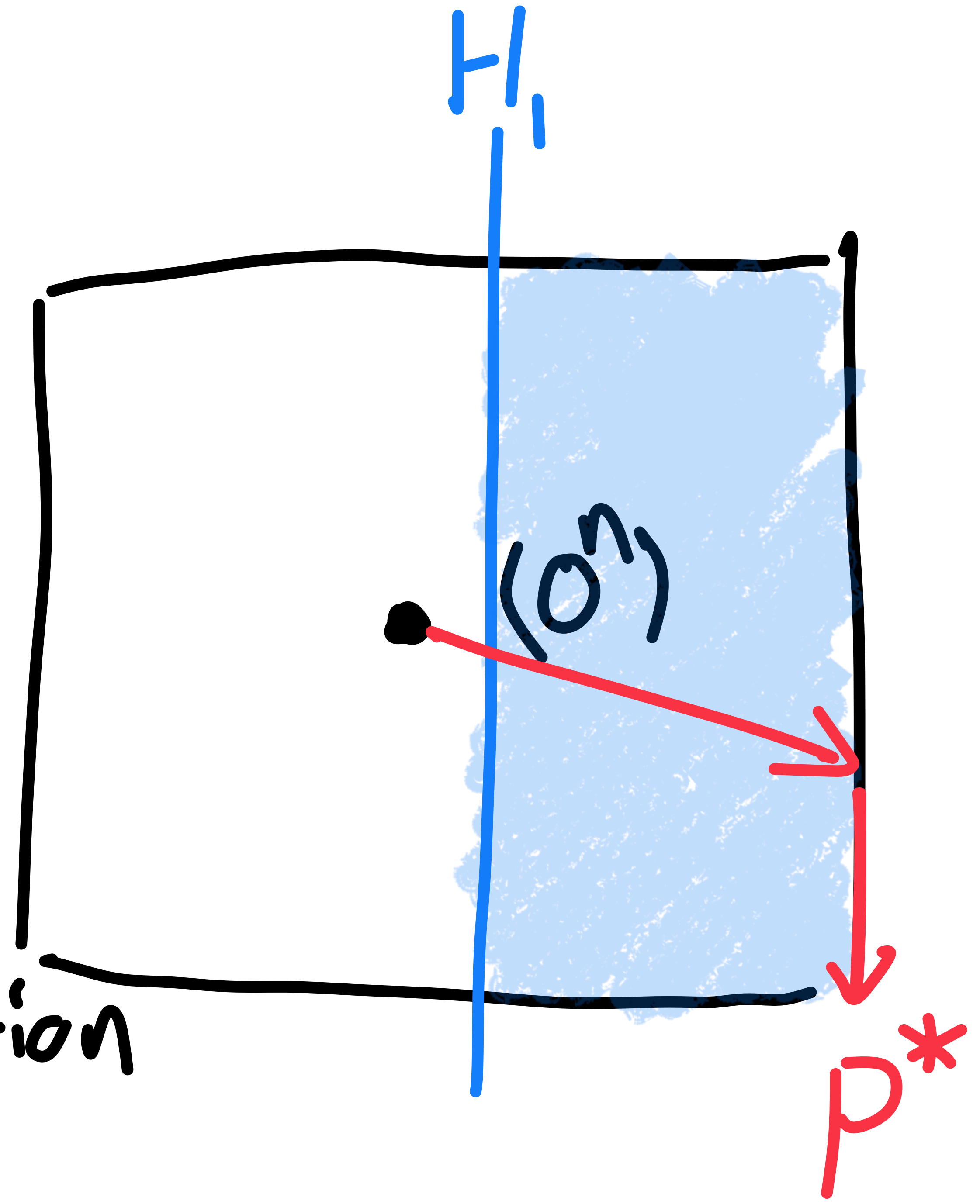


Proof Idea of Technical Lemma

∴ We have found a point $p^* \in \{-1,0,1\}^n$ s.t.

- 2 coordinates are in $\{-1,1\}$
- $H_1(p^*) = 0$

→ take the $\{-1,1\}$ coordinates to be our restriction p'

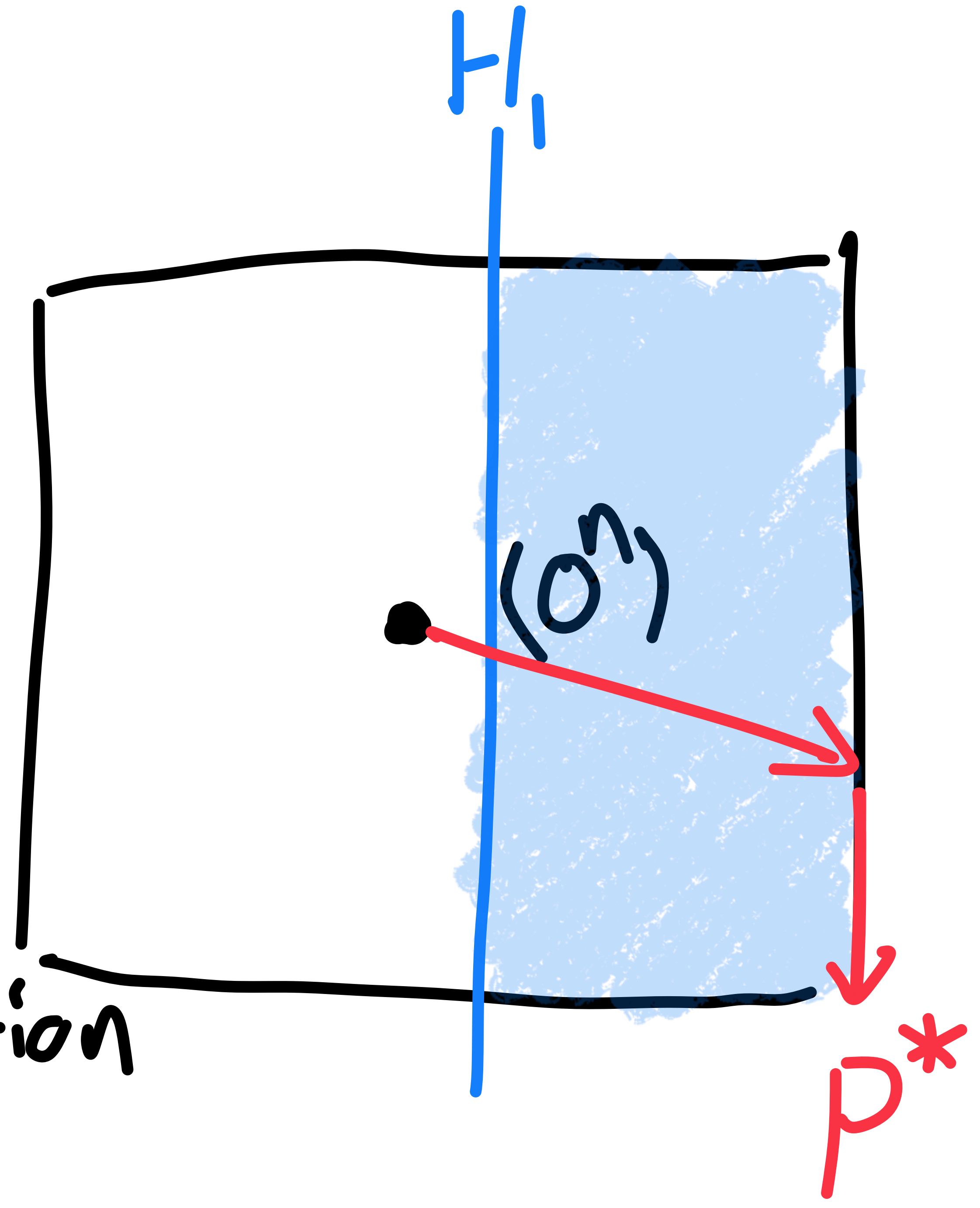


Proof Idea of Technical Lemma

$\therefore$ We have found a point $p^* \in \{-1,0,1\}^n$ s.t.

- 2 coordinates are in $\{-1,1\}$
- $H_1(p^*) = 0$

$\rightarrow$ take the $\{-1,1\}$ coordinates to be our restriction $p' \longrightarrow H_1\restriction p'$ is good.



Open Problems

- Can CP p-simulate SP*?
- Can CP (quasipolynomially) simulate SP?
- Can CP* (quasipolynomially) simulate SP*?
- Can SP or CP simulate dag-like SP (R(CP))?
  - → CP cannot p-simulate R(CP) [ABE02]
- Can treelike CP refute systems of $\mathbb{F}_2$ linear equations?