The Proof Complexity of Integer Programming

Noah Fleming

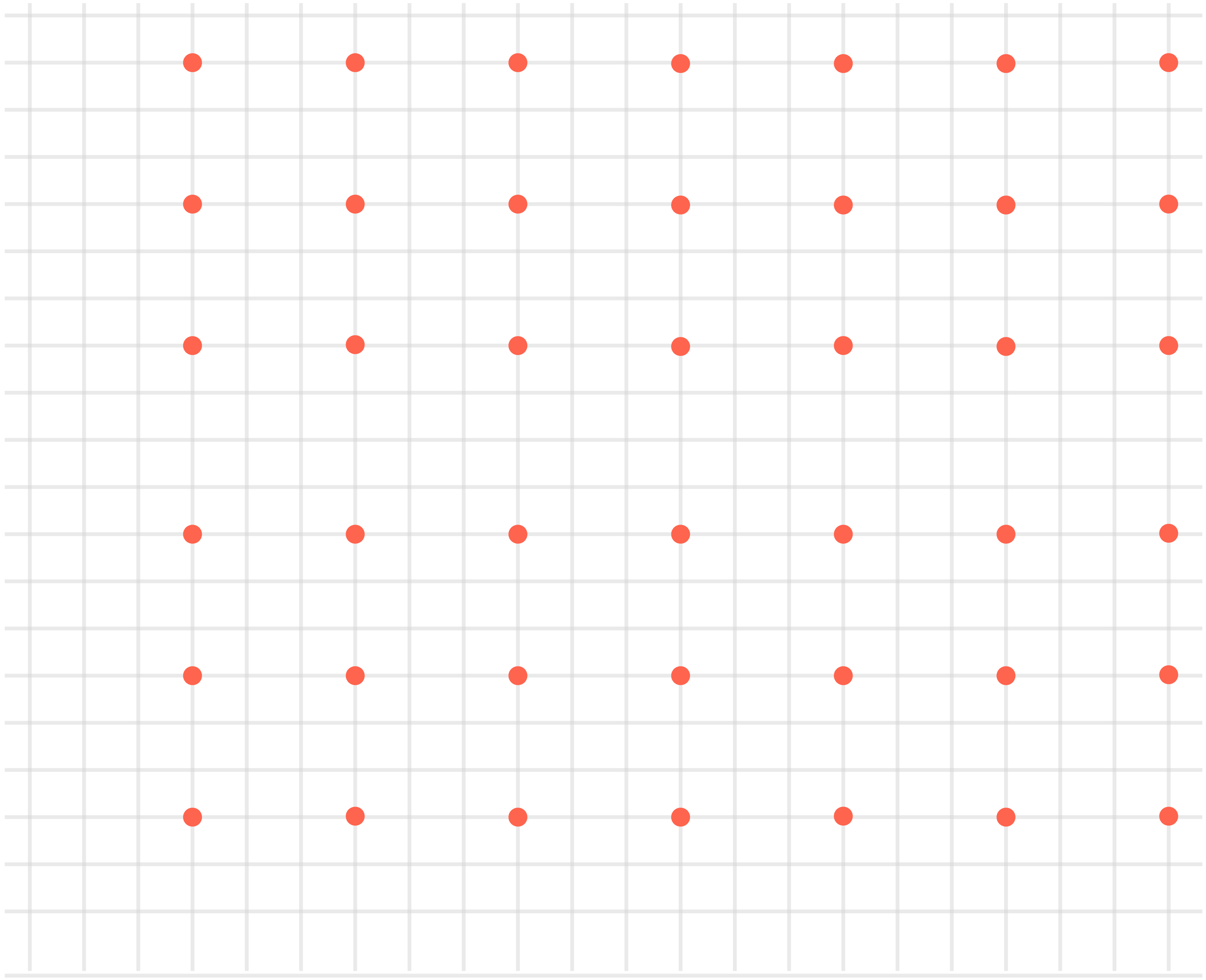
Memorial University

Integer Programming

Input

- A system of linear inequalities

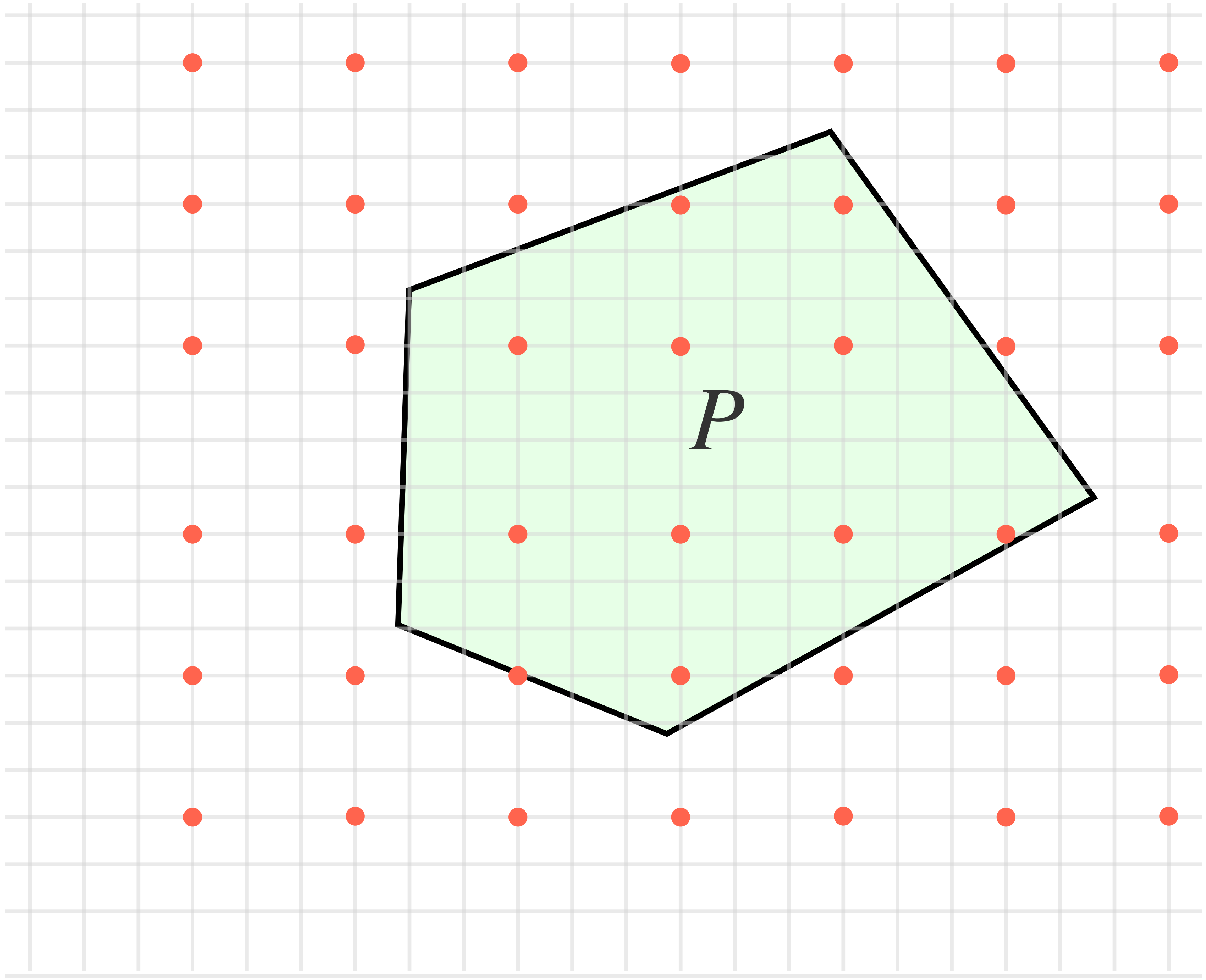
$$a_1 x \geq b_1, \ldots, a_m x \geq b_m$$



Integer Programming

Input

- A system of linear inequalities
  $$P = \{x : a_1 x \geq b_1, \ldots, a_m x \geq b_m\}$$
  defining a polytope P in $\mathbb{R}^n$

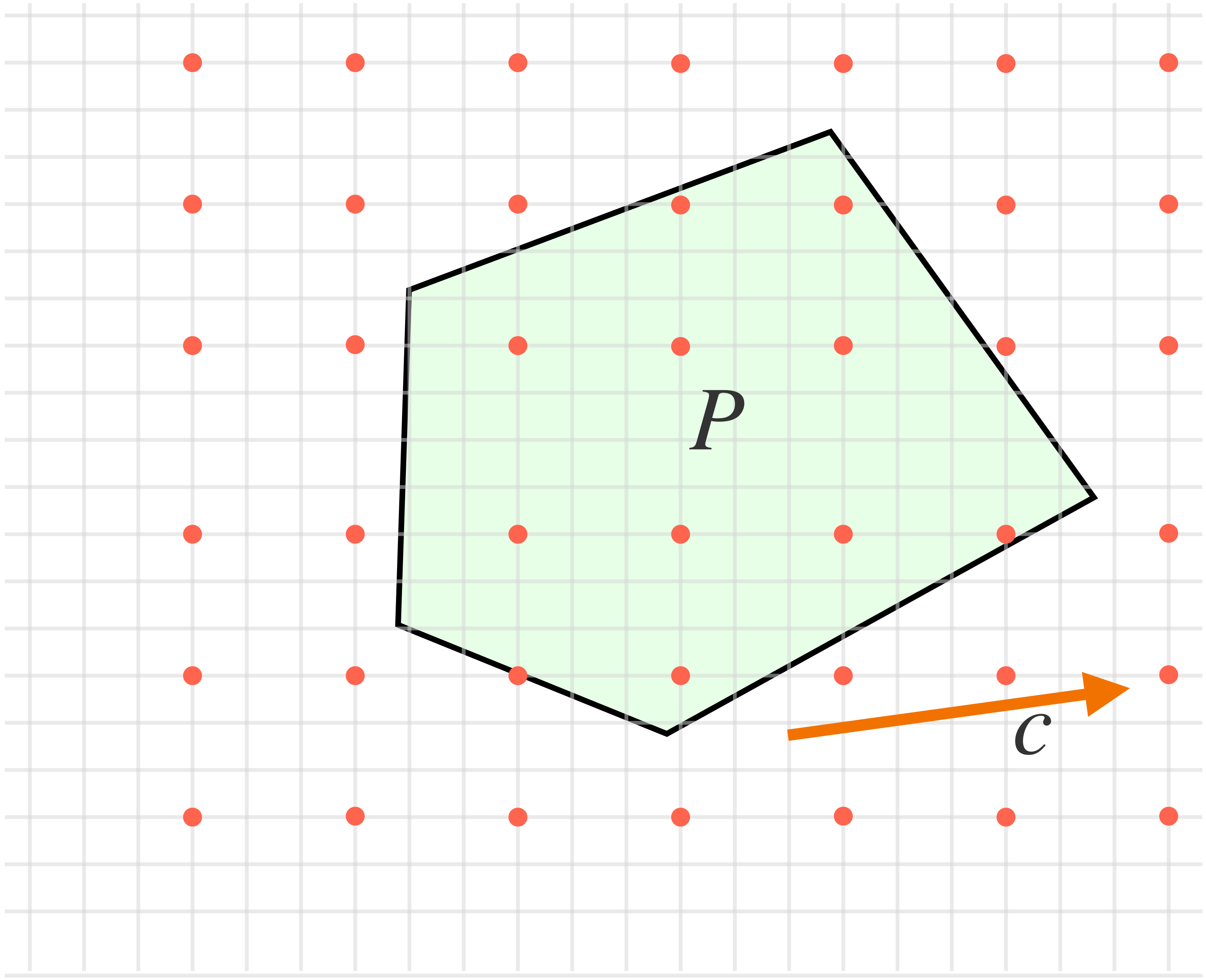


Integer Programming

Input

- A system of linear inequalities
$$P = \{x : a_1 x \geq b_1, \ldots, a_m x \geq b_m\}$$
defining a polytope P in $\mathbb{R}^n$

- A "direction" $c \in \mathbb{R}^n$



Integer Programming

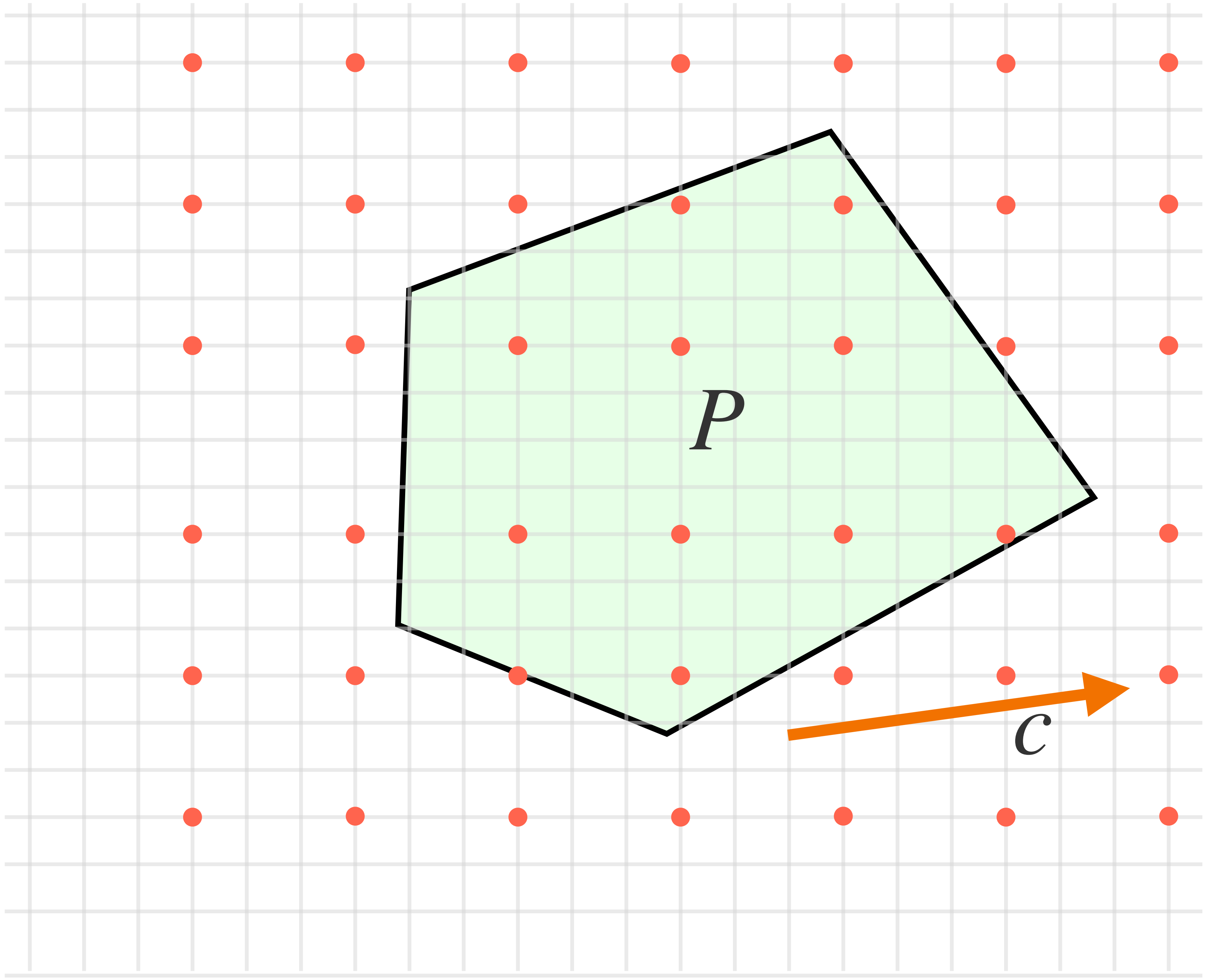
Input

- A system of linear inequalities
$$P = \{x : a_1 x \geq b_1, \ldots, a_m x \geq b_m\}$$
defining a polytope P in $\mathbb{R}^n$

- A "direction" $c \in \mathbb{R}^n$

Output

- An integer point $x \in P$ maximizing cx



Integer Programming

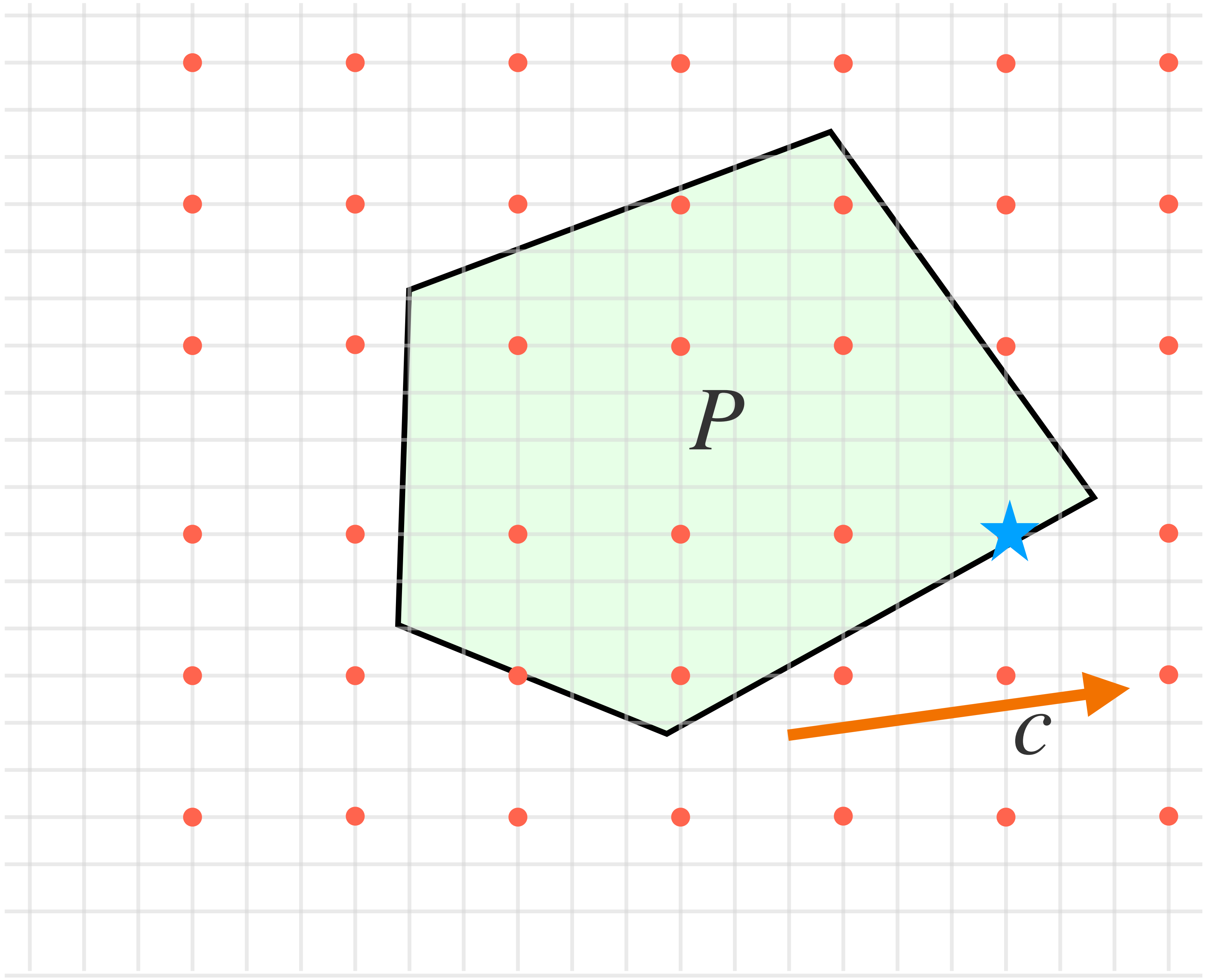
Input

- A system of linear inequalities
  $$P = \{x : a_1 x \geq b_1, \ldots, a_m x \geq b_m\}$$
  defining a polytope P in $\mathbb{R}^n$

- A "direction" $c \in \mathbb{R}^n$

Output

- An integer point $x \in P$ maximizing cx



Integer Programming

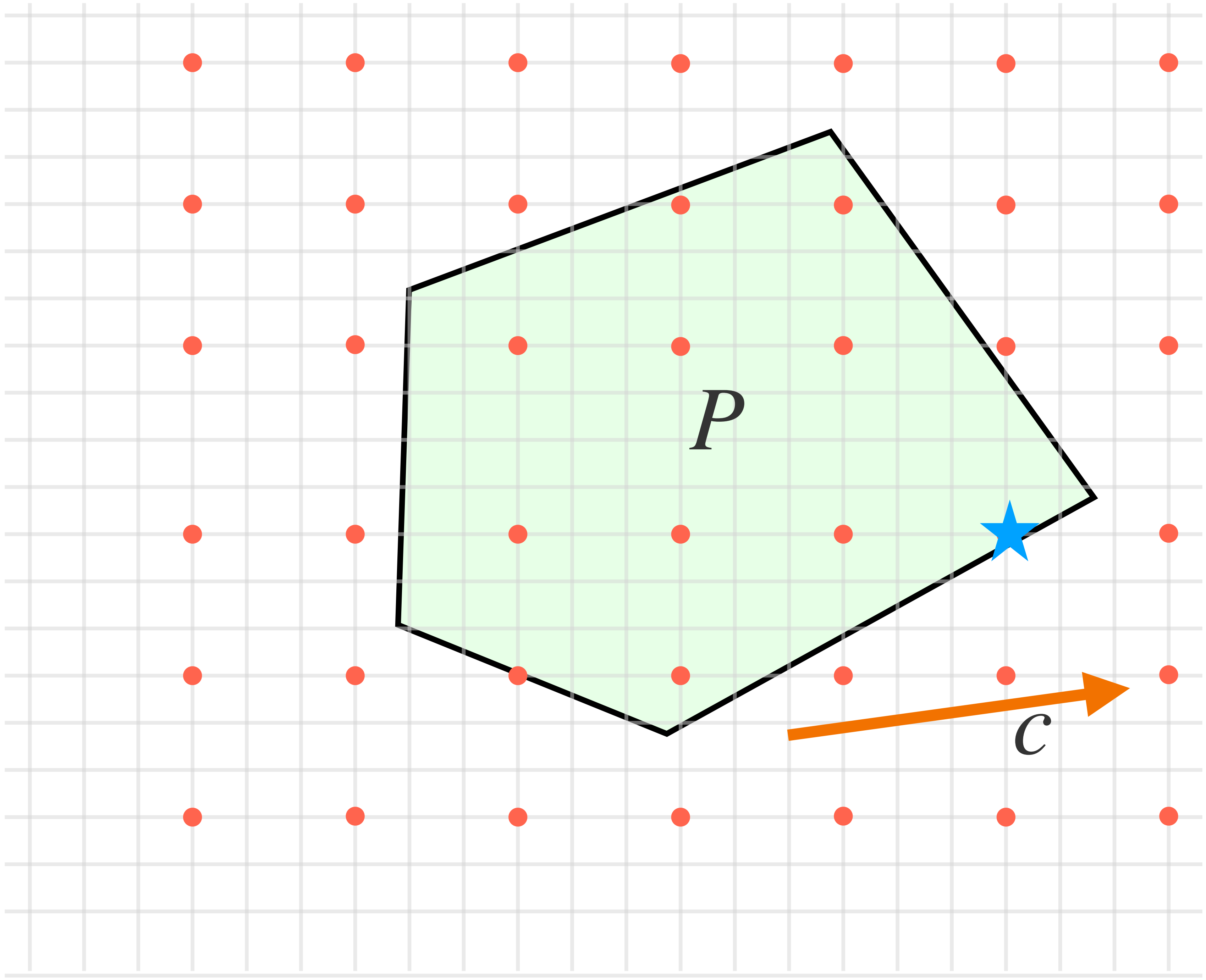
Input

- A system of linear inequalities
$$P = \{x : a_1 x \geq b_1, \ldots, a_m x \geq b_m\}$$
defining a polytope P in $\mathbb{R}^n$

- A "direction" $c \in \mathbb{R}^n$

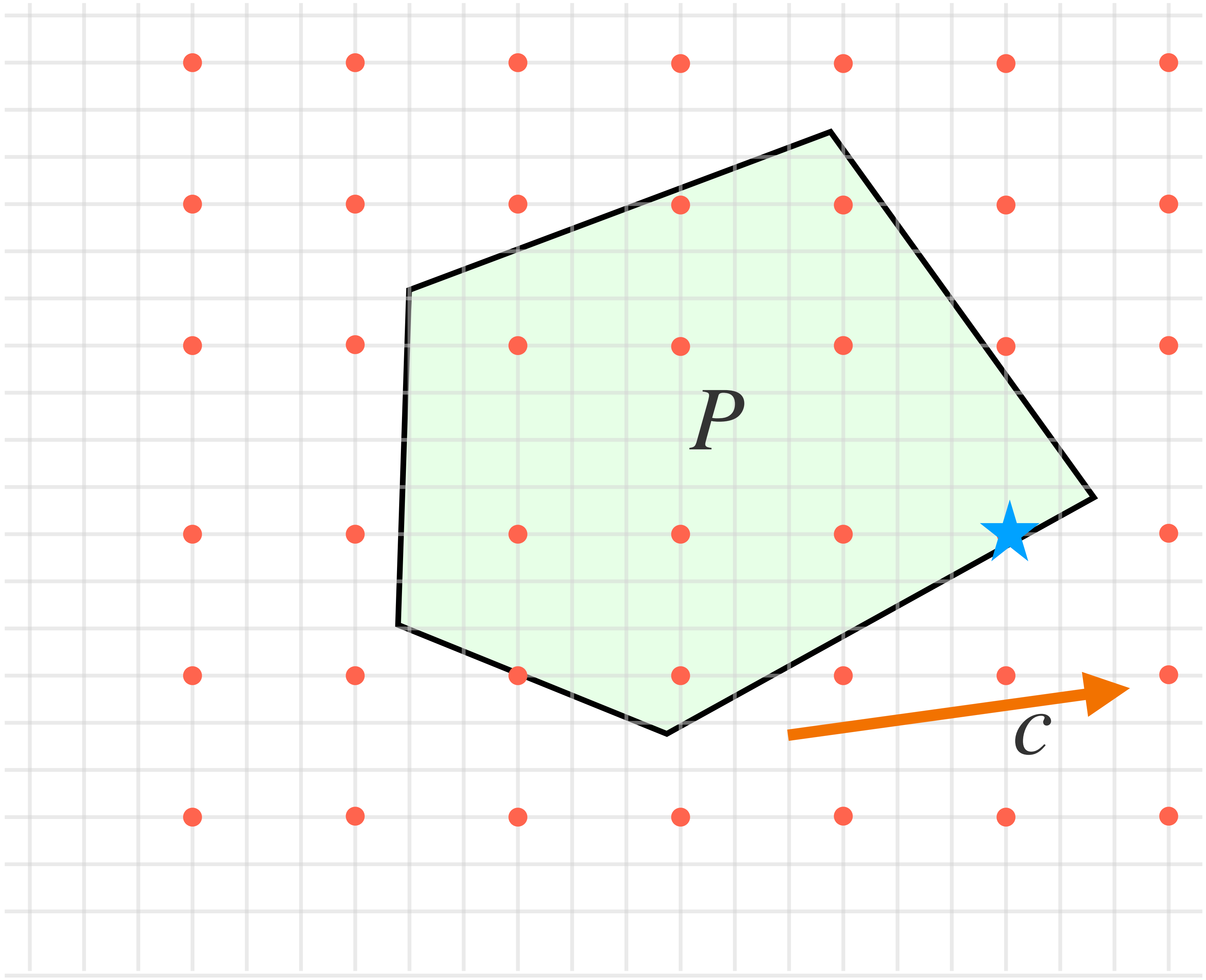
Output

- An integer point $x \in P$ maximizing cx



Very general framework. However, NP-hard to solve

Integer Programming



Input

- A system of linear inequalities
  $$P = \{x : a_1 x \geq b_1, \ldots, a_m x \geq b_m\}$$
  defining a polytope P in $\mathbb{R}^n$

- A "direction" $c \in \mathbb{R}^n$

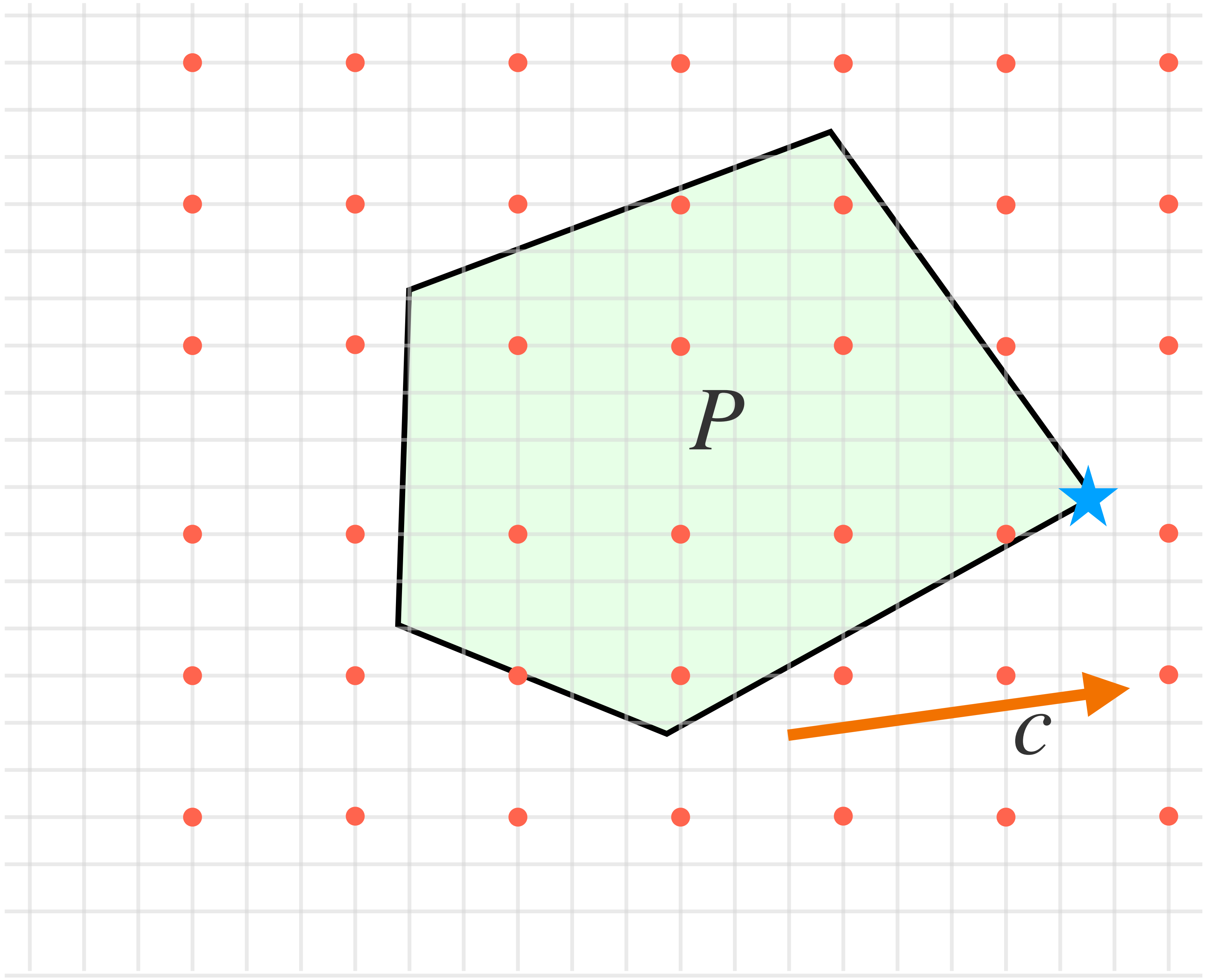
Output

- An integer point $x \in P$ maximizing cx

Very general framework. However, NP-hard to solve

→ Contrast with linear programming: output any $x \in P$ maximizing cx

Integer Programming



Input

- A system of linear inequalities
  $$P = \{x : a_1 x \geq b_1, \ldots, a_m x \geq b_m\}$$
  defining a polytope P in $\mathbb{R}^n$

- A "direction" $c \in \mathbb{R}^n$

Output

- An integer point $x \in P$ maximizing cx

Very general framework. However, NP-hard to solve

→ Contrast with linear programming: output any $x \in P$ maximizing cx

Integer Programming

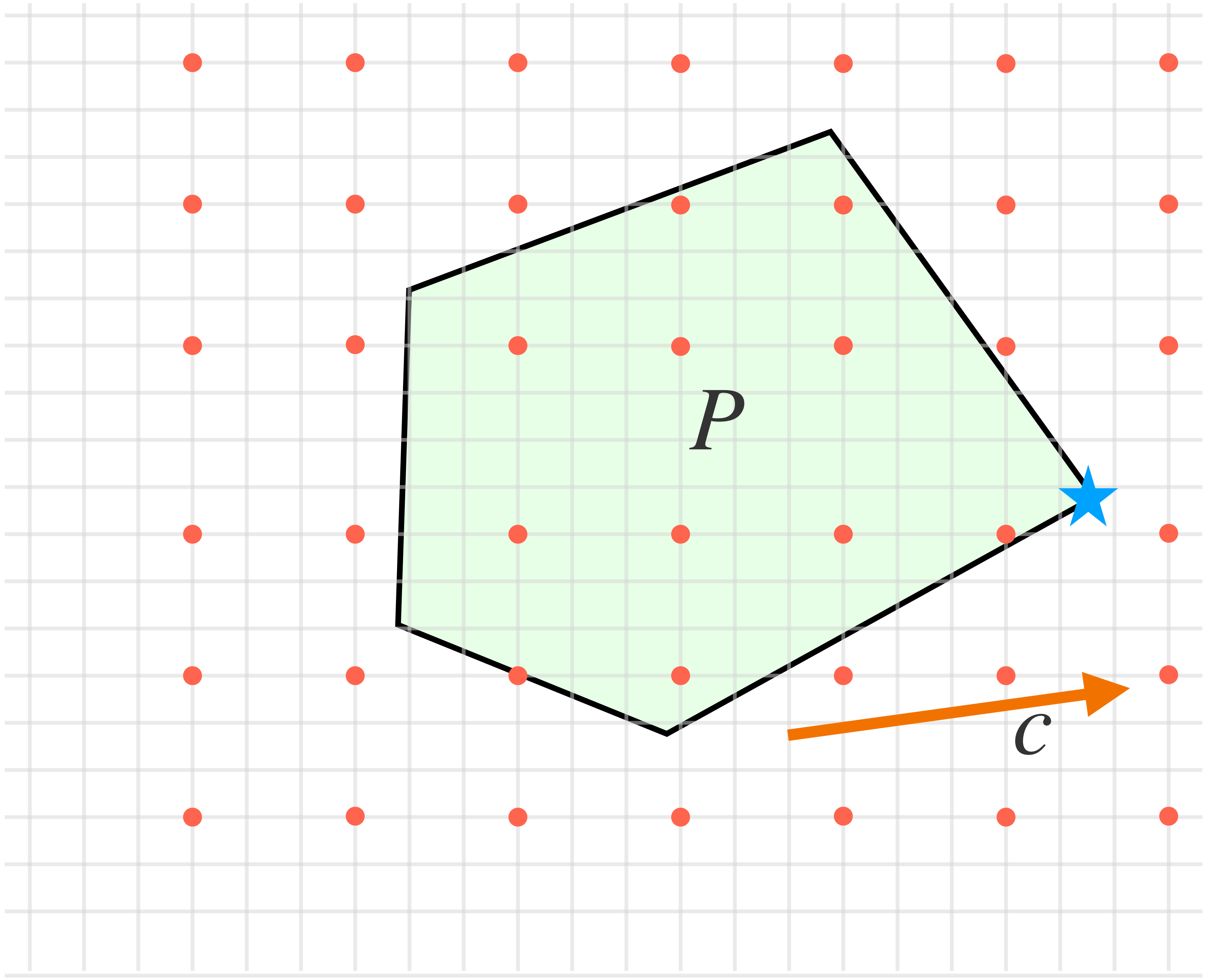
Input

- A system of linear inequalities
$$P = \{x : a_1x \geq b_1, \ldots, a_mx \geq b_m\}$$
defining a polytope P in $\mathbb{R}^n$

- A "direction" $c \in \mathbb{R}^n$

Output

- An integer point $x \in P$ maximizing cx



Very general framework. However, NP-hard to solve

→ Contrast with linear programming: output any $x \in P$ maximizing cx

Like SAT, practitioners routinely able to solve practical instances of IP

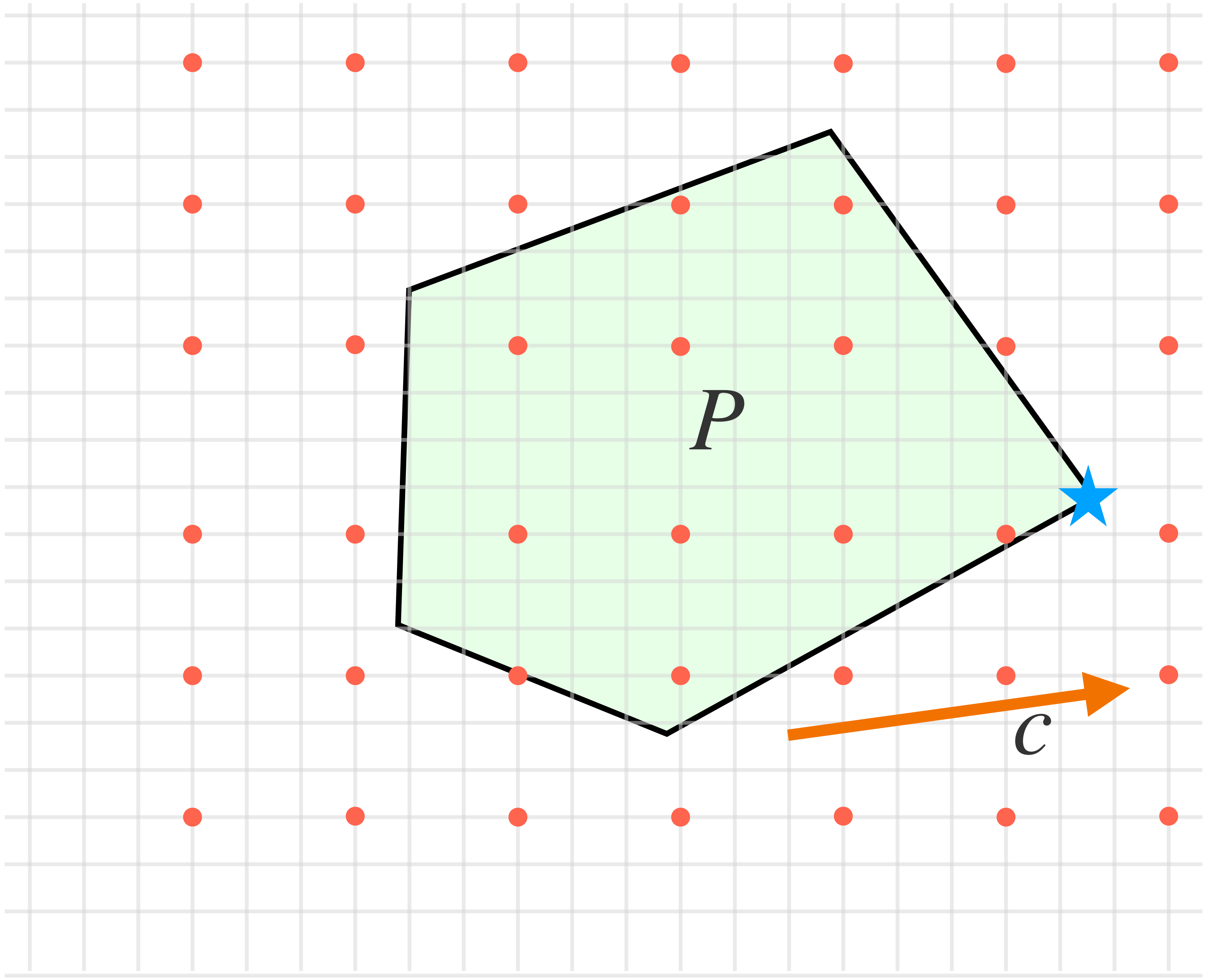
Integer Programming

Input

- A system of linear inequalities
  $$P = \{x : a_1 x \geq b_1, \ldots, a_m x \geq b_m\}$$
  defining a polytope P in $\mathbb{R}^n$
- A "direction" $c \in \mathbb{R}^n$

Output

- An integer point $x \in P$ maximizing cx



Very general framework. However, NP-hard to solve

→ Contrast with linear programming: output any $x \in P$ maximizing cx

Like SAT, practitioners routinely able to solve practical instances of IP

→ How?

Integer Programming

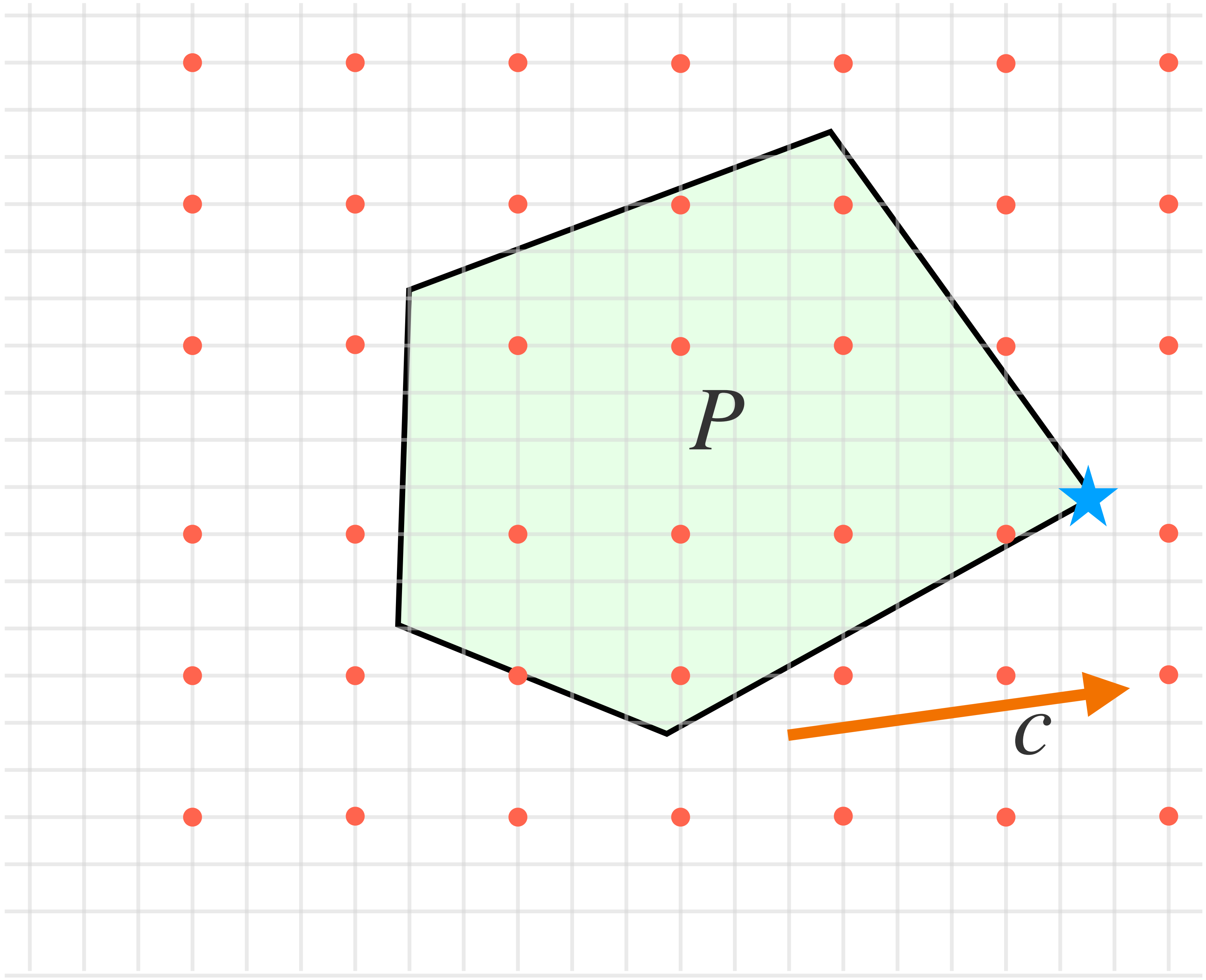
Input

- A system of linear inequalities
$$P = \{x : a_1 x \geq b_1, \ldots, a_m x \geq b_m\}$$
defining a polytope P in $\mathbb{R}^n$

- A "direction" $c \in \mathbb{R}^n$

Output

- An integer point $x \in P$ maximizing cx



Very general framework. However, NP-hard to solve

→ Contrast with linear programming: output any $x \in P$ maximizing cx

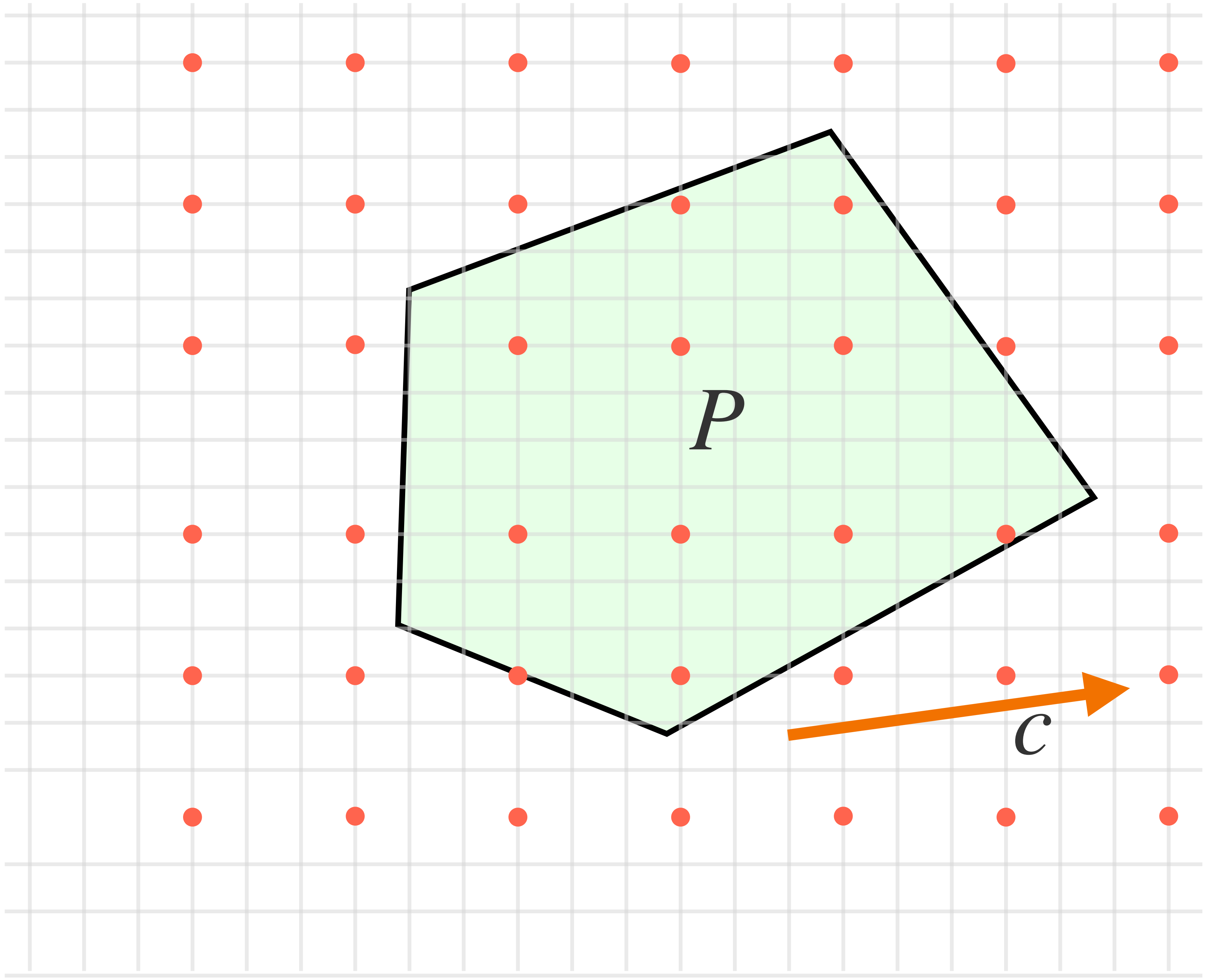
Like SAT, practitioners routinely able to solve practical instances of IP

→ How? — Reduce to linear programming!

Integer Programming in Practice

Integer Hull of a polytope P is

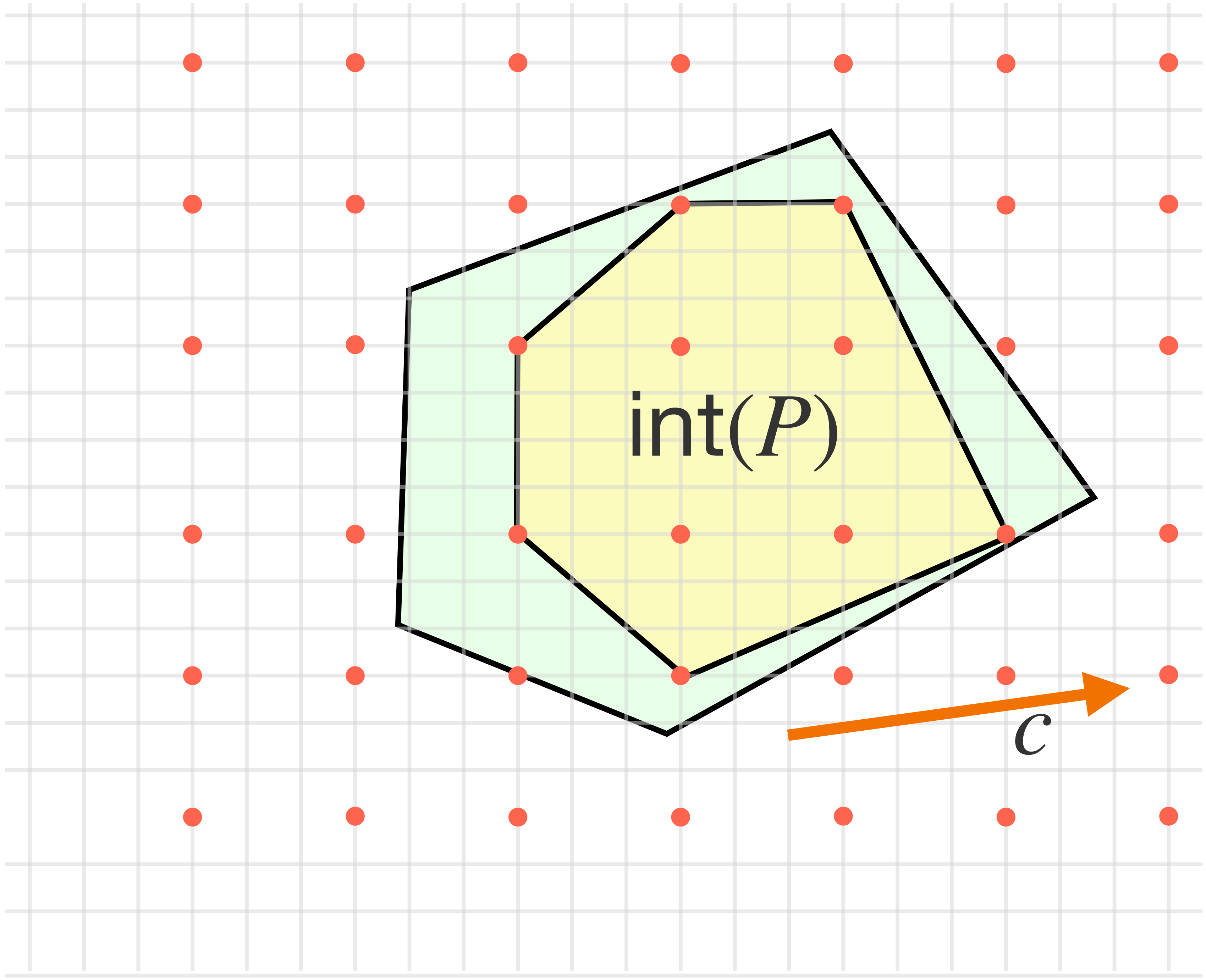
$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$



Integer Programming in Practice

Integer Hull of a polytope P is

$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$

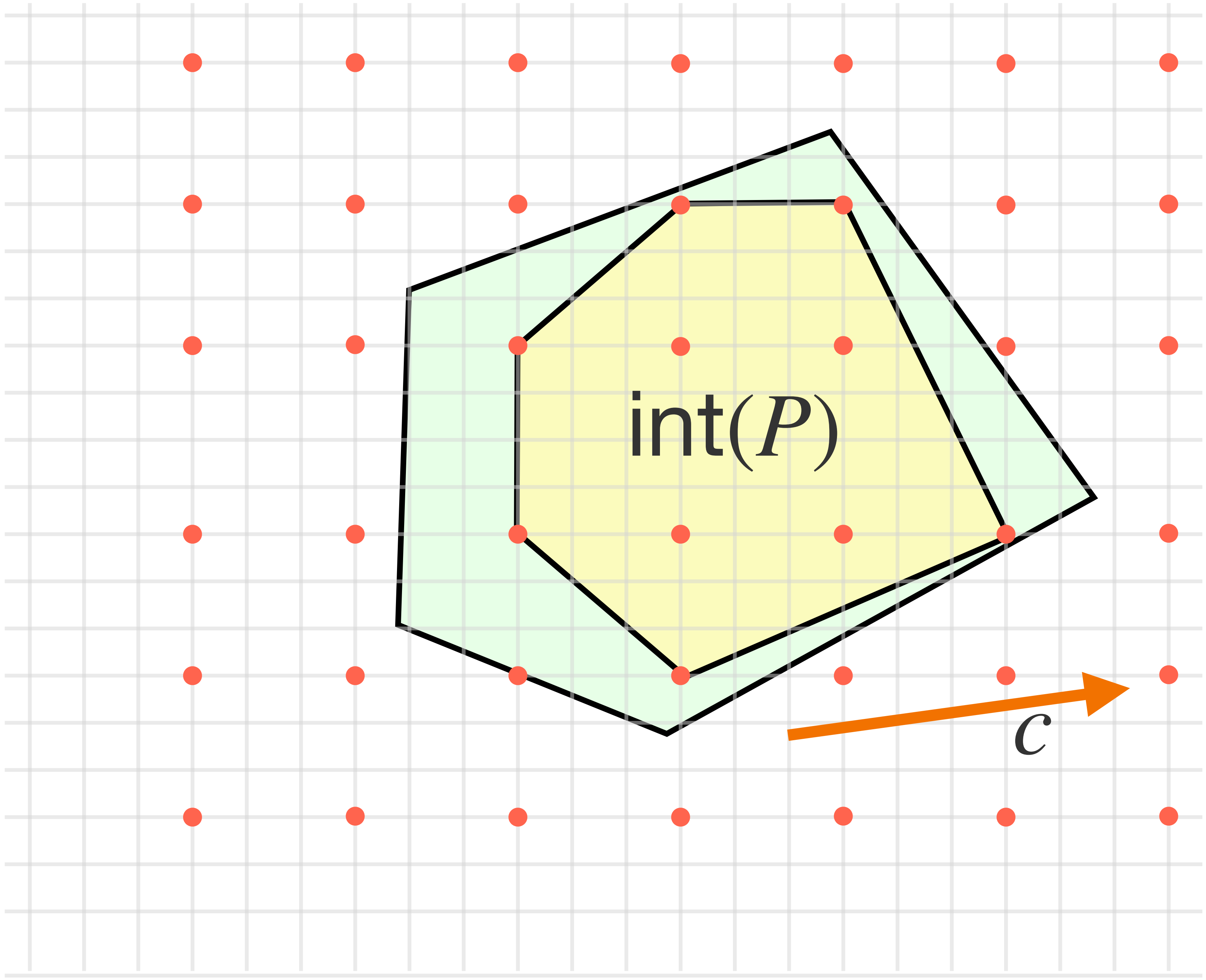


Integer Programming in Practice

Integer Hull of a polytope P is

$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$

- $\text{int}(P)$ is a polytope



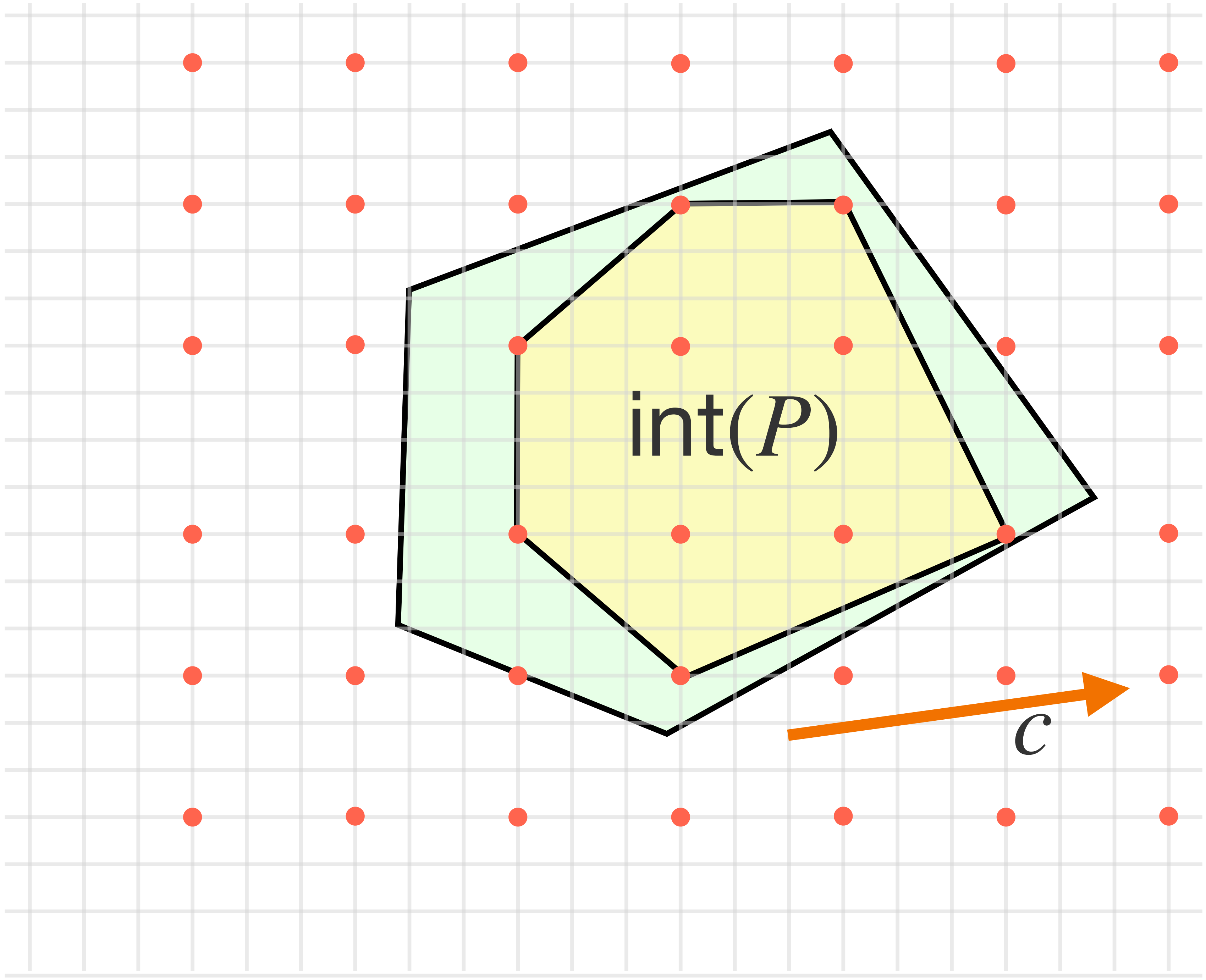
Integer Programming in Practice

Integer Hull of a polytope P is

$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$

- $\text{int}(P)$ is a polytope
- An LP solution to $\text{int}(P)$ is an ILP solution to P

$$LP(\text{int}(P), c) = ILP(P, c)$$



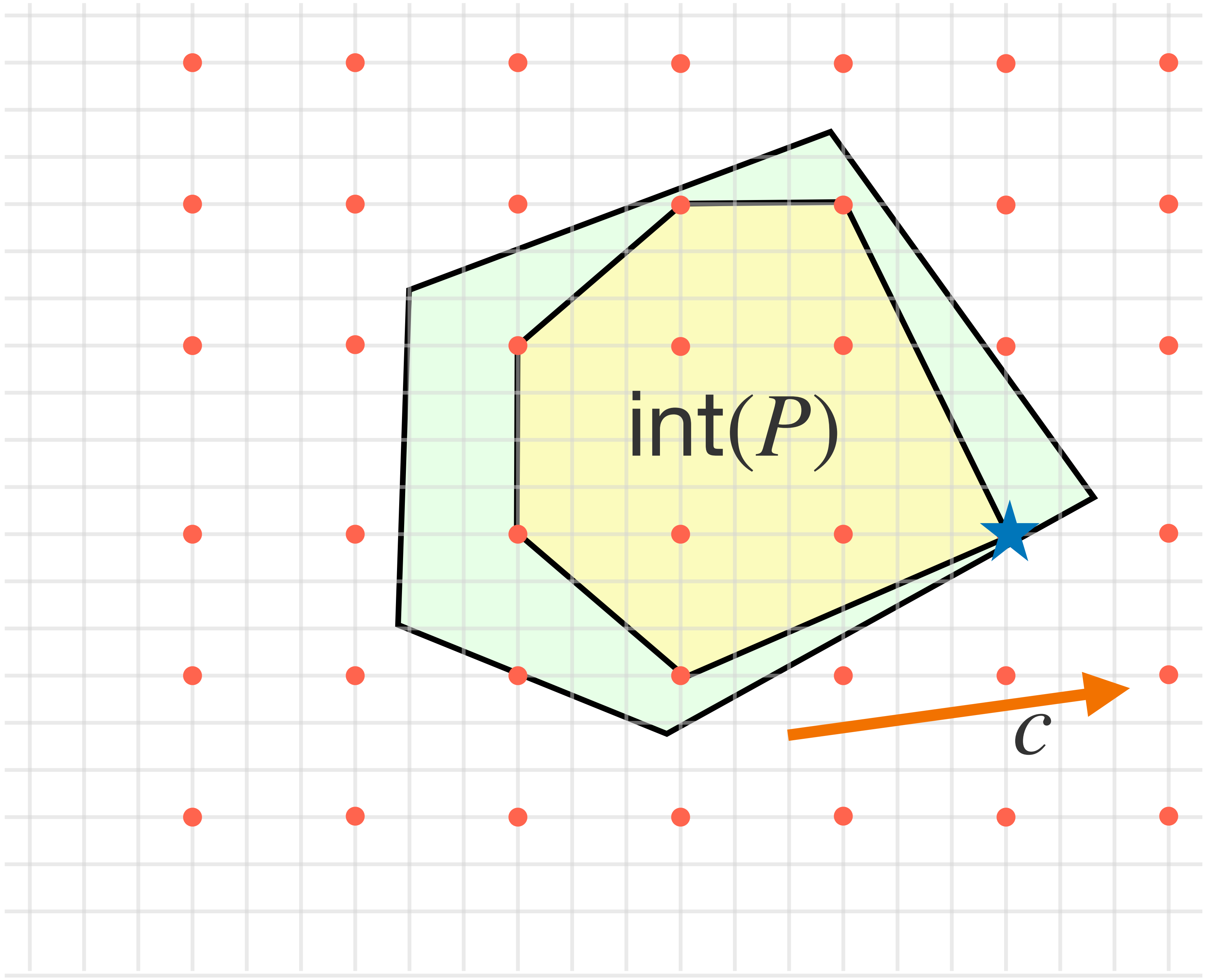
Integer Programming in Practice

Integer Hull of a polytope P is

$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$

- $\text{int}(P)$ is a polytope
- An LP solution to $\text{int}(P)$ is an ILP solution to P

$$LP(\text{int}(P), c) = ILP(P, c)$$



Integer Programming in Practice

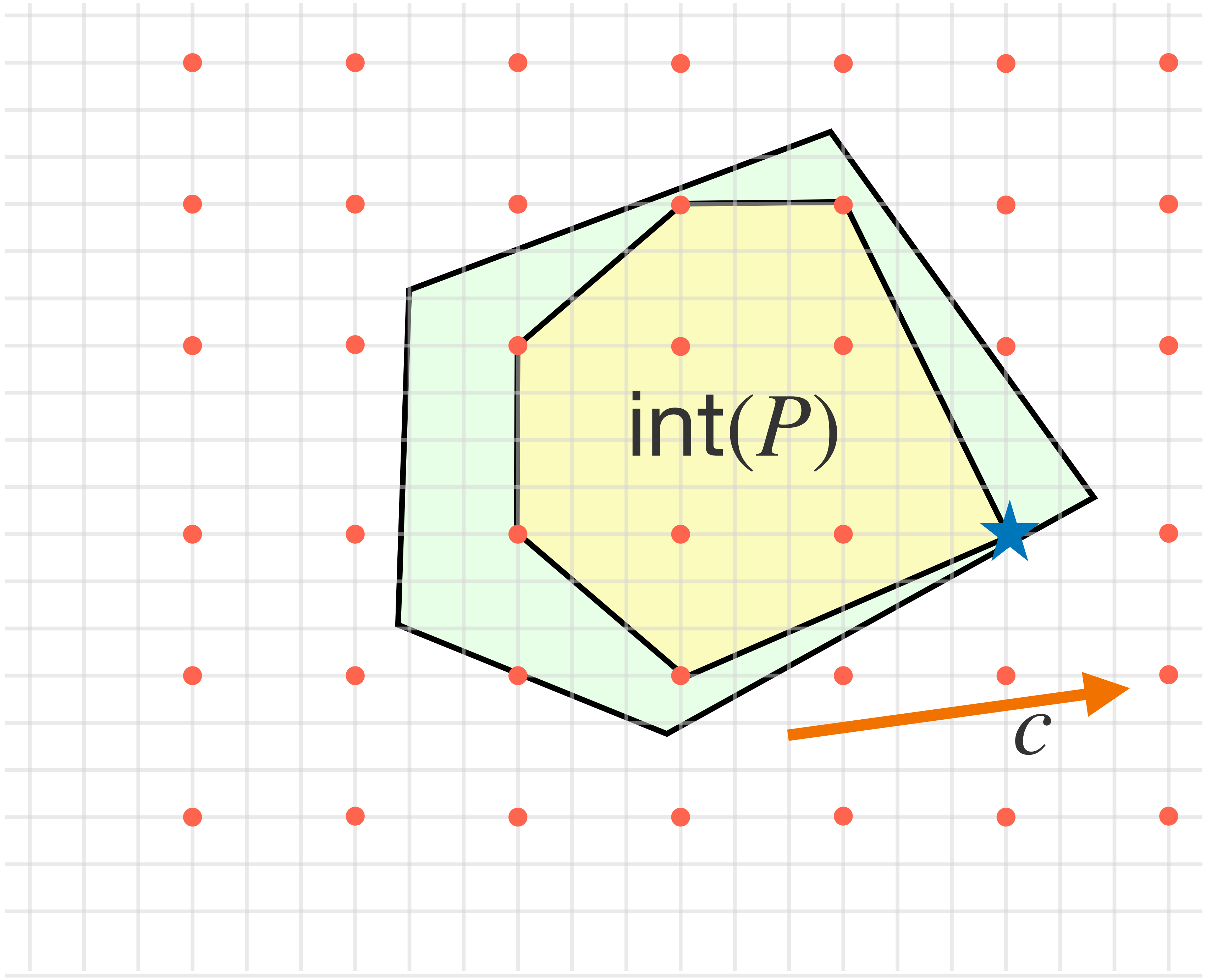
Integer Hull of a polytope P is

$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$

- $\text{int}(P)$ is a polytope
- An LP solution to $\text{int}(P)$ is an ILP solution to P

$$LP(\text{int}(P), c) = ILP(P, c)$$

Modern IP-algorithms try to reduce $P \rightarrow \text{int}(P)$



Integer Programming in Practice

Integer Hull of a polytope P is

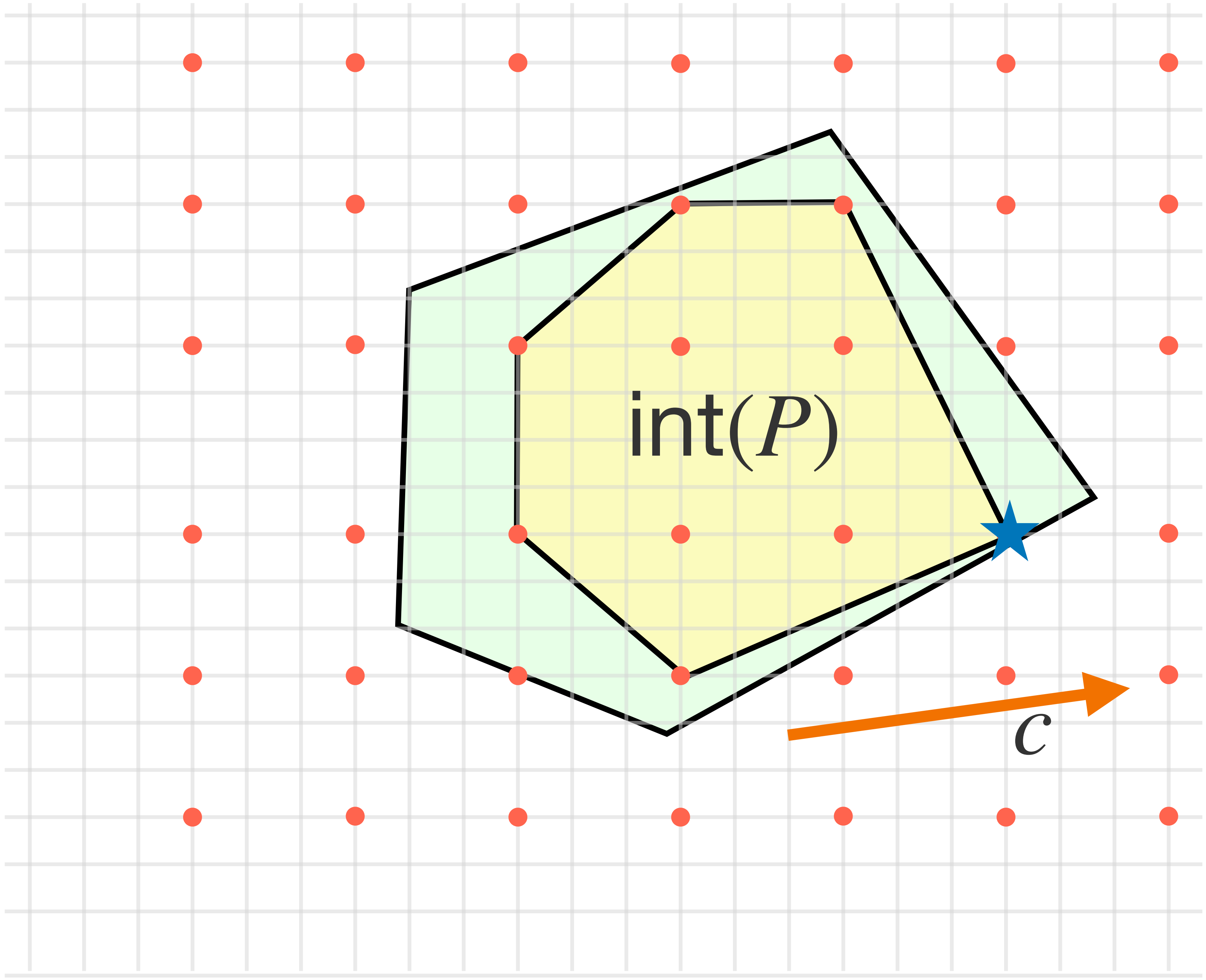
$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$

- $\text{int}(P)$ is a polytope
- An LP solution to $\text{int}(P)$ is an ILP solution to P

$$LP(\text{int}(P), c) = ILP(P, c)$$

Modern IP-algorithms try to reduce $P \to \text{int}(P)$

How? — [Gomory63, Chvátal73]: Add cutting planes



Integer Programming in Practice

Integer Hull of a polytope P is

$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$

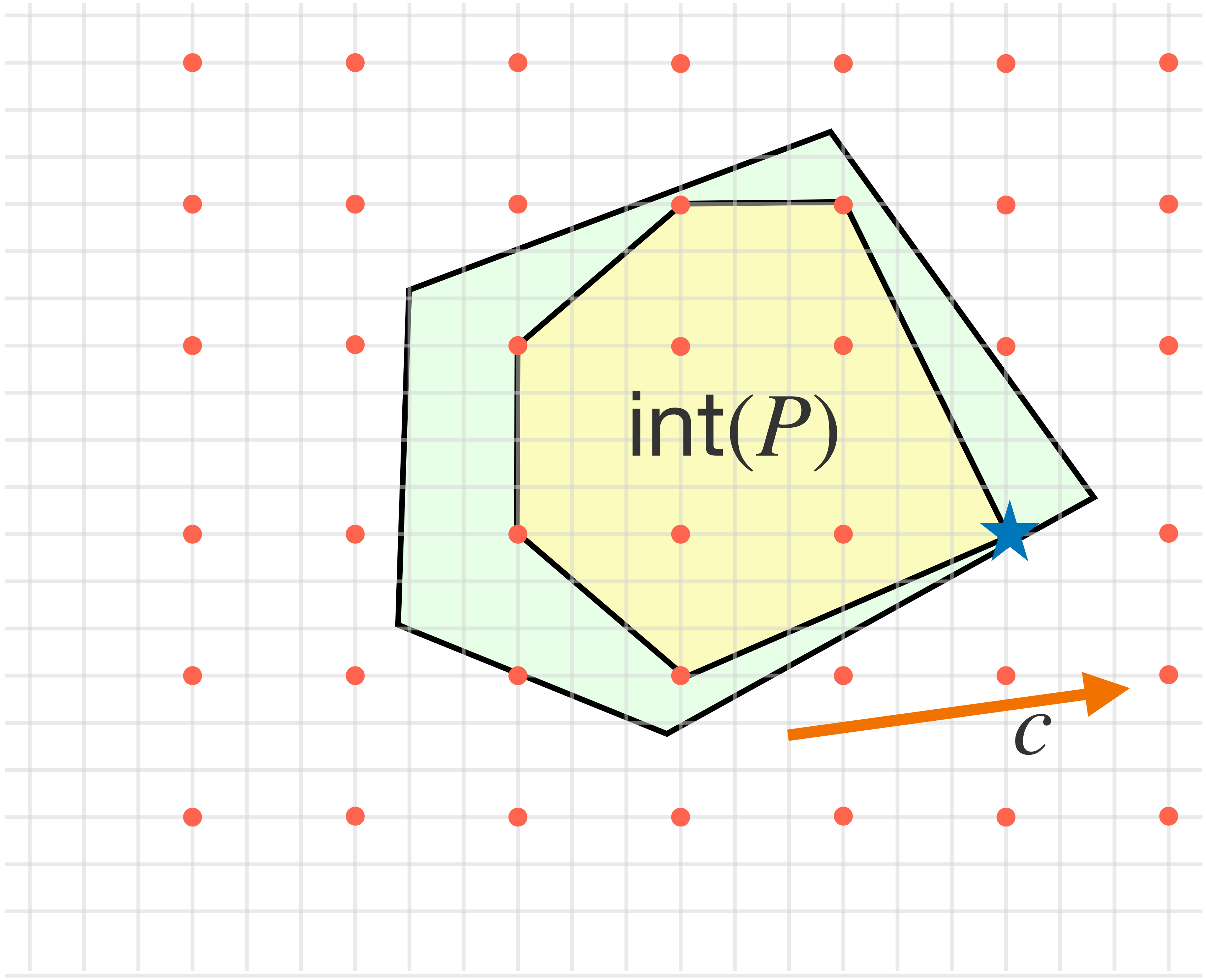
- $\text{int}(P)$ is a polytope
- An LP solution to $\text{int}(P)$ is an ILP solution to P

$$LP(\text{int}(P), c) = ILP(P, c)$$

Modern IP-algorithms try to reduce $P \to \text{int}(P)$

How? — [Gomory63, Chvátal73]: Add cutting planes

If $ax \geq b$ is valid for P with $a \in \mathbb{Z}^n$ relatively prime and $b \in \mathbb{R}$ then $ax \geq \lceil b \rceil$ is a **CG-cut** for P



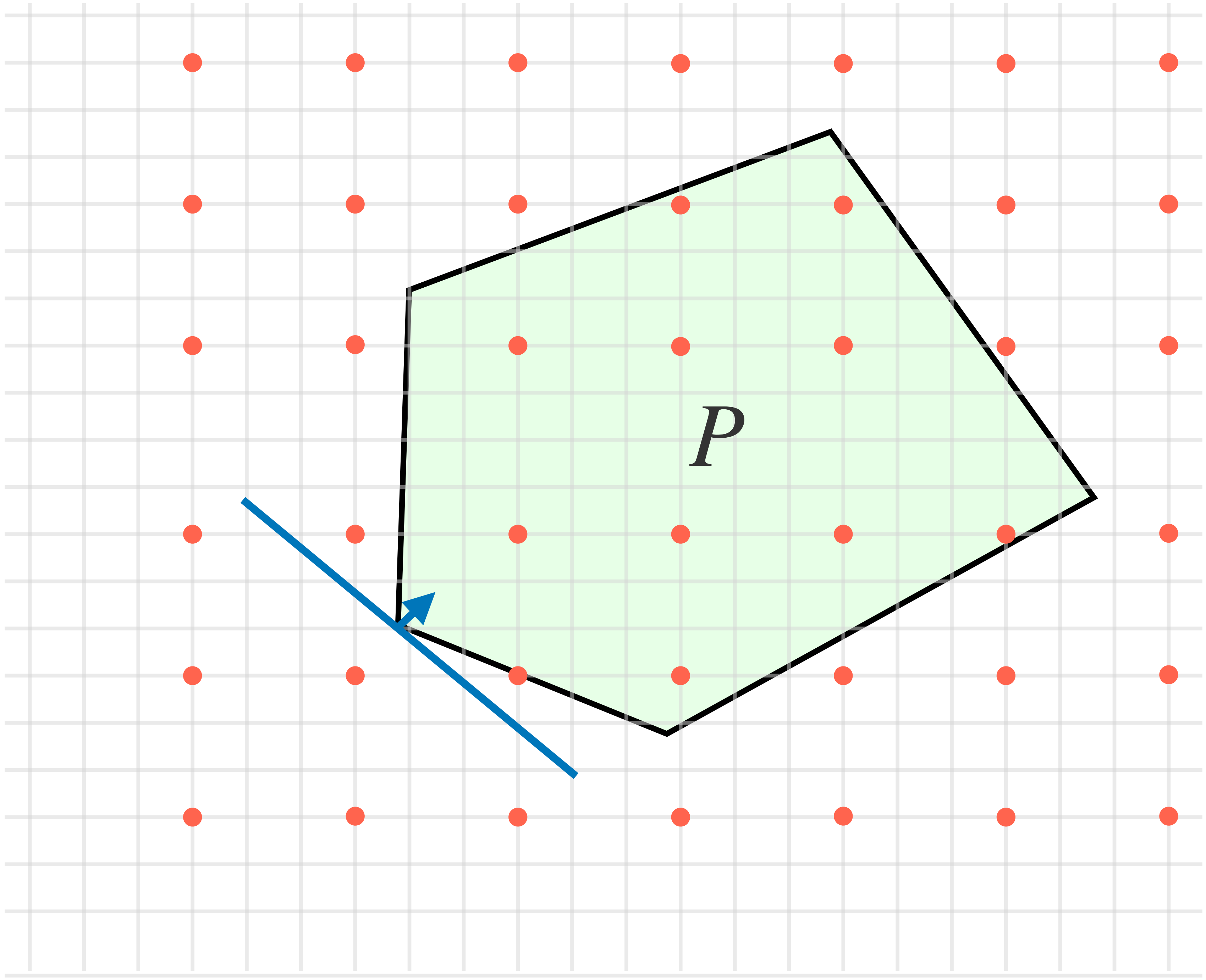
Integer Programming in Practice

Integer Hull of a polytope P is

$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$

- $\text{int}(P)$ is a polytope
- An LP solution to $\text{int}(P)$ is an ILP solution to P

$$LP(\text{int}(P), c) = ILP(P, c)$$



Modern IP-algorithms try to reduce $P \rightarrow \text{int}(P)$

How? — [Gomory63, Chvátal73]: Add cutting planes

If $ax \geq b$ is valid for P with $a \in \mathbb{Z}^n$ relatively prime and $b \in \mathbb{R}$ then $ax \geq \lceil b \rceil$ is a **CG-cut** for P

Integer Programming in Practice

Integer Hull of a polytope P is

$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$

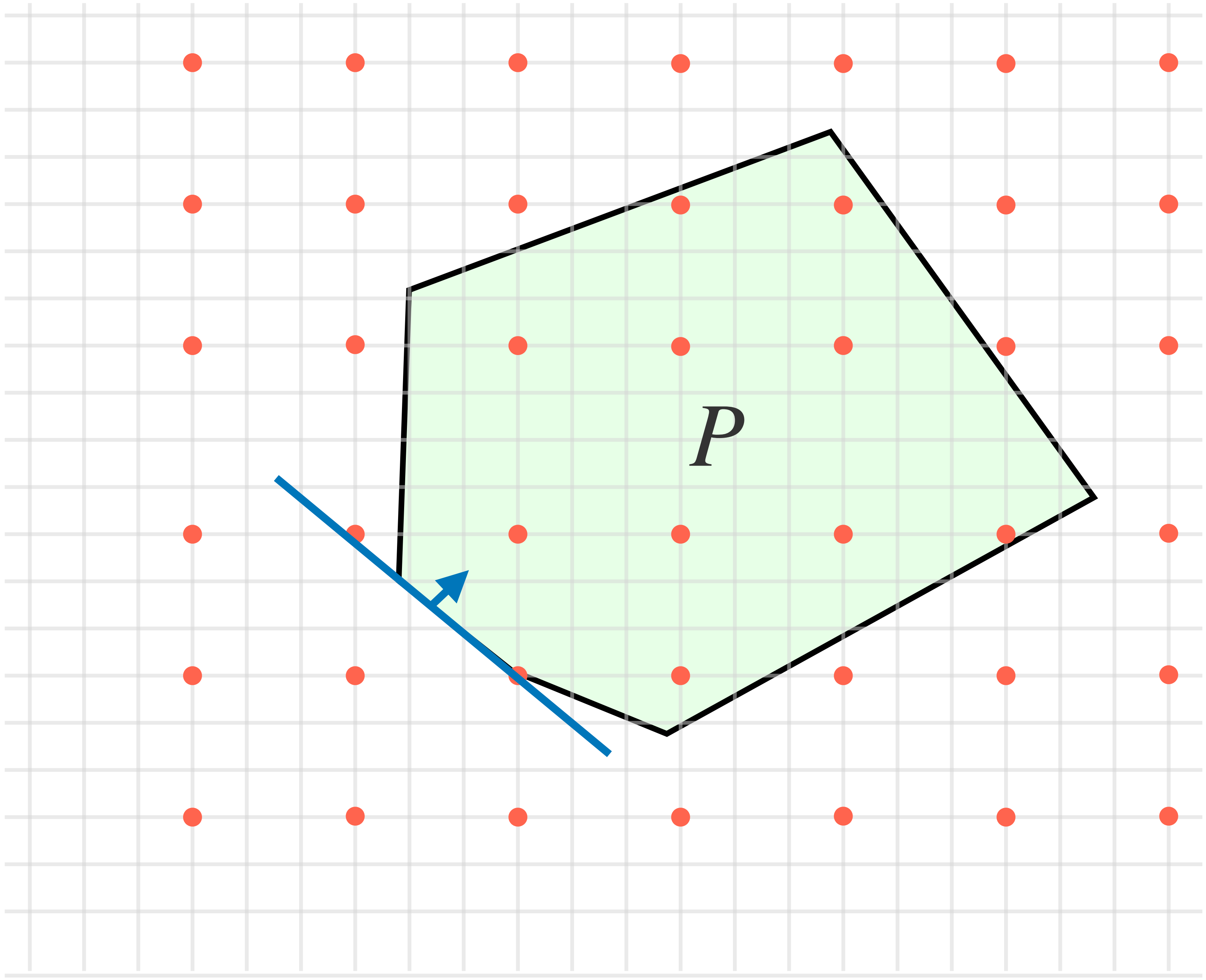
- $\text{int}(P)$ is a polytope
- An LP solution to $\text{int}(P)$ is an ILP solution to P

$$LP(\text{int}(P), c) = ILP(P, c)$$

Modern IP-algorithms try to reduce $P \to \text{int}(P)$

How? — [Gomory63, Chvátal73]: Add cutting planes

If $ax \geq b$ is valid for P with $a \in \mathbb{Z}^n$ relatively prime and $b \in \mathbb{R}$ then $ax \geq \lceil b \rceil$ is a **CG-cut** for P



Integer Programming in Practice

Integer Hull of a polytope P is

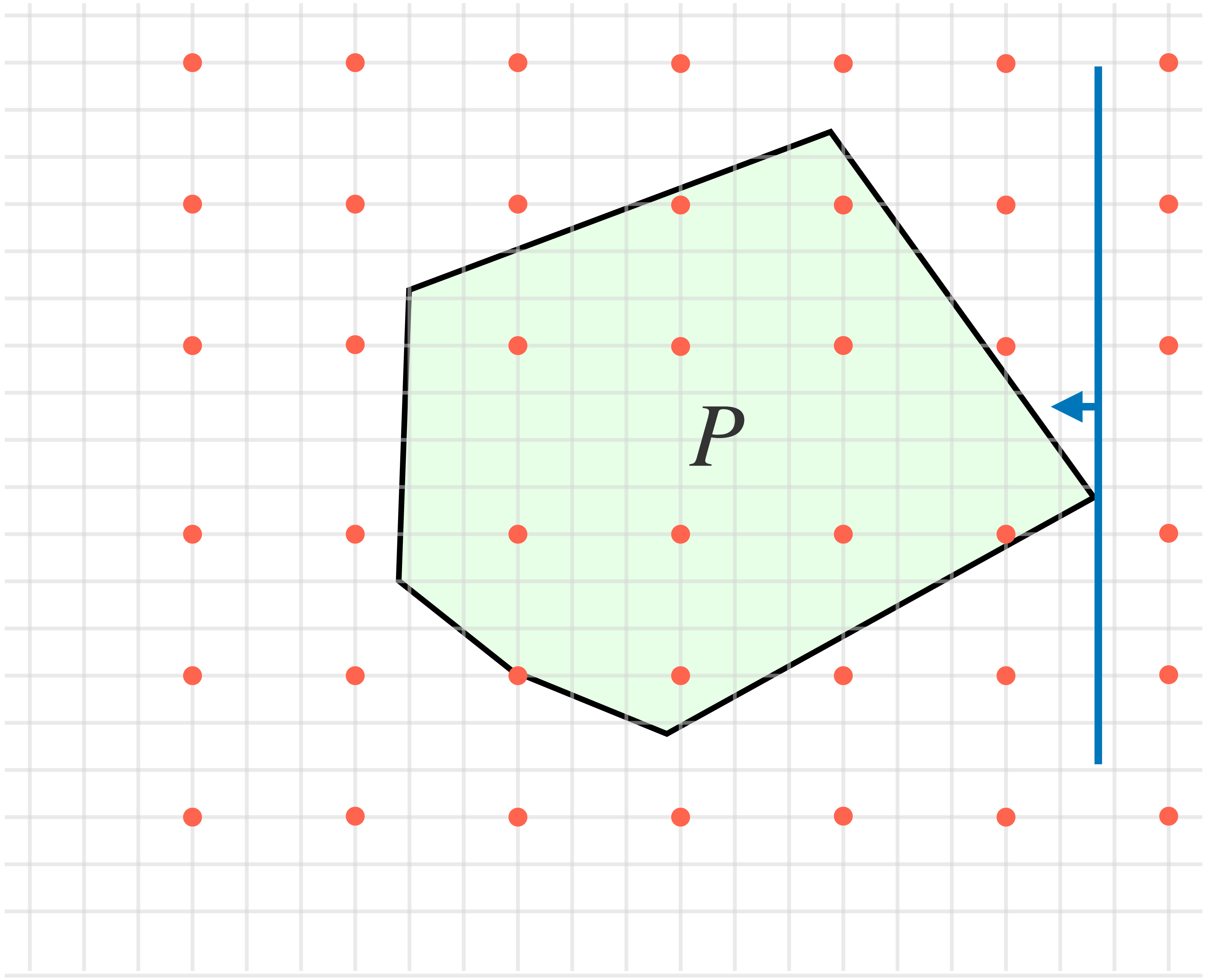
$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$

- $\text{int}(P)$ is a polytope
- An LP solution to $\text{int}(P)$ is an ILP solution to P

$$LP(\text{int}(P), c) = ILP(P, c)$$

Modern IP-algorithms try to reduce $P \to \text{int}(P)$

How? — [Gomory63, Chvátal73]: Add cutting planes



If $ax \geq b$ is valid for P with $a \in \mathbb{Z}^n$ relatively prime and $b \in \mathbb{R}$ then $ax \geq \lceil b \rceil$ is a **CG-cut** for P

Integer Programming in Practice

Integer Hull of a polytope P is

$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$

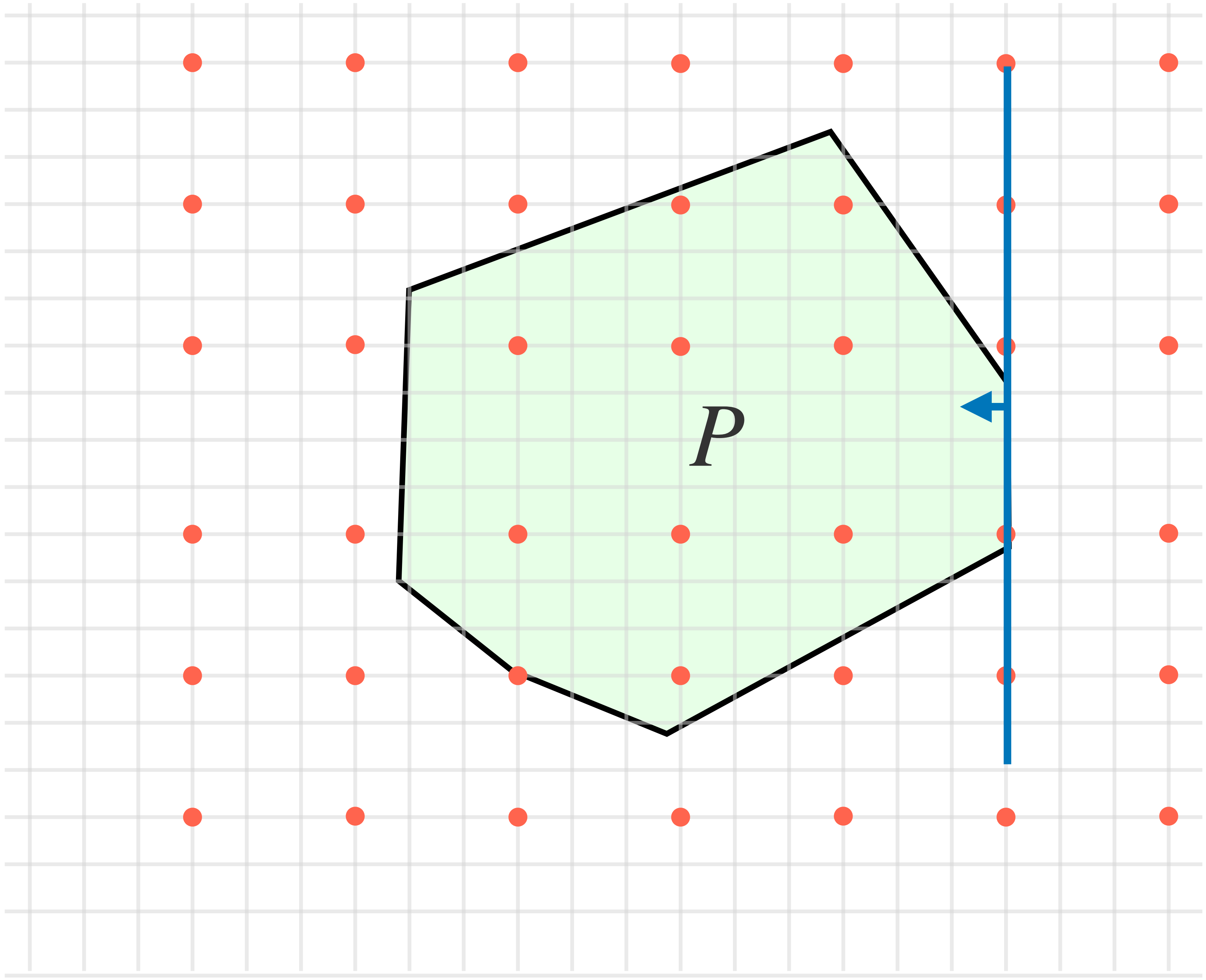
- $\text{int}(P)$ is a polytope
- An LP solution to $\text{int}(P)$ is an ILP solution to P

$$LP(\text{int}(P), c) = ILP(P, c)$$

Modern IP-algorithms try to reduce $P \rightarrow \text{int}(P)$

How? — [Gomory63, Chvátal73]: Add cutting planes

If $ax \geq b$ is valid for P with $a \in \mathbb{Z}^n$ relatively prime and $b \in \mathbb{R}$ then $ax \geq \lceil b \rceil$ is a **CG-cut** for P



Integer Programming in Practice

Integer Hull of a polytope P is

$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$

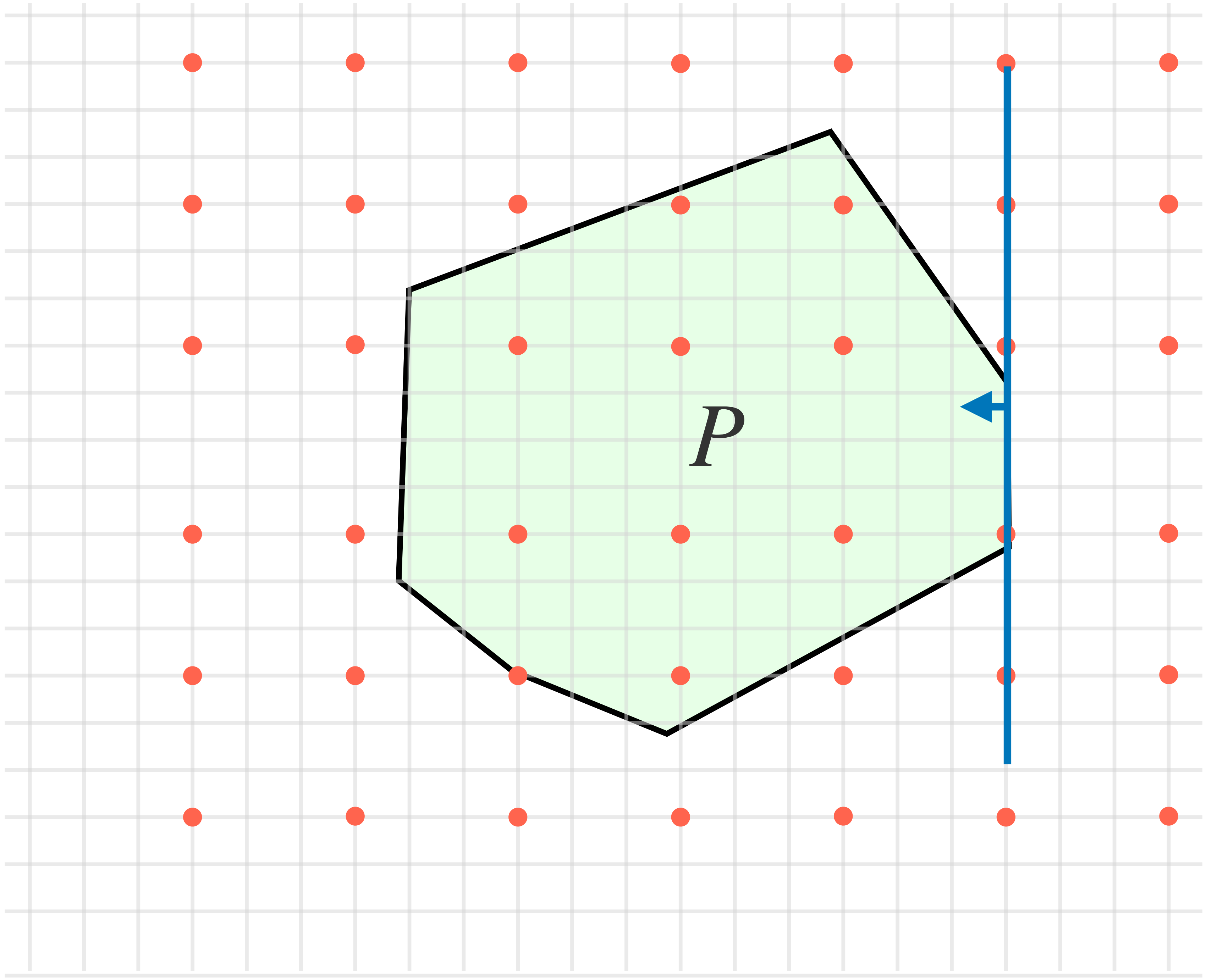
- $\text{int}(P)$ is a polytope
- An LP solution to $\text{int}(P)$ is an ILP solution to P

$$LP(\text{int}(P), c) = ILP(P, c)$$

Modern IP-algorithms try to reduce $P \to \text{int}(P)$

How? — [Gomory63, Chvátal73]: Add cutting planes

If $ax \geq b$ is valid for P with $a \in \mathbb{Z}^n$ relatively prime and $b \in \mathbb{R}$ then $ax \geq \lceil b \rceil$ is a **CG-cut** for P



- May take exponentially many CG-cuts [Pudlák93, BPR93]

Integer Programming in Practice

Integer Hull of a polytope P is

$$\text{int}(P) := \text{conv}(P \cap \mathbb{Z}^n)$$

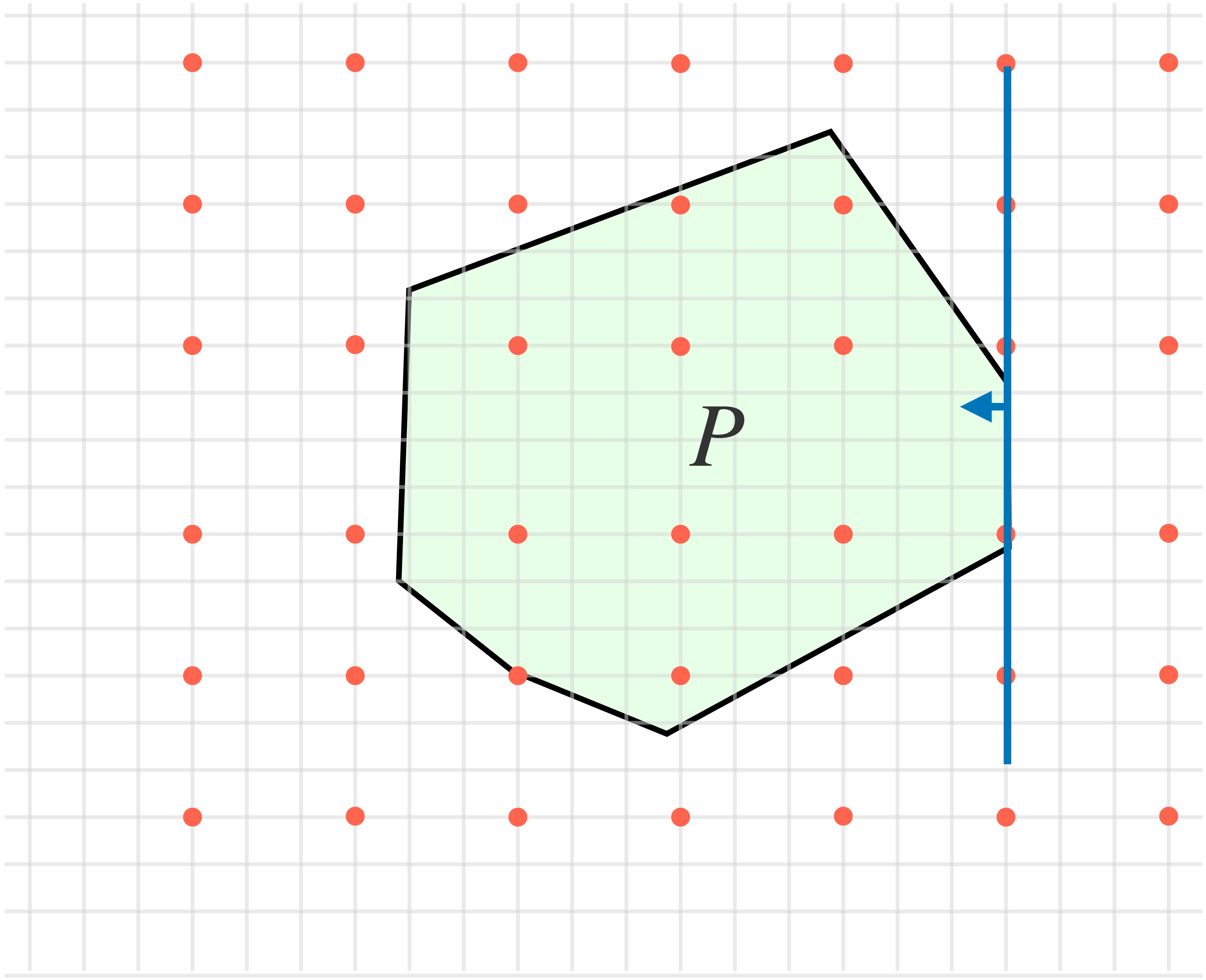
- $\text{int}(P)$ is a polytope
- An LP solution to $\text{int}(P)$ is an ILP solution to P

$$LP(\text{int}(P), c) = ILP(P, c)$$

Modern IP-algorithms try to reduce $P \to \text{int}(P)$

How? — [Gomory63, Chvátal73]: Add cutting planes

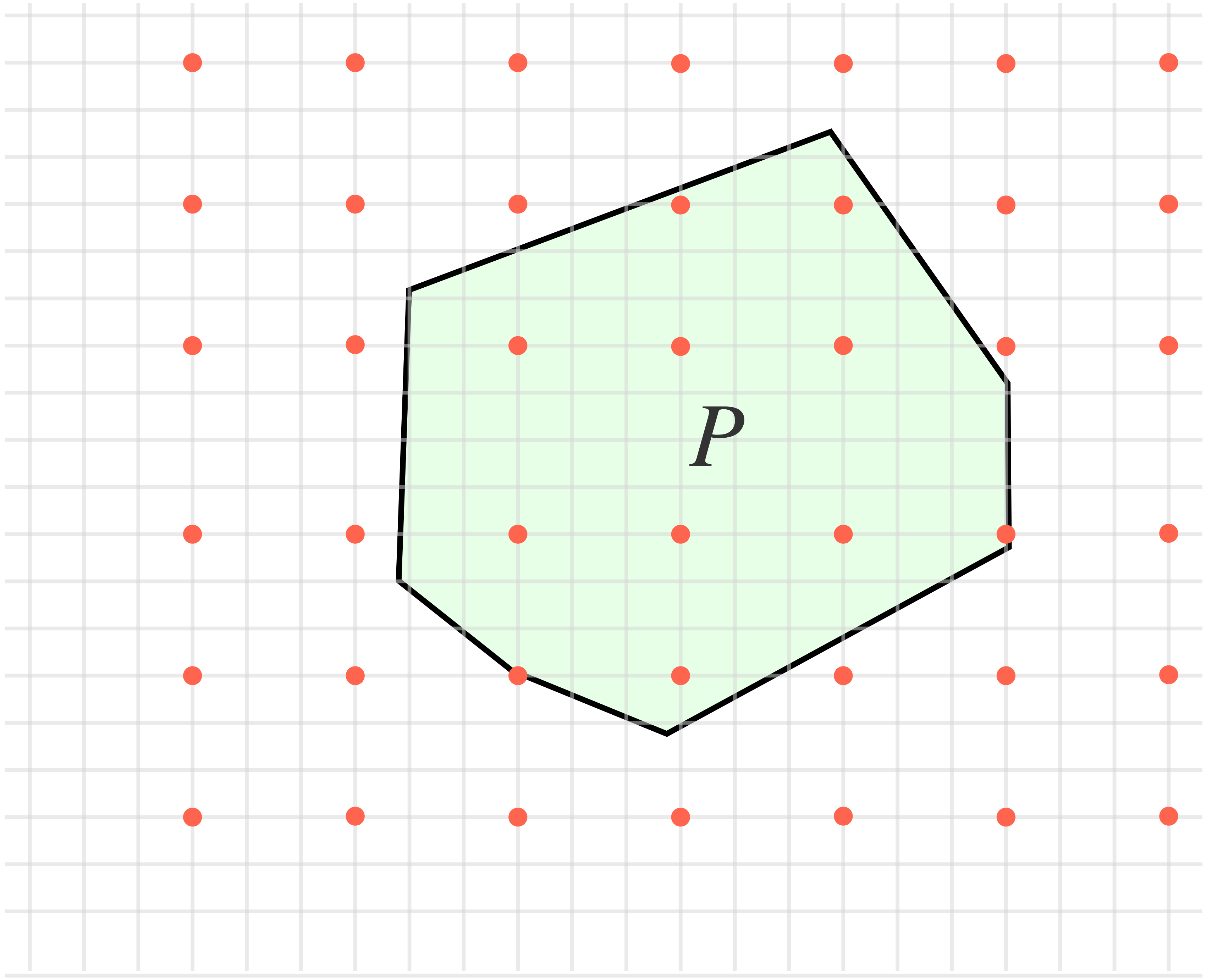
If $ax \geq b$ is valid for P with $a \in \mathbb{Z}^n$ relatively prime and $b \in \mathbb{R}$ then $ax \geq \lceil b \rceil$ is a **CG-cut** for P



- May take exponentially many CG-cuts [Pudlák93, BPR93]
- Numerically unstable to implement

Modern IP Algorithms

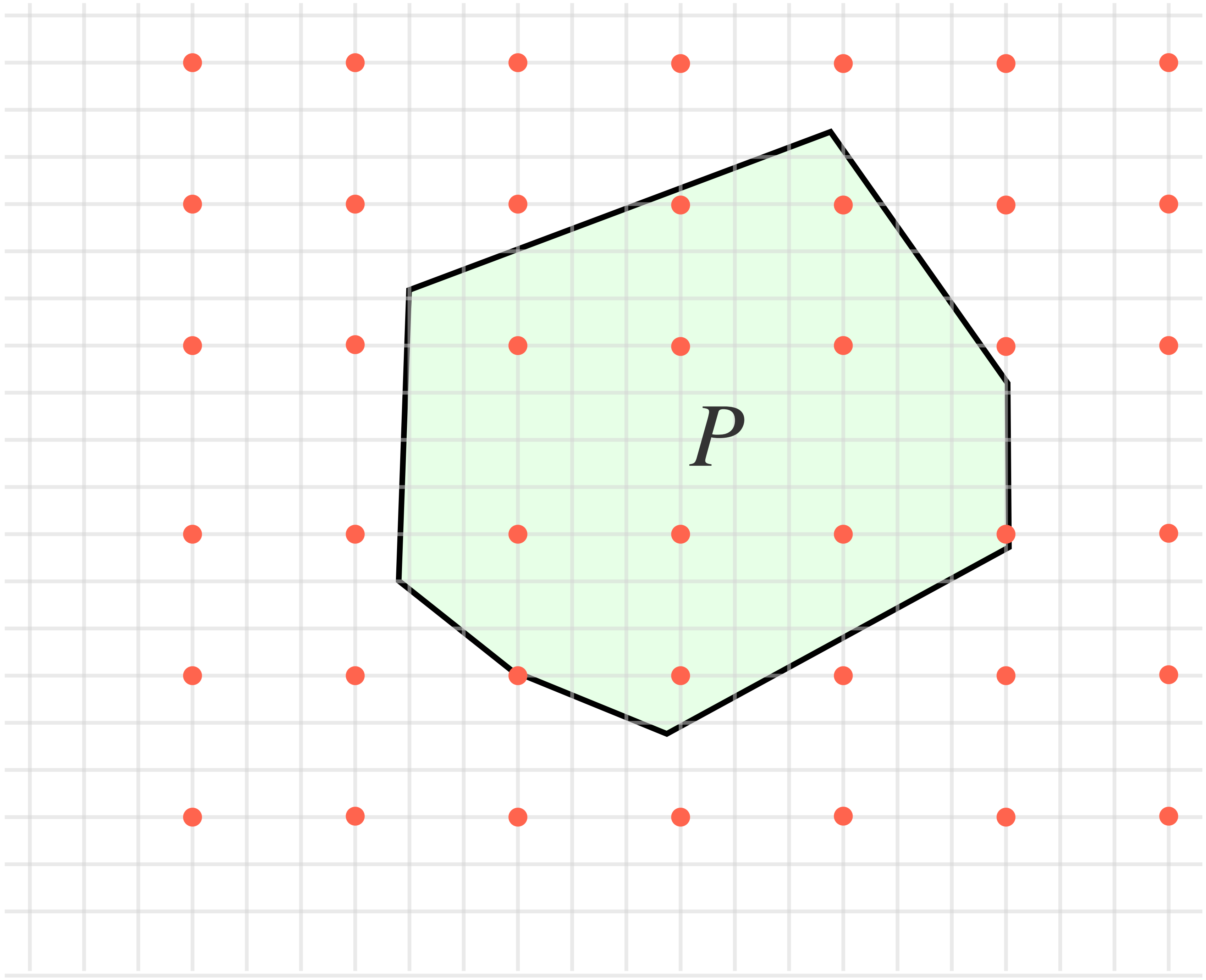
Combine cutting planes with branch-and-bound



Modern IP Algorithms

Combine cutting planes with branch-and-bound

Branch and Cut



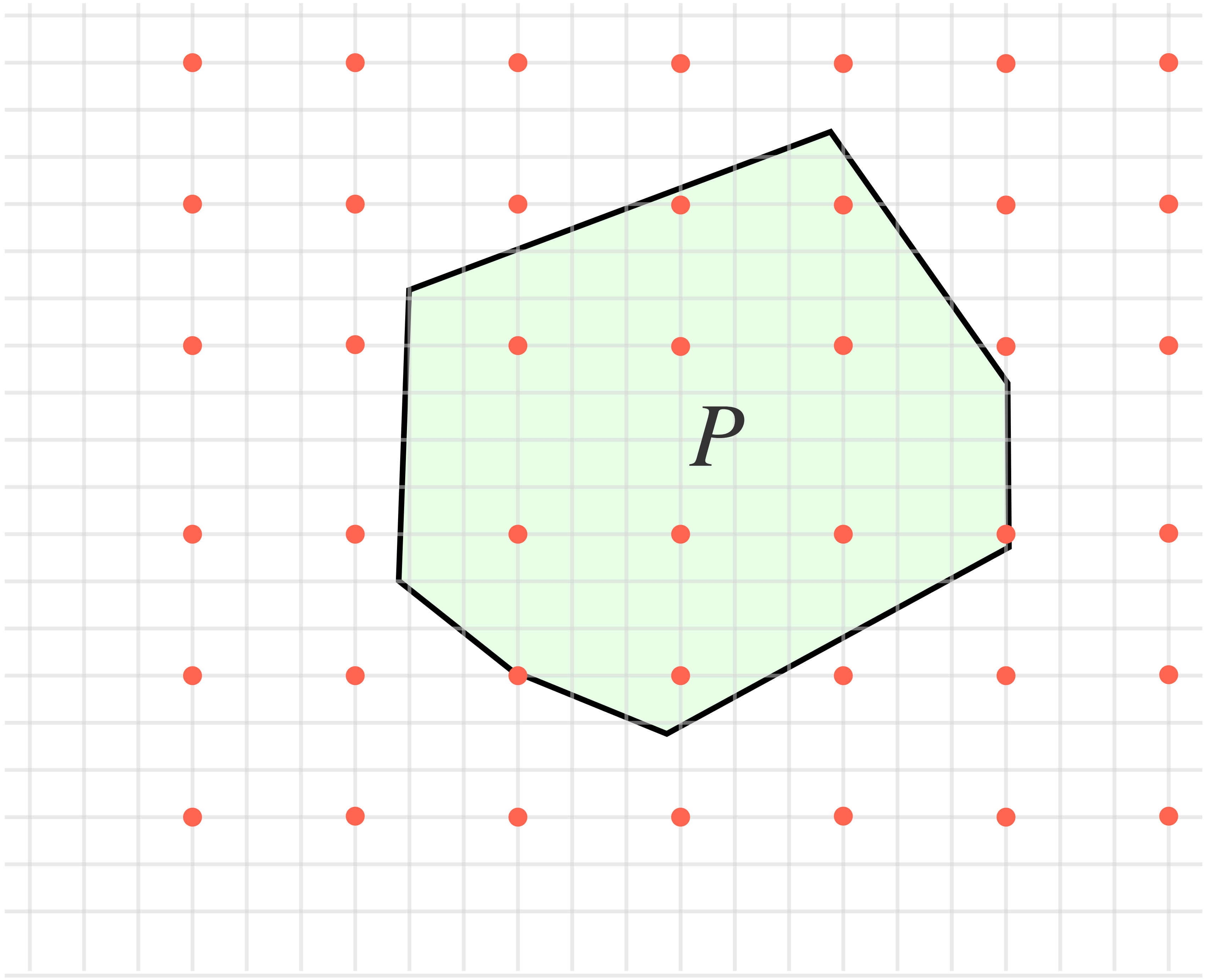
Modern IP Algorithms

Combine cutting planes with branch-and-bound

Branch and Cut

Alternate

- **Branch:** Choose $P_1, \ldots, P_k$ such that
$$P \cap \mathbb{Z}^n \subseteq \cup_i P_i.$$



Modern IP Algorithms

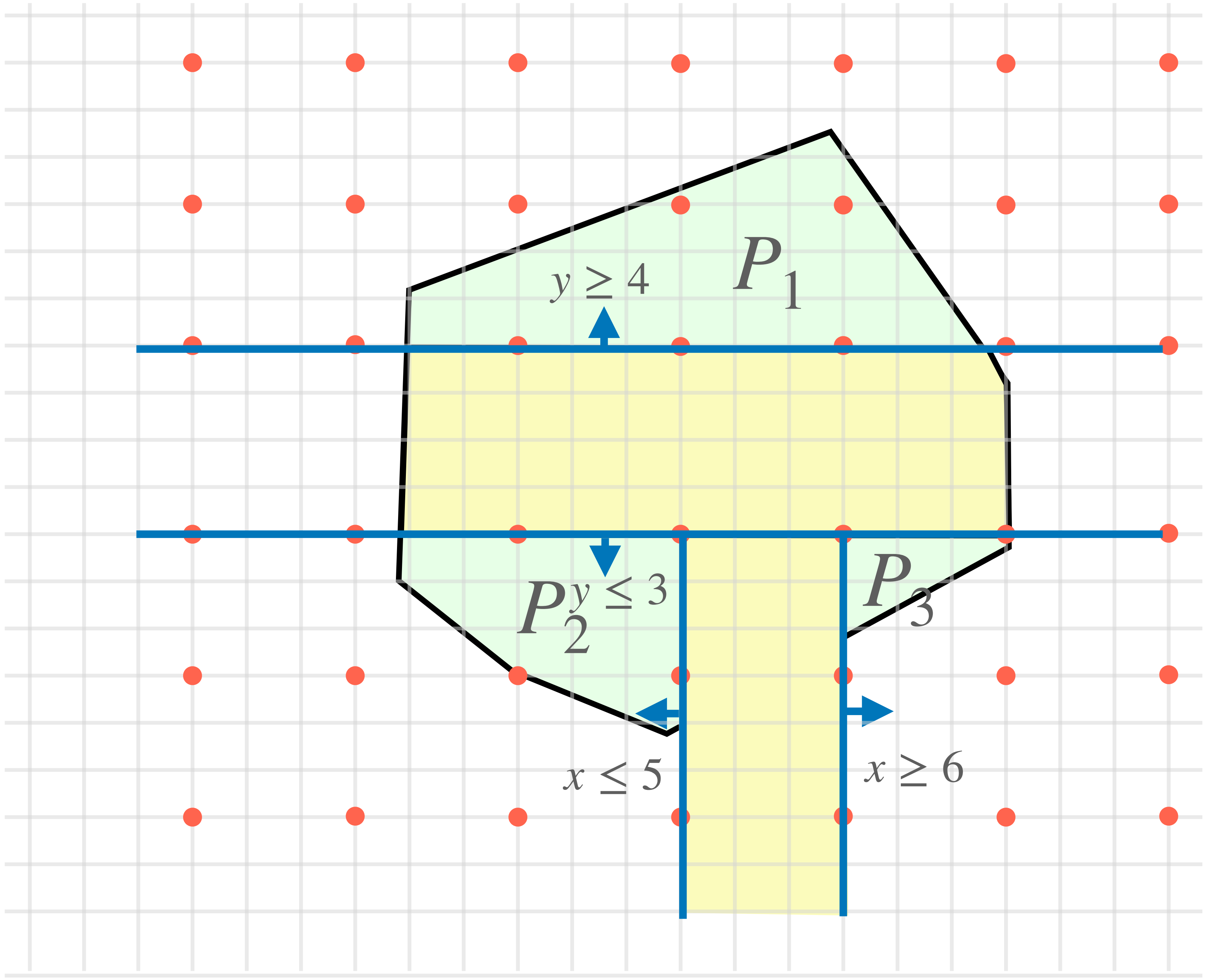
Combine cutting planes with branch-and-bound

Branch and Cut

Alternate

- **Branch:** Choose $P_1, \ldots, P_k$ such that

$$P \cap \mathbb{Z}^n \subseteq \cup_i P_i.$$



Modern IP Algorithms

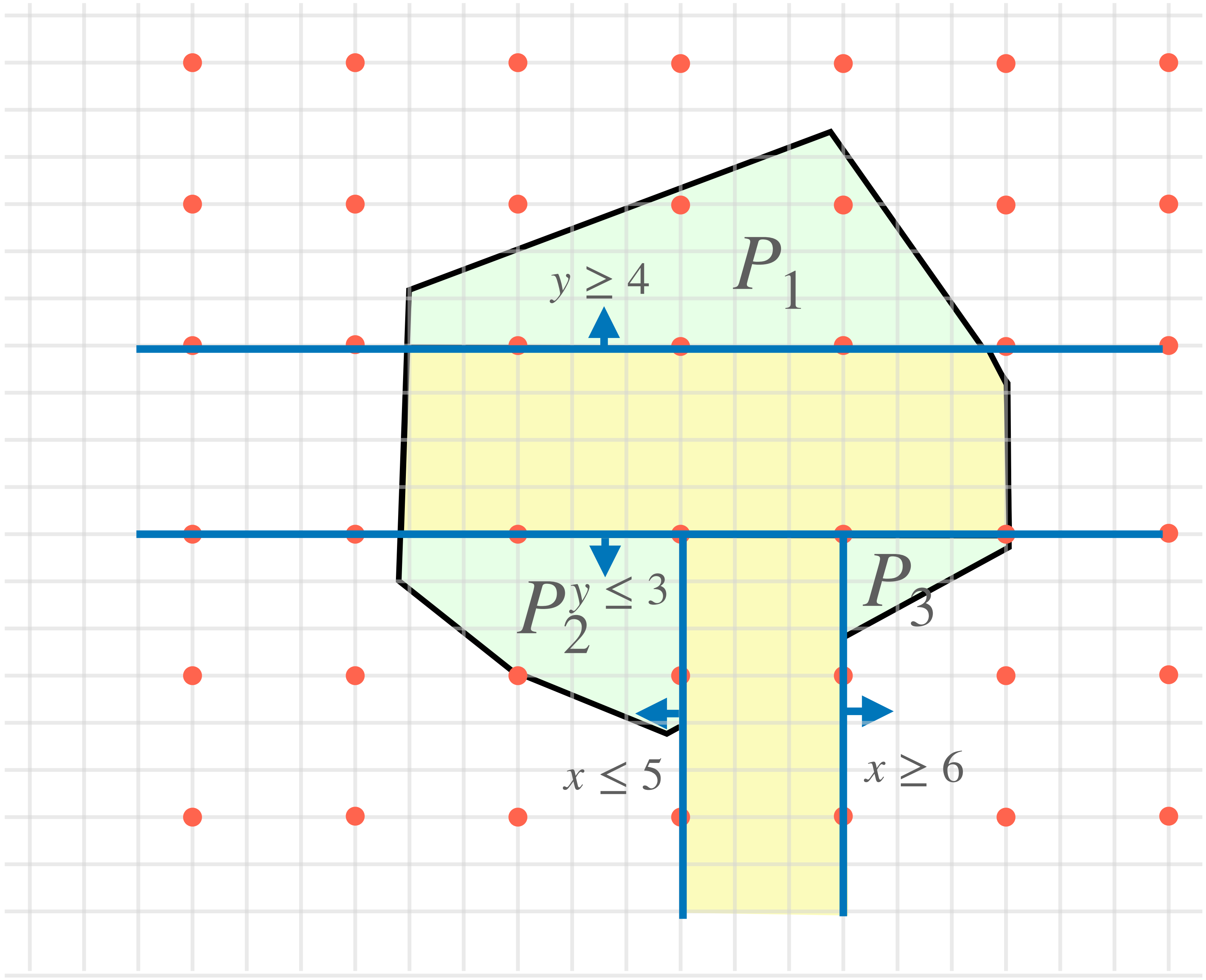
Combine cutting planes with branch-and-bound

Branch and Cut

Alternate

- **Branch:** Choose $P_1, \ldots, P_k$ such that

$$P \cap \mathbb{Z}^n \subseteq \cup_i P_i.$$



$P \cap \{y \geq 4\}, P \cap \{x \leq 3\} \cap \{x \leq 5\}, P \cap \{y \leq 3\} \cap \{x \geq 6\}$

All integer points preserved!

Modern IP Algorithms

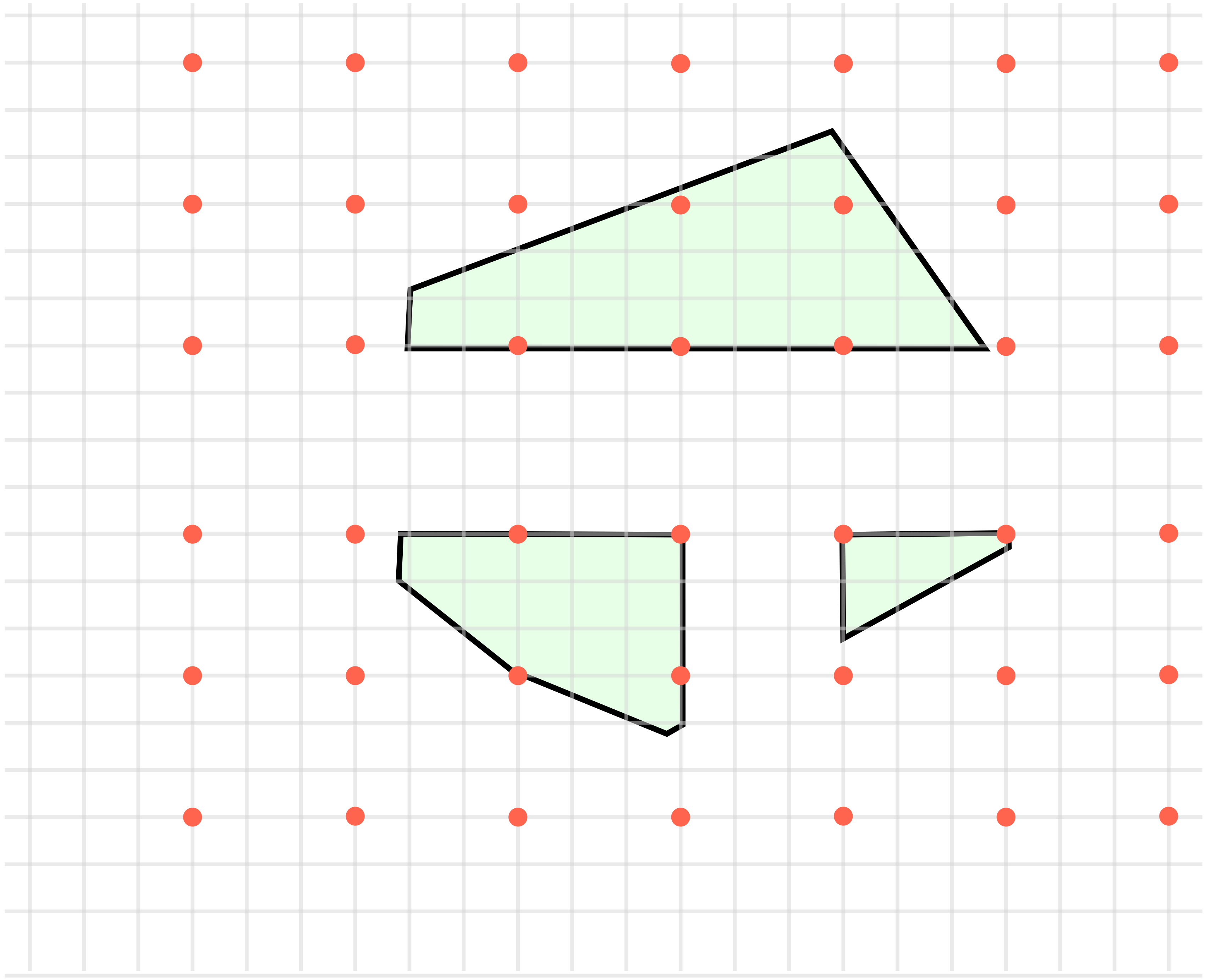
Combine cutting planes with branch-and-bound

Branch and Cut

Alternate

- **Branch:** Choose $P_1, \ldots, P_k$ such that

$$P \cap \mathbb{Z}^n \subseteq \cup_i P_i.$$



$P \cap \{y \geq 4\}, P \cap \{x \leq 3\} \cap \{x \leq 5\}, P \cap \{y \leq 3\} \cap \{x \geq 6\}$

All integer points preserved!

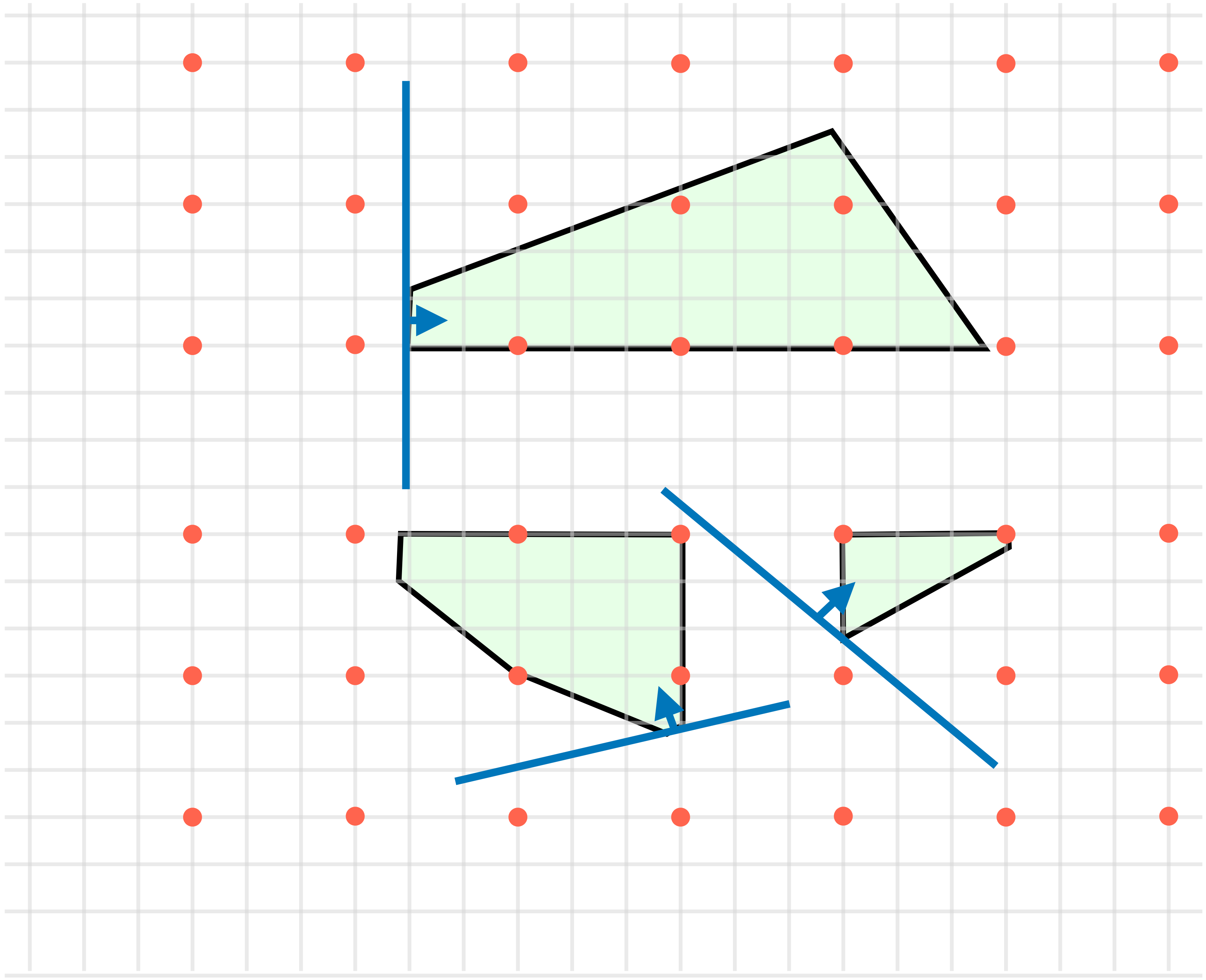
Modern IP Algorithms

Combine cutting planes with branch-and-bound

Branch and Cut

Alternate

- **Branch:** Choose $P_1, \ldots, P_k$ such that
$$P \cap \mathbb{Z}^n \subseteq \cup_i P_i.$$
- **Cut:** Refine $P_1, \ldots, P_k$ by adding additional cutting planes.



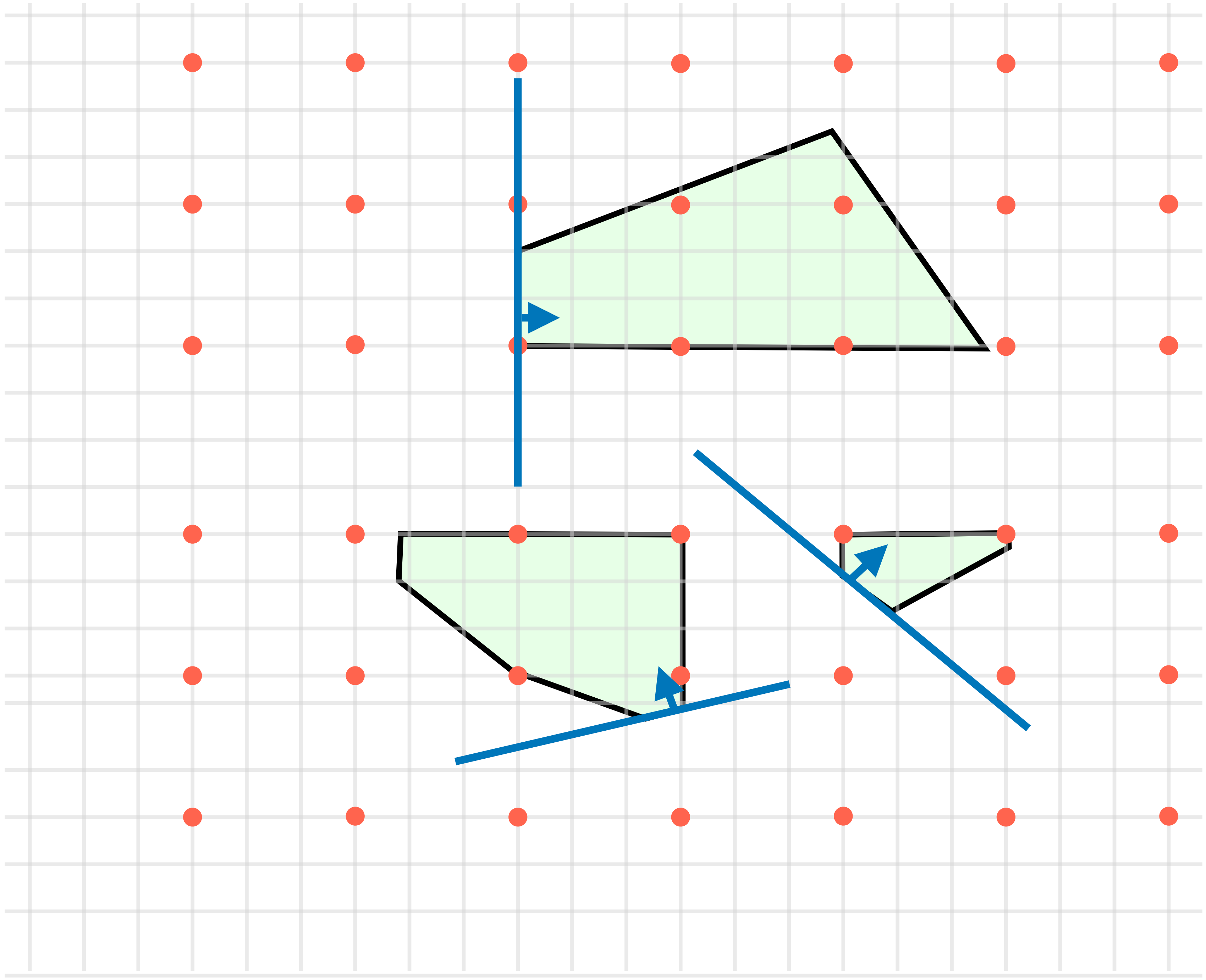
Modern IP Algorithms

Combine cutting planes with branch-and-bound

Branch and Cut

Alternate

- **Branch:** Choose $P_1, \ldots, P_k$ such that
$$P \cap \mathbb{Z}^n \subseteq \cup_i P_i.$$
- **Cut:** Refine $P_1, \ldots, P_k$ by adding additional cutting planes.



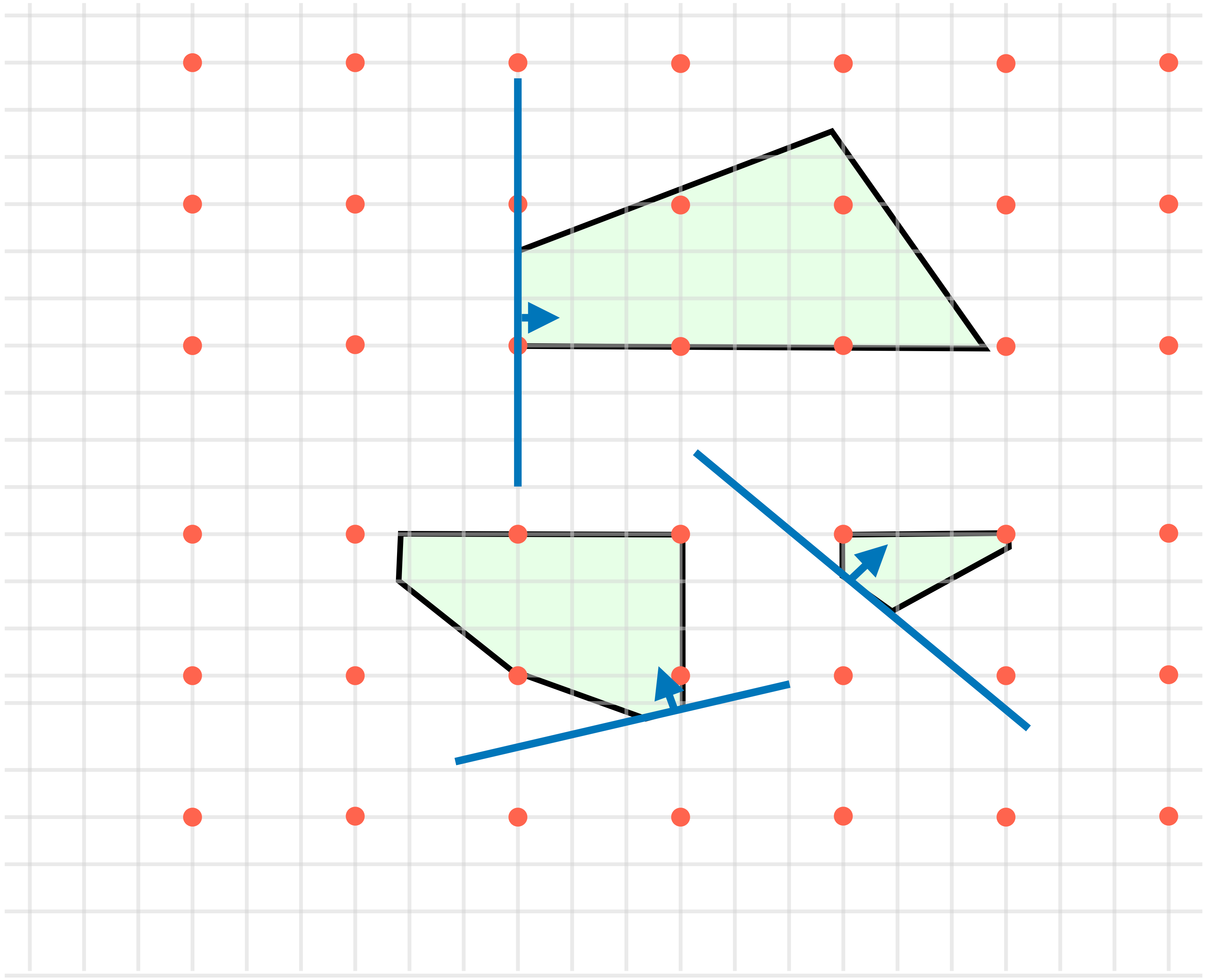
Modern IP Algorithms

Combine cutting planes with branch-and-bound

Branch and Cut

Alternate

- **Branch:** Choose $P_1, \ldots, P_k$ such that
$$P \cap \mathbb{Z}^n \subseteq \cup_i P_i.$$
- **Cut:** Refine $P_1, \ldots, P_k$ by adding additional cutting planes.



In practice branching is done by splitting on $ax \leq b$ and $ax \geq b + 1$ for $a \in \mathbb{Z}^n, b \in \mathbb{Z}$

Analyzing Modern IP Algorithms

Modern IP algorithms are a complicated mess of heuristics:

- Choosing how to branch,
- Choosing which cuts to add.

Makes analyzing these algorithms directly challenging!

Analyzing Modern IP Algorithms

Modern IP algorithms are a complicated mess of heuristics:

- Choosing how to branch,
- Choosing which cuts to add.

Makes analyzing these algorithms directly challenging!

Observation: If P contains no integer points then any correct IP algorithm running on P must identify this fact.

Analyzing Modern IP Algorithms

Modern IP algorithms are a complicated mess of heuristics:

- Choosing how to branch,
- Choosing which cuts to add.

Makes analyzing these algorithms directly challenging!

Observation: If P contains no integer points then any correct IP algorithm running on P must identify this fact.

$\implies$ the transcript of the algorithm's execution is a **proof** that $P \cap \mathbb{Z}^n = \emptyset$.

Analyzing Modern IP Algorithms

Modern IP algorithms are a complicated mess of heuristics:

- Choosing how to branch,
- Choosing which cuts to add.

Makes analyzing these algorithms directly challenging!

Observation: If P contains no integer points then any correct IP algorithm running on P must identify this fact.

$\implies$ the transcript of the algorithm's execution is a **proof** that $P \cap \mathbb{Z}^n = \varnothing$.

Even works in optimization! Algorithm has to "prove" that no better solution exists

Analyzing Modern IP Algorithms

Modern IP algorithms are a complicated mess of heuristics:

- Choosing how to branch,
- Choosing which cuts to add.

Makes analyzing these algorithms directly challenging!

Observation: If P contains no integer points then any correct IP algorithm running on P must identify this fact.

$\implies$ the transcript of the algorithm's execution is a **proof** that $P \cap \mathbb{Z}^n = \emptyset$.

Even works in optimization! Algorithm has to "prove" that no better solution exists

Instead of trying to understand an algorithm A directly, formalize the techniques used by the algorithms into a proof system S.

Analyzing Modern IP Algorithms

Modern IP algorithms are a complicated mess of heuristics:

- Choosing how to branch,
- Choosing which cuts to add.

Makes analyzing these algorithms directly challenging!

Observation: If P contains no integer points then any correct IP algorithm running on P must identify this fact.

$\implies$ the transcript of the algorithm's execution is a **proof** that $P \cap \mathbb{Z}^n = \emptyset$.

Even works in optimization! Algorithm has to “prove” that no better solution exists

Instead of trying to understand an algorithm A directly, formalize the techniques used by the algorithms into a proof system S.

$\rightarrow$ Lower bounds on the size of S-proofs imply runtime lower bounds for A

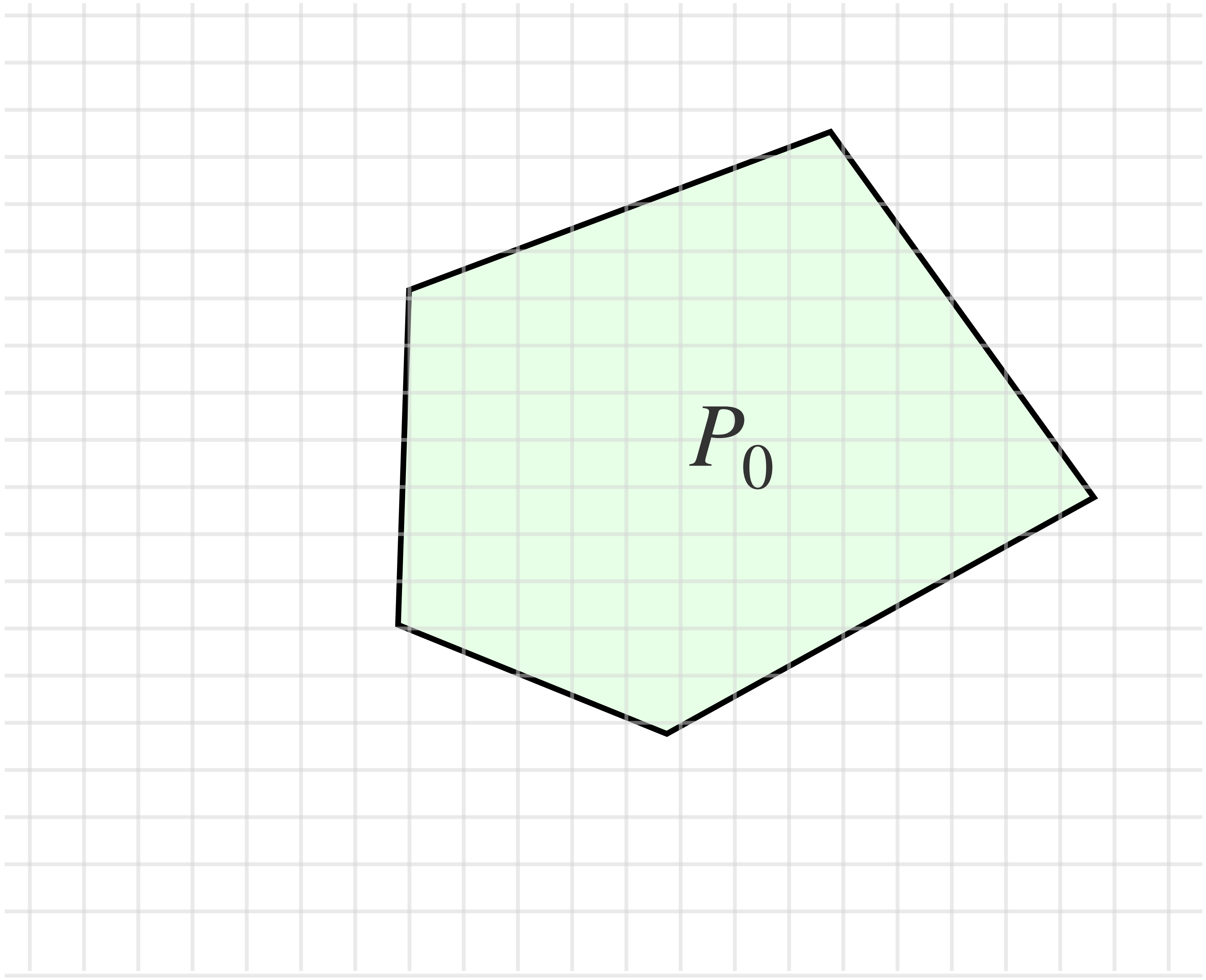
Cutting Planes

A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \emptyset)$$

where P_i is obtained by a CG-cut from P_{i-1}.

The size of the proof is s.



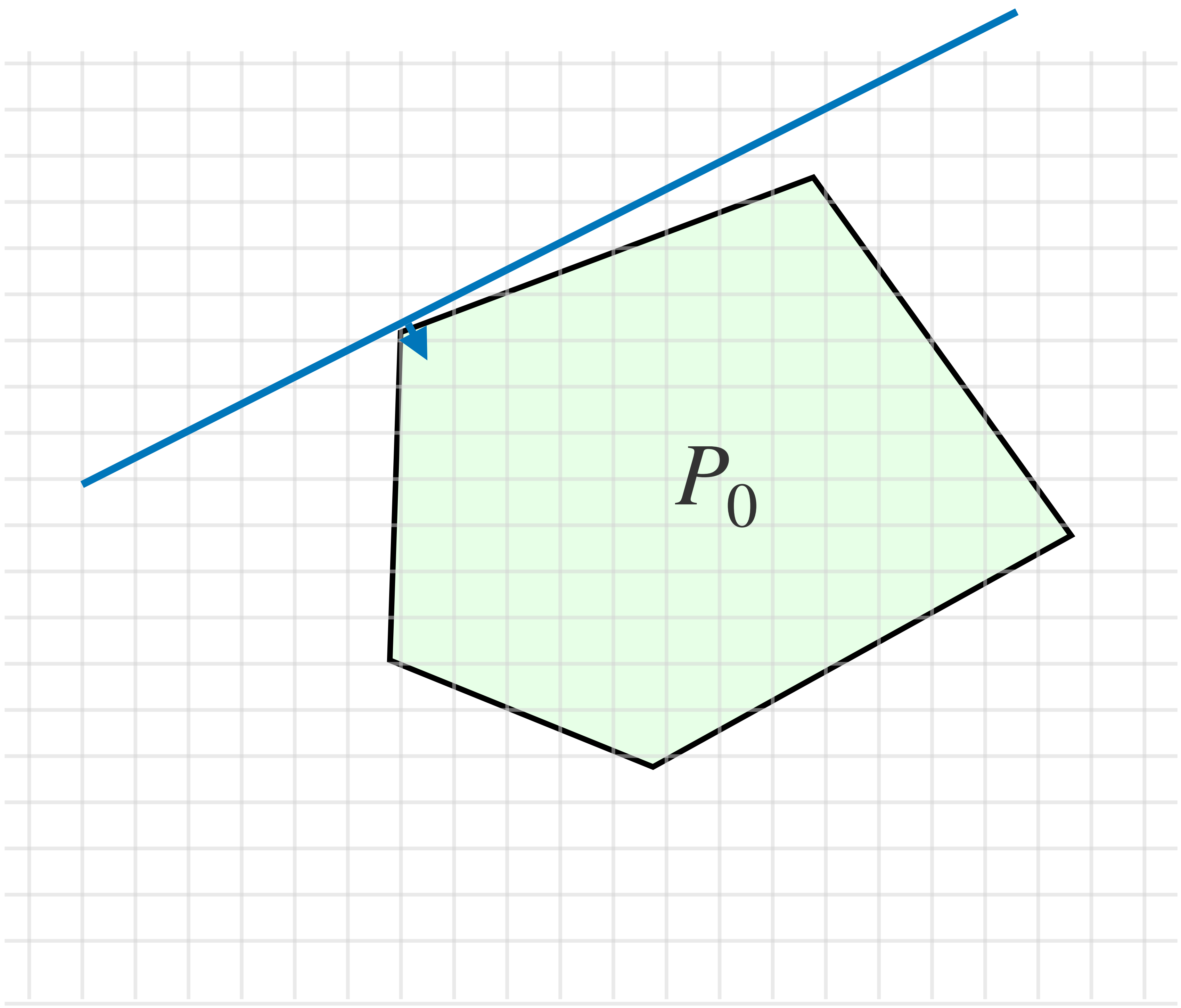
Cutting Planes

A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \emptyset)$$

where P_i is obtained by a CG-cut from P_{i-1}.

The size of the proof is s.



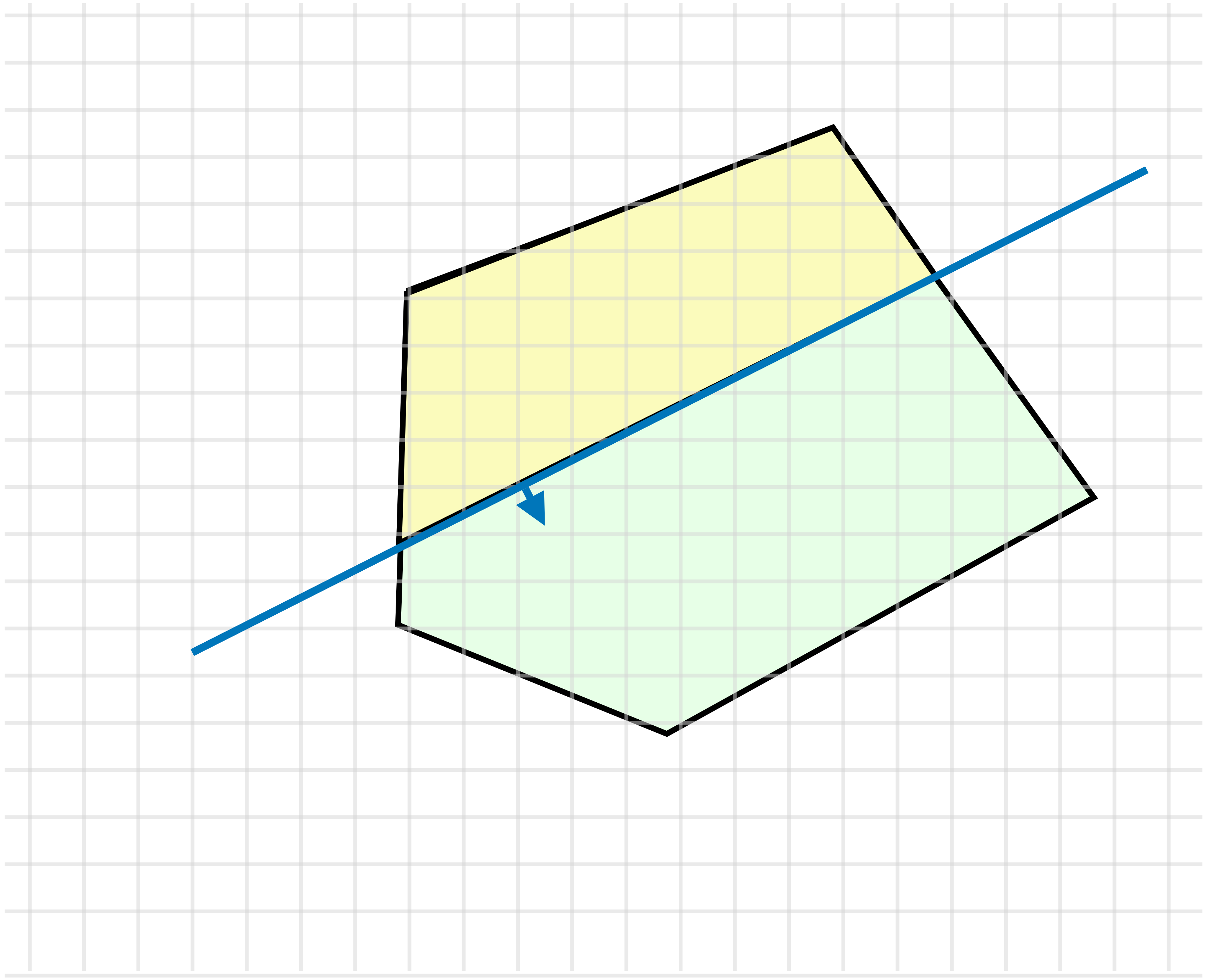
Cutting Planes

A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \emptyset)$$

where P_i is obtained by a CG-cut from P_{i-1}.

The size of the proof is s.



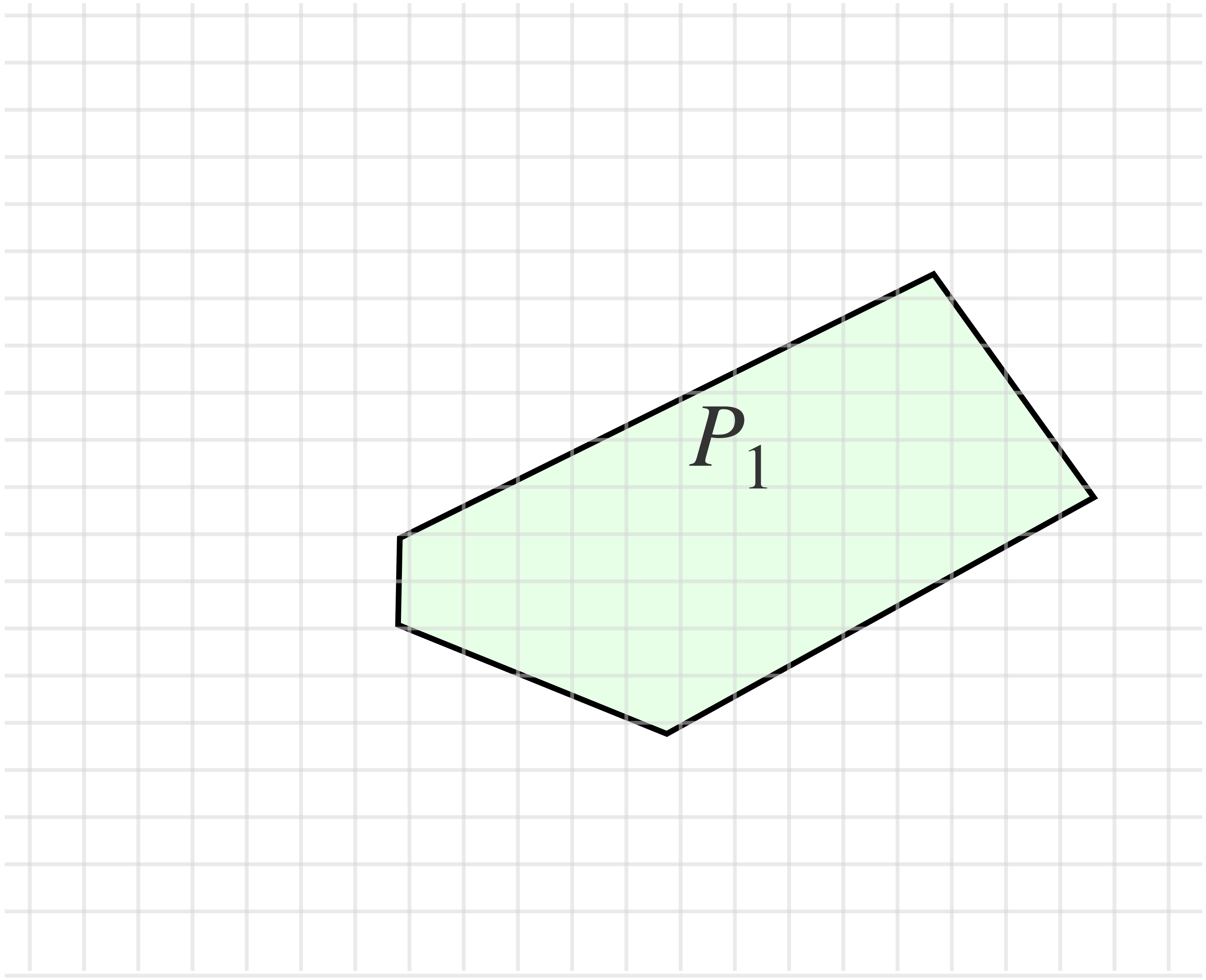
Cutting Planes

A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \varnothing$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \varnothing)$$

where P_i is obtained by a CG-cut from P_{i-1}.

The size of the proof is s.



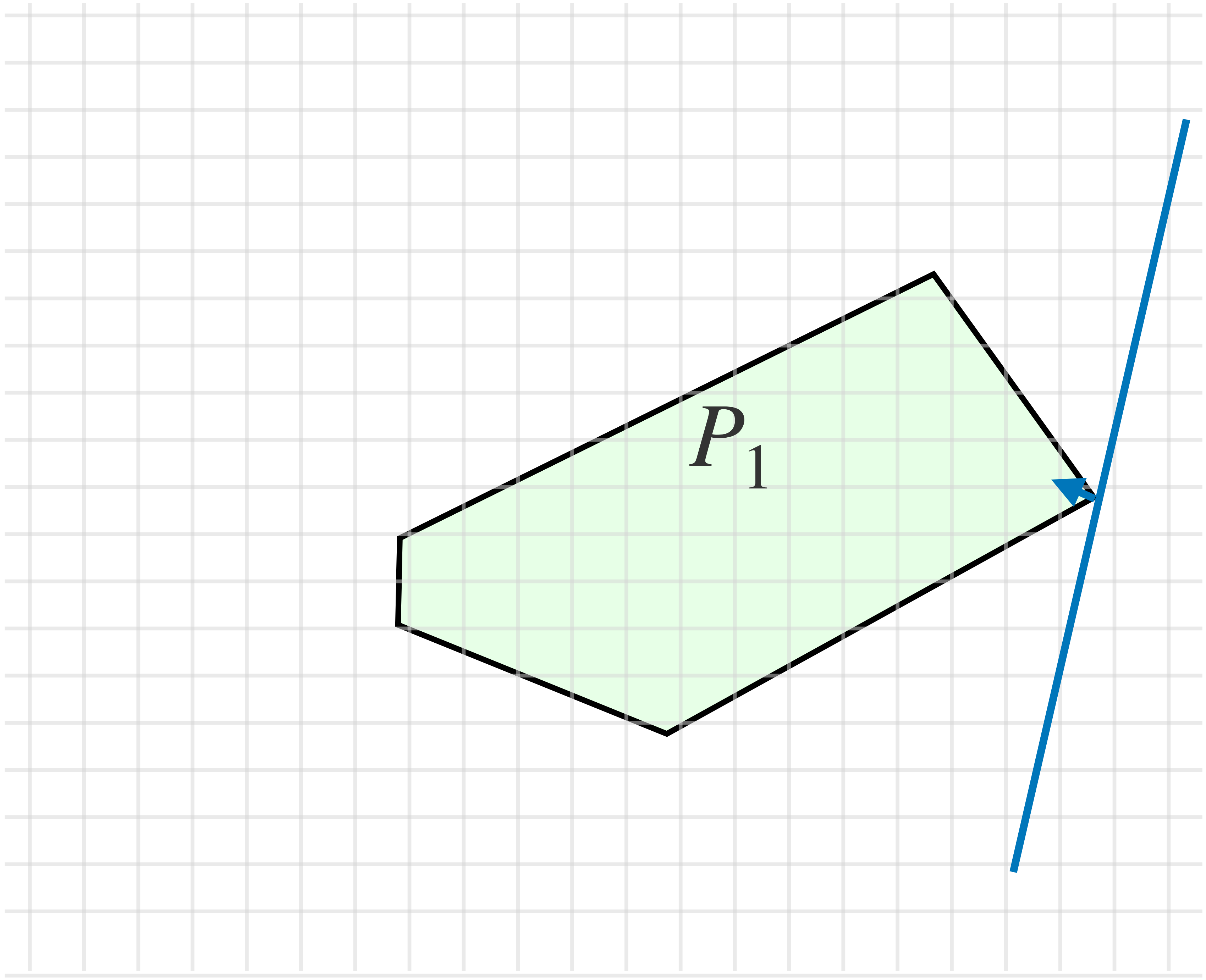
Cutting Planes

A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \emptyset)$$

where P_i is obtained by a CG-cut from P_{i-1}.

The size of the proof is s.



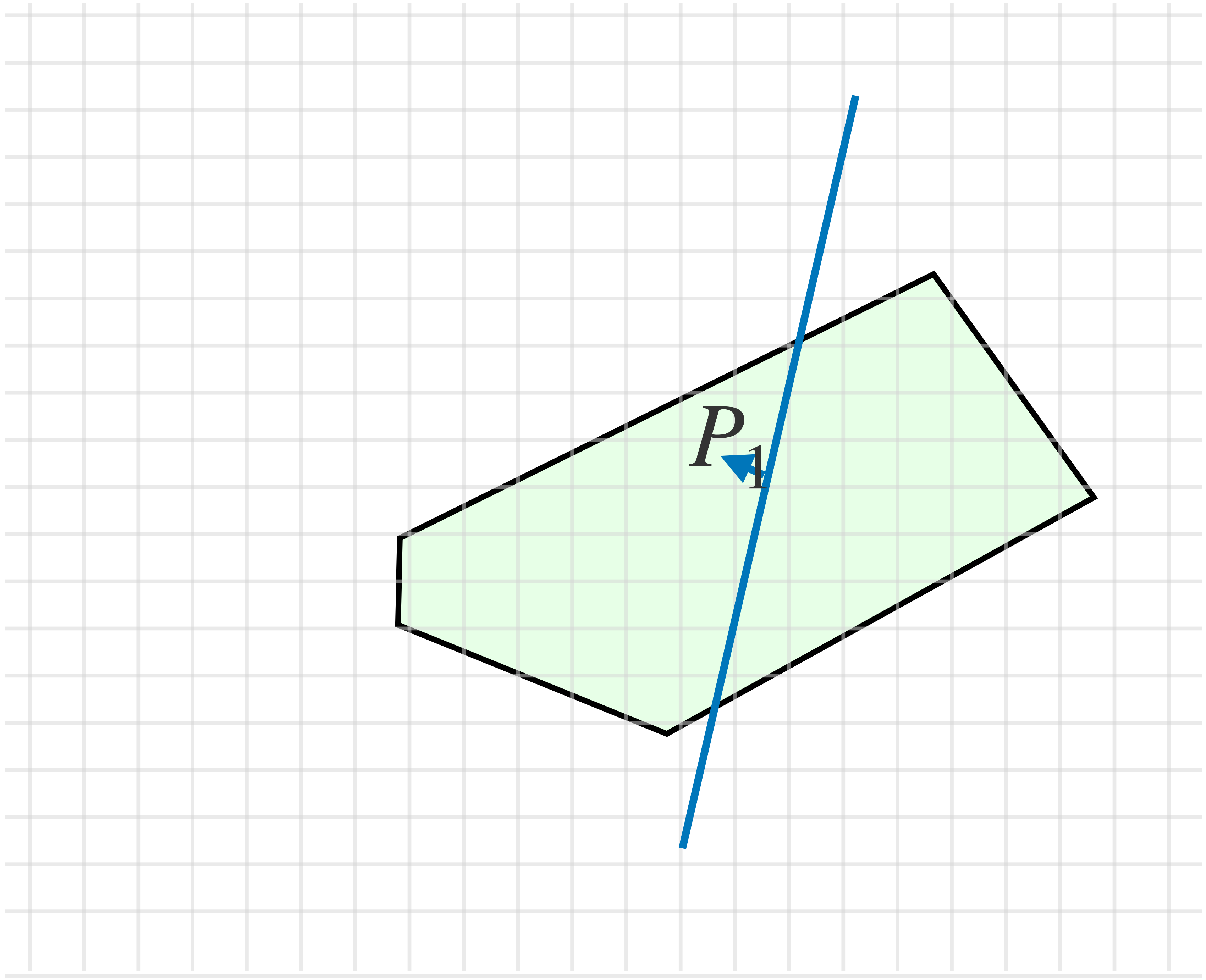
Cutting Planes

A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \varnothing$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \varnothing)$$

where P_i is obtained by a CG-cut from P_{i-1}.

The size of the proof is s.



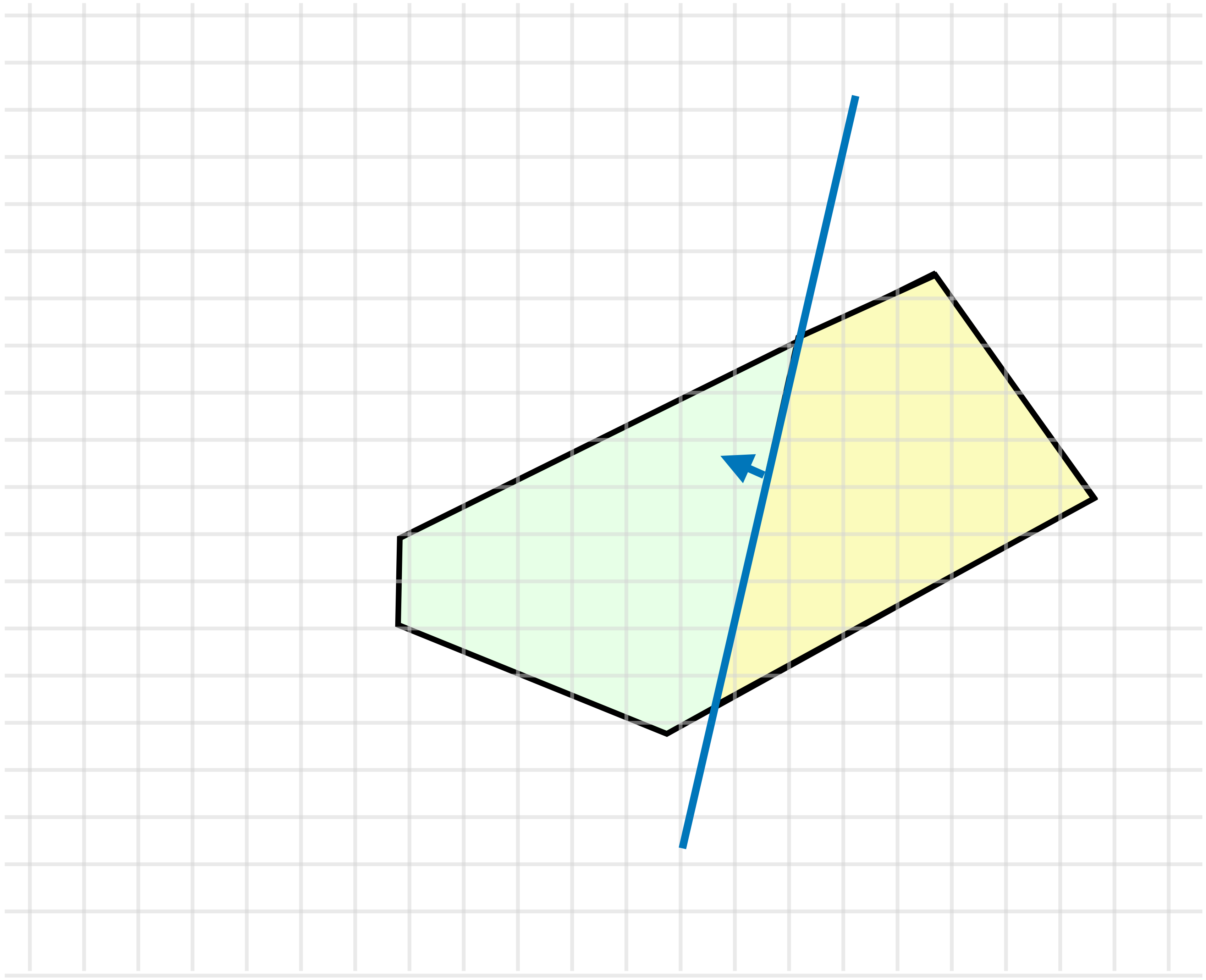
Cutting Planes

A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \emptyset)$$

where P_i is obtained by a CG-cut from P_{i-1}.

The size of the proof is s.



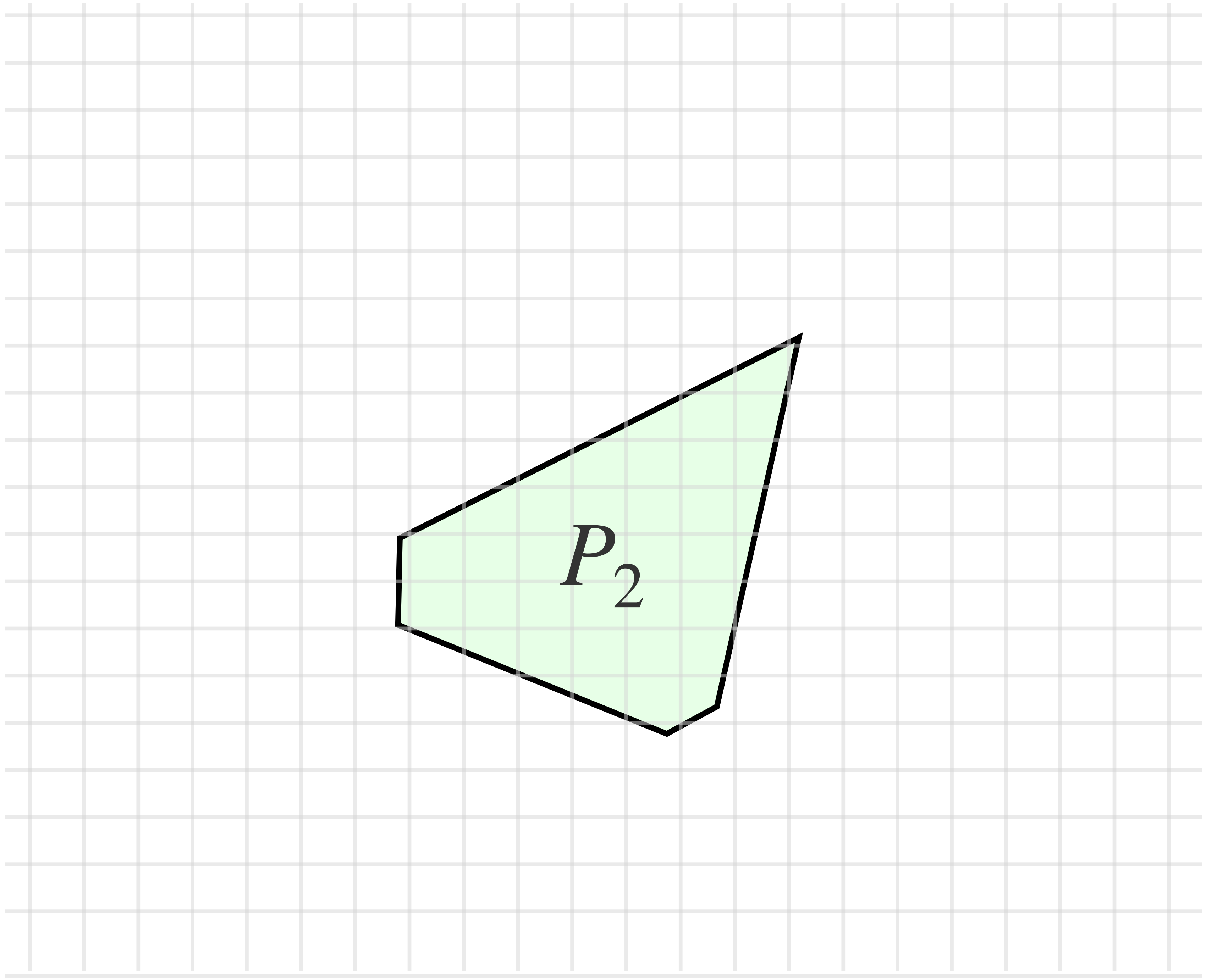
Cutting Planes

A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \emptyset)$$

where P_i is obtained by a CG-cut from P_{i-1}.

The size of the proof is s.



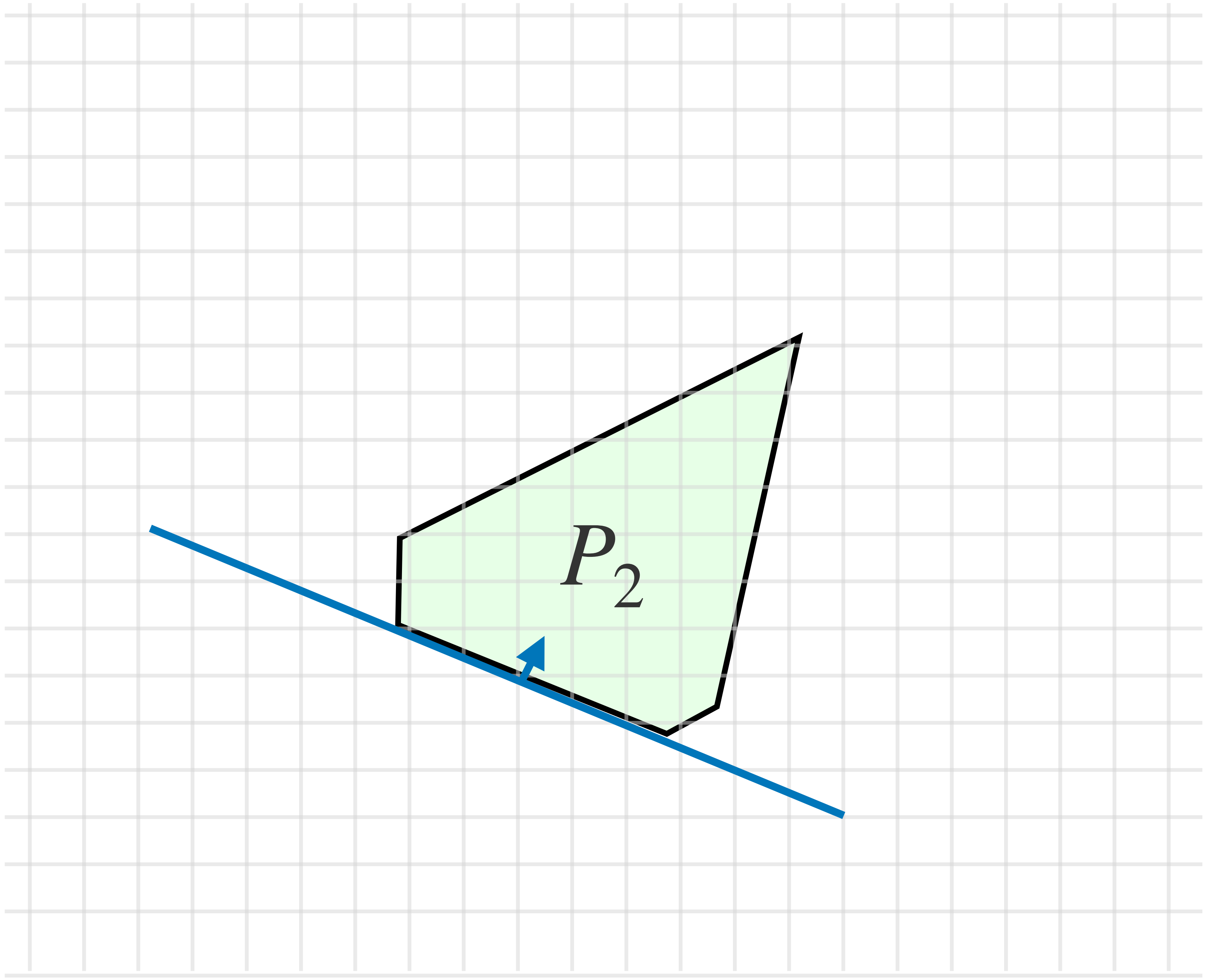
Cutting Planes

A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \varnothing$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \varnothing)$$

where P_i is obtained by a CG-cut from P_{i-1}.

The size of the proof is s.



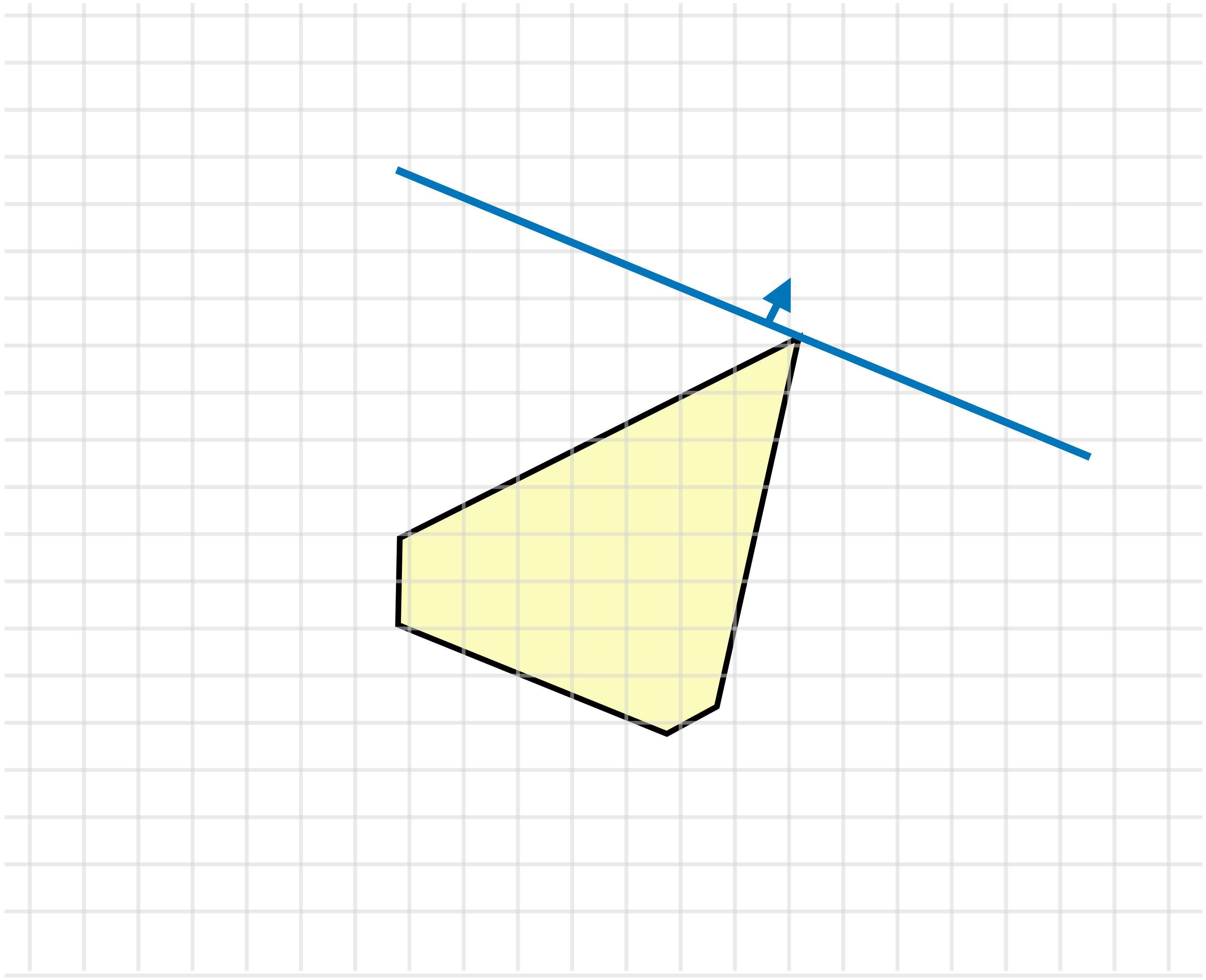
Cutting Planes

A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \emptyset)$$

where P_i is obtained by a CG-cut from P_{i-1}.

The size of the proof is s.



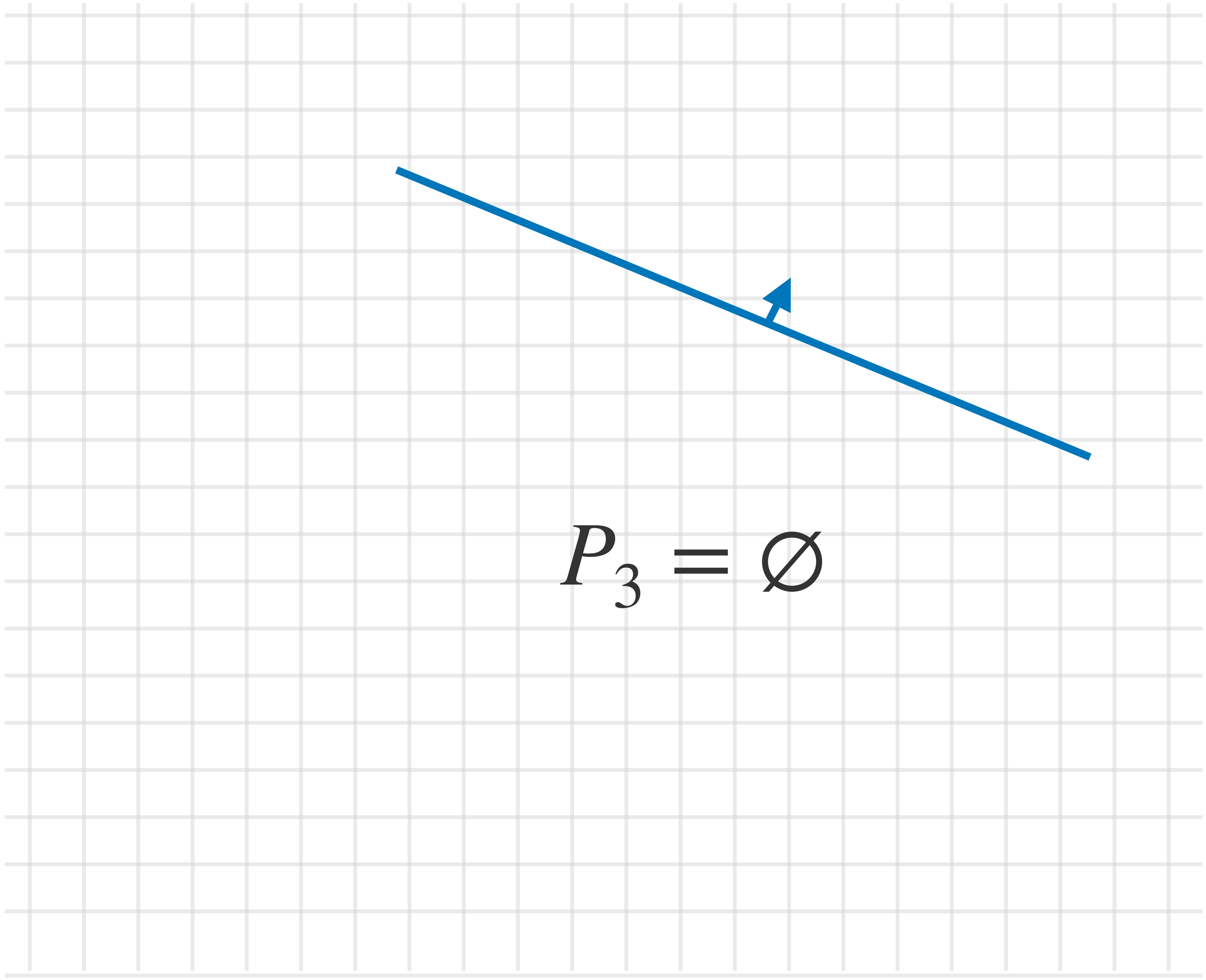
Cutting Planes

A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \varnothing$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \varnothing)$$

where P_i is obtained by a CG-cut from P_{i-1}.

The size of the proof is s.



Cutting Planes

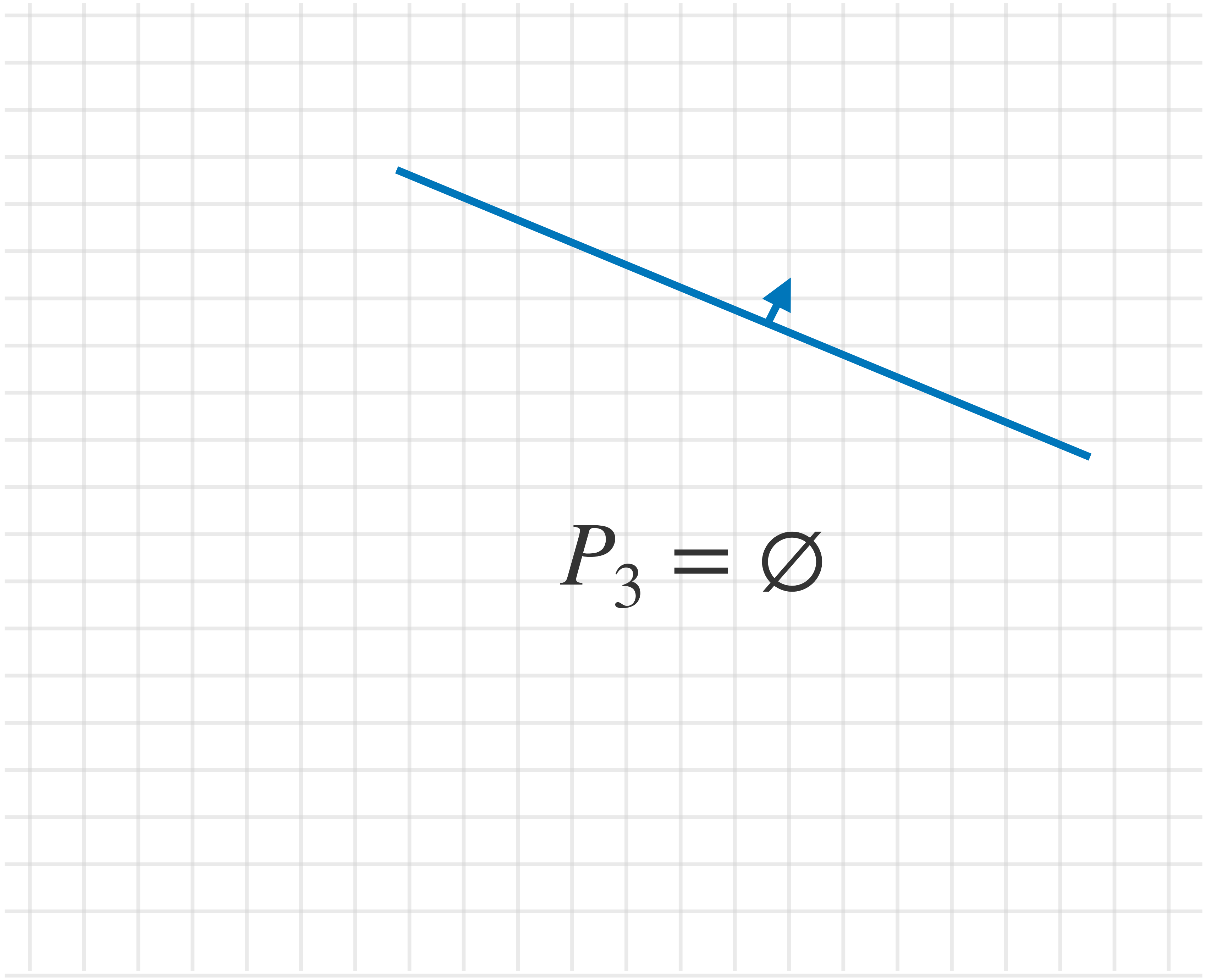
A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \emptyset)$$

where P_i is obtained by a CG-cut from P_{i-1}.

The size of the proof is s.

- Introduced in [Chvátal73].



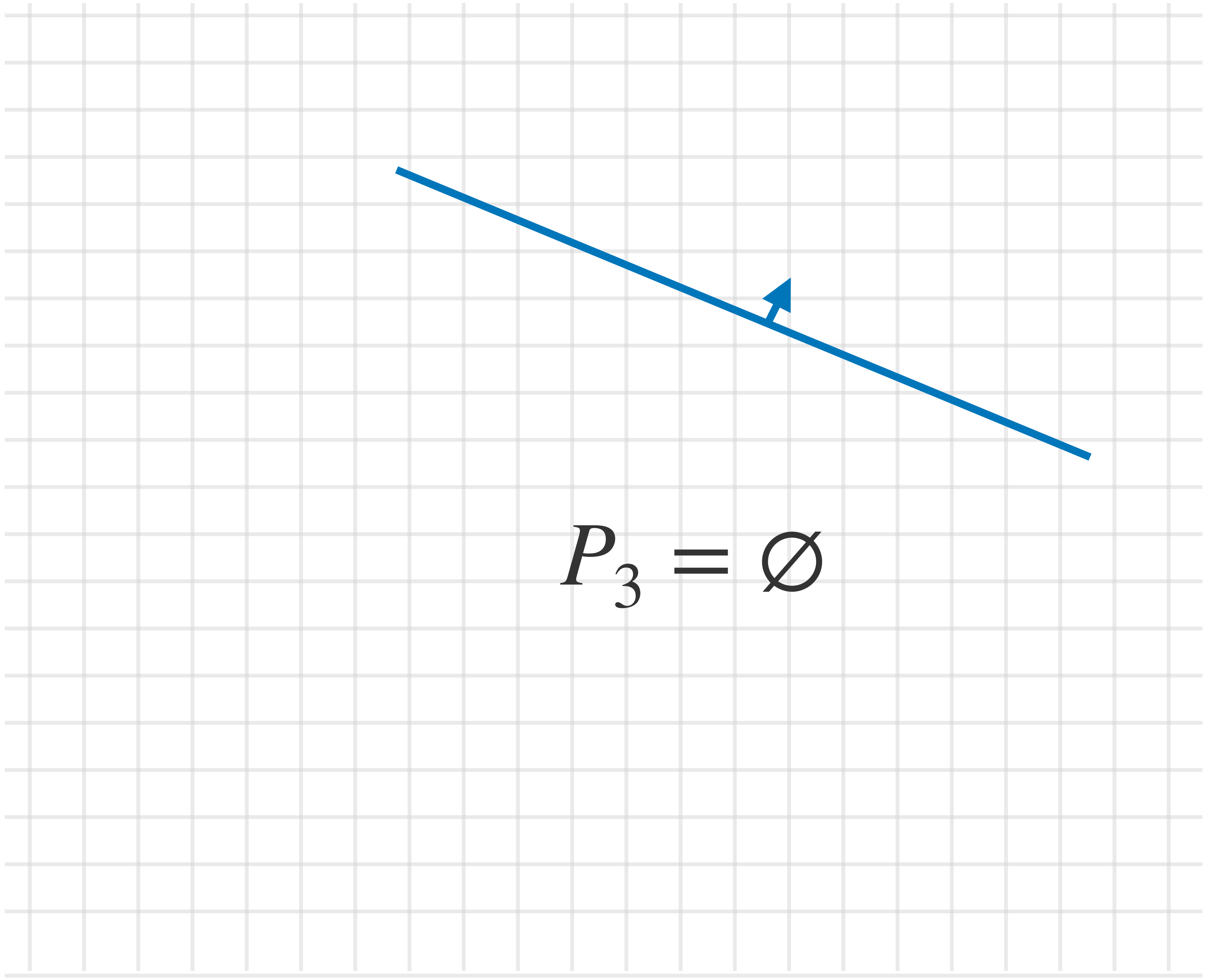
Cutting Planes

A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \emptyset)$$

where P_i is obtained by a CG-cut from P_{i-1}.

The size of the proof is s.



- Introduced in [Chvátal73].
- First exponential lower bounds in [Pudlák93] and [BPR93] for a restricted variant.

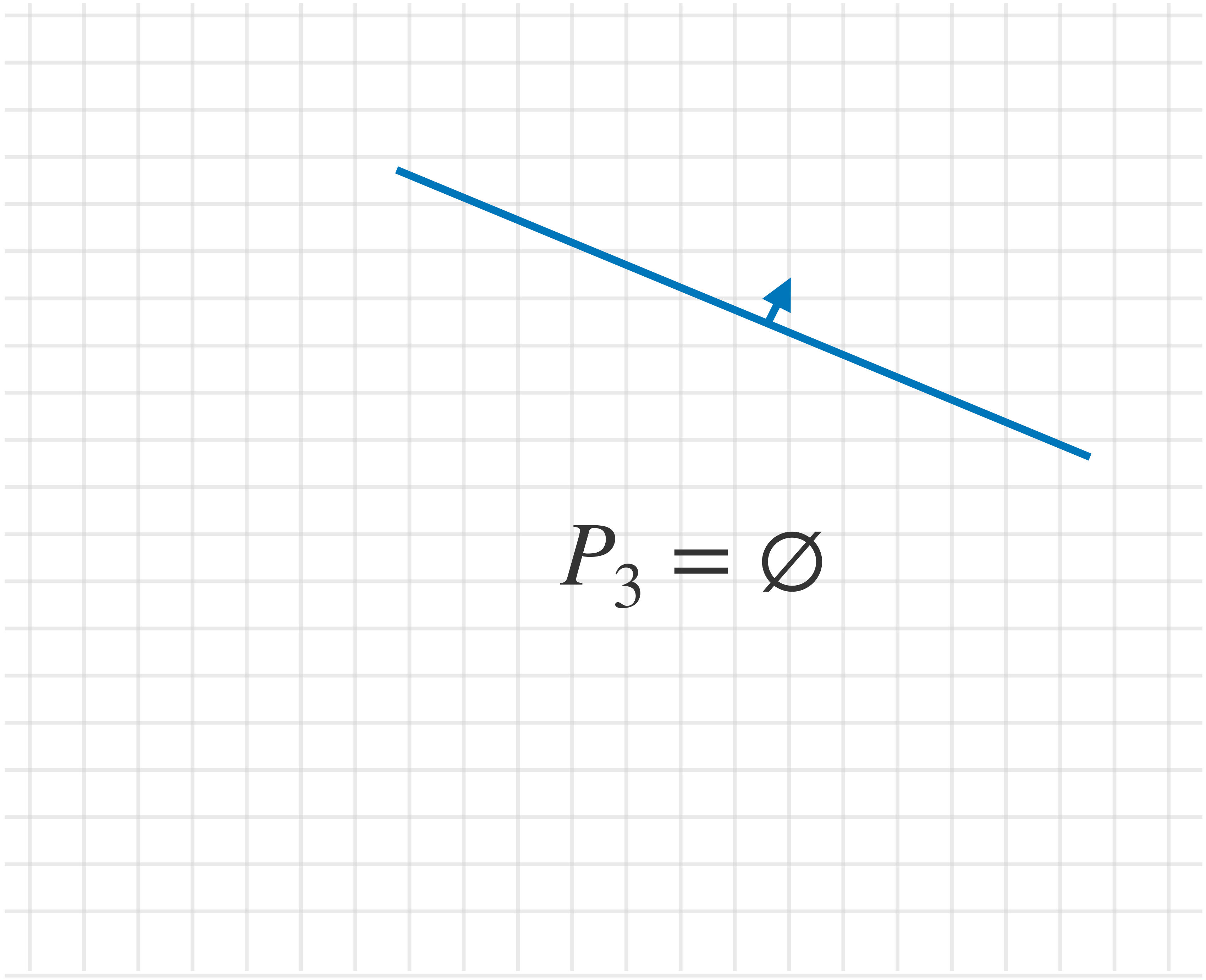
Cutting Planes

A Cutting Planes refutation of P with $P \cap \mathbb{Z}^n = \varnothing$ is a sequence of polytopes

$$(P =)\ P_0, \ldots, P_s\ (= \varnothing)$$

where P_i is obtained by a CG-cut from P_{i-1}.

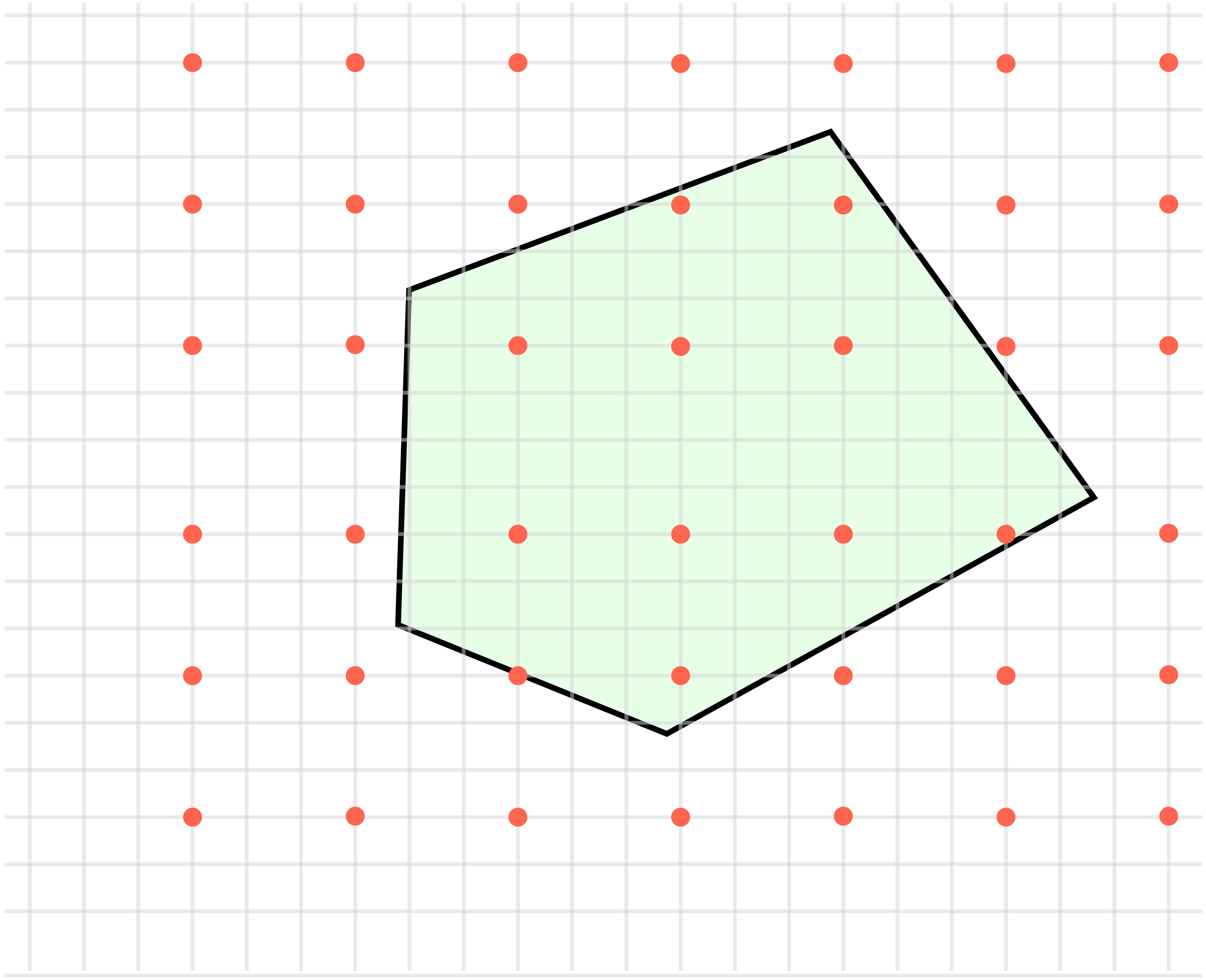
The size of the proof is s.



- Introduced in [Chvátal73].
- First exponential lower bounds in [Pudlák93] and [BPR93] for a restricted variant.
- Captures IP algorithms which use only CG-cuts (no branching).

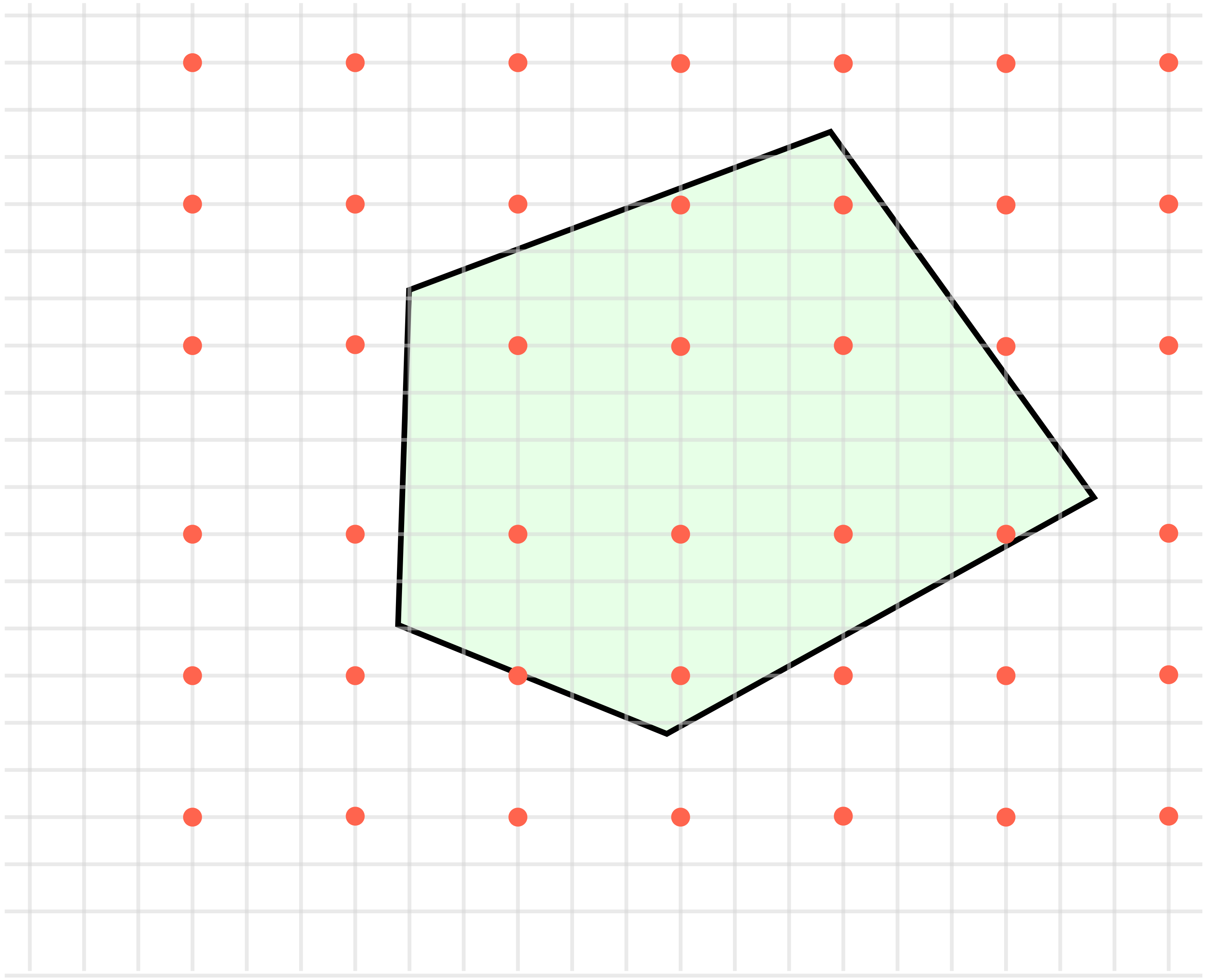
Stabbing Planes

- Introduced to model branch-and-cut algorithms [BFI+18].



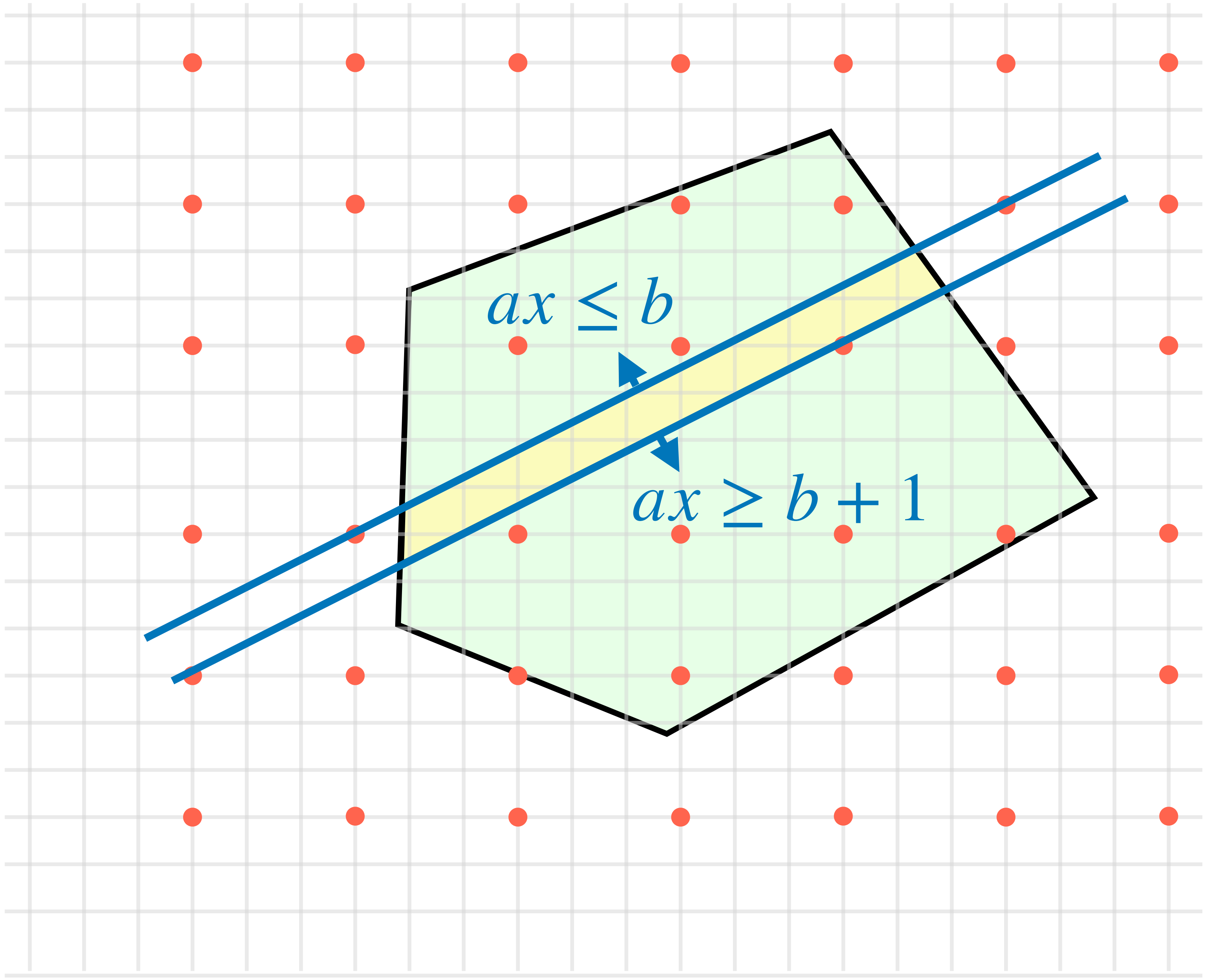
Stabbing Planes

- Introduced to model branch-and-cut algorithms [BFI+18].
- At each step one chooses an integer-linear inequality $ax \leq b$ recurses on $P \cap \{ax \leq b\}$ and $P \cap \{ax \geq b + 1\}$.



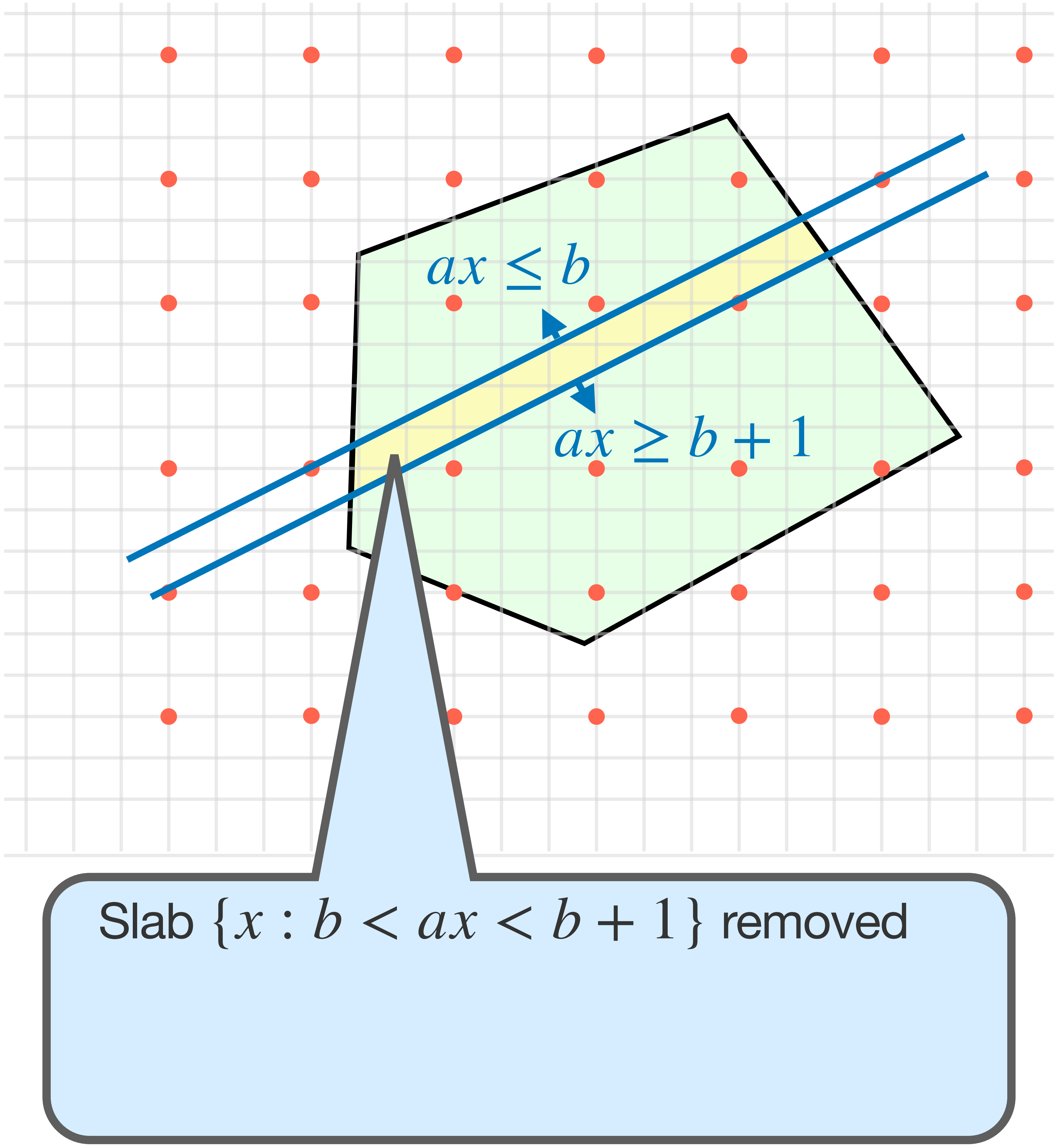
Stabbing Planes

- Introduced to model branch-and-cut algorithms [BFI+18].
- At each step one chooses an integer-linear inequality $ax \leq b$ recurses on $P \cap \{ax \leq b\}$ and $P \cap \{ax \geq b+1\}$.



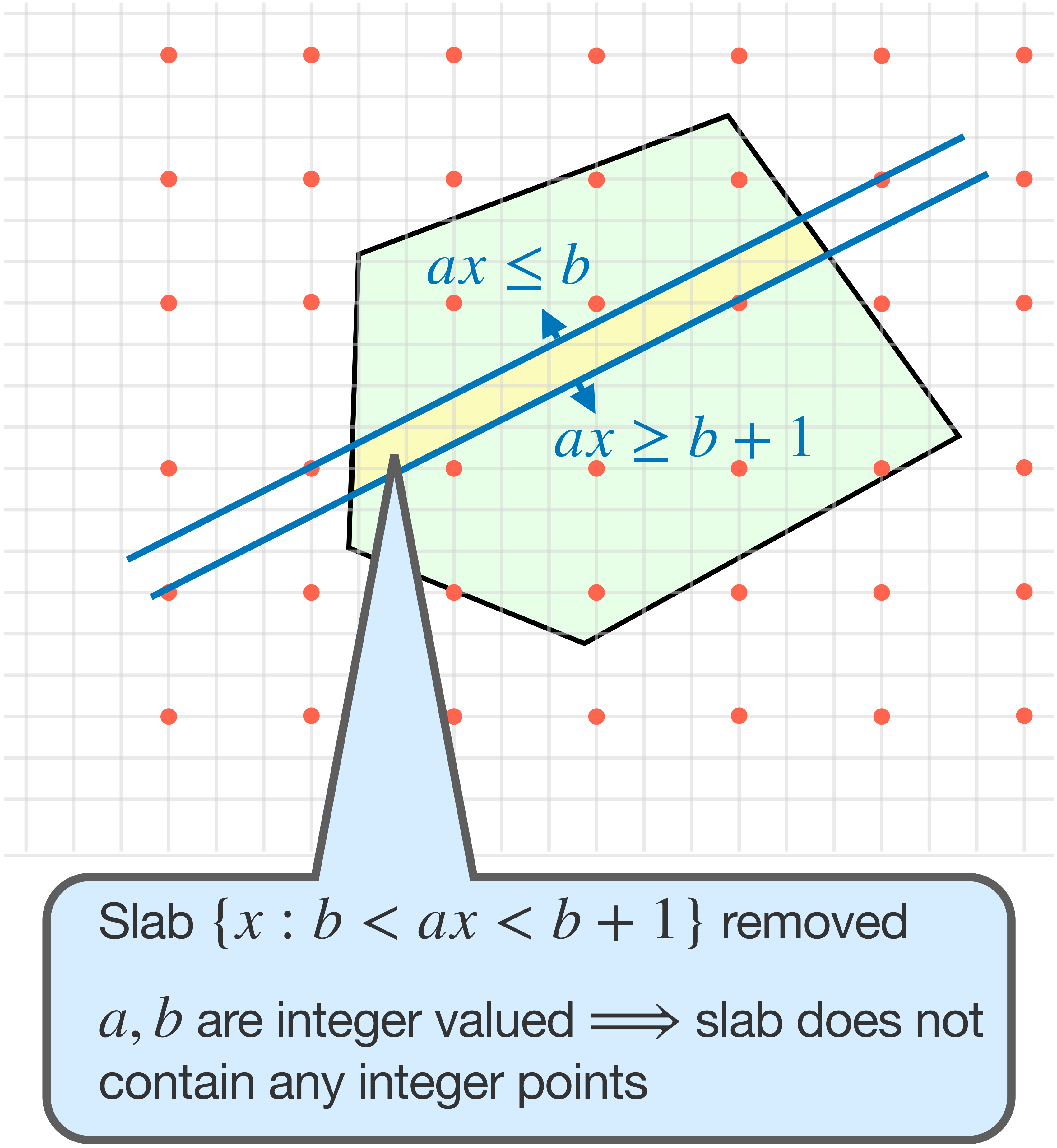
Stabbing Planes

- Introduced to model branch-and-cut algorithms [BFI+18].
- At each step one chooses an integer-linear inequality $ax \leq b$ recurses on $P \cap \{ax \leq b\}$ and $P \cap \{ax \geq b+1\}$.



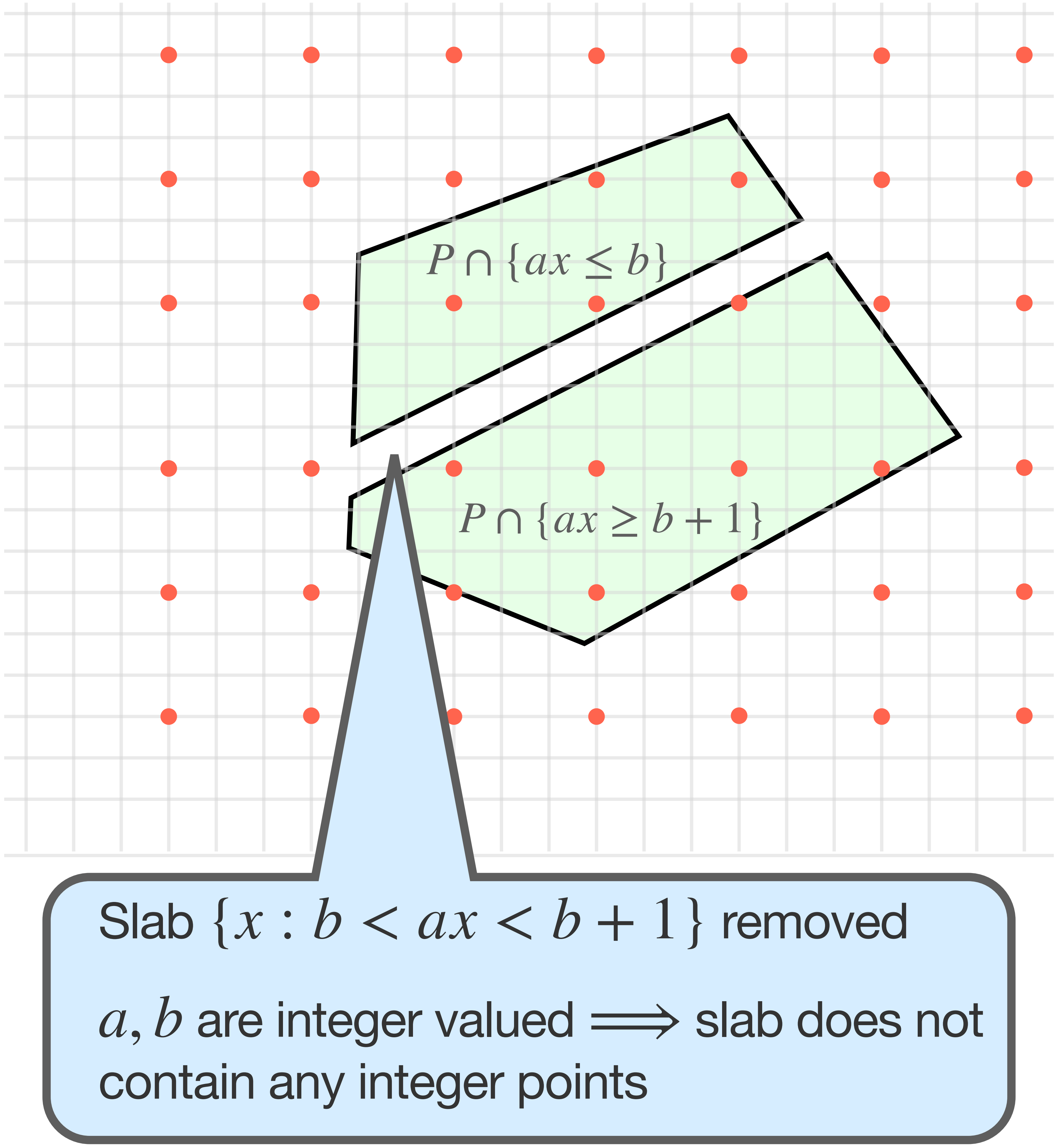
Stabbing Planes

- Introduced to model branch-and-cut algorithms [BFI+18].
- At each step one chooses an integer-linear inequality $ax \leq b$ recurses on $P \cap \{ax \leq b\}$ and $P \cap \{ax \geq b+1\}$.



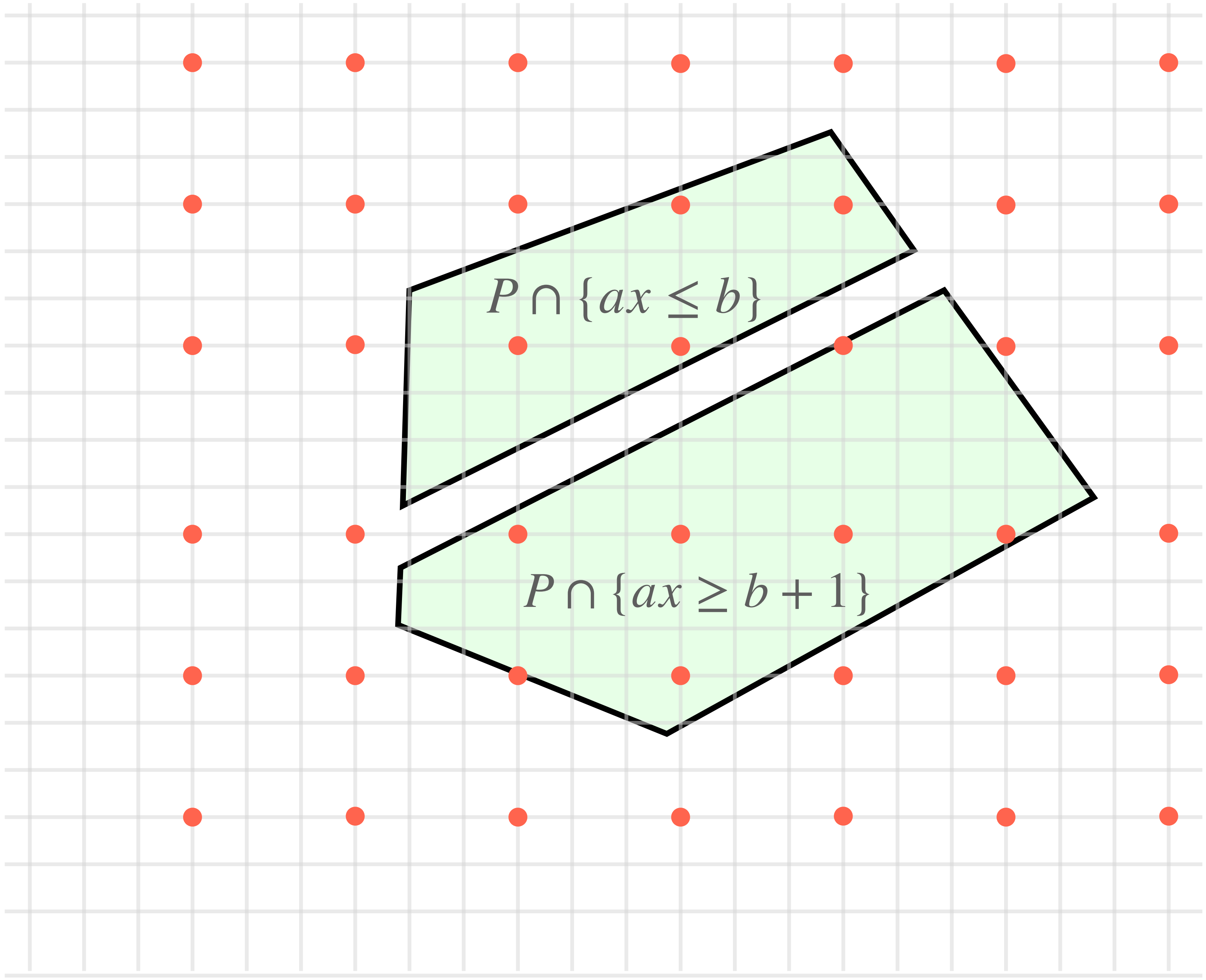
Stabbing Planes

- Introduced to model branch-and-cut algorithms [BFI+18].
- At each step one chooses an integer-linear inequality $ax \leq b$ recurses on $P \cap \{ax \leq b\}$ and $P \cap \{ax \geq b + 1\}$.



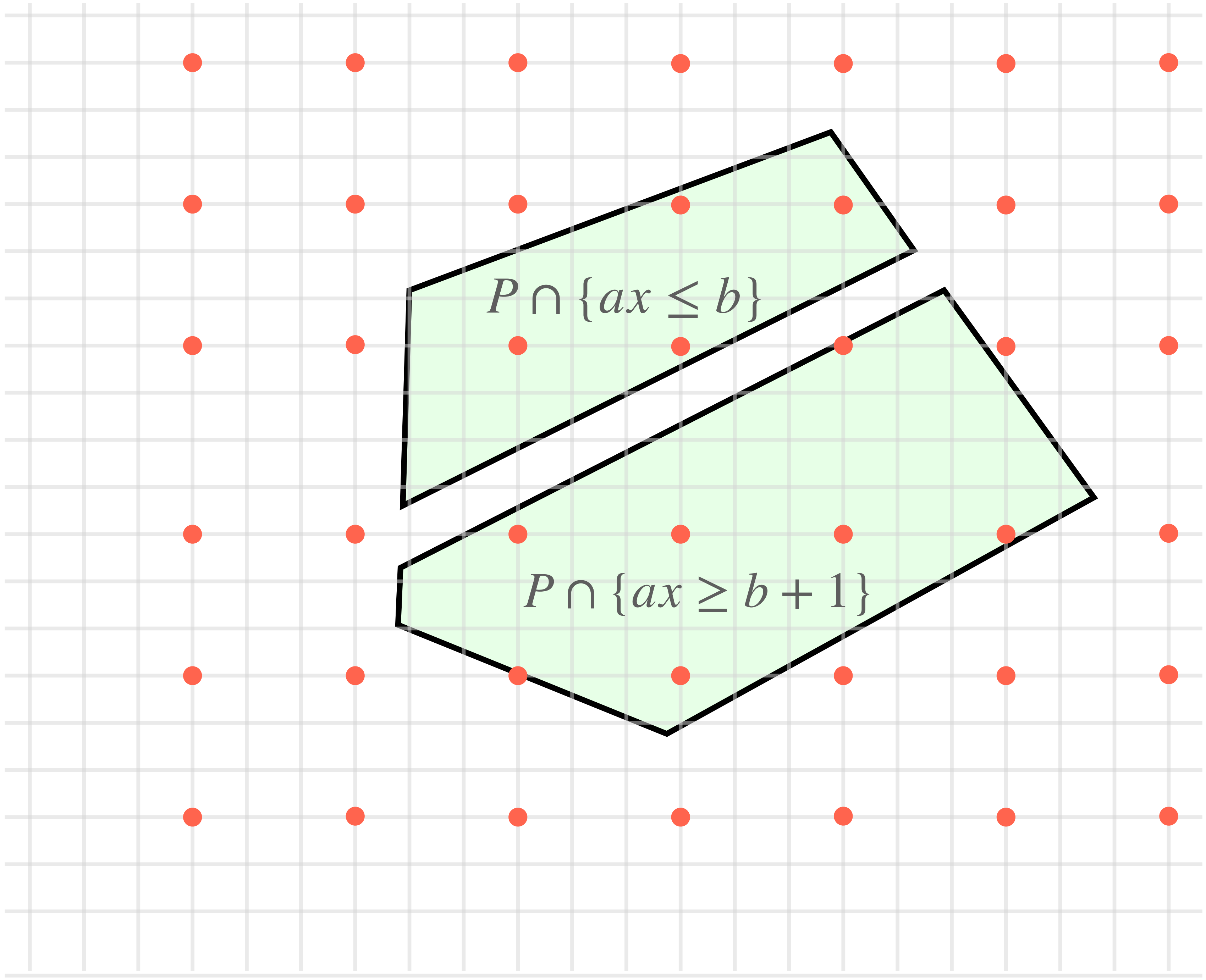
Stabbing Planes

- Introduced to model branch-and-cut algorithms [BFI+18].
- At each step one chooses an integer-linear inequality $ax \leq b$ recurses on $P \cap \{ax \leq b\}$ and $P \cap \{ax \geq b + 1\}$.
- Terminate a recursive branch when the polytope is empty.

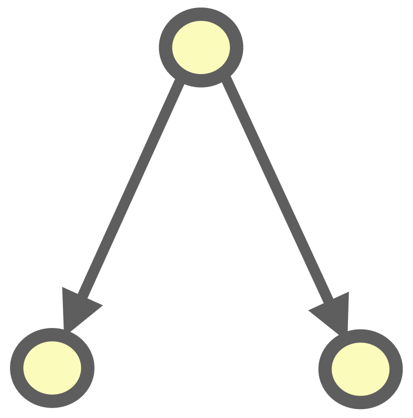


Stabbing Planes

- Introduced to model branch-and-cut algorithms [BFI+18].
- At each step one chooses an integer-linear inequality $ax \leq b$ recurses on $P \cap \{ax \leq b\}$ and $P \cap \{ax \geq b+1\}$.
- Terminate a recursive branch when the polytope is empty.

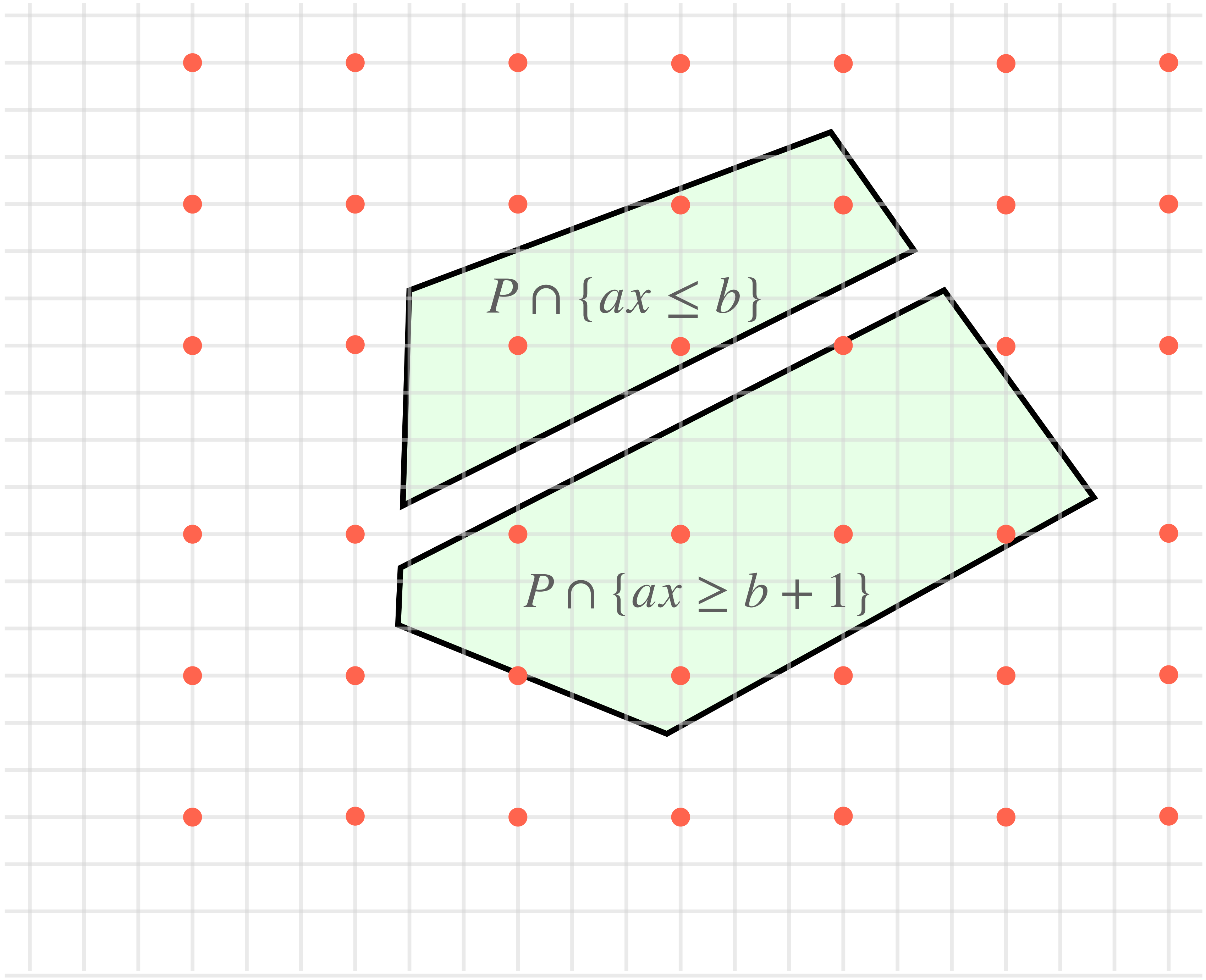


A Stabbing Planes refutation of P with $P \cap \mathbb{Z}^n$ is a binary tree:



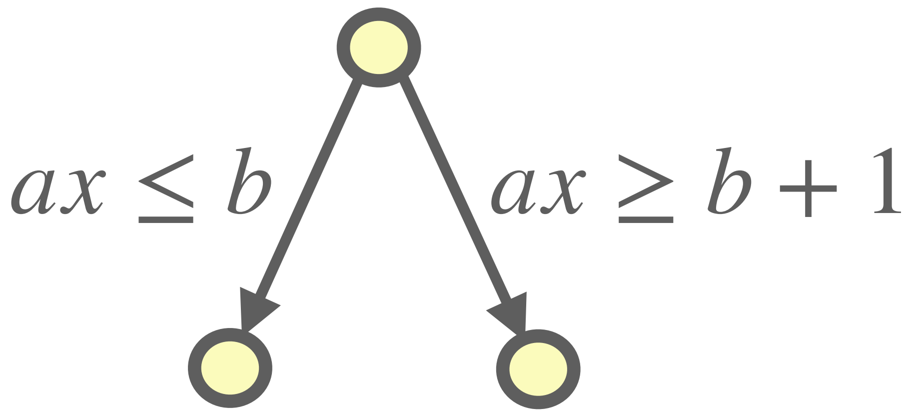
Stabbing Planes

- Introduced to model branch-and-cut algorithms [BFI+18].
- At each step one chooses an integer-linear inequality $ax \leq b$ recurses on $P \cap \{ax \leq b\}$ and $P \cap \{ax \geq b+1\}$.
- Terminate a recursive branch when the polytope is empty.



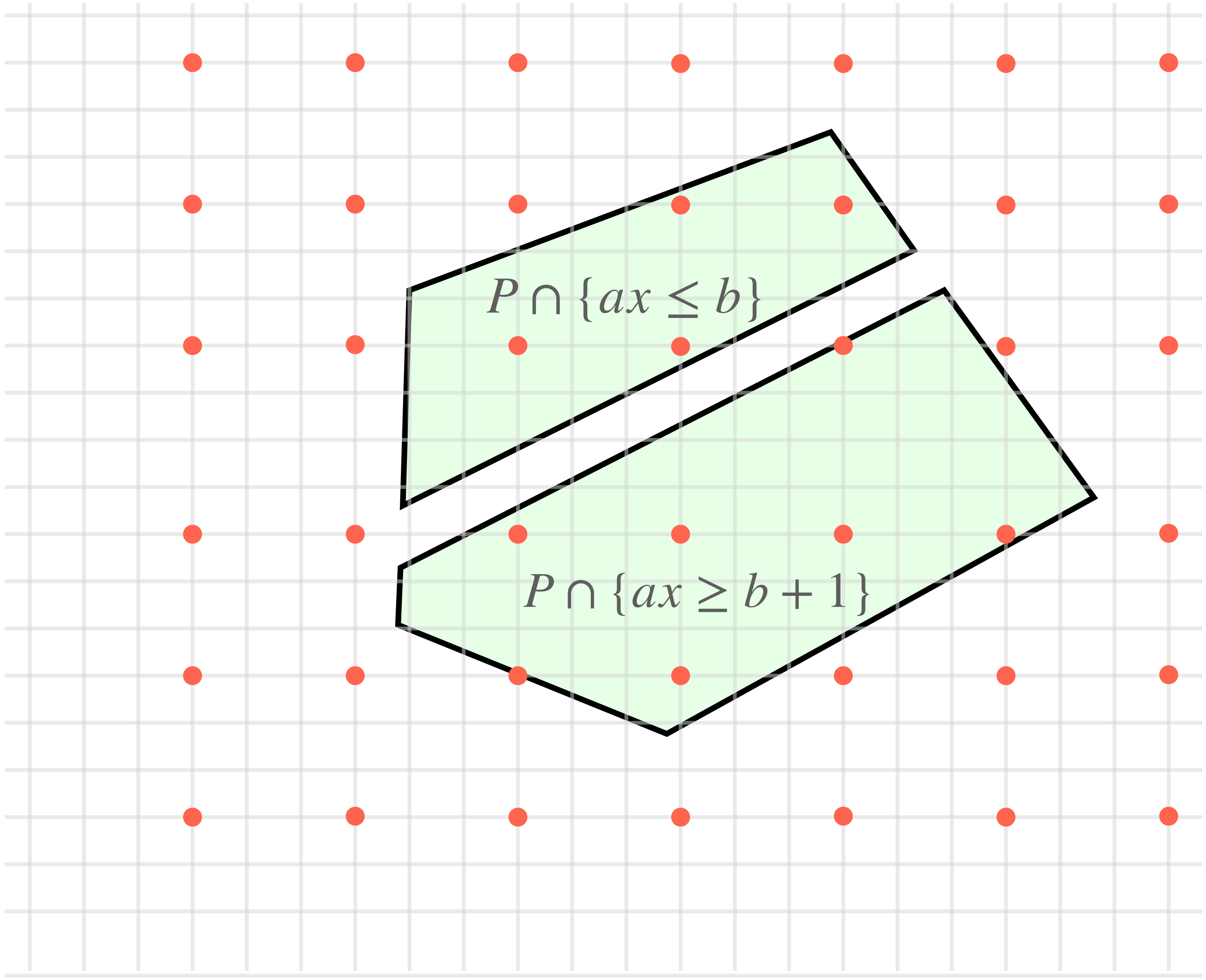
A Stabbing Planes refutation of P with $P \cap \mathbb{Z}^n$ is a binary tree:

- Each internal node has two outgoing edges labelled $ax \leq b$ and $ax \geq b+1$ for some $a \in \mathbb{Z}^n, b \in \mathbb{Z}$



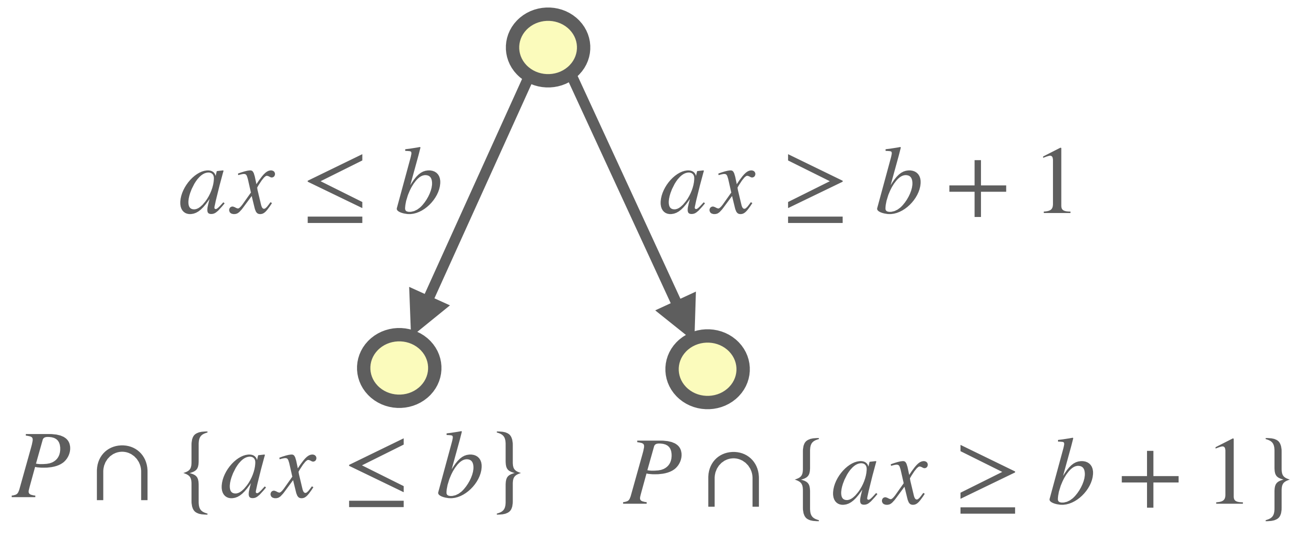
Stabbing Planes

- Introduced to model branch-and-cut algorithms [BFI+18].
- At each step one chooses an integer-linear inequality $ax \leq b$ recurses on $P \cap \{ax \leq b\}$ and $P \cap \{ax \geq b+1\}$.
- Terminate a recursive branch when the polytope is empty.



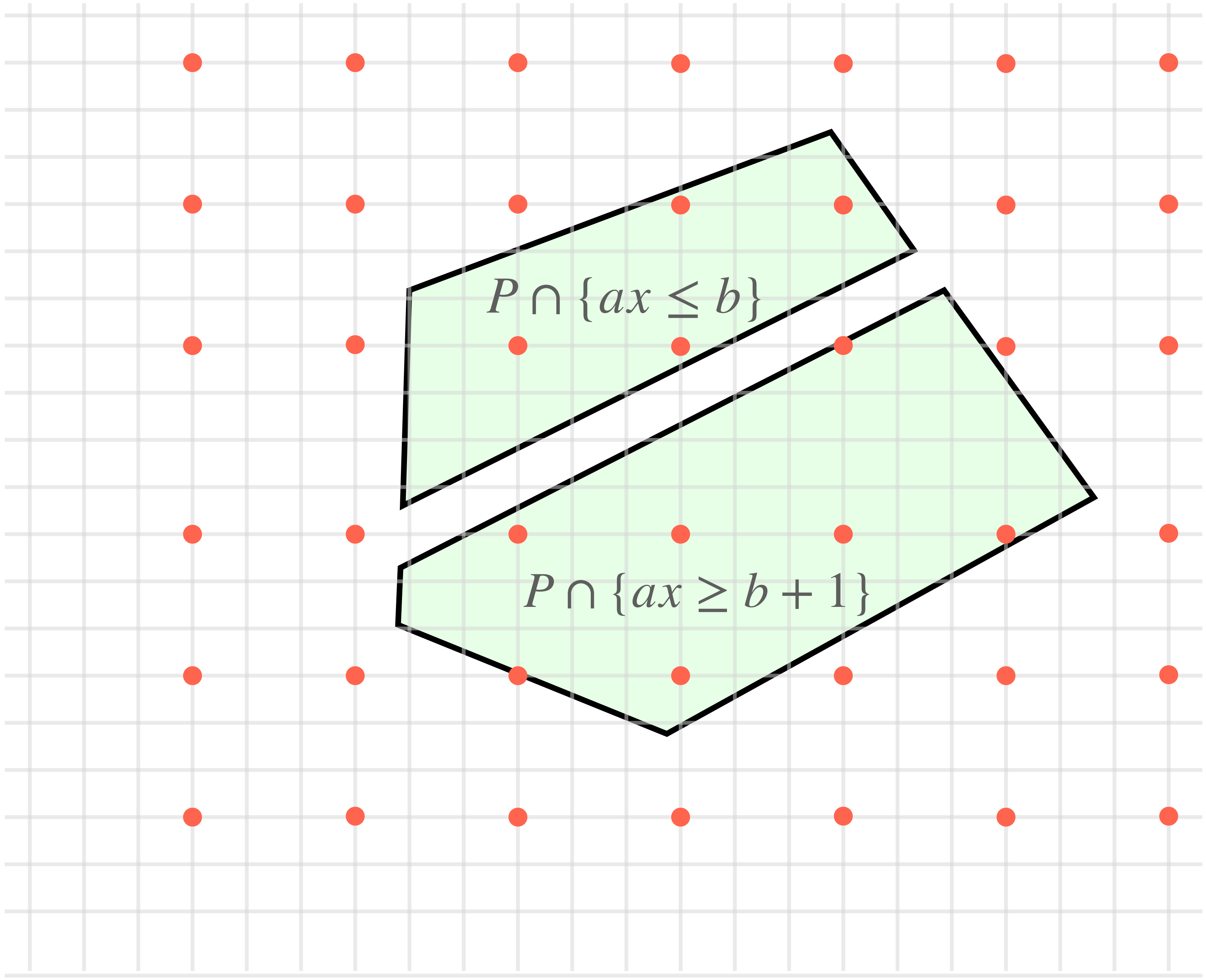
A Stabbing Planes refutation of P with $P \cap \mathbb{Z}^n$ is a binary tree:

- Each internal node has two outgoing edges labelled $ax \leq b$ and $ax \geq b+1$ for some $a \in \mathbb{Z}^n, b \in \mathbb{Z}$
- For each node v let P_v be the polytope obtained by intersecting P with the inequalities labelling the root-to-v path.



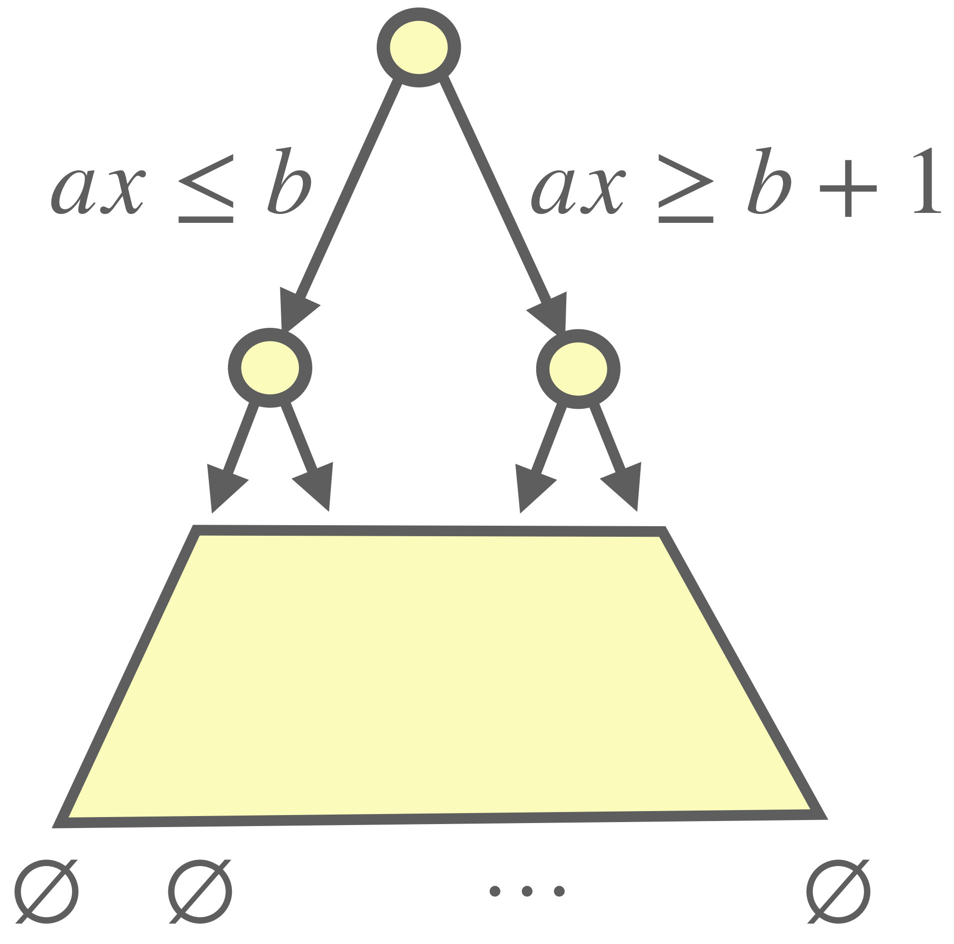
Stabbing Planes

- Introduced to model branch-and-cut algorithms [BFI+18].
- At each step one chooses an integer-linear inequality $ax \leq b$ recurses on $P \cap \{ax \leq b\}$ and $P \cap \{ax \geq b+1\}$.
- Terminate a recursive branch when the polytope is empty.



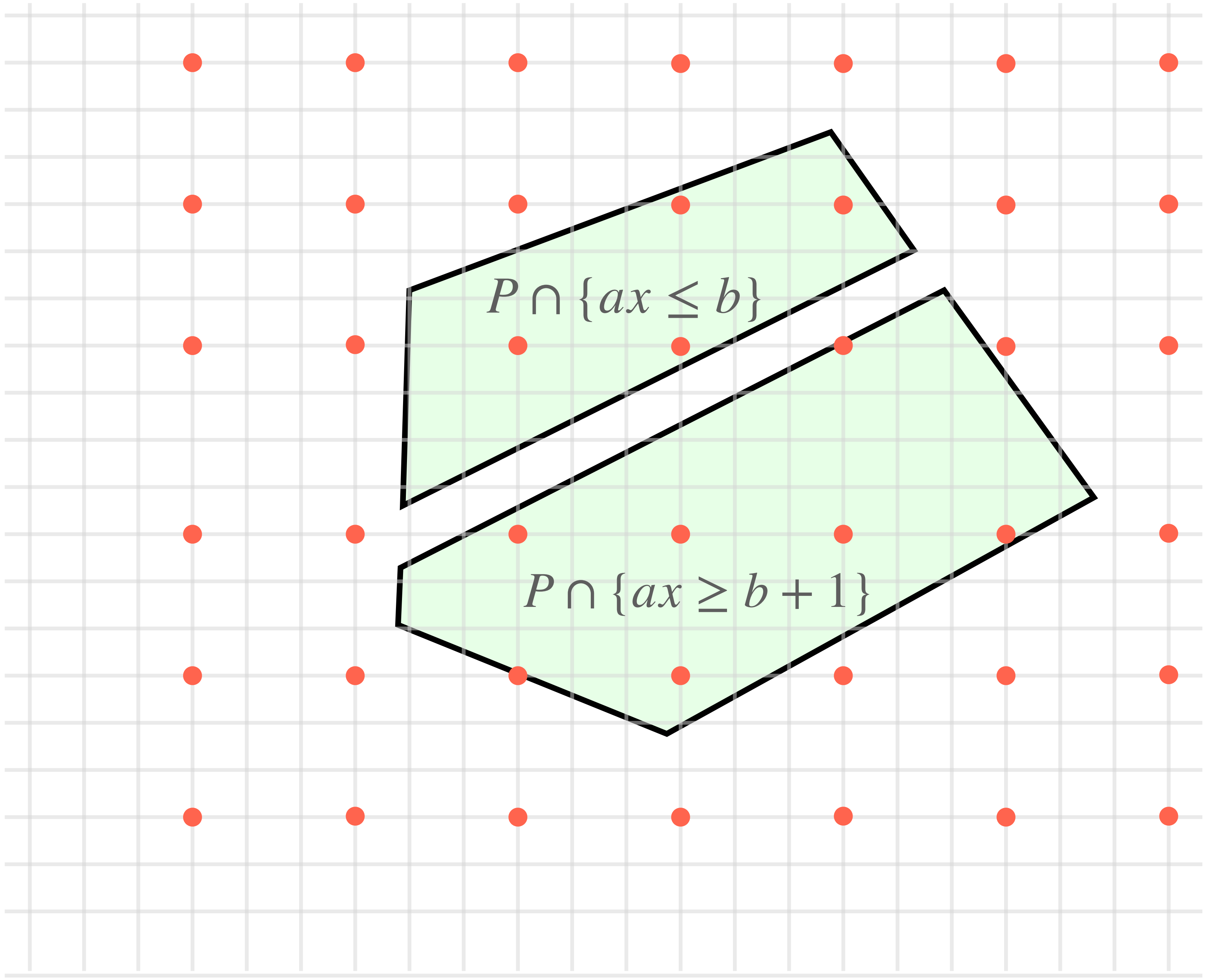
A Stabbing Planes refutation of P with $P \cap \mathbb{Z}^n$ is a binary tree:

- Each internal node has two outgoing edges labelled $ax \leq b$ and $ax \geq b+1$ for some $a \in \mathbb{Z}^n, b \in \mathbb{Z}$
- For each node v let P_v be the polytope obtained by intersecting P with the inequalities labelling the root-to-v path. Each leaf ℓ satisfies $P_\ell = \emptyset$.



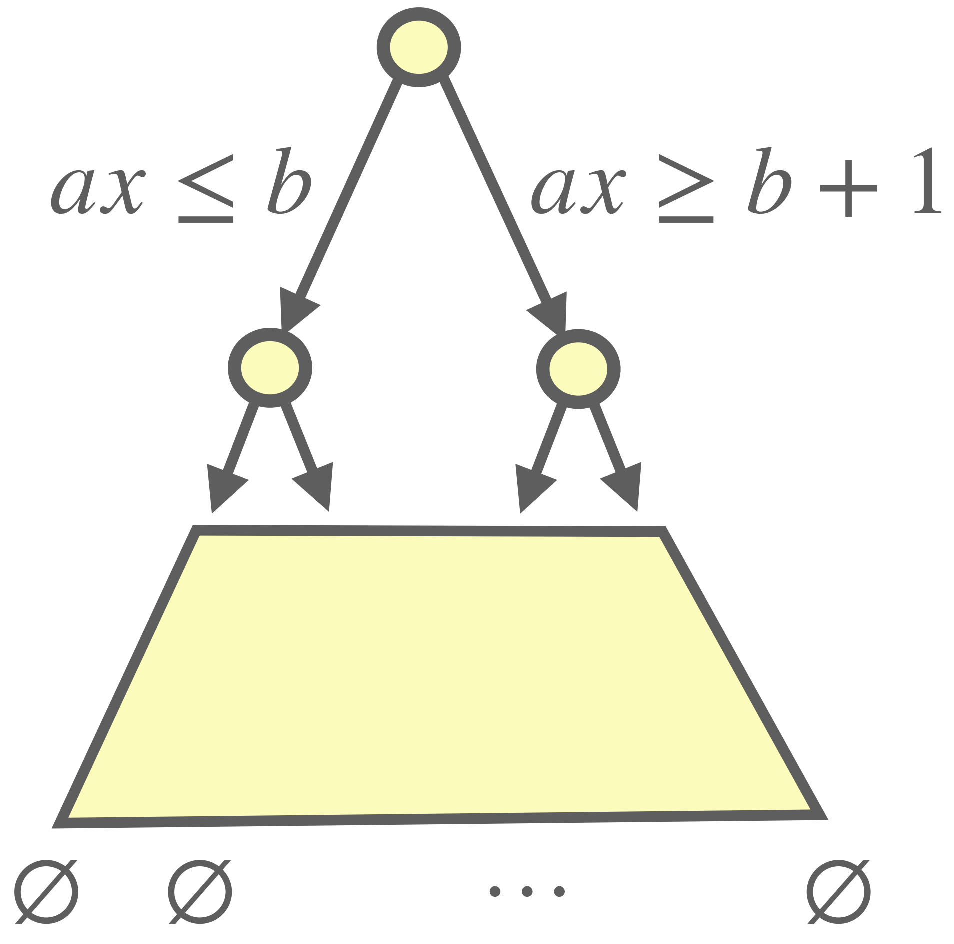
Stabbing Planes

- Introduced to model branch-and-cut algorithms [BFI+18].
- At each step one chooses an integer-linear inequality $ax \leq b$ recurses on $P \cap \{ax \leq b\}$ and $P \cap \{ax \geq b+1\}$.
- Terminate a recursive branch when the polytope is empty.



A Stabbing Planes refutation of P with $P \cap \mathbb{Z}^n$ is a binary tree:

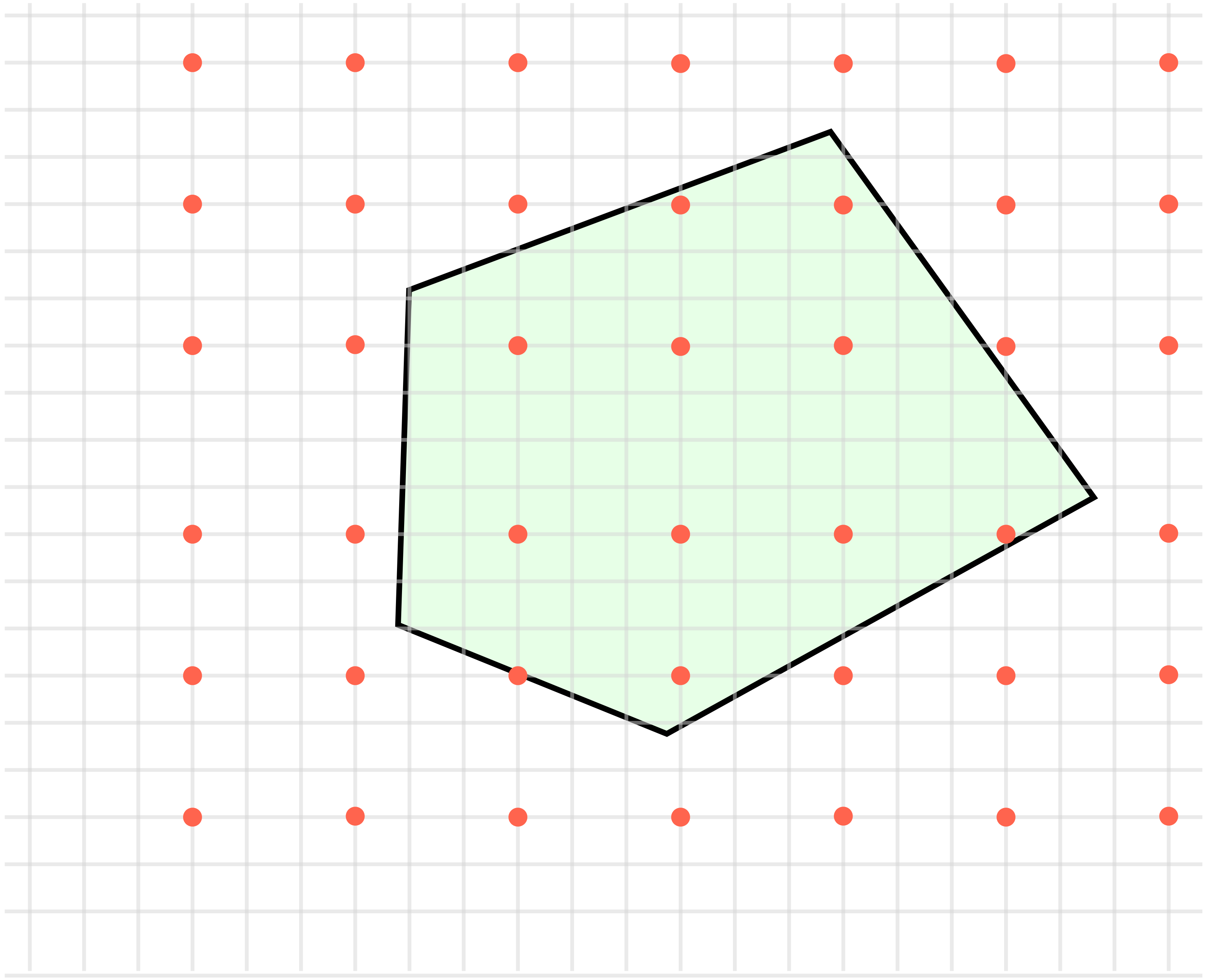
- Each internal node has two outgoing edges labelled $ax \leq b$ and $ax \geq b+1$ for some $a \in \mathbb{Z}^n, b \in \mathbb{Z}$
- For each node v let P_v be the polytope obtained by intersecting P with the inequalities labelling the root-to-v path. Each leaf ℓ satisfies $P_\ell = \emptyset$.



- The size is the number of nodes in the tree

Cutting Planes vs. Stabbing Planes

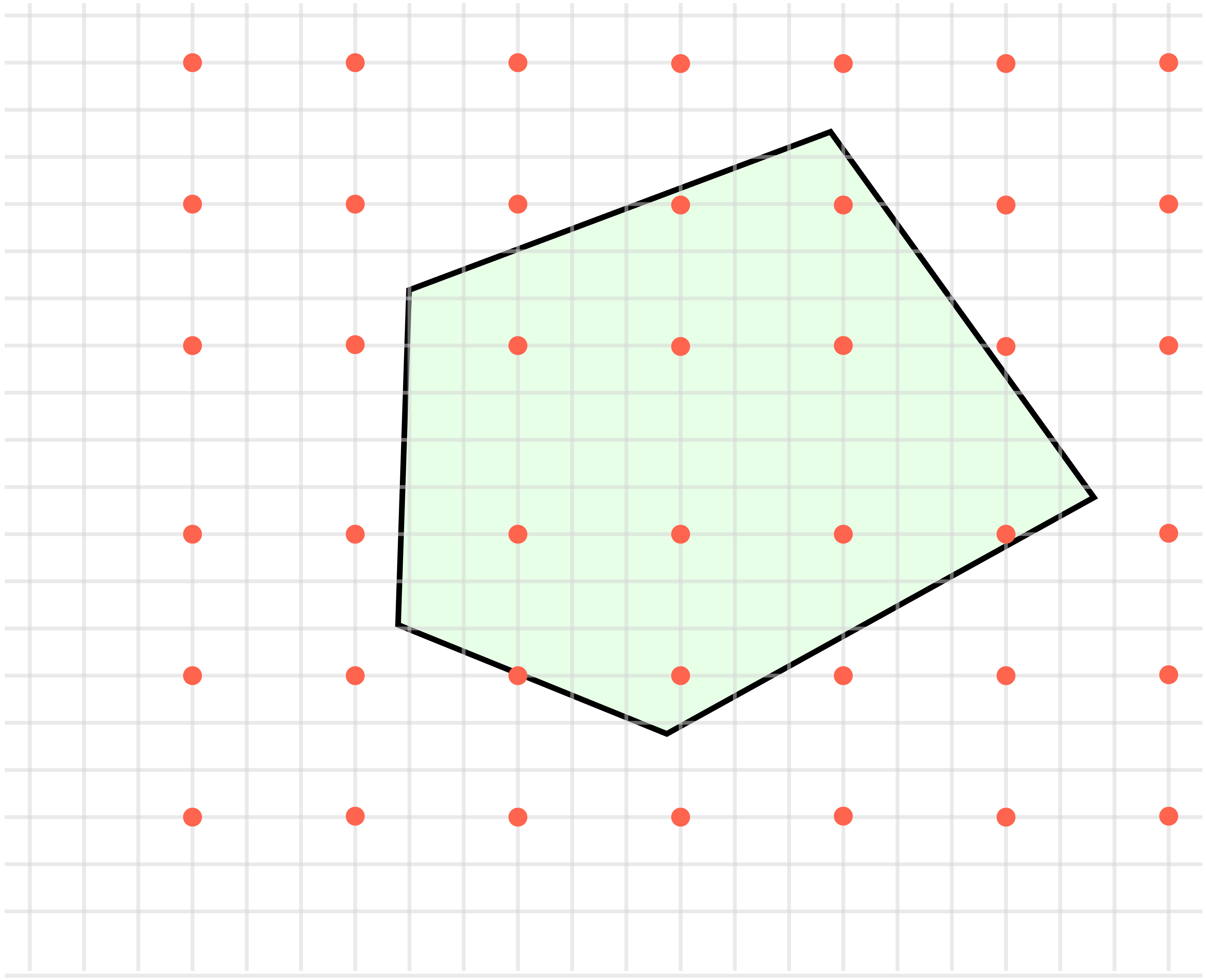
CP and SP capture separate parts of branch-and-cut



Cutting Planes vs. Stabbing Planes

CP and SP capture separate parts of branch-and-cut

- CP captures CG-cuts,
- SP captures branching.

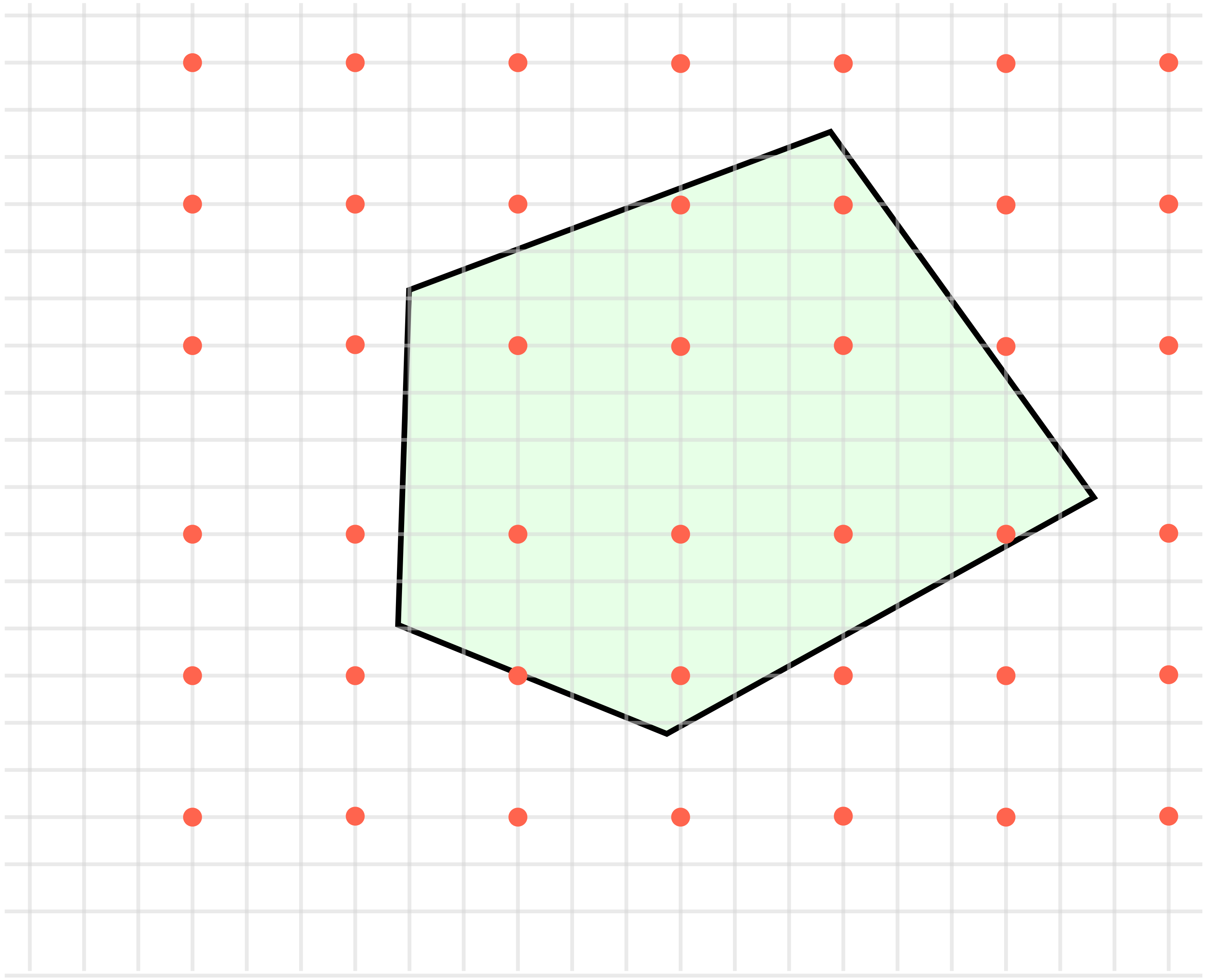


Cutting Planes vs. Stabbing Planes

CP and SP capture separate parts of branch-and-cut

- CP captures CG-cuts,
- SP captures branching.

Theorem: If P has a size s Cutting Planes proof then there is a size $O(s)$ Stabbing Planes refutation of P.



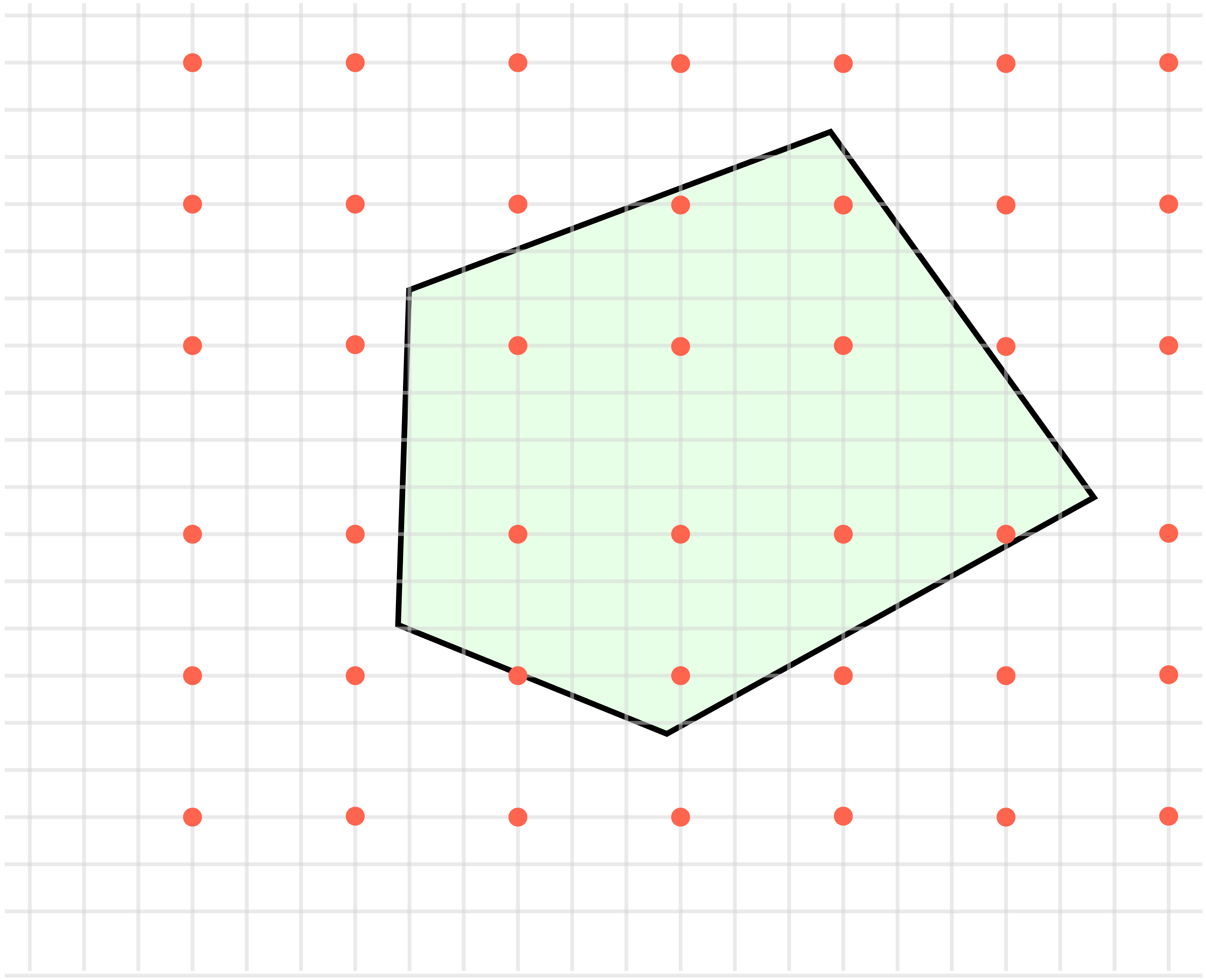
Cutting Planes vs. Stabbing Planes

CP and SP capture separate parts of branch-and-cut

- CP captures CG-cuts,
- SP captures branching.

Theorem: If P has a size s Cutting Planes proof then there is a size $O(s)$ Stabbing Planes refutation of P.

A Stabbing Planes query $ax \leq b, ax \geq b+1$ is **pathlike** if either $P_u \cap \{ax \leq b\} = \emptyset$ or $P_u \cap \{ax \geq b+1\} = \emptyset$.



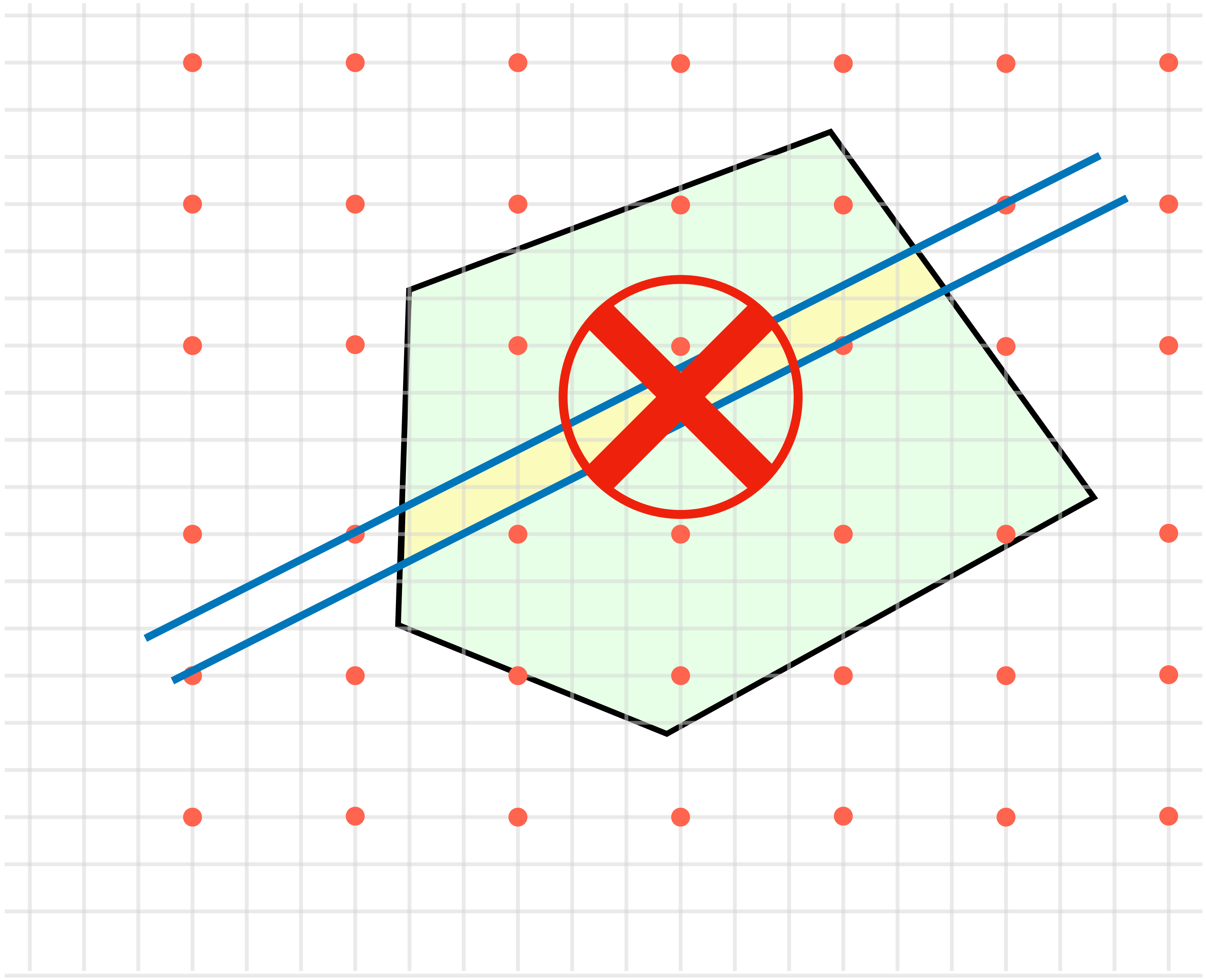
Cutting Planes vs. Stabbing Planes

CP and SP capture separate parts of branch-and-cut

- CP captures CG-cuts,
- SP captures branching.

Theorem: If P has a size s Cutting Planes proof then there is a size $O(s)$ Stabbing Planes refutation of P.

A Stabbing Planes query $ax \leq b, ax \geq b+1$ is **pathlike** if either $P_u \cap \{ax \leq b\} = \varnothing$ or $P_u \cap \{ax \geq b+1\} = \varnothing$.



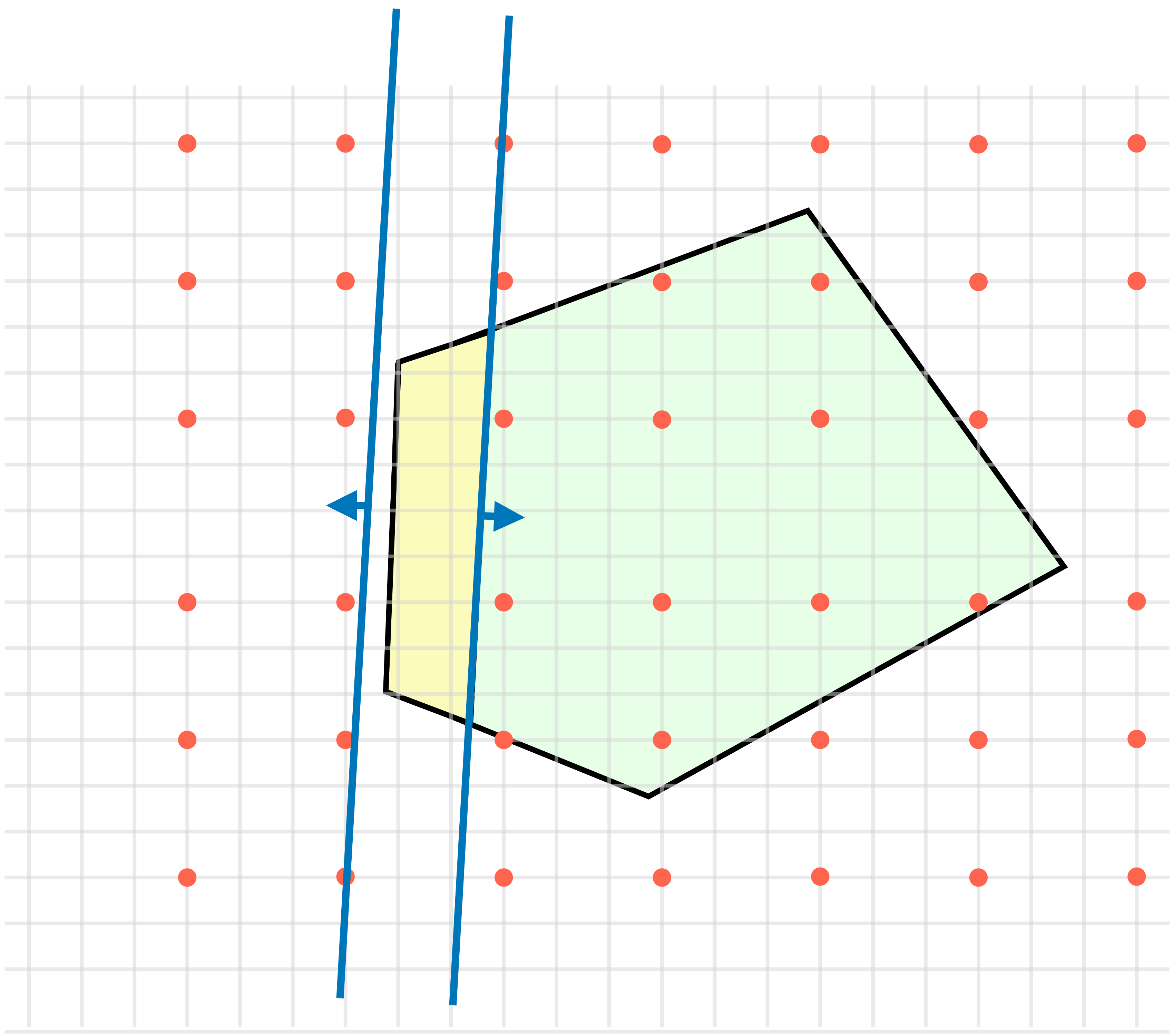
Cutting Planes vs. Stabbing Planes

CP and SP capture separate parts of branch-and-cut

- CP captures CG-cuts,
- SP captures branching.

Theorem: If P has a size s Cutting Planes proof then there is a size $O(s)$ Stabbing Planes refutation of P.

A Stabbing Planes query $ax \leq b, ax \geq b+1$ is **pathlike** if either $P_u \cap \{ax \leq b\} = \varnothing$ or $P_u \cap \{ax \geq b+1\} = \varnothing$.



Cutting Planes vs. Stabbing Planes

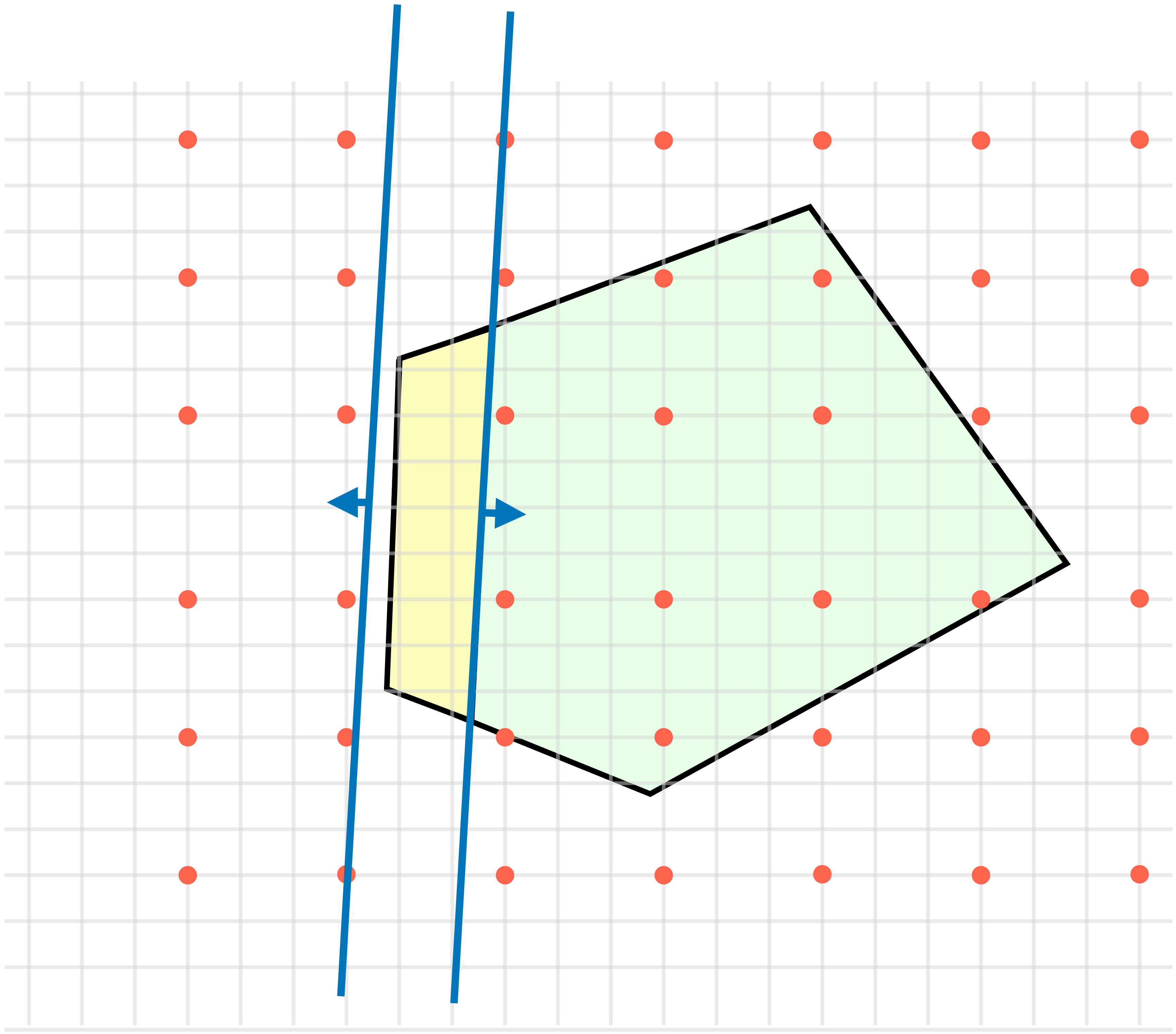
CP and SP capture separate parts of branch-and-cut

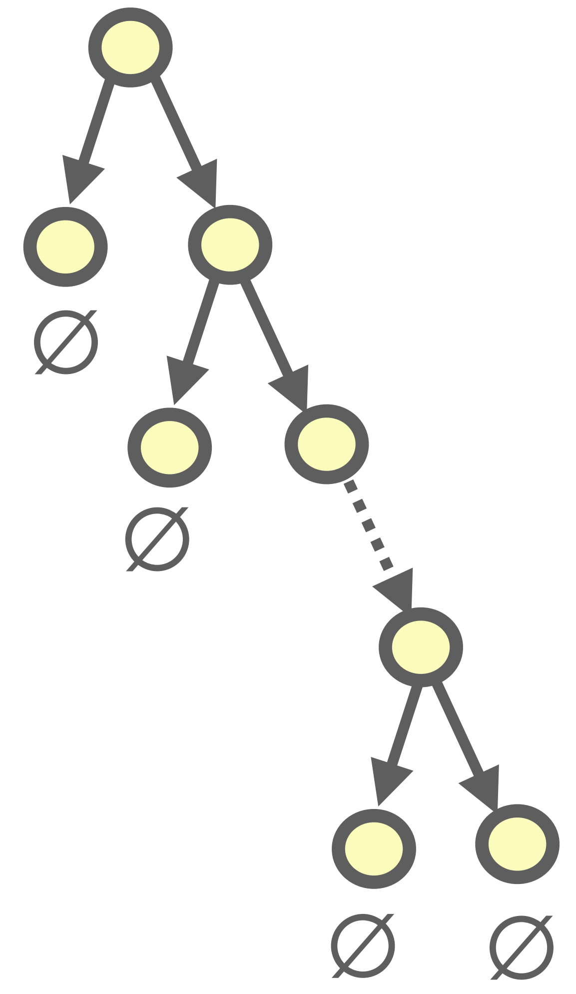
- CP captures CG-cuts,
- SP captures branching.

Theorem: If P has a size s Cutting Planes proof then there is a size $O(s)$ Stabbing Planes refutation of P.

A Stabbing Planes query $ax \leq b, ax \geq b+1$ is **pathlike** if either $P_u \cap \{ax \leq b\} = \varnothing$ or $P_u \cap \{ax \geq b+1\} = \varnothing$.

A Stabbing Planes proof is **pathlike** if every query is pathlike.





Cutting Planes vs. Stabbing Planes

CP and SP capture separate parts of branch-and-cut

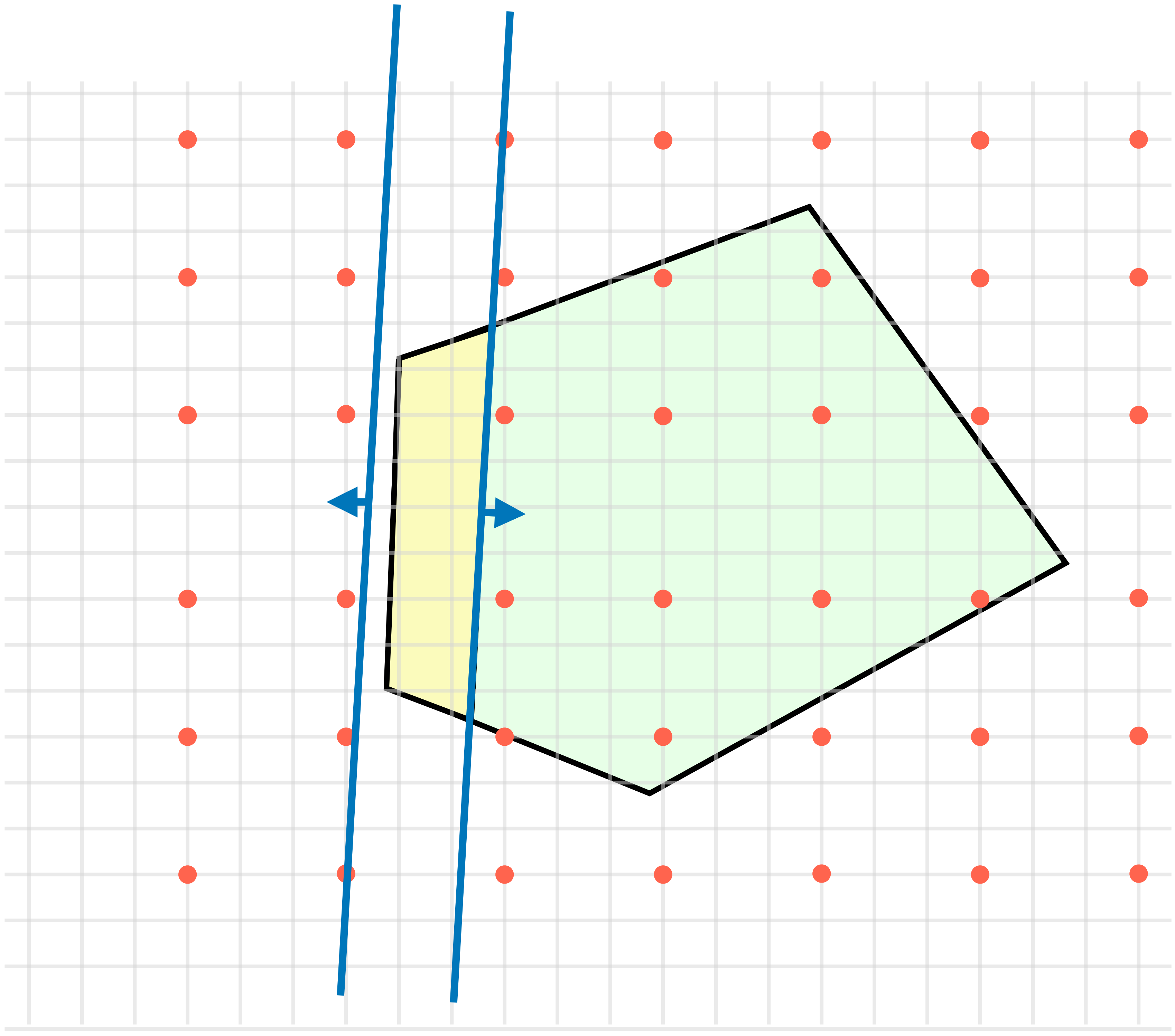
- CP captures CG-cuts,
- SP captures branching.

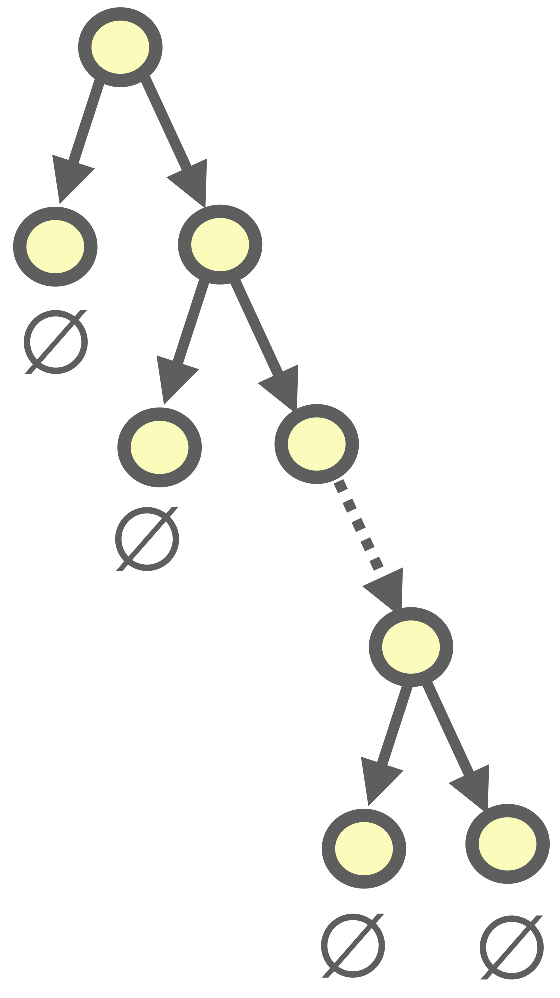
Theorem: If P has a size s Cutting Planes proof then there is a size $O(s)$ Stabbing Planes refutation of P.

A Stabbing Planes query $ax \leq b, ax \geq b+1$ is **pathlike** if either $P_u \cap \{ax \leq b\} = \varnothing$ or $P_u \cap \{ax \geq b+1\} = \varnothing$.

A Stabbing Planes proof is **pathlike** if every query is pathlike.

Theorem: Pathlike Stabbing Planes = Cutting Planes





Cutting Planes vs. Stabbing Planes

CP and SP capture separate parts of branch-and-cut

- CP captures CG-cuts,
- SP captures branching.

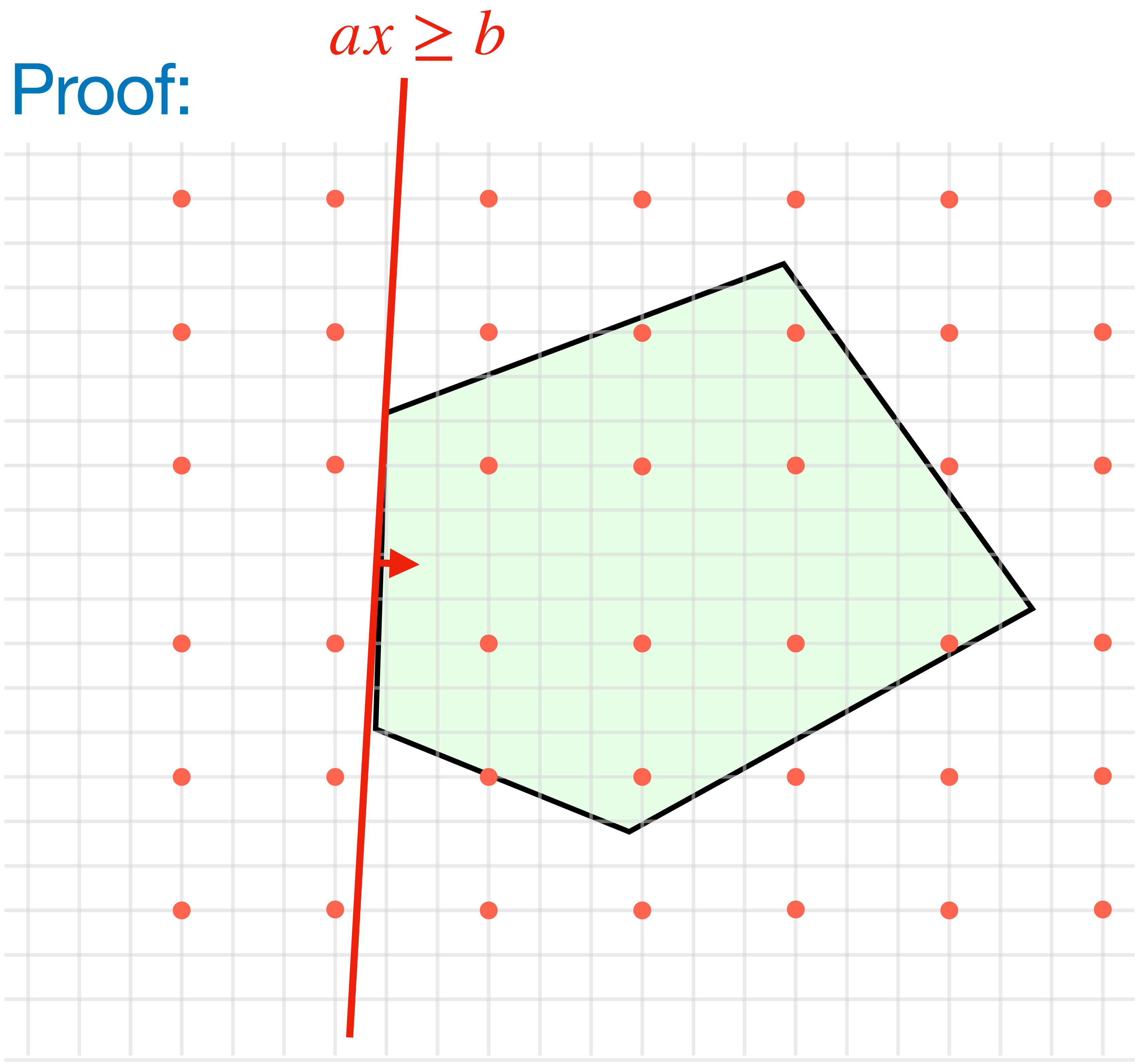
Theorem: If P has a size s Cutting Planes proof then there is a size $O(s)$ Stabbing Planes refutation of P.

A Stabbing Planes query $ax \leq b, ax \geq b+1$ is **pathlike** if either $P_u \cap \{ax \leq b\} = \emptyset$ or $P_u \cap \{ax \geq b+1\} = \emptyset$.

A Stabbing Planes proof is **pathlike** if every query is pathlike.

Theorem: Pathlike Stabbing Planes = Cutting Planes

Proof:



Cutting Planes vs. Stabbing Planes

CP and SP capture separate parts of branch-and-cut

- CP captures CG-cuts,
- SP captures branching.

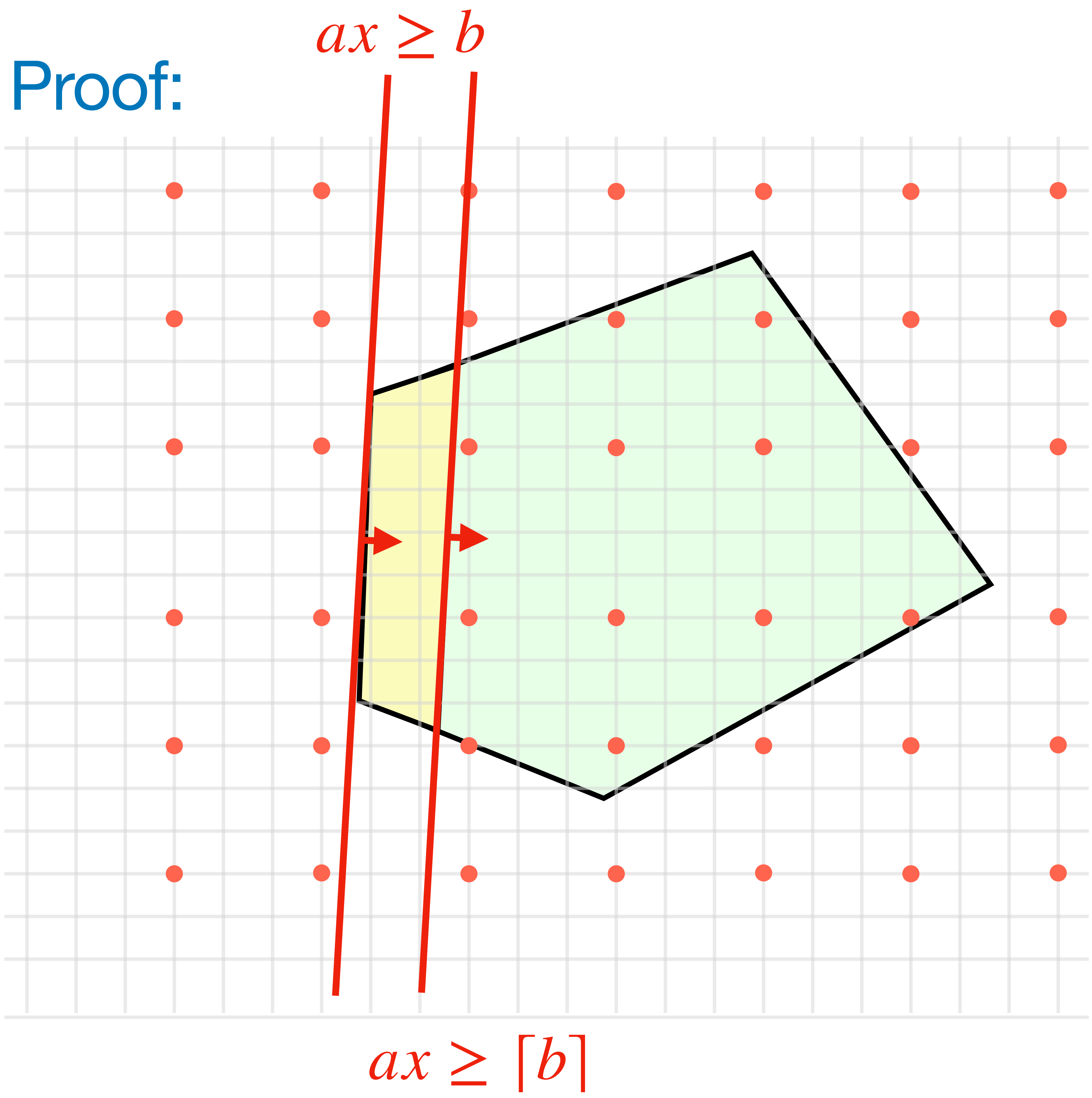
Theorem: If P has a size s Cutting Planes proof then there is a size $O(s)$ Stabbing Planes refutation of P.

A Stabbing Planes query $ax \leq b, ax \geq b+1$ is **pathlike** if either $P_u \cap \{ax \leq b\} = \emptyset$ or $P_u \cap \{ax \geq b+1\} = \emptyset$.

A Stabbing Planes proof is **pathlike** if every query is pathlike.

Theorem: Pathlike Stabbing Planes = Cutting Planes

Proof:



Cutting Planes vs. Stabbing Planes

CP and SP capture separate parts of branch-and-cut

- CP captures CG-cuts,
- SP captures branching.

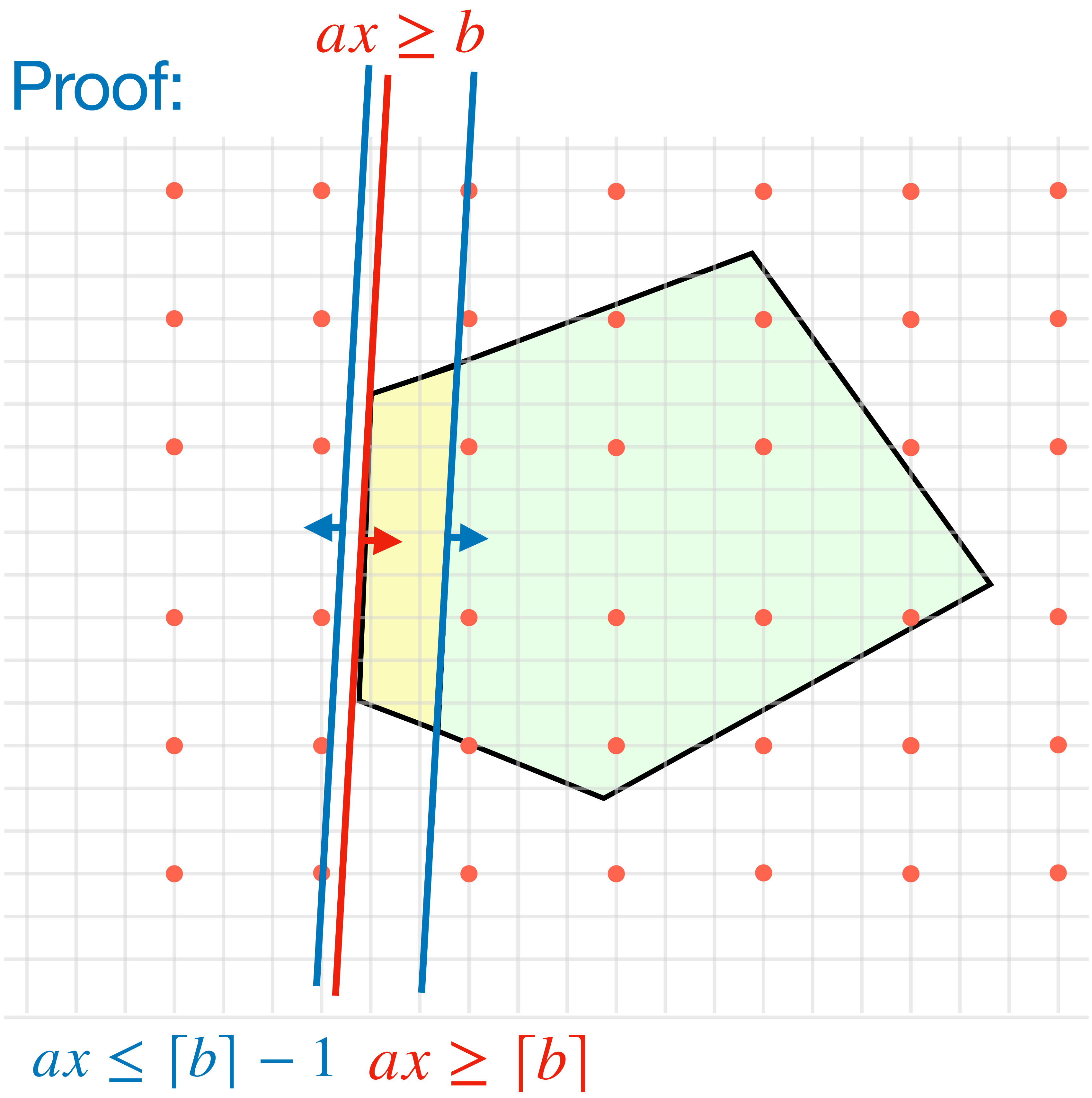
Theorem: If P has a size s Cutting Planes proof then there is a size $O(s)$ Stabbing Planes refutation of P.

A Stabbing Planes query $ax \leq b, ax \geq b+1$ is **pathlike** if either $P_u \cap \{ax \leq b\} = \varnothing$ or $P_u \cap \{ax \geq b+1\} = \varnothing$.

A Stabbing Planes proof is **pathlike** if every query is pathlike.

Theorem: Pathlike Stabbing Planes = Cutting Planes

Proof:



Cutting Planes vs. Stabbing Planes

Is Cutting Planes weaker than Stabbing Planes?

Cutting Planes vs. Stabbing Planes

Is Cutting Planes weaker than Stabbing Planes?

- [BFI+18] conjectured that the **Tseitin formulas** are a separating example.

Cutting Planes vs. Stabbing Planes

Is Cutting Planes weaker than Stabbing Planes?

- [BFI+18] conjectured that the **Tseitin formulas** are a separating example.

Let $G = (V, E)$ be a connected graph with an odd number of vertices. The Tseitin formula for G is the system of $\mathbb{F}_2$-linear equations

$$\forall v \in V : \bigoplus_{uv \in E} x_{uv} = 1$$

asserting that there is a way to assign edges so that each vertex has an **odd** number of neighbours.

Cutting Planes vs. Stabbing Planes

Is Cutting Planes weaker than Stabbing Planes?

- [BFI+18] conjectured that the **Tseitin formulas** are a separating example.

Let $G = (V, E)$ be a connected graph with an odd number of vertices. The Tseitin formula for G is the system of $\mathbb{F}_2$-linear equations

$$\forall v \in V : \bigoplus_{uv \in E} x_{uv} = 1$$

asserting that there is a way to assign edges so that each vertex has an **odd** number of neighbours.

- Conjectured in the 80s to require exponential Cutting Planes proofs.

Cutting Planes vs. Stabbing Planes

Is Cutting Planes weaker than Stabbing Planes?

- [BFI+18] conjectured that the **Tseitin formulas** are a separating example.

Let $G = (V, E)$ be a connected graph with an odd number of vertices. The Tseitin formula for G is the system of $\mathbb{F}_2$-linear equations

$$\forall v \in V: \bigoplus_{uv \in E} x_{uv} = 1$$

asserting that there is a way to assign edges so that each vertex has an **odd** number of neighbours.

- Conjectured in the 80s to require exponential Cutting Planes proofs.
- [BFI+18] There are $n^{O(\log n)}$-size Cutting Planes proofs of Tseitin.

Cutting Planes vs. Stabbing Planes

Is Cutting Planes weaker than Stabbing Planes?

- [BFI+18] conjectured that the **Tseitin formulas** are a separating example.

Let $G = (V, E)$ be a connected graph with an odd number of vertices. The Tseitin formula for G is the system of $\mathbb{F}_2$-linear equations

$$\forall v \in V : \bigoplus_{uv \in E} x_{uv} = 1$$

asserting that there is a way to assign edges so that each vertex has an **odd** number of neighbours.

- Conjectured in the 80s to require exponential Cutting Planes proofs.
- [BFI+18] There are $n^{O(\log n)}$-size Cutting Planes proofs of Tseitin.

[DT20] The quasi-polynomial size Stabbing Planes proofs of Tseitin can be translated into quasi-polynomial size Cutting Planes proofs!

Cutting Planes vs. Stabbing Planes

Can **every** Stabbing Planes proof be efficiently translated into Cutting Planes?

Cutting Planes vs. Stabbing Planes

Can **every** Stabbing Planes proof be efficiently translated into Cutting Planes?

- **Yes!** Provided the coefficients of the inequalities are not too large.

Cutting Planes vs. Stabbing Planes

Can **every** Stabbing Planes proof be efficiently translated into Cutting Planes?

- **Yes!** Provided the coefficients of the inequalities are not too large.

Theorem [FGI+ 22]. Let $P \subseteq \mathbf{R}^n$ be a polytope, and suppose that there is a Stabbing Planes refutation of P with size s and where every coefficient has magnitude at most c. Then there is a Cutting Planes refutation of P of size

$$s(cd(P)\sqrt{n})^{\log s}$$

where $d(P)$ is the diameter of P.

Cutting Planes vs. Stabbing Planes

Can **every** Stabbing Planes proof be efficiently translated into Cutting Planes?

- **Yes!** Provided the coefficients of the inequalities are not too large.

Theorem [FGI+ 22]. Let $P \subseteq \mathbf{R}^n$ be a polytope, and suppose that there is a Stabbing Planes refutation of P with size s and where every coefficient has magnitude at most c. Then there is a Cutting Planes refutation of P of size

$$s(cd(P)\sqrt{n})^{\log s}$$

where $d(P)$ is the diameter of P.

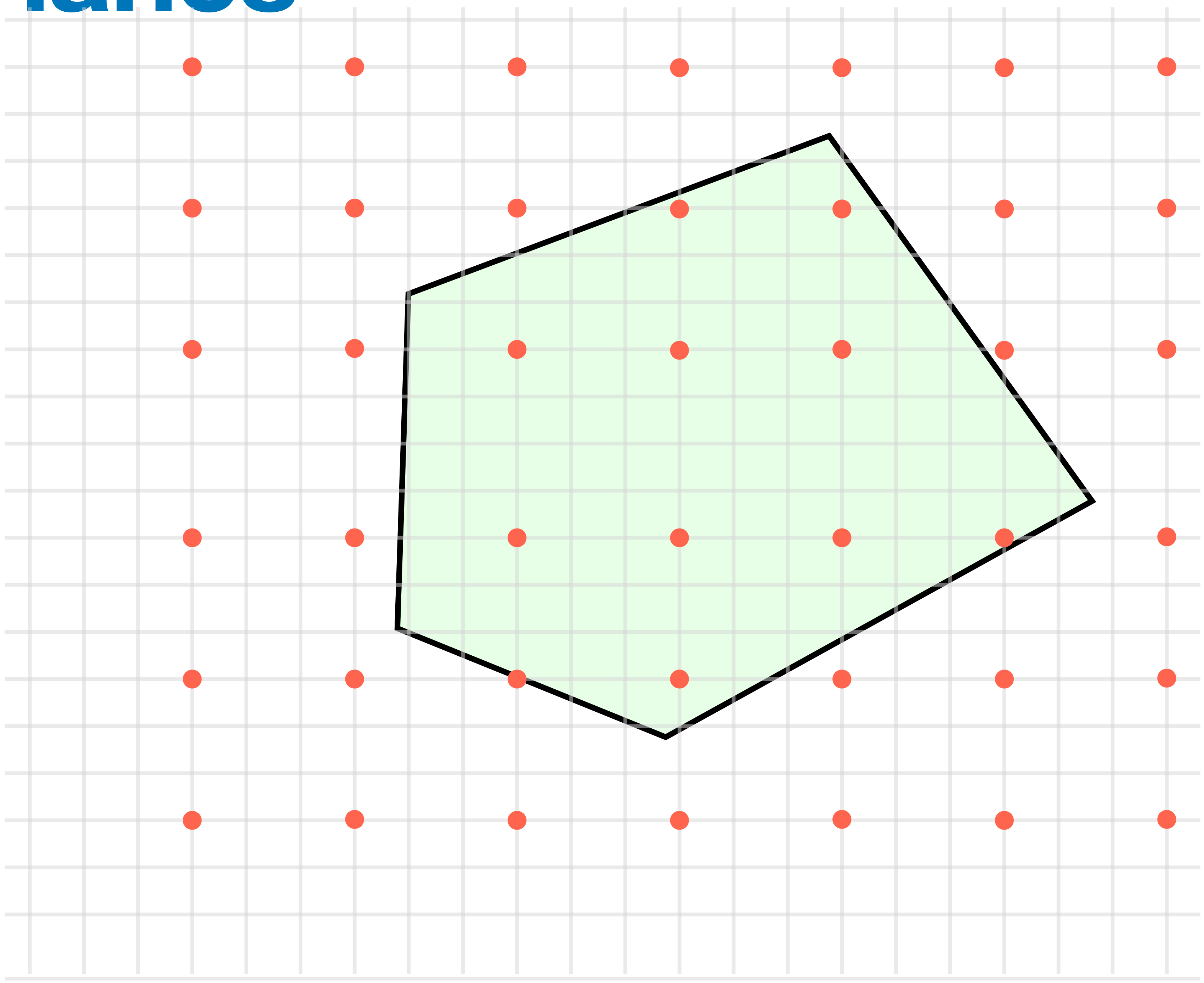
Corollary: Applying existing lower bounds for Cutting Planes proofs [P93, HP17, FPPR17]:

- The clique-colour formulas requires exponential size bounded-coefficient SP proofs.
- Random $\Theta(\log n)$-CNF formulas require exponential size bounded-coefficient SP proofs.

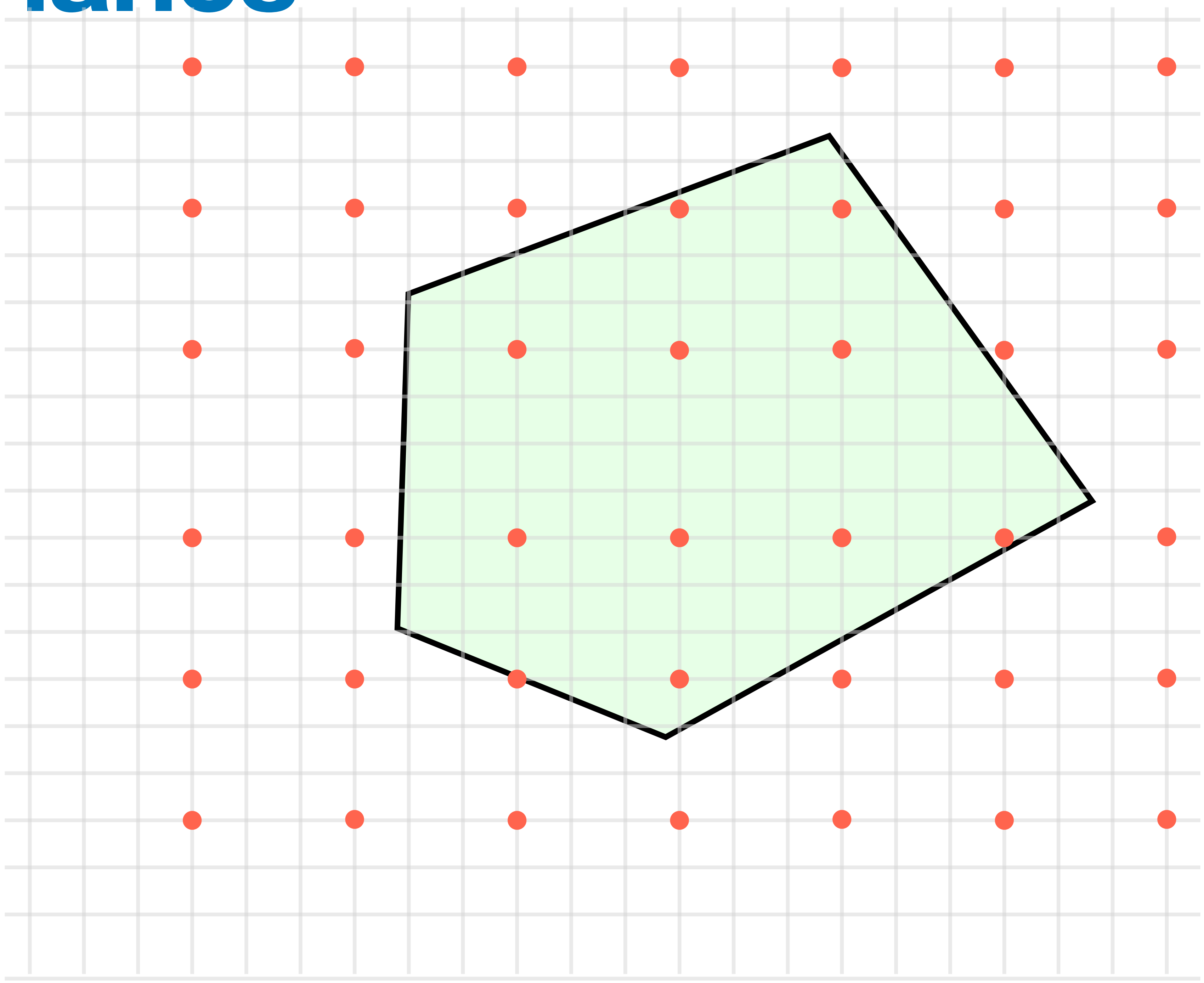
Proof Idea: Stabbing Planes* → Cutting Planes

Two steps

1. CP = Pathlike SP



Proof Idea: Stabbing Planes* → Cutting Planes



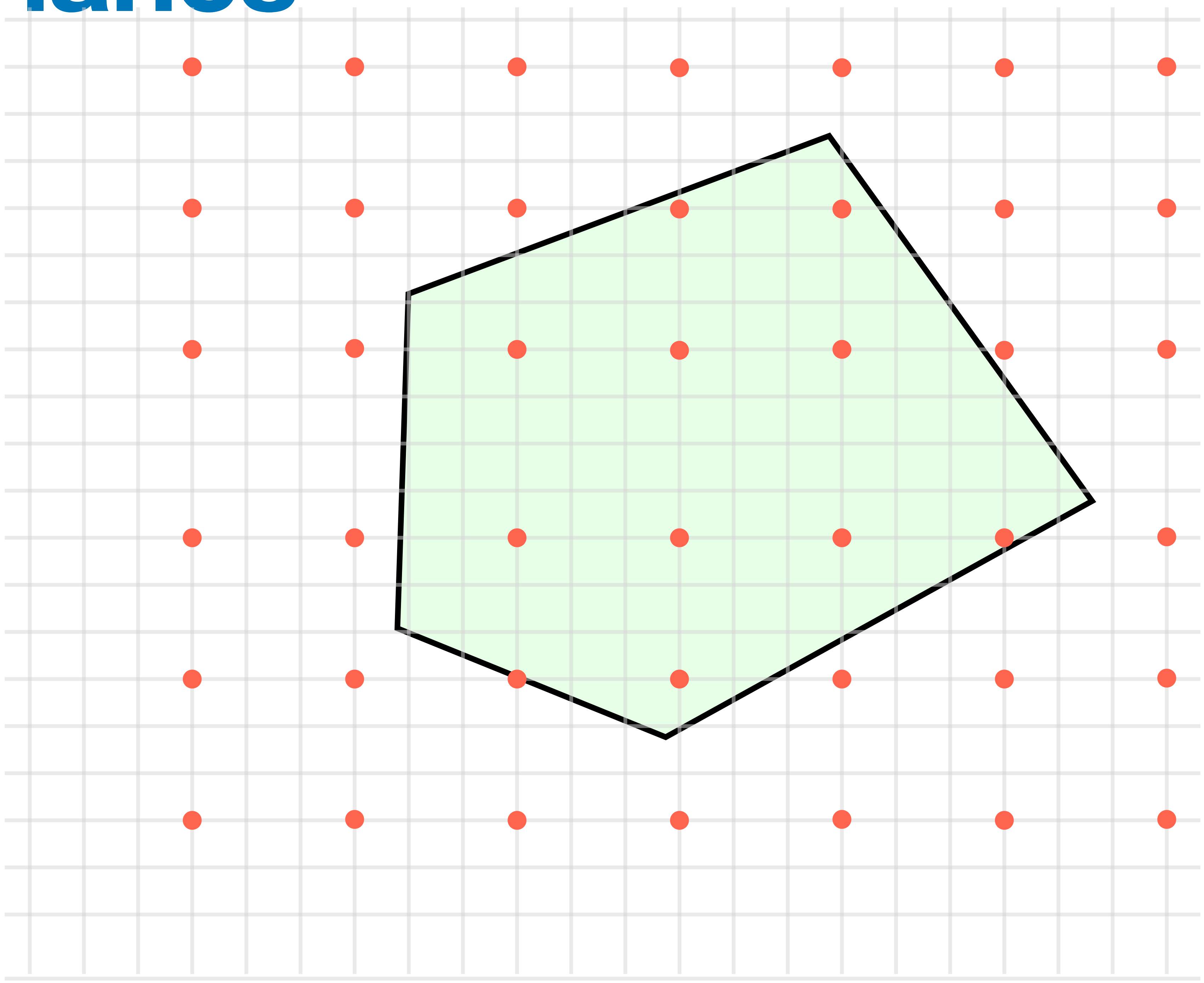
Two steps

1. CP = Pathlike SP = **Face**like SP.

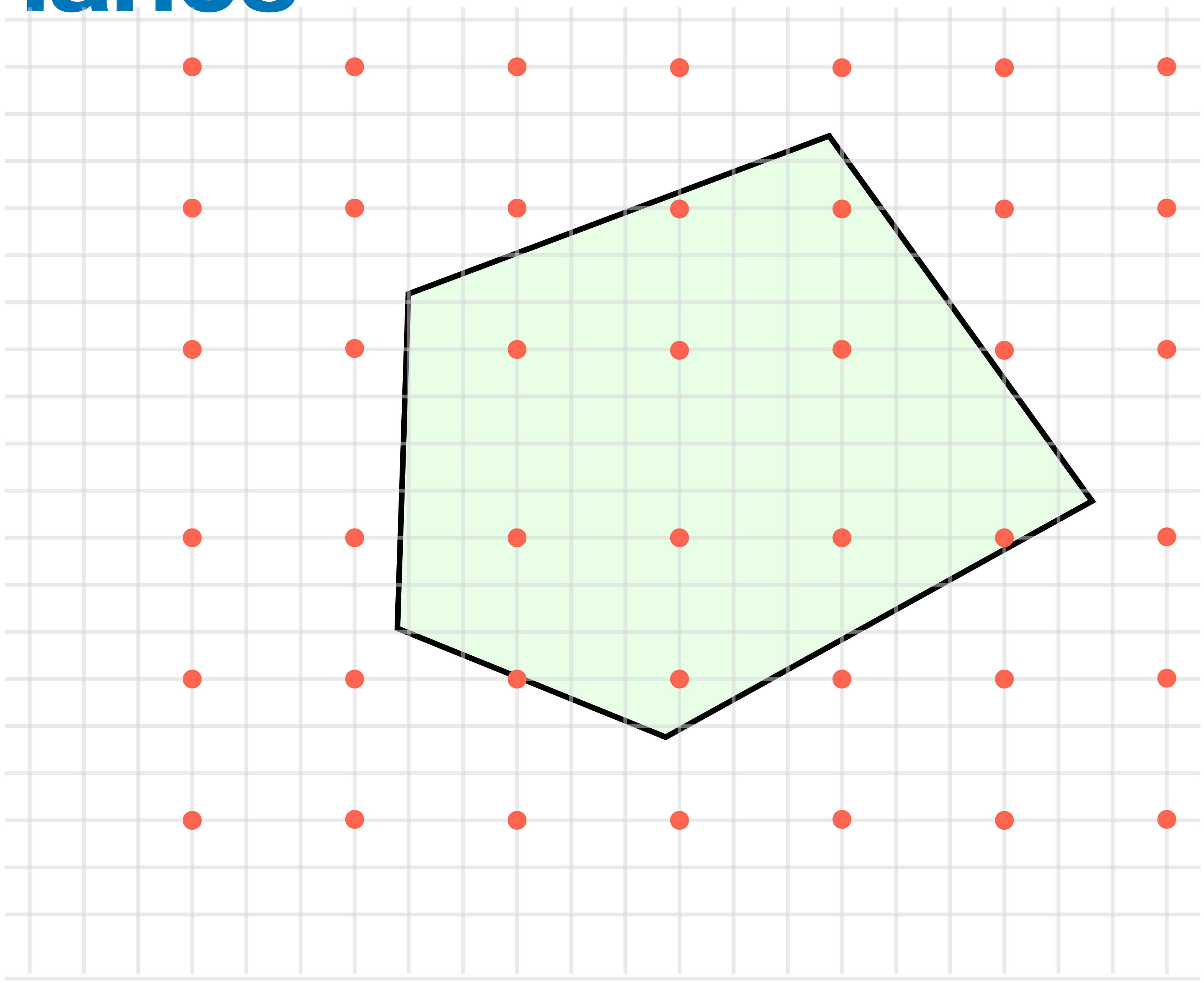
Proof Idea: Stabbing Planes* → Cutting Planes

Two steps

1. CP = Pathlike SP = **Face**like SP.
2. Bounded-coefficient SP proofs can be made Facelike.



Proof Idea: Stabbing Planes* $\longrightarrow$ Cutting Planes



Two steps

1. CP = Pathlike SP = **Face**like SP.
2. Bounded-coefficient SP proofs can be made Facelike.

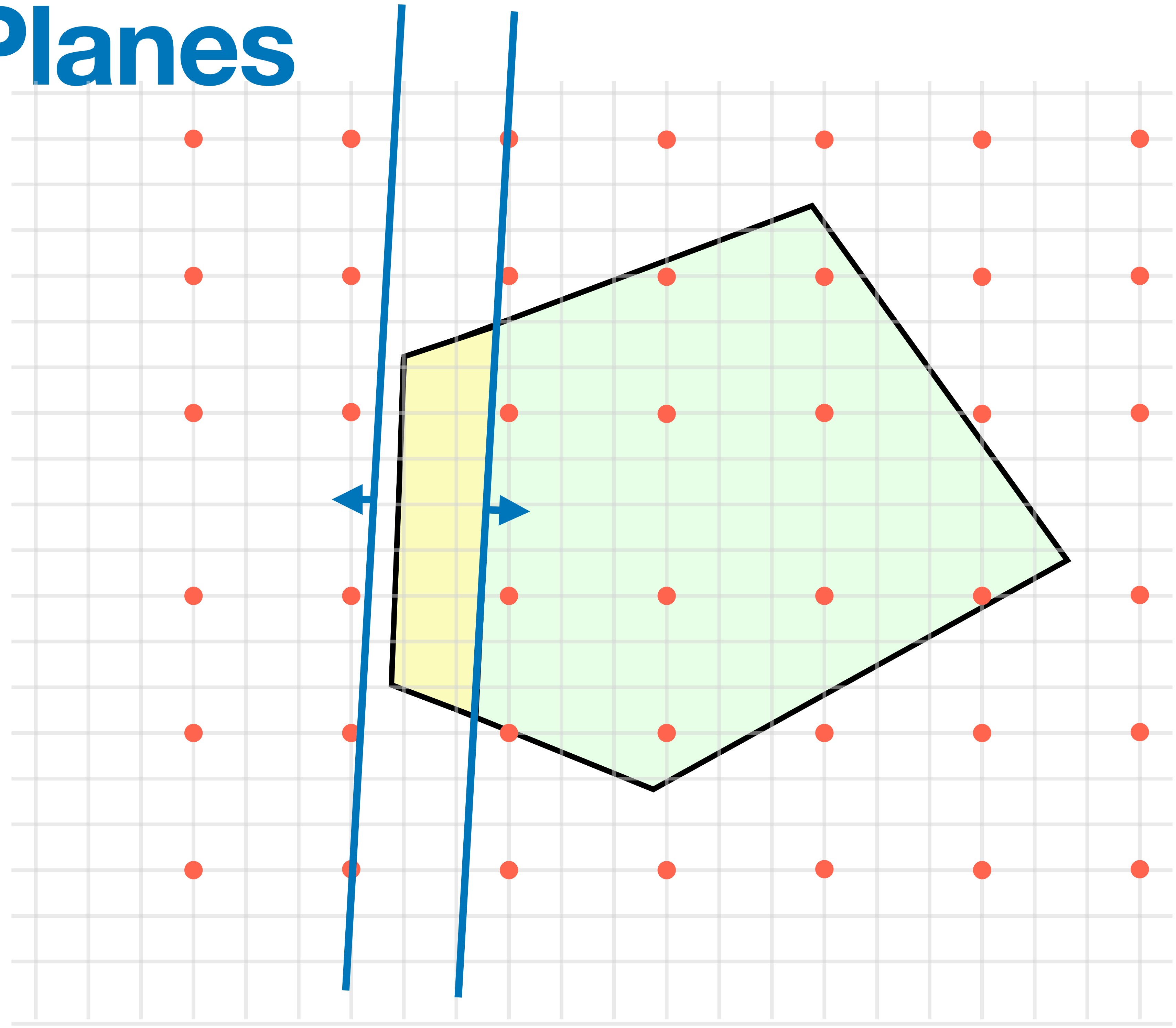
A Stabbing Planes query $ax \leq b, ax \geq b+1$ is **facelike** if at least one of $P_u \cap \{ax \leq b\}$ or $P_u \cap \{ax \geq b+1\}$ is a face of P_u.

Proof Idea: Stabbing Planes* $\longrightarrow$ Cutting Planes

Two steps

1. CP = Pathlike SP = **Face**like SP.
2. Bounded-coefficient SP proofs can be made Facelike.

A Stabbing Planes query $ax \leq b, ax \geq b+1$ is **facelike** if at least one of $P_u \cap \{ax \leq b\}$ or $P_u \cap \{ax \geq b+1\}$ is a face of P_u.



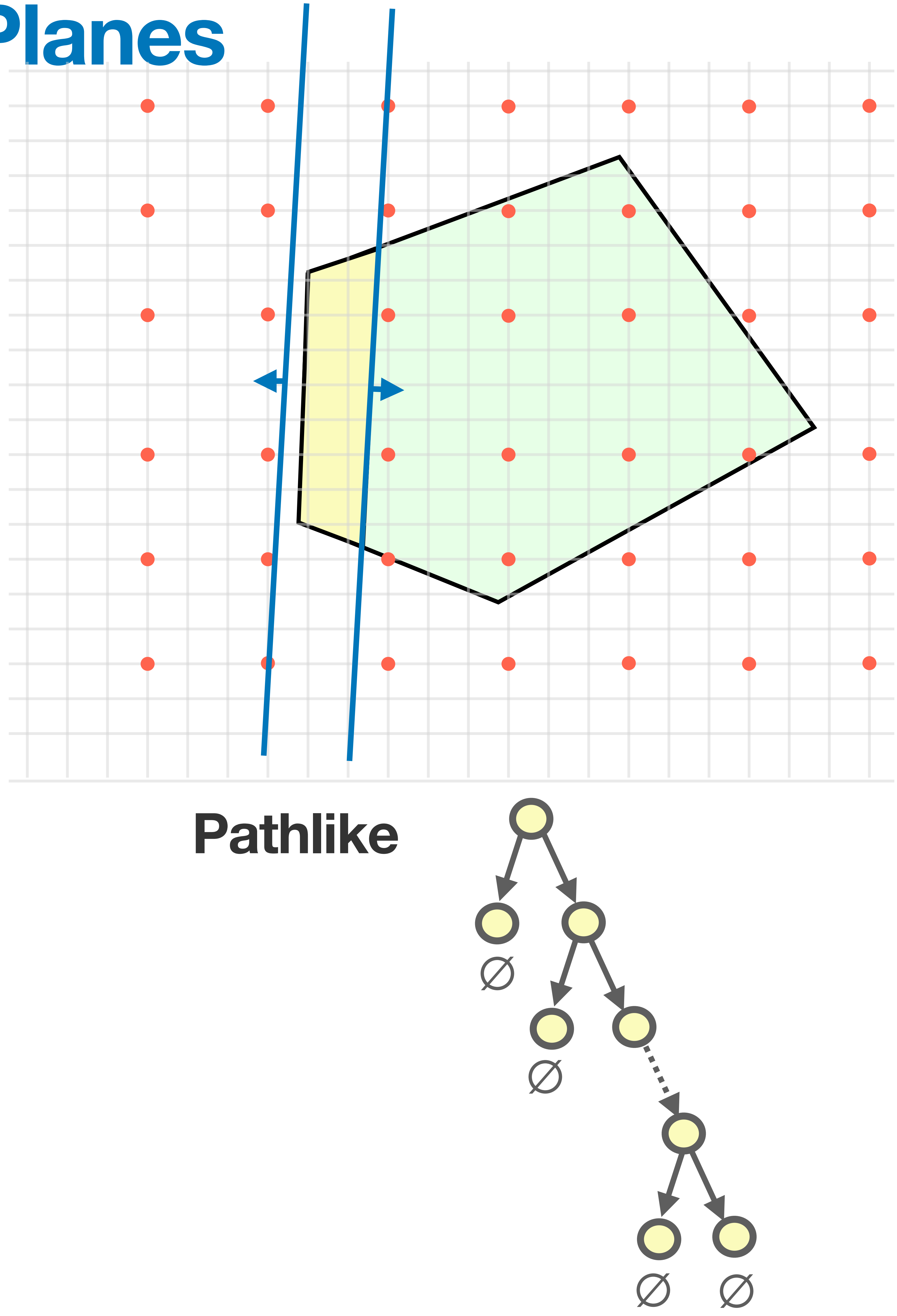
Pathlike

Proof Idea: Stabbing Planes* → Cutting Planes

Two steps

1. CP = Pathlike SP = **Face**like SP.
2. Bounded-coefficient SP proofs can be made Facelike.

A Stabbing Planes query $ax \leq b, ax \geq b + 1$ is **facelike** if at least one of $P_u \cap \{ax \leq b\}$ or $P_u \cap \{ax \geq b + 1\}$ is a face of P_u.



Proof Idea: Stabbing Planes* → Cutting Planes

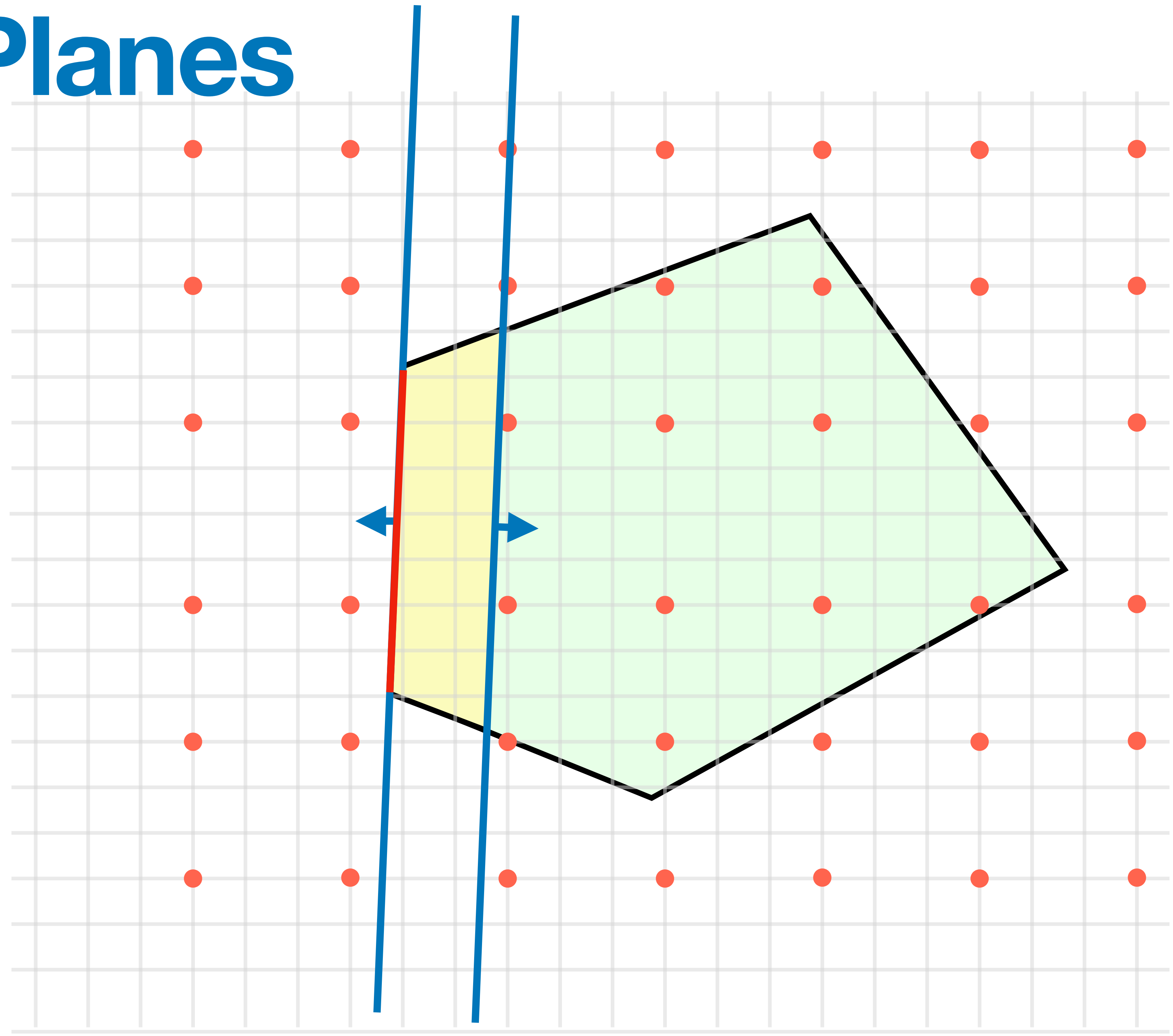
Two steps

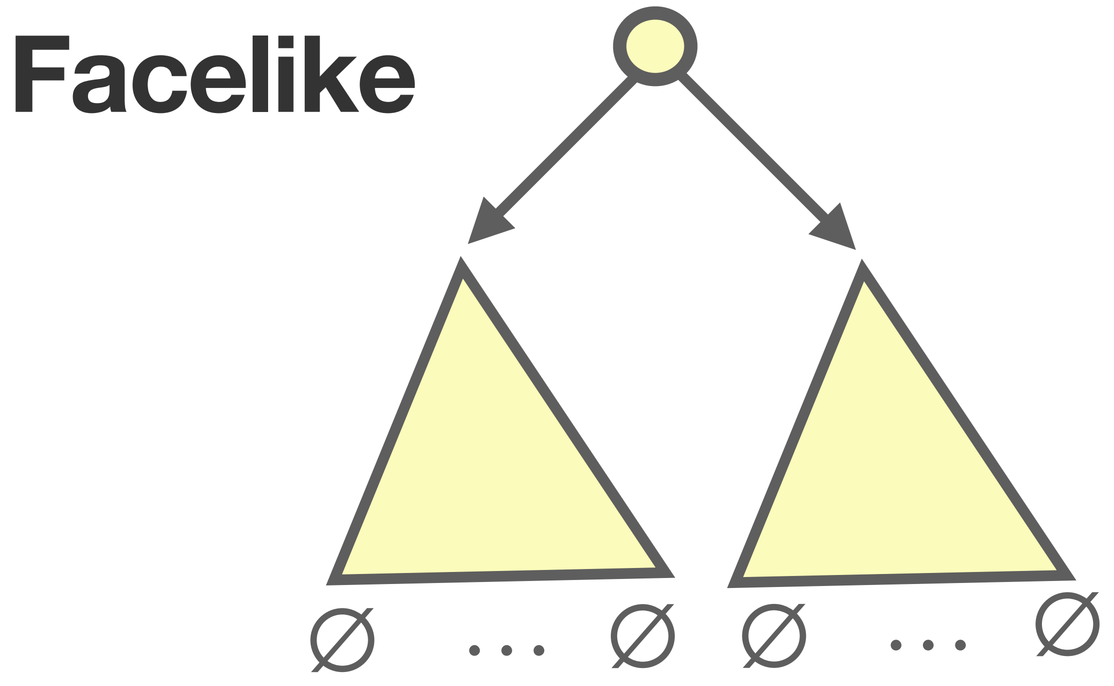
1. CP = Pathlike SP = **Face**like SP.
2. Bounded-coefficient SP proofs can be made Facelike.

A Stabbing Planes query $ax \leq b, ax \geq b+1$ is **facelike** if at least one of $P_u \cap \{ax \leq b\}$ or $P_u \cap \{ax \geq b+1\}$ is a face of P_u.

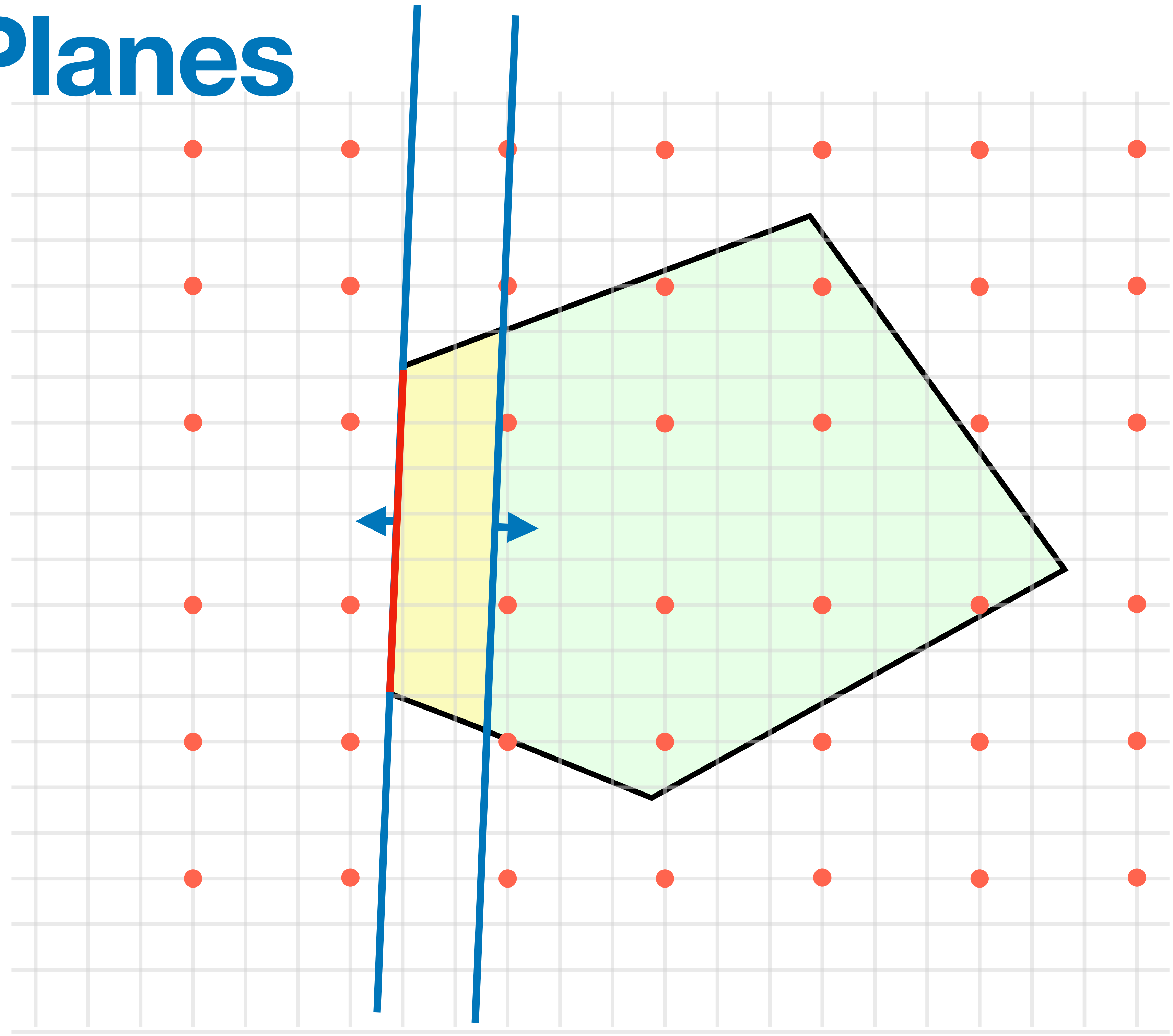
A Stabbing Planes proof is **facelike** if every query is facelike.

- In a facelike query, one child has **lower dimension**!





Proof Idea: Stabbing Planes* $\rightarrow$ Cutting Planes



Two steps

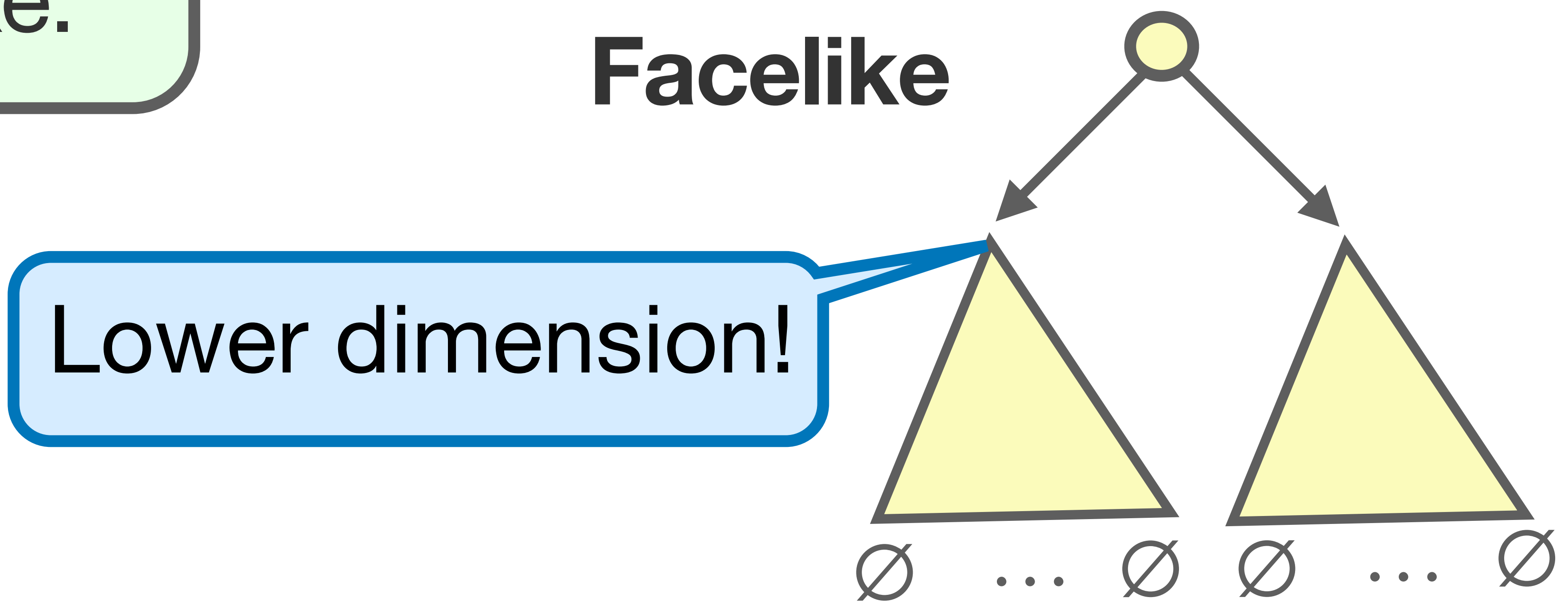
1. CP = Pathlike SP = **Face**like SP.
2. Bounded-coefficient SP proofs can be made Facelike.

A Stabbing Planes query $ax \leq b, ax \geq b + 1$ is **facelike** if at least one of $P_u \cap \{ax \leq b\}$ or $P_u \cap \{ax \geq b + 1\}$ is a face of P_u.

A Stabbing Planes proof is **facelike** if every query is facelike.

- In a facelike query, one child has **lower dimension**!

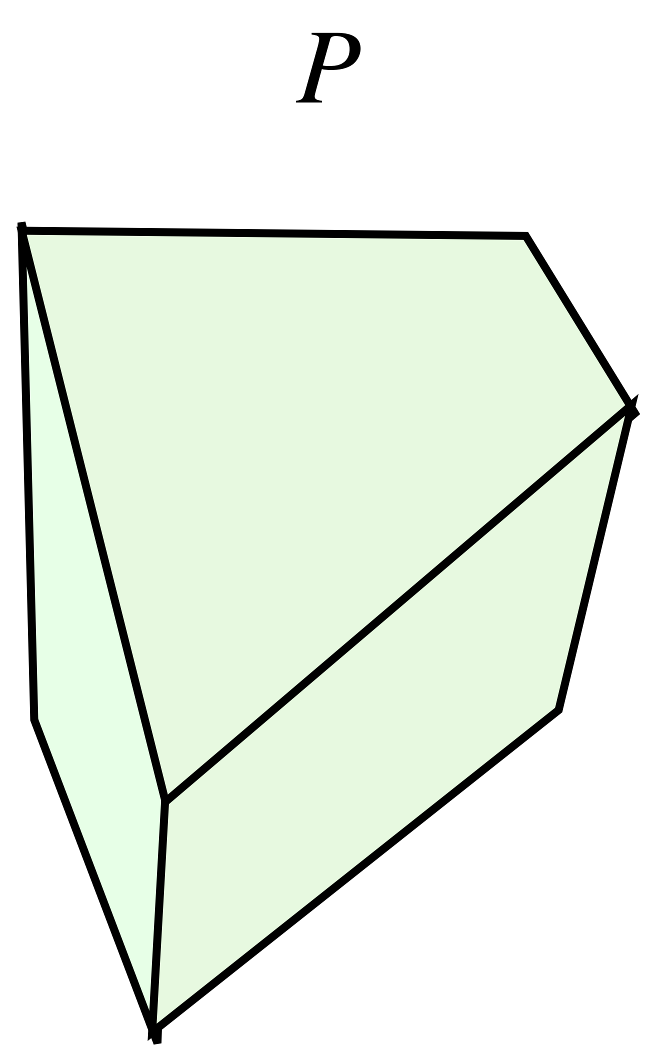
Facelike



Step 1. Facelike Stabbing Planes = Cutting Planes

Idea:

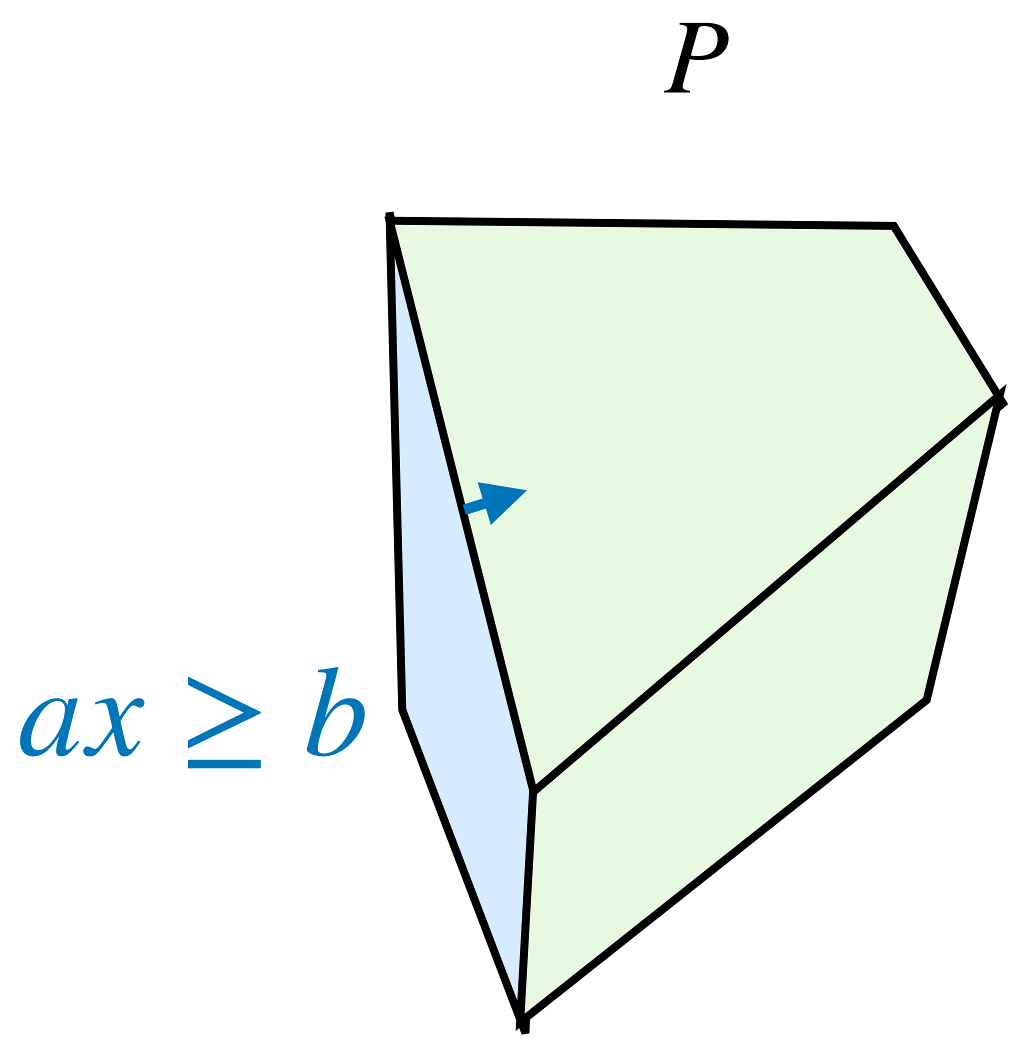
Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b+1\}$ from P of the same size!



Step 1. Facelike Stabbing Planes = Cutting Planes

Idea:

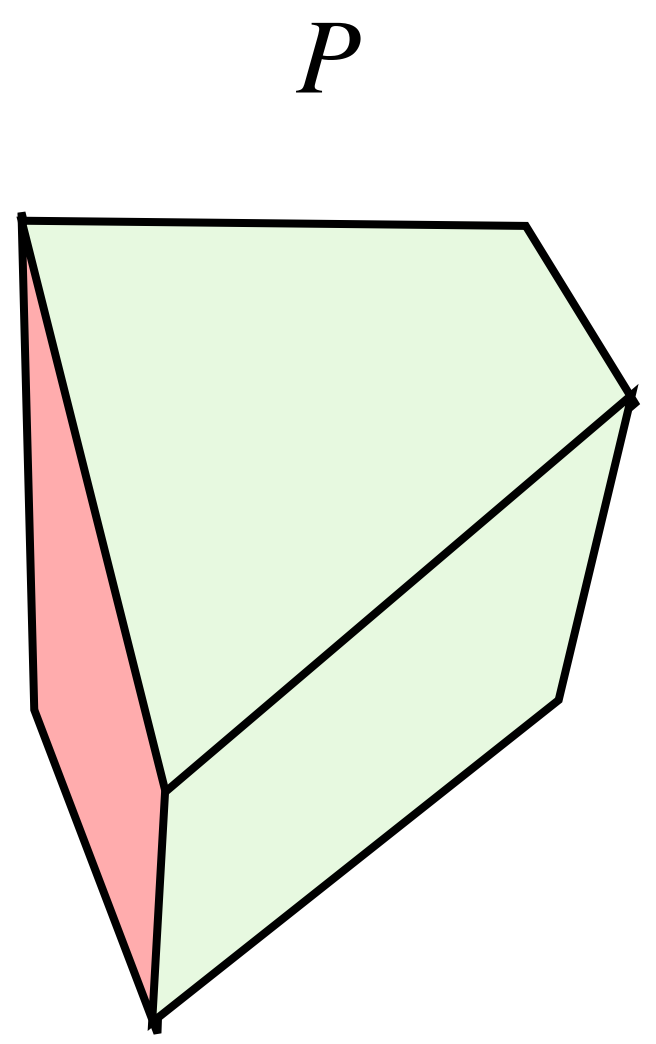
Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b+1\}$ from P of the same size!



Step 1. Facelike Stabbing Planes = Cutting Planes

Idea:

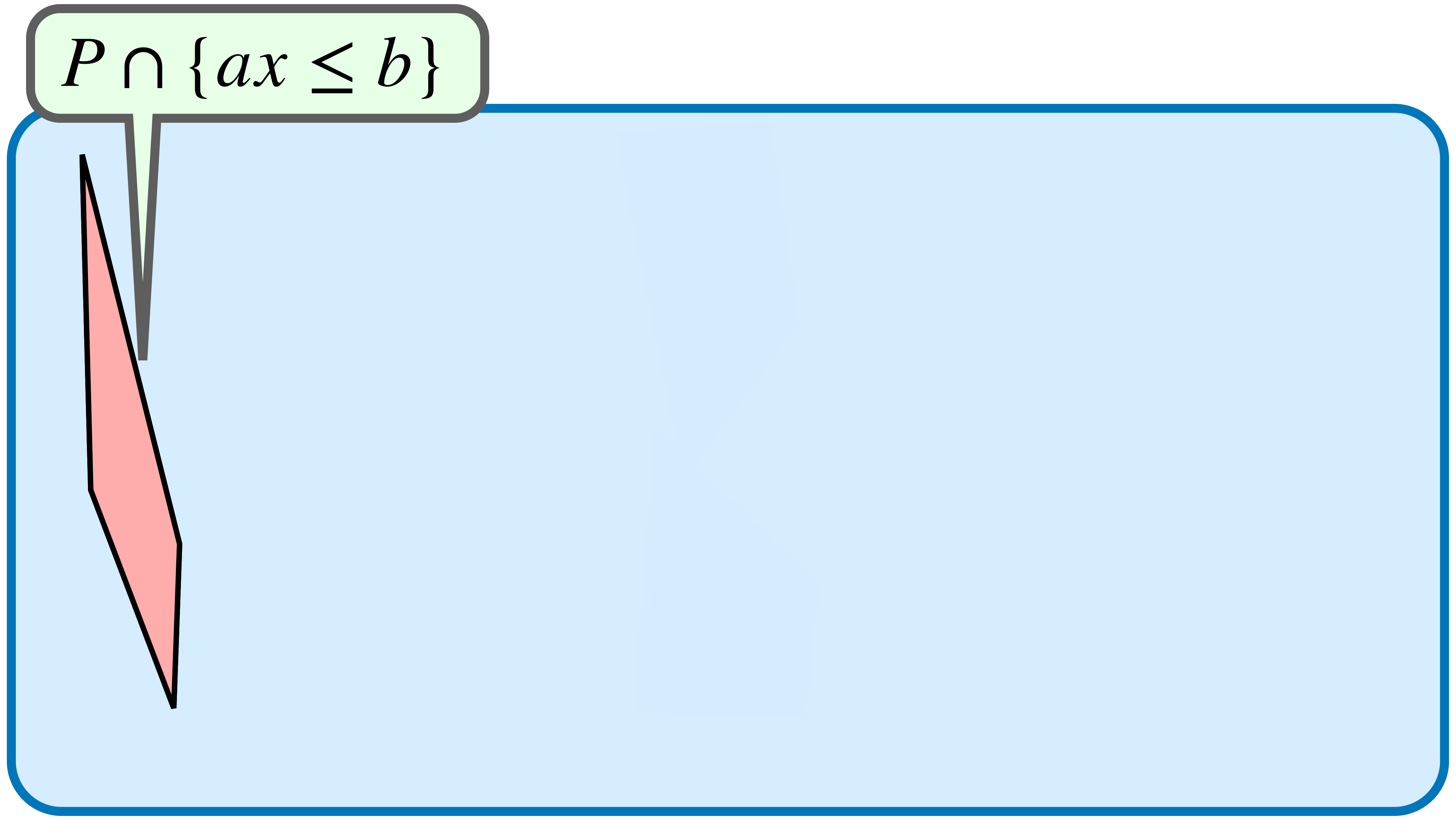
Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b + 1\}$ from P of the same size!

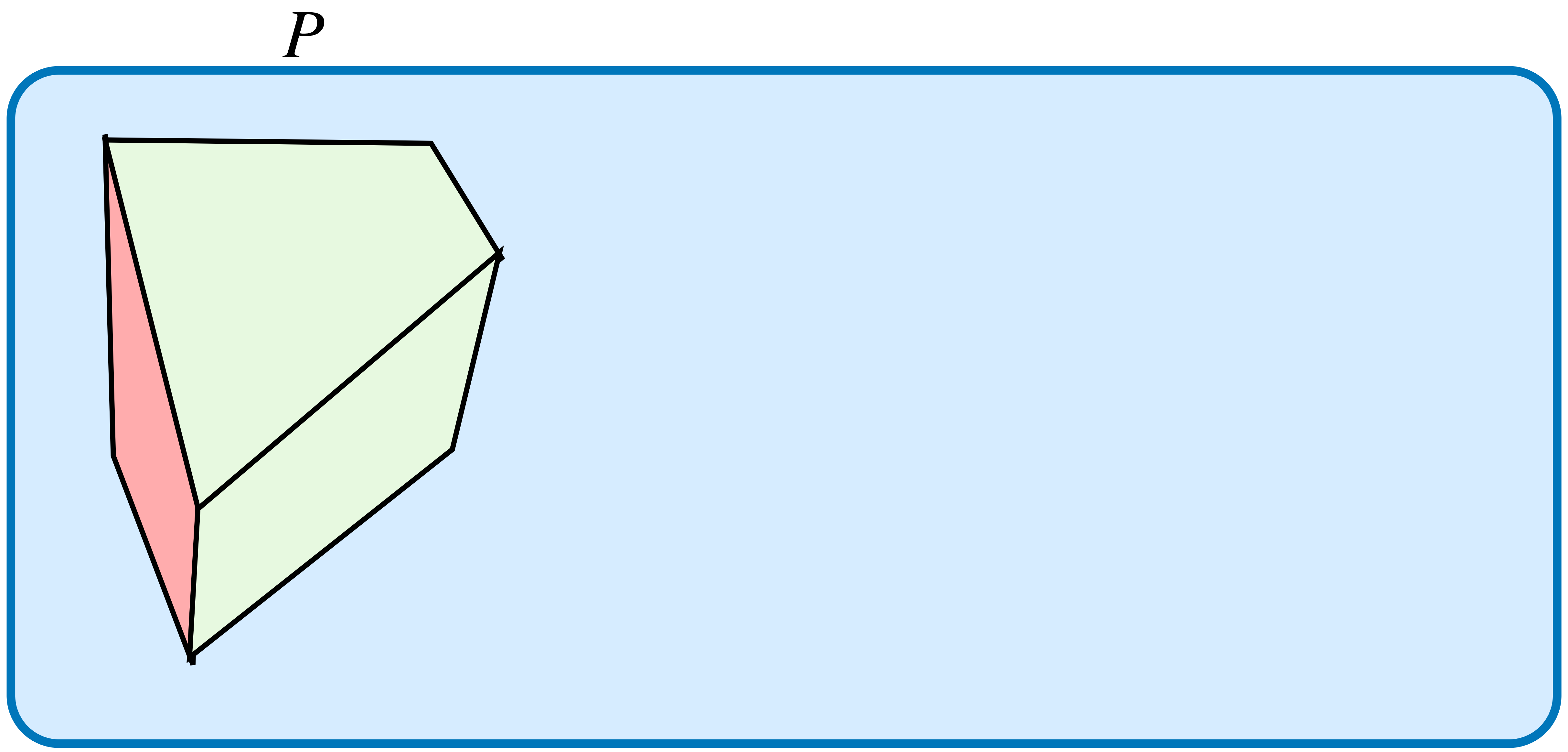


Step 1. Facelike Stabbing Planes = Cutting Planes

Idea:

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b+1\}$ from P of the same size!

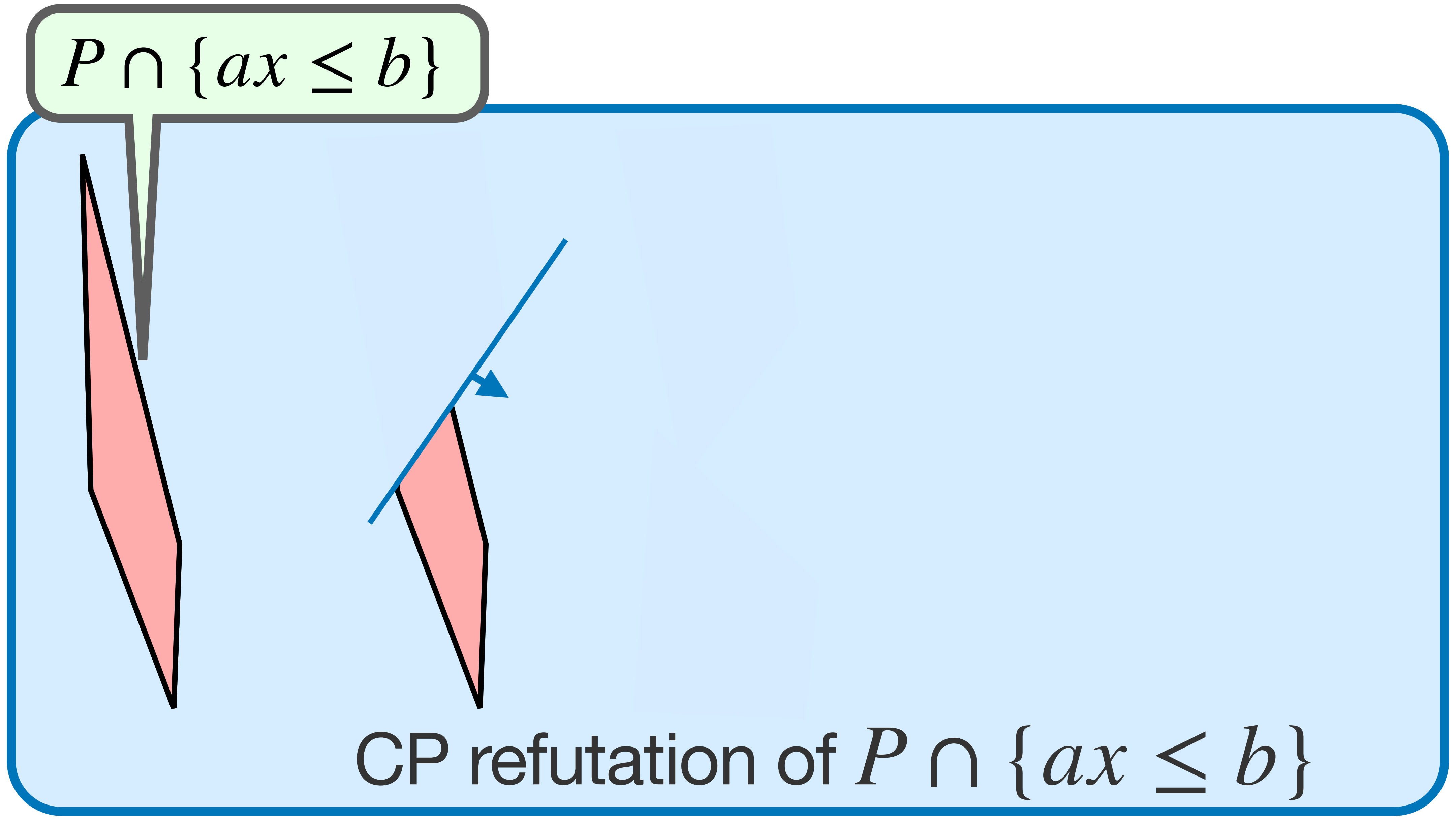


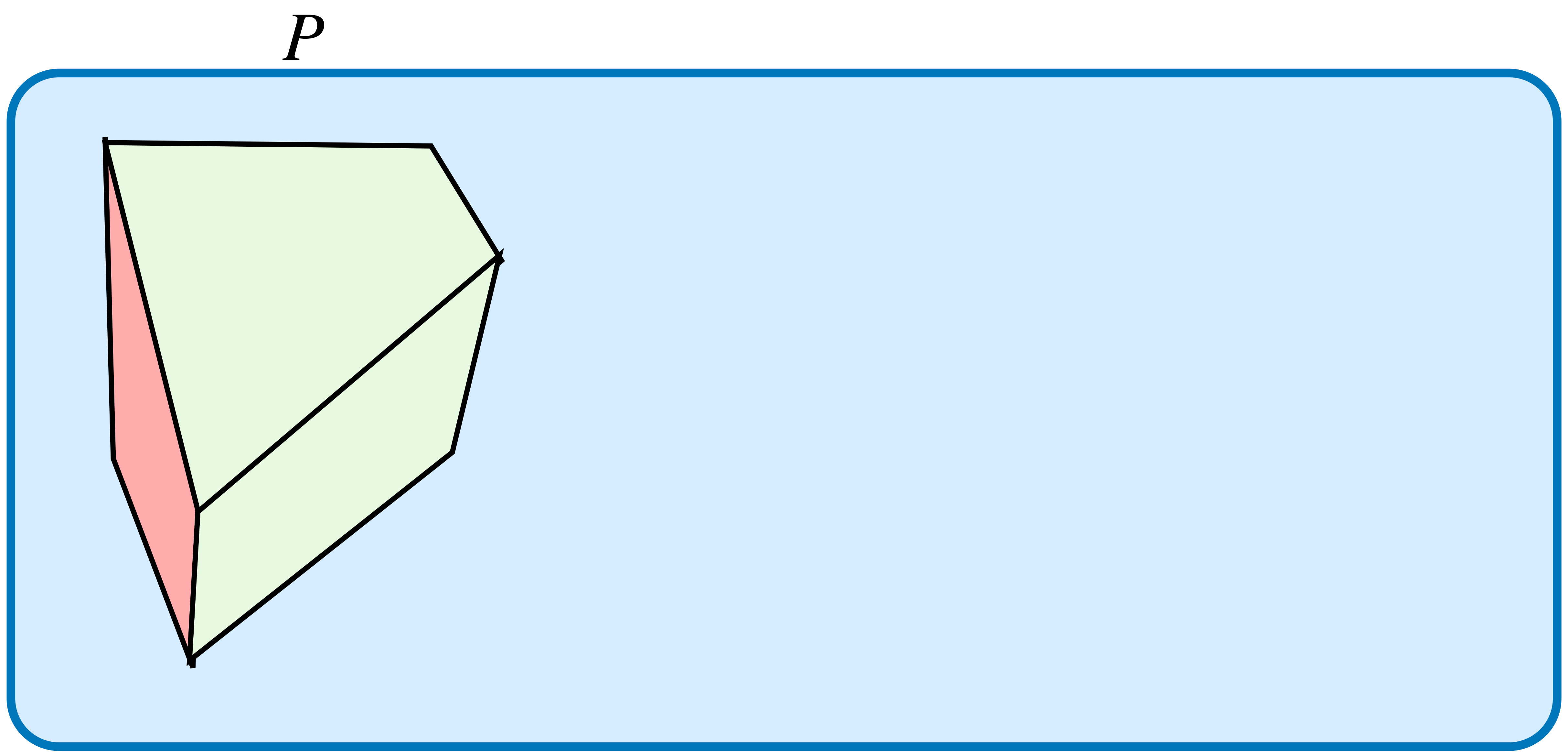


Step 1. Facelike Stabbing Planes = Cutting Planes

Idea:

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b+1\}$ from P of the same size!

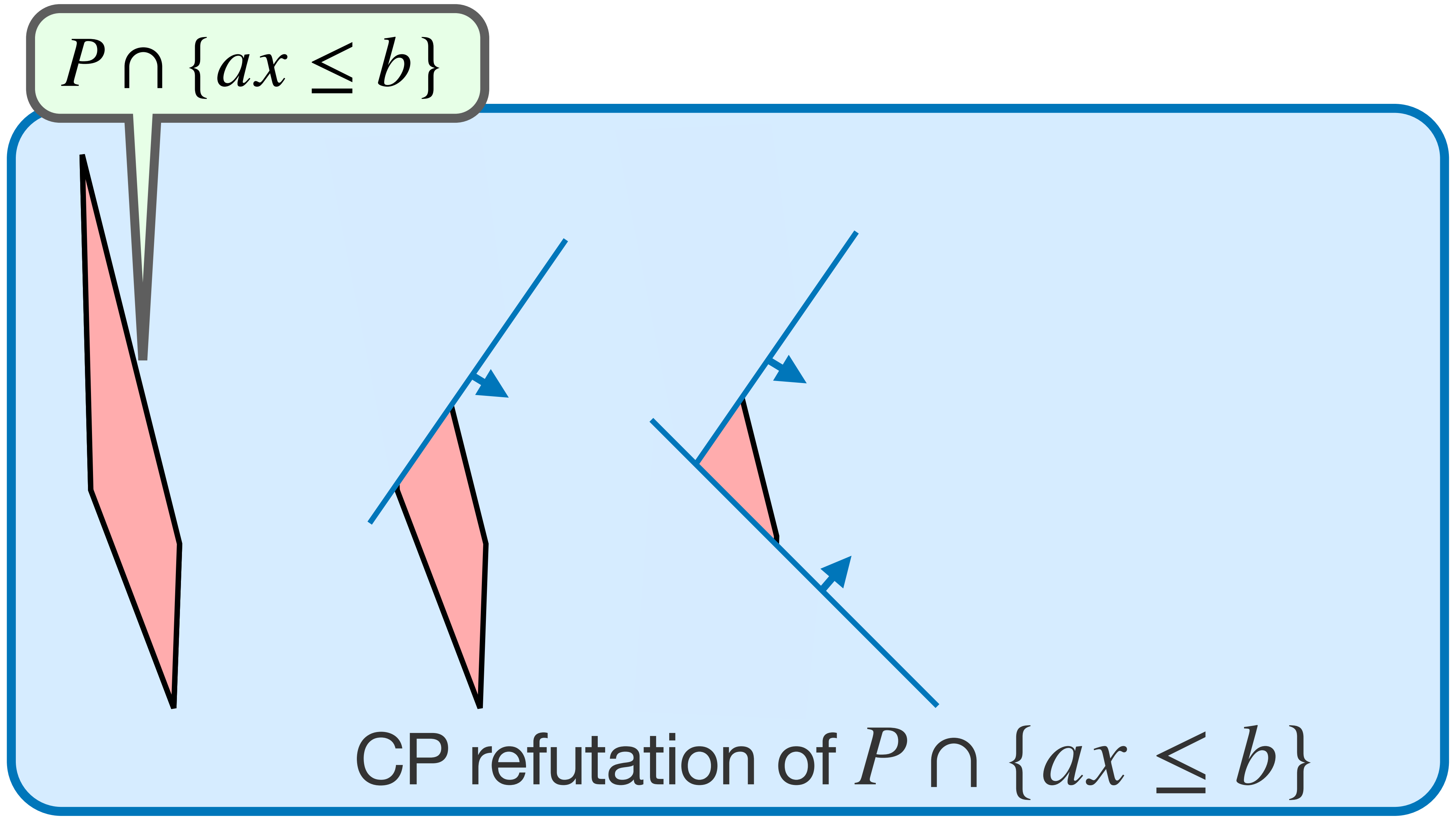


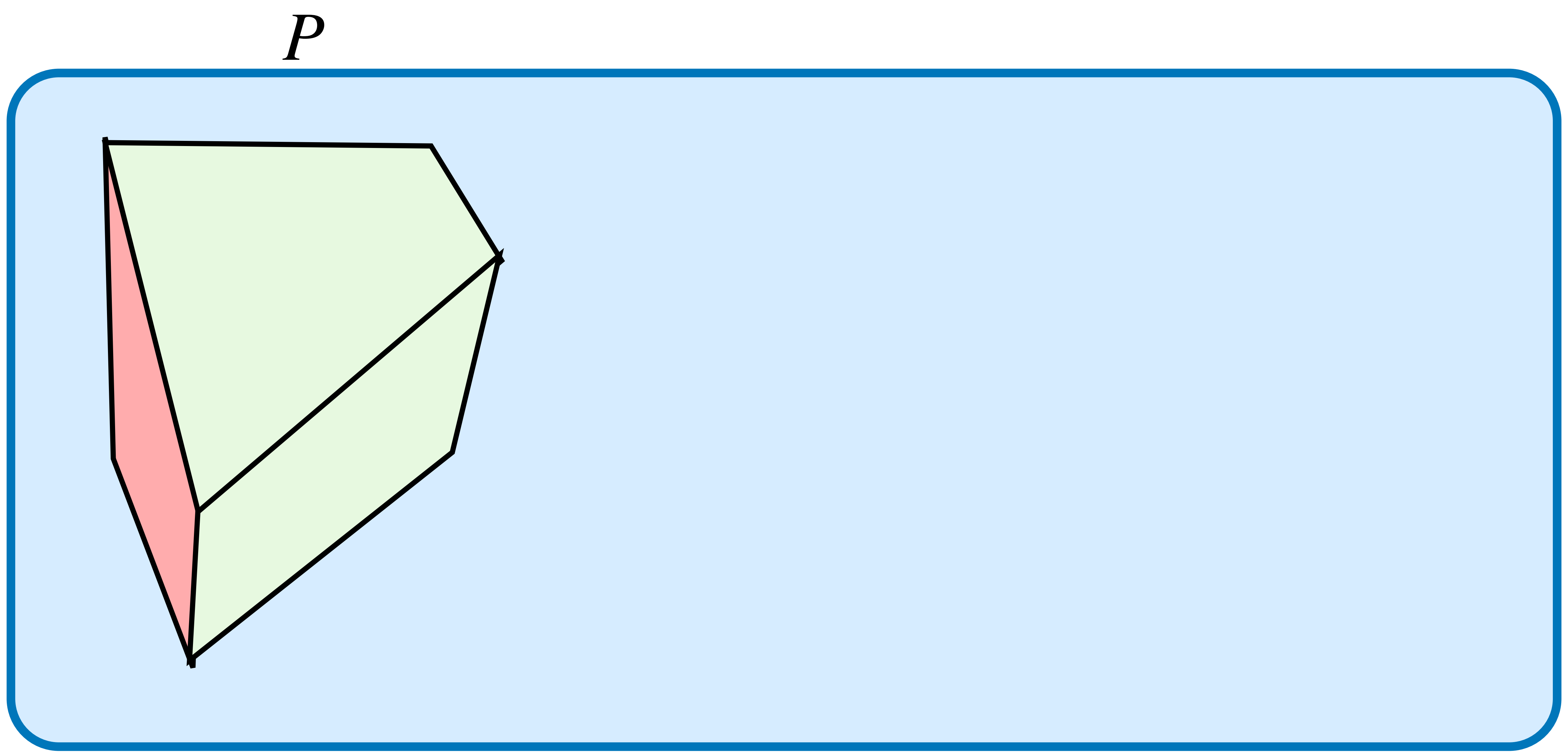


Step 1. Facelike Stabbing Planes = Cutting Planes

Idea:

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b+1\}$ from P of the same size!

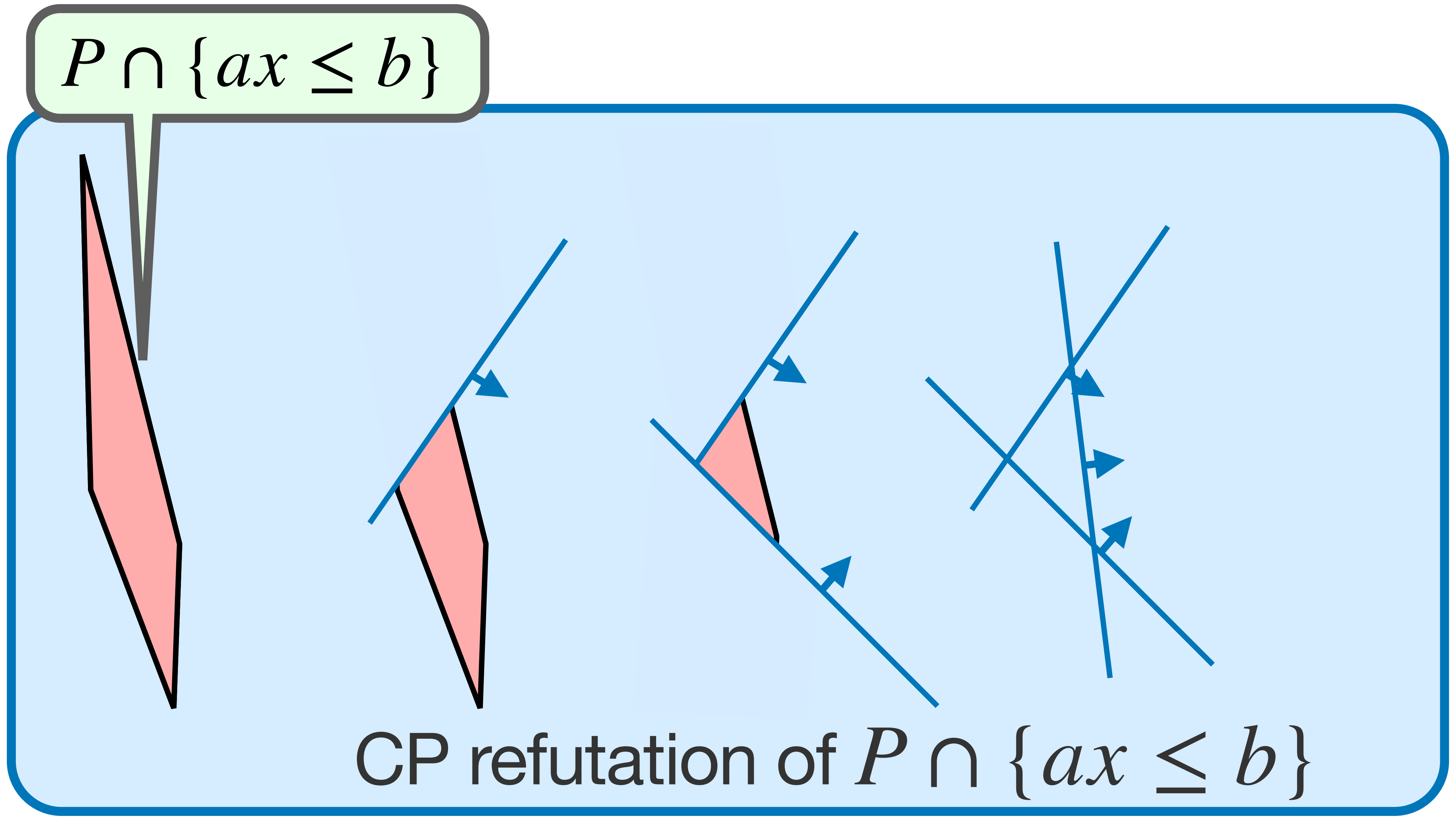


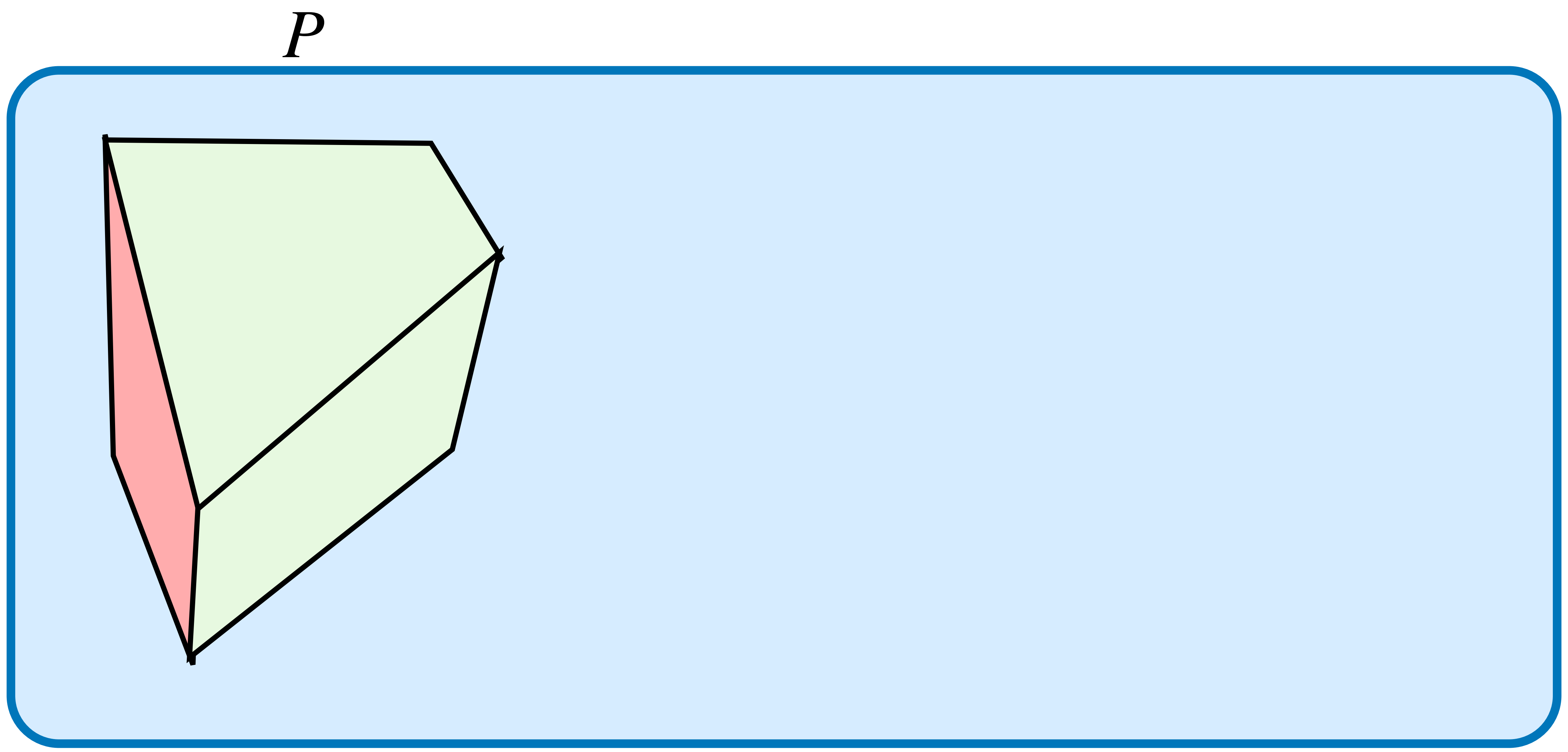


Step 1. Facelike Stabbing Planes = Cutting Planes

Idea:

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b+1\}$ from P of the same size!

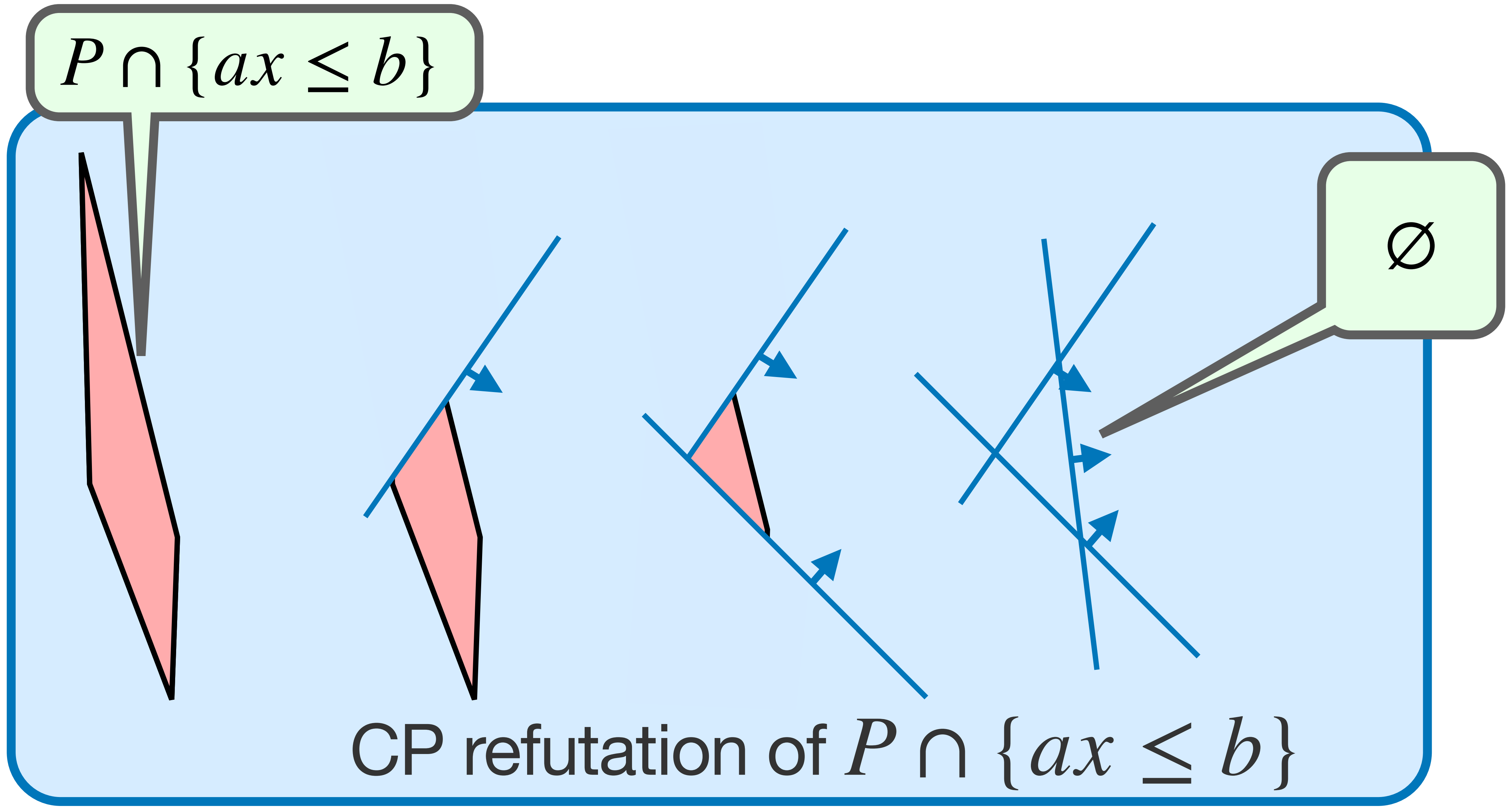


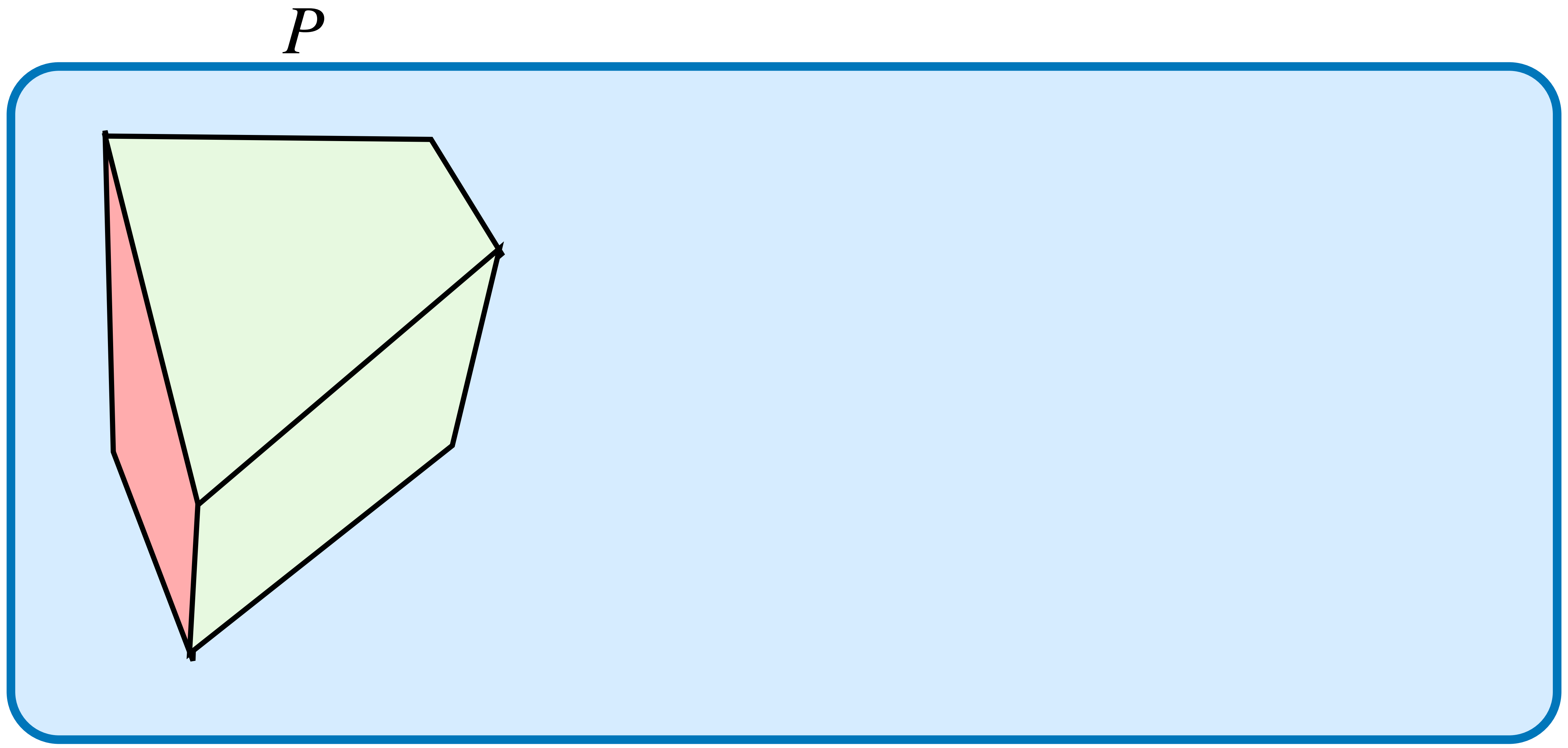


Step 1. Facelike Stabbing Planes = Cutting Planes

Idea:

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b+1\}$ from P of the same size!

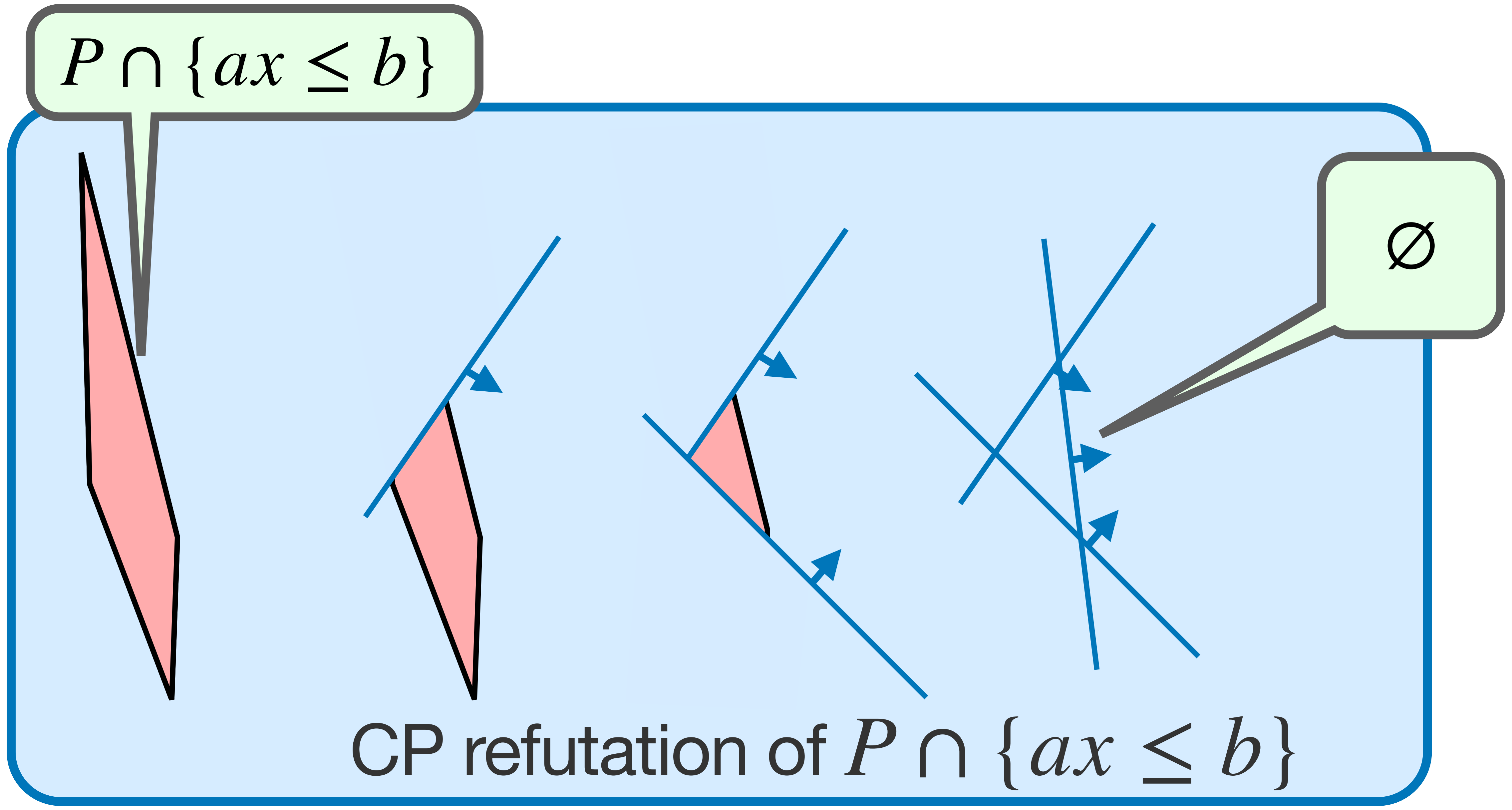


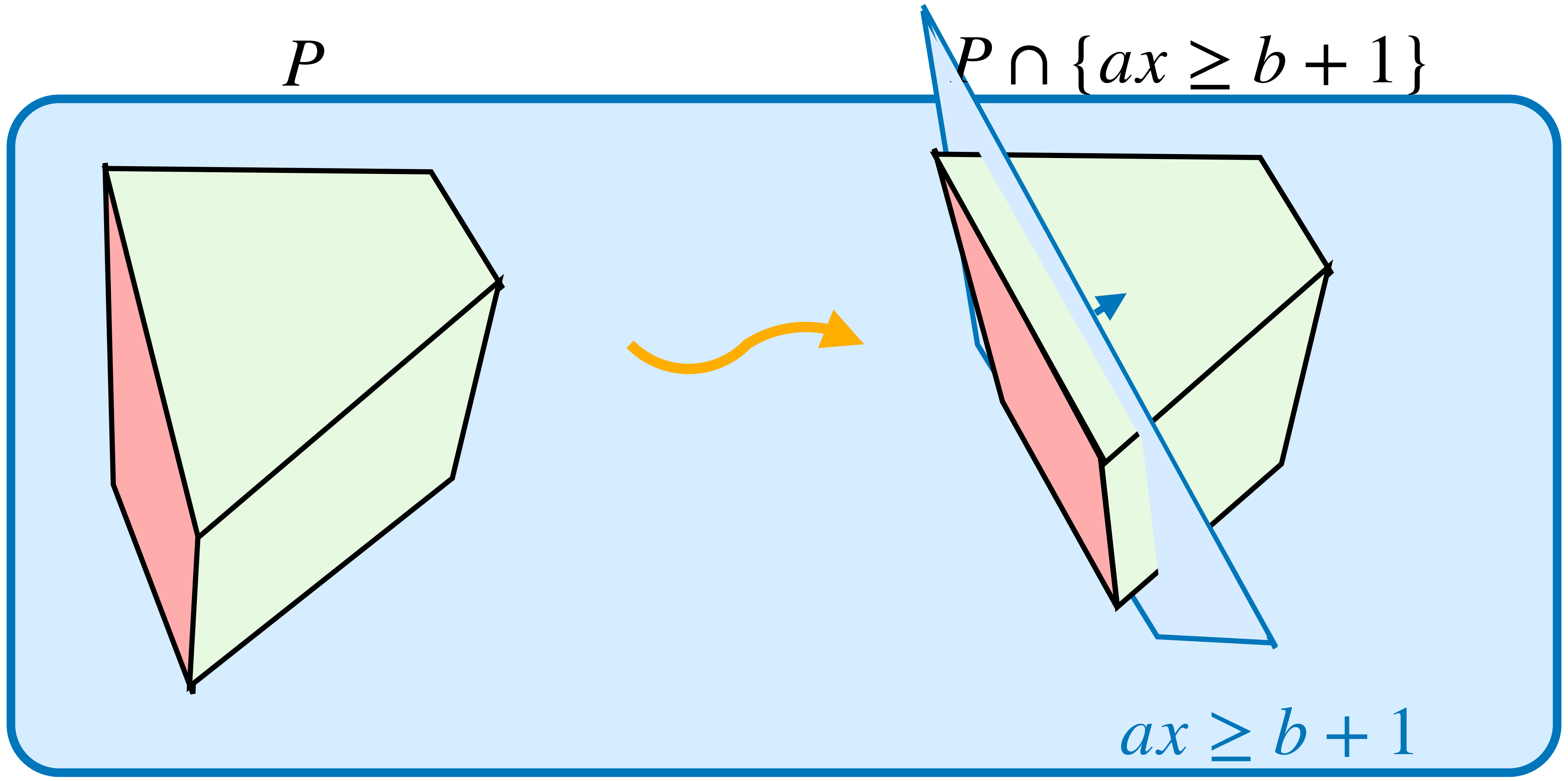


Step 1. Facelike Stabbing Planes = Cutting Planes

Idea:

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b+1\}$ from P of the same size!





Step 1. Facelike Stabbing Planes = Cutting Planes

Idea:

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \le b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \ge b+1\}$ from P of the same size!

Step 1. Facelike Stabbing Planes = Cutting Planes

Idea:

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b+1\}$ from P of the same size!

Theorem: Facelike Stabbing Planes = Cutting Planes.

Step 1. Facelike Stabbing Planes = Cutting Planes

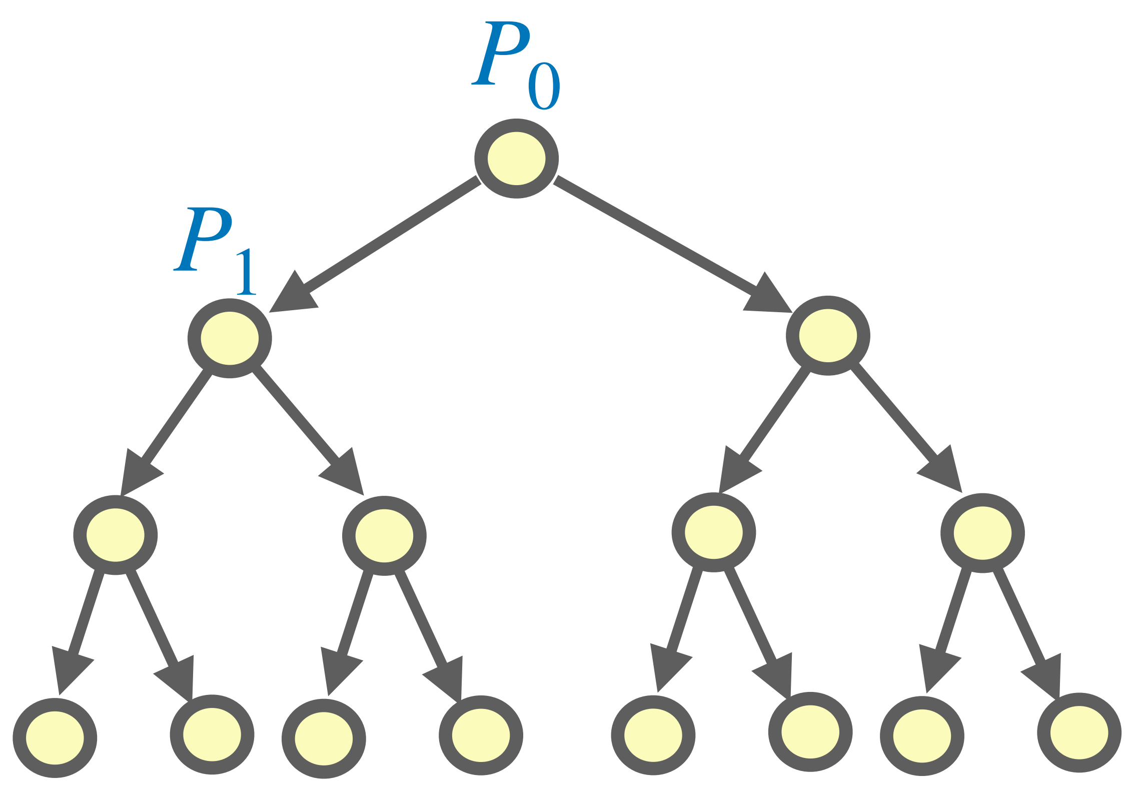
Idea:

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b + 1\}$ from P of the same size!

Theorem: Facelike Stabbing Planes = Cutting Planes.

Proof sketch.

- Fix a Facelike Stabbing Planes proof.



Step 1. Facelike Stabbing Planes = Cutting Planes

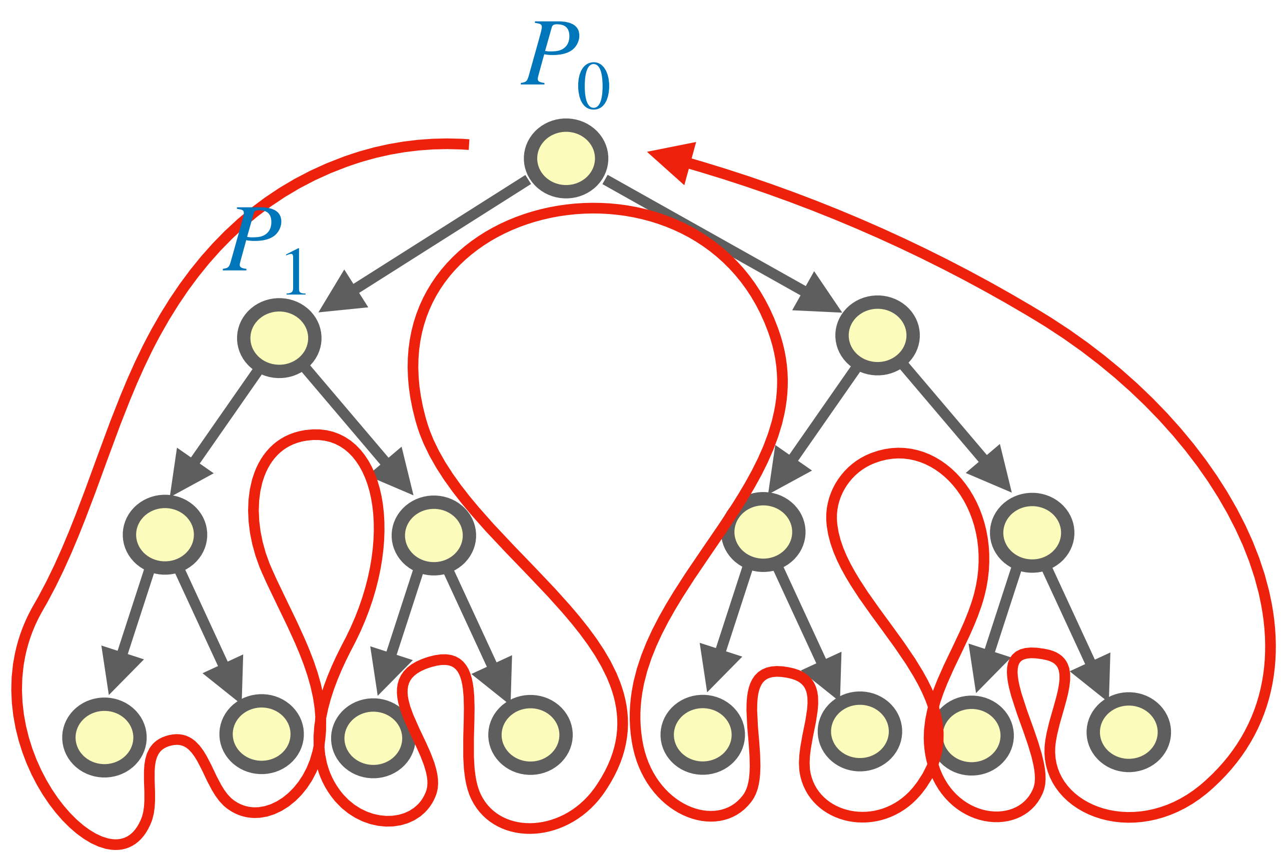
Idea:

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b+1\}$ from P of the same size!

Theorem: Facelike Stabbing Planes = Cutting Planes.

Proof sketch.

- Fix a Facelike Stabbing Planes proof.
- Take an **in-order** traversal, repeatedly applying the lemma.



Step 1. Facelike Stabbing Planes = Cutting Planes

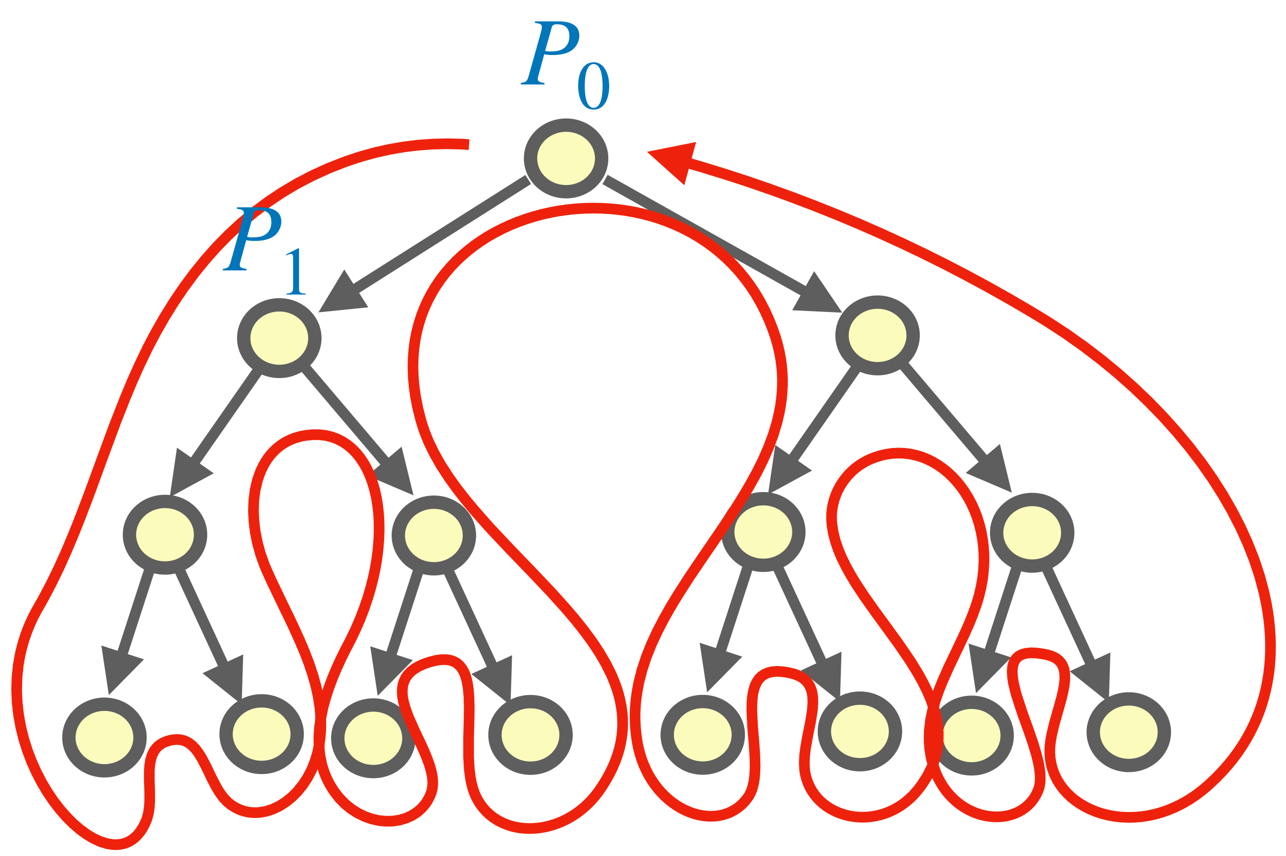
Idea:

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b+1\}$ from P of the same size!

Theorem: Facelike Stabbing Planes = Cutting Planes.

Proof sketch.

- Fix a Facelike Stabbing Planes proof.
- Take an **in-order** traversal, repeatedly applying the lemma.
  - Repeatedly lift refutations of faces to derivations using the lemma.



Step 1. Facelike Stabbing Planes = Cutting Planes

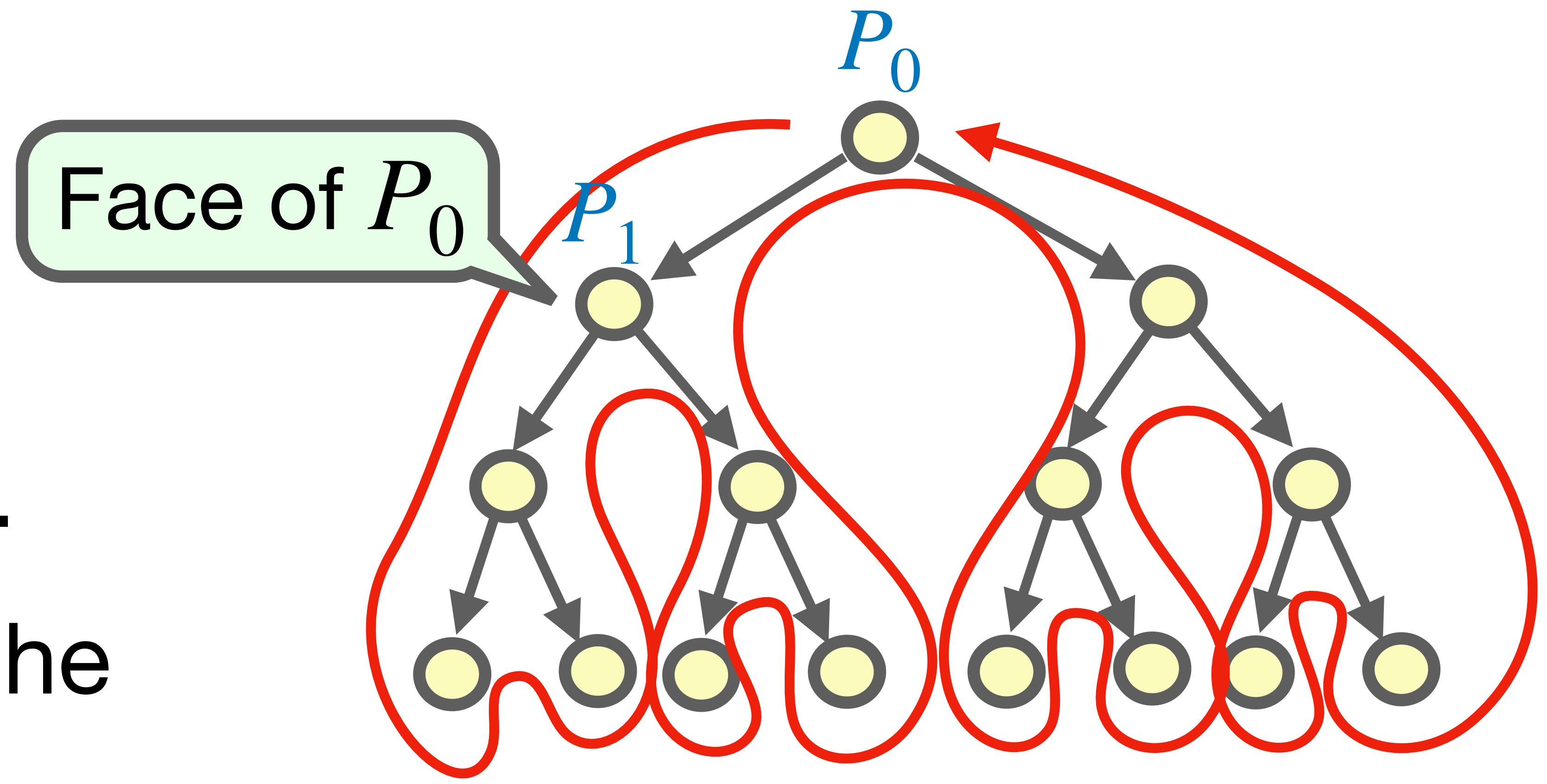
Idea:

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b+1\}$ from P of the same size!

Theorem: Facelike Stabbing Planes = Cutting Planes.

Proof sketch.

- Fix a Facelike Stabbing Planes proof.
- Take an **in-order** traversal, repeatedly applying the lemma.
  - Repeatedly lift refutations of faces to derivations using the lemma.



Step 1. Facelike Stabbing Planes = Cutting Planes

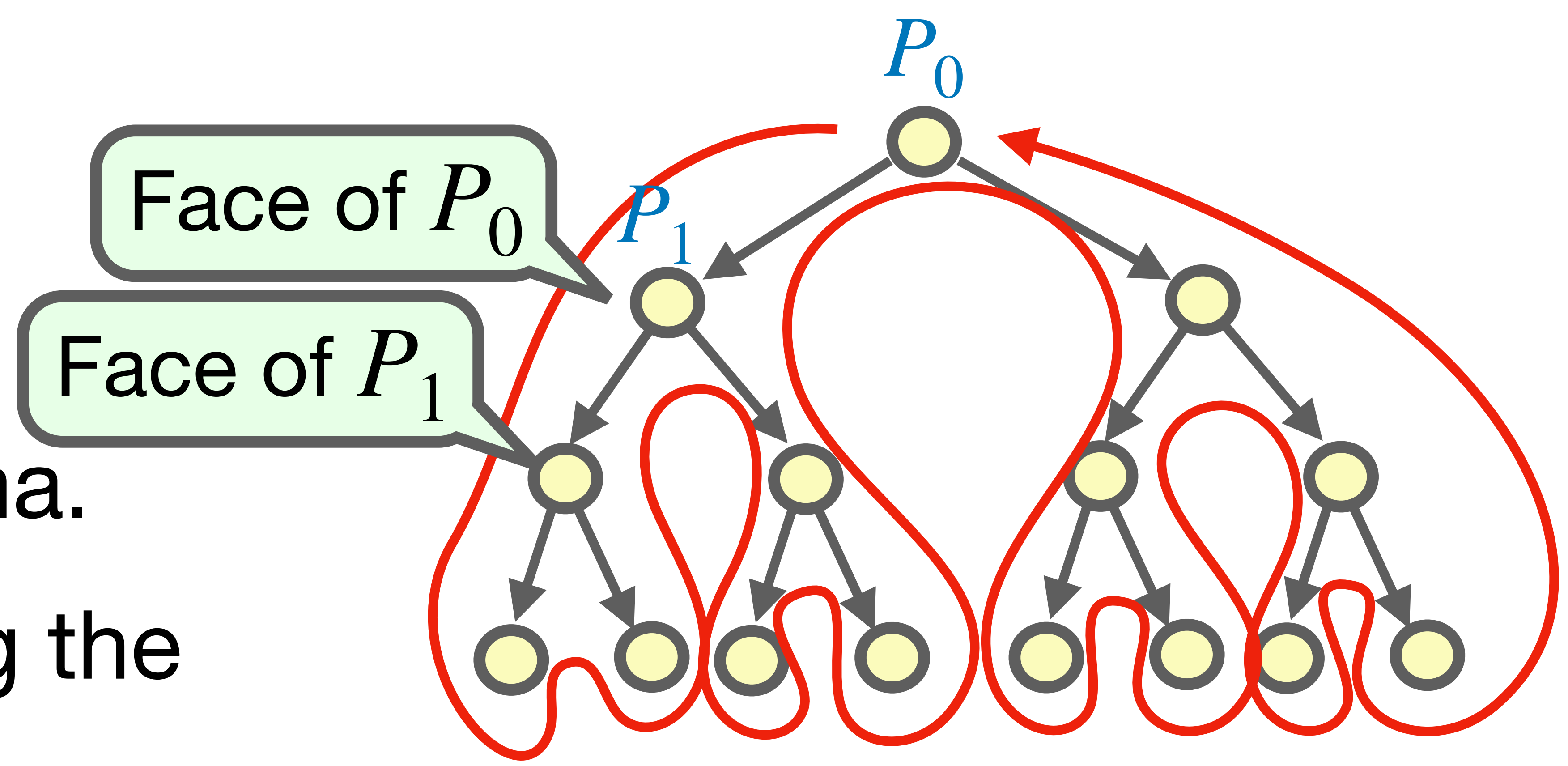
Idea:

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b+1\}$ from P of the same size!

Theorem: Facelike Stabbing Planes = Cutting Planes.

Proof sketch.

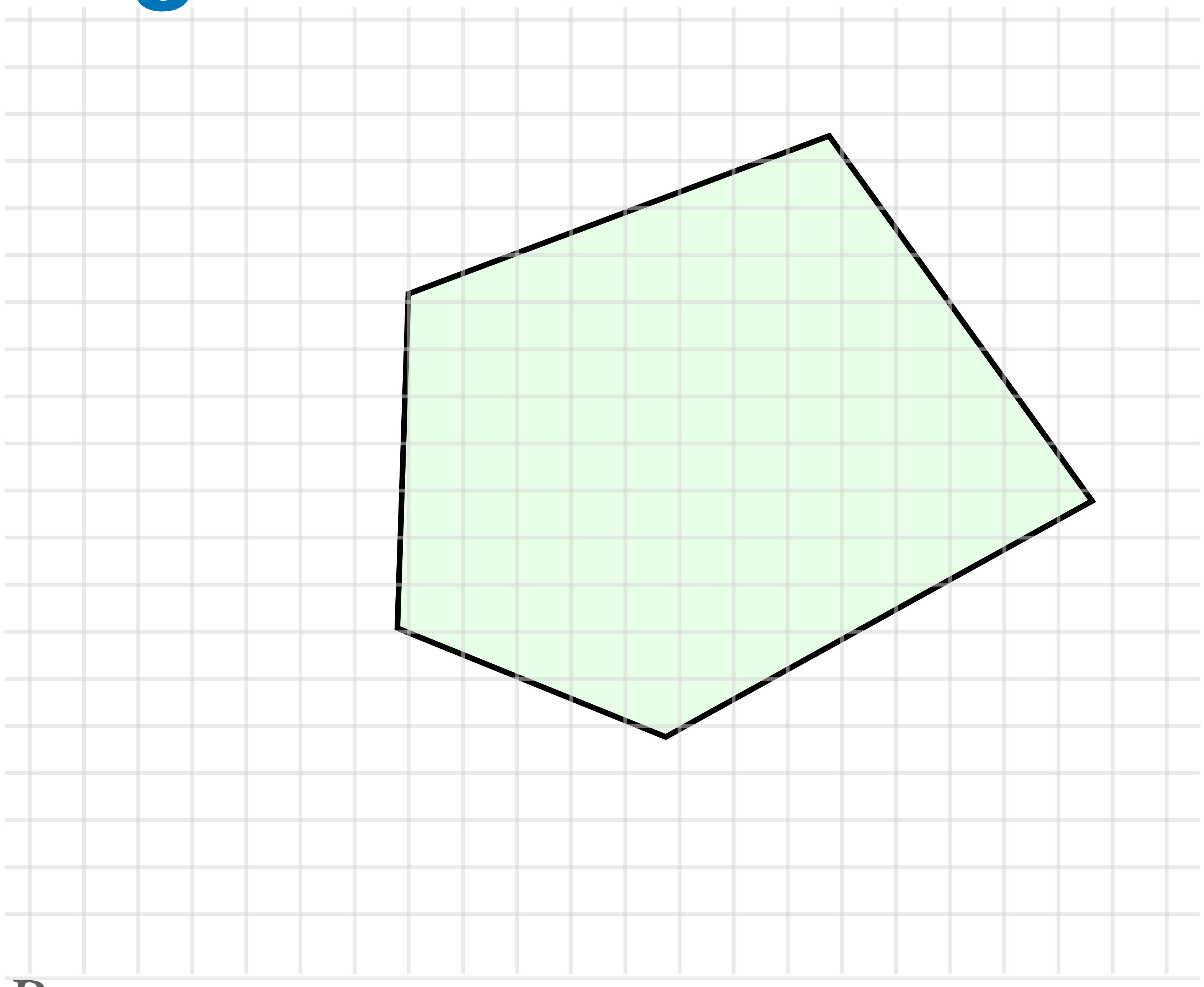
- Fix a Facelike Stabbing Planes proof.
- Take an **in-order** traversal, repeatedly applying the lemma.
  - Repeatedly lift refutations of faces to derivations using the lemma.

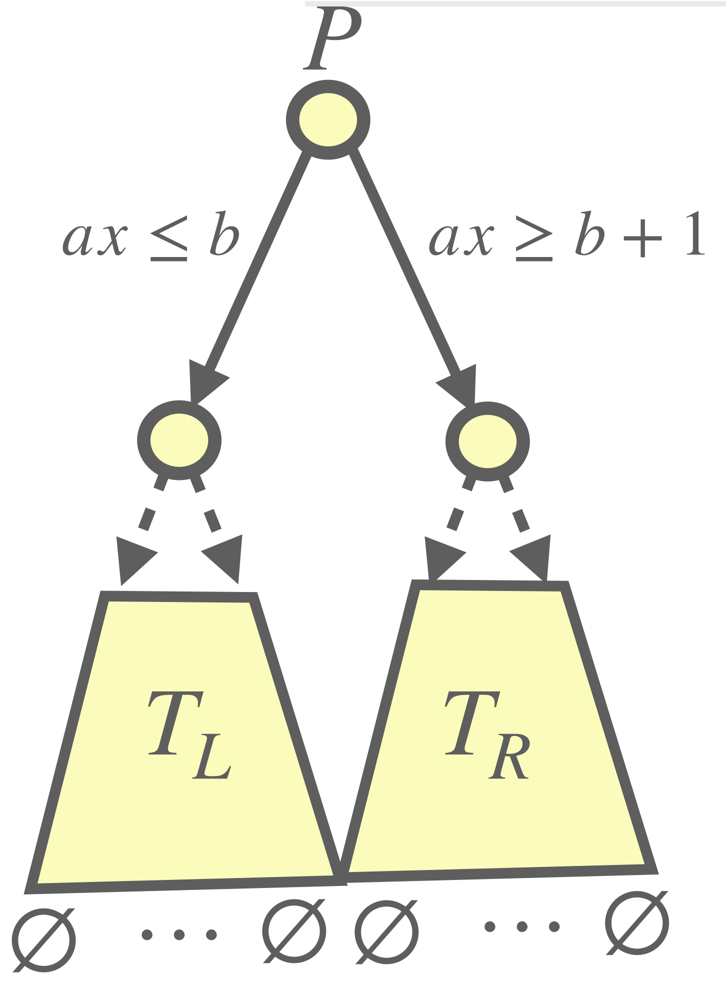


Step 2. Stabbing Planes* = Facelike Stabbing Planes

Proof sketch.

- Consider a bounded-coefficient Stabbing Planes proof.

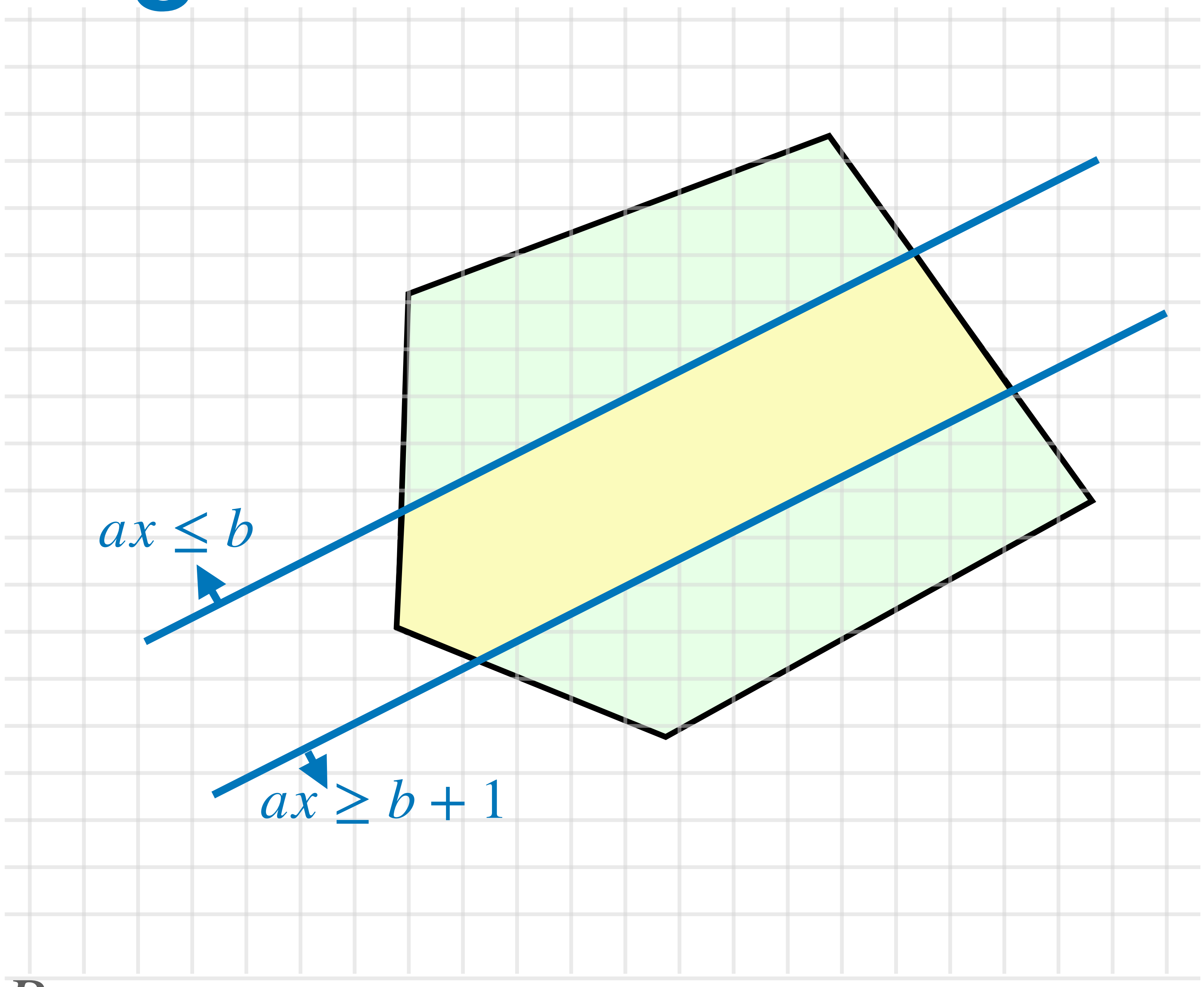


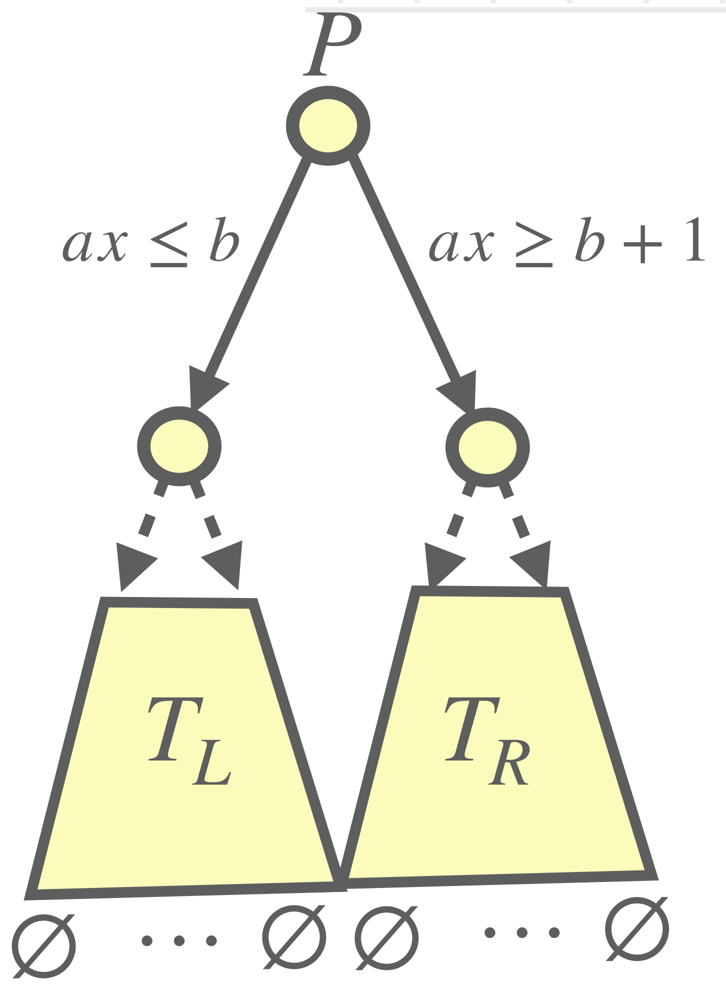


Step 2. Stabbing Planes* = Facelike Stabbing Planes

Proof sketch.

- Consider a bounded-coefficient Stabbing Planes proof.
- Look at the first query: $ax \leq b,\ ax \geq b+1$.



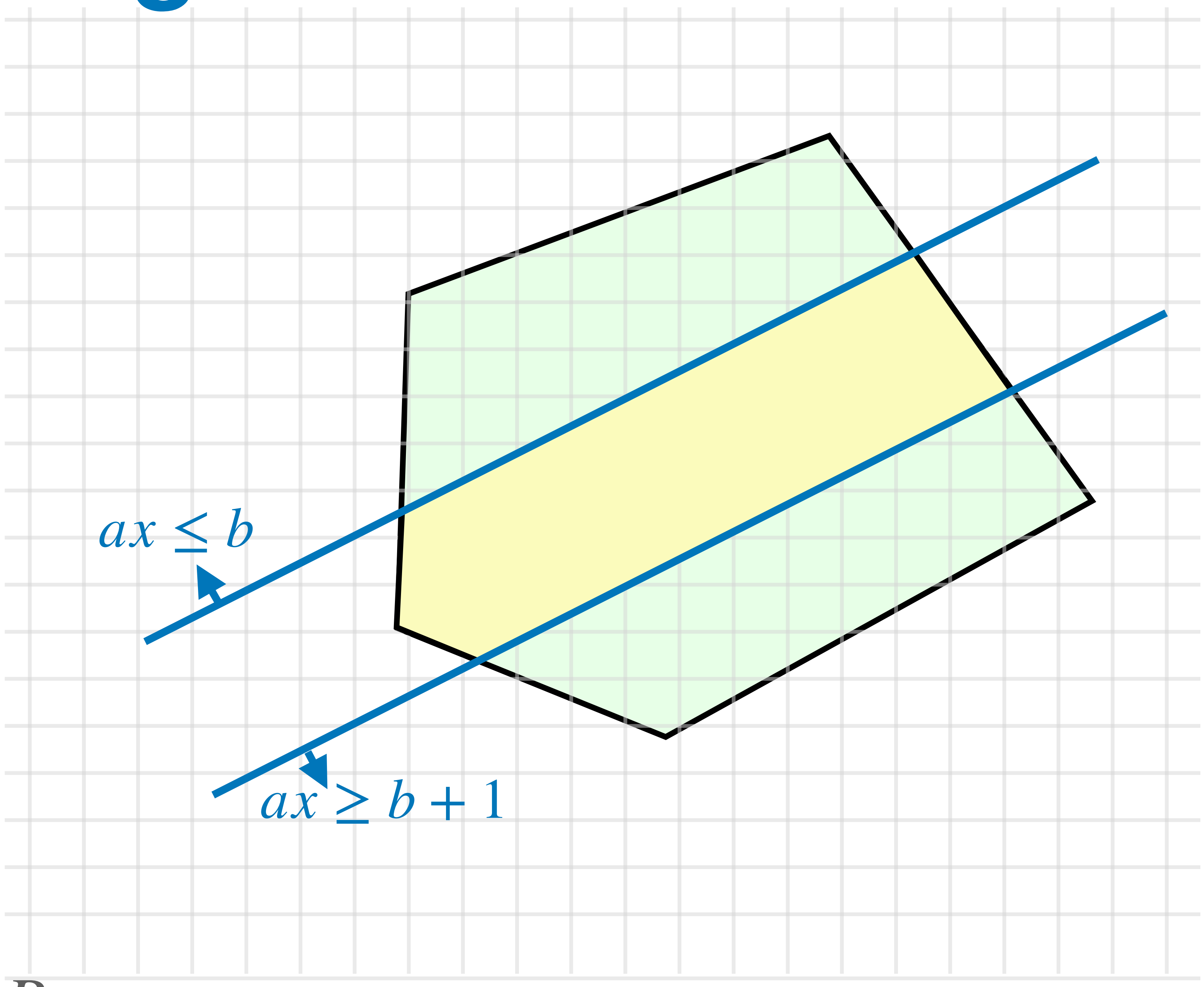


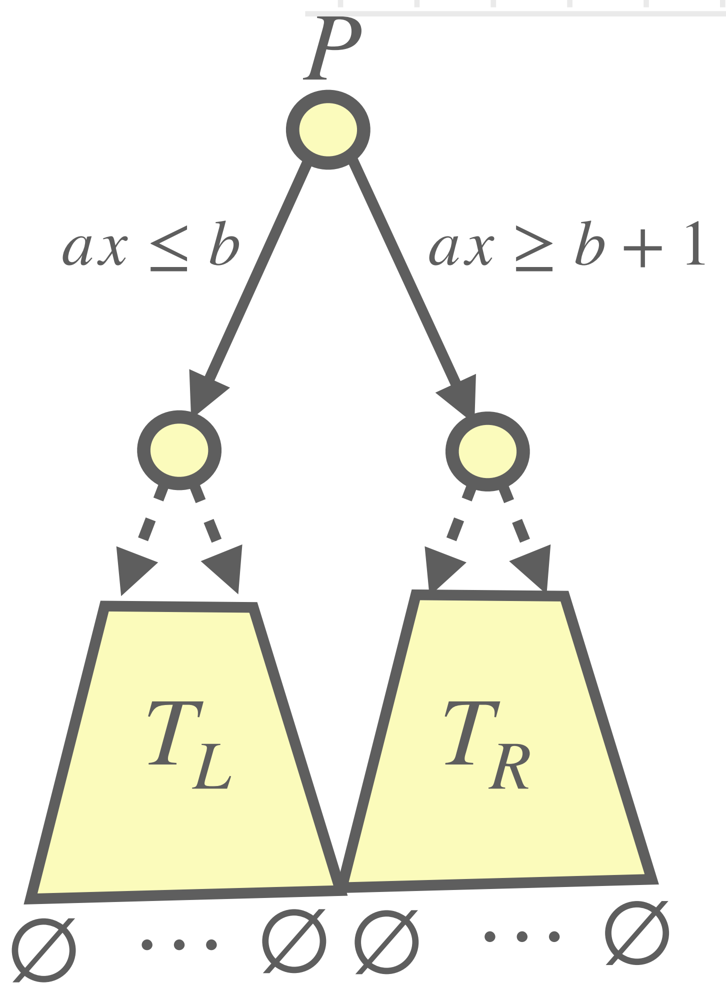
Step 2. Stabbing Planes* = Facelike Stabbing Planes

Proof sketch.

- Consider a bounded-coefficient Stabbing Planes proof.
- Look at the first query: $ax \leq b,\ ax \geq b + 1$.

Goal: convert to a facelike query.





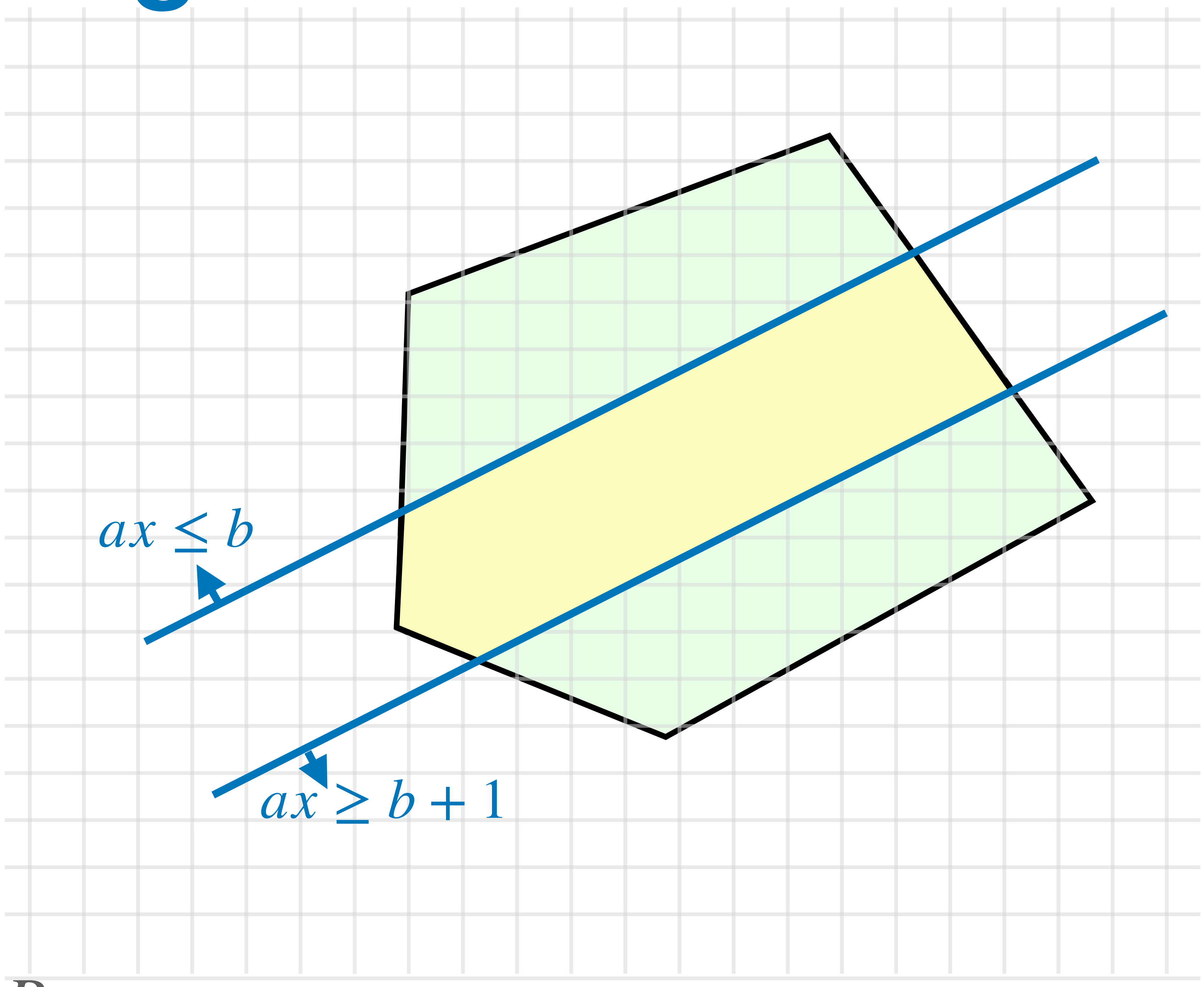
Step 2. Stabbing Planes* = Facelike Stabbing Planes

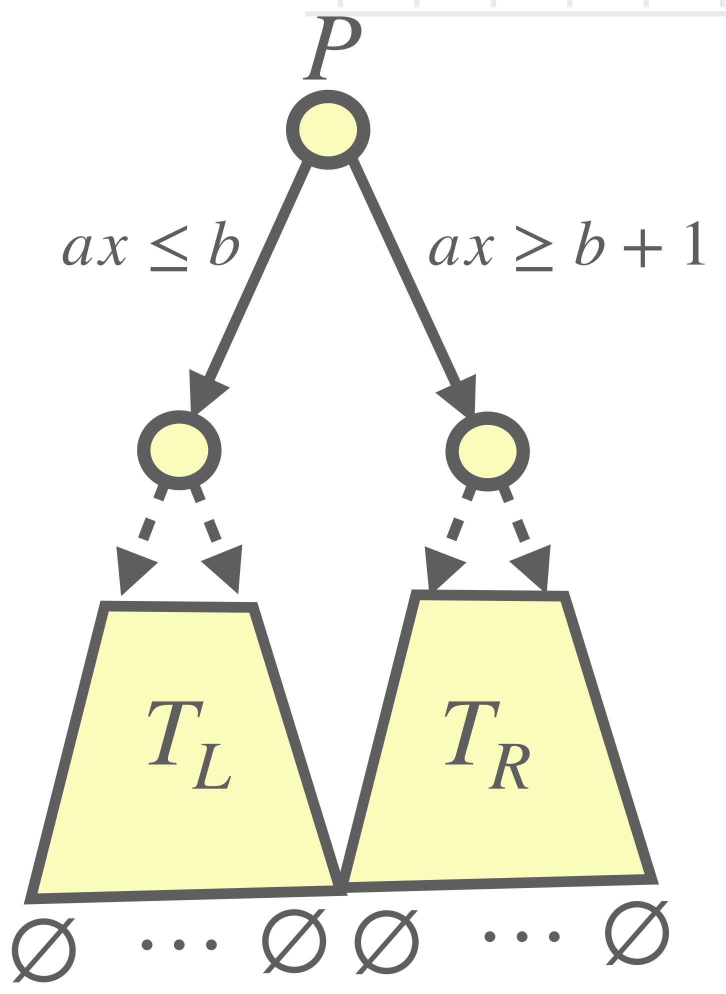
Proof sketch.

- Consider a bounded-coefficient Stabbing Planes proof.
- Look at the first query: $ax \leq b$, $ax \geq b + 1$.

Goal: convert to a facelike query.

- Add translates of the slab until we lie on the face!





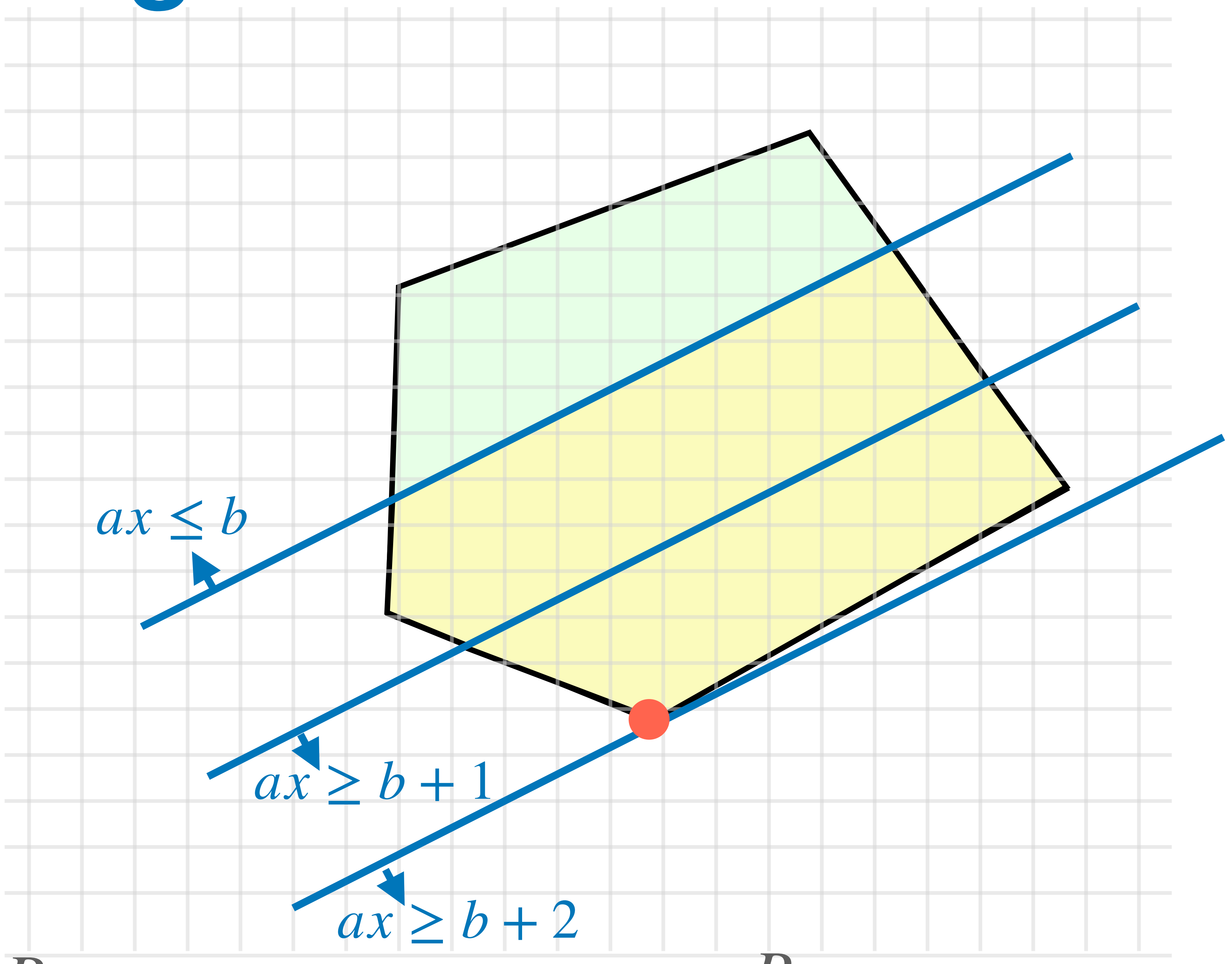
Step 2. Stabbing Planes* = Facelike Stabbing Planes

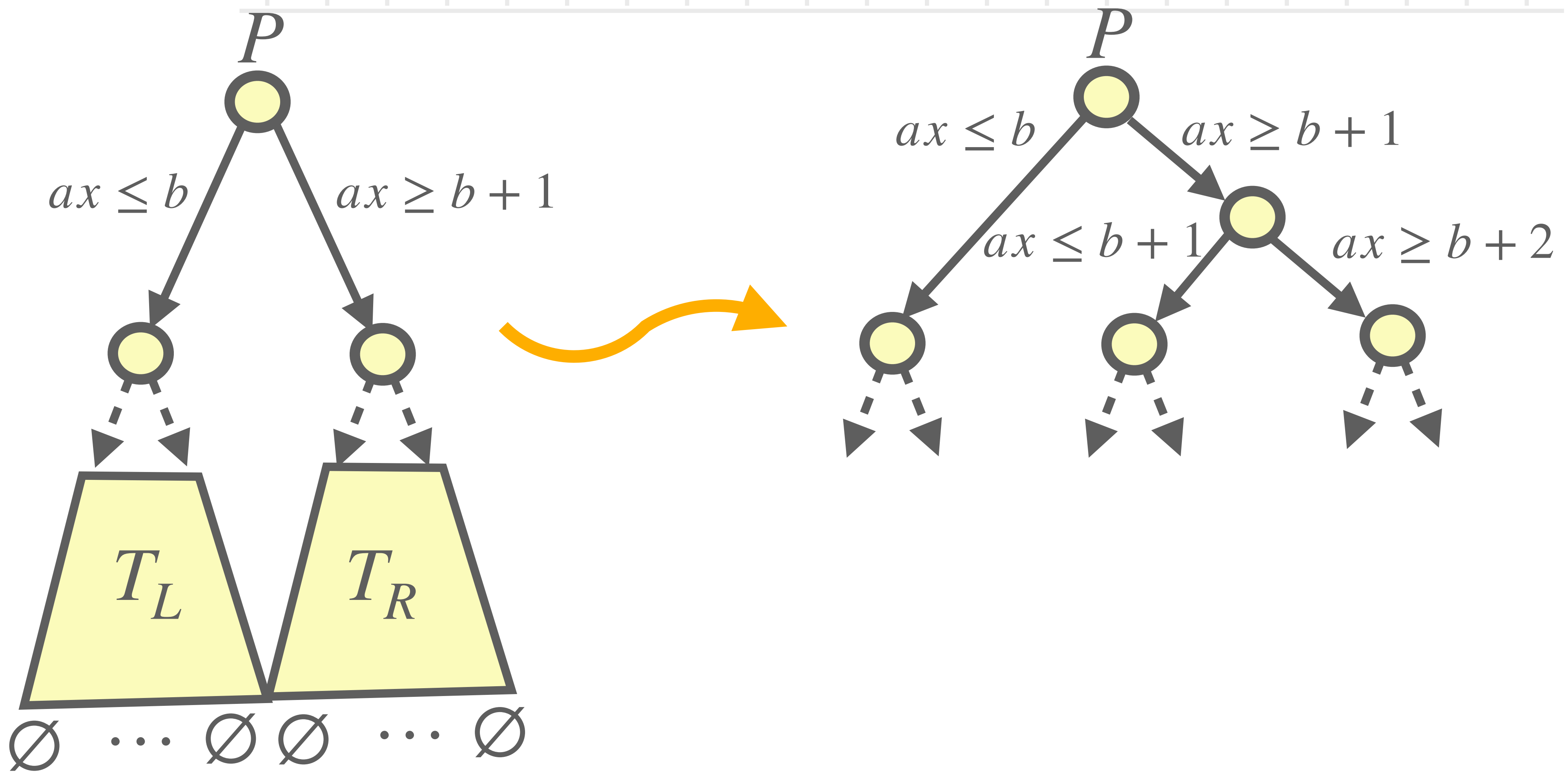
Proof sketch.

- Consider a bounded-coefficient Stabbing Planes proof.
- Look at the first query: $ax \leq b$, $ax \geq b + 1$.

Goal: convert to a facelike query.

- Add translates of the slab until we lie on the face!





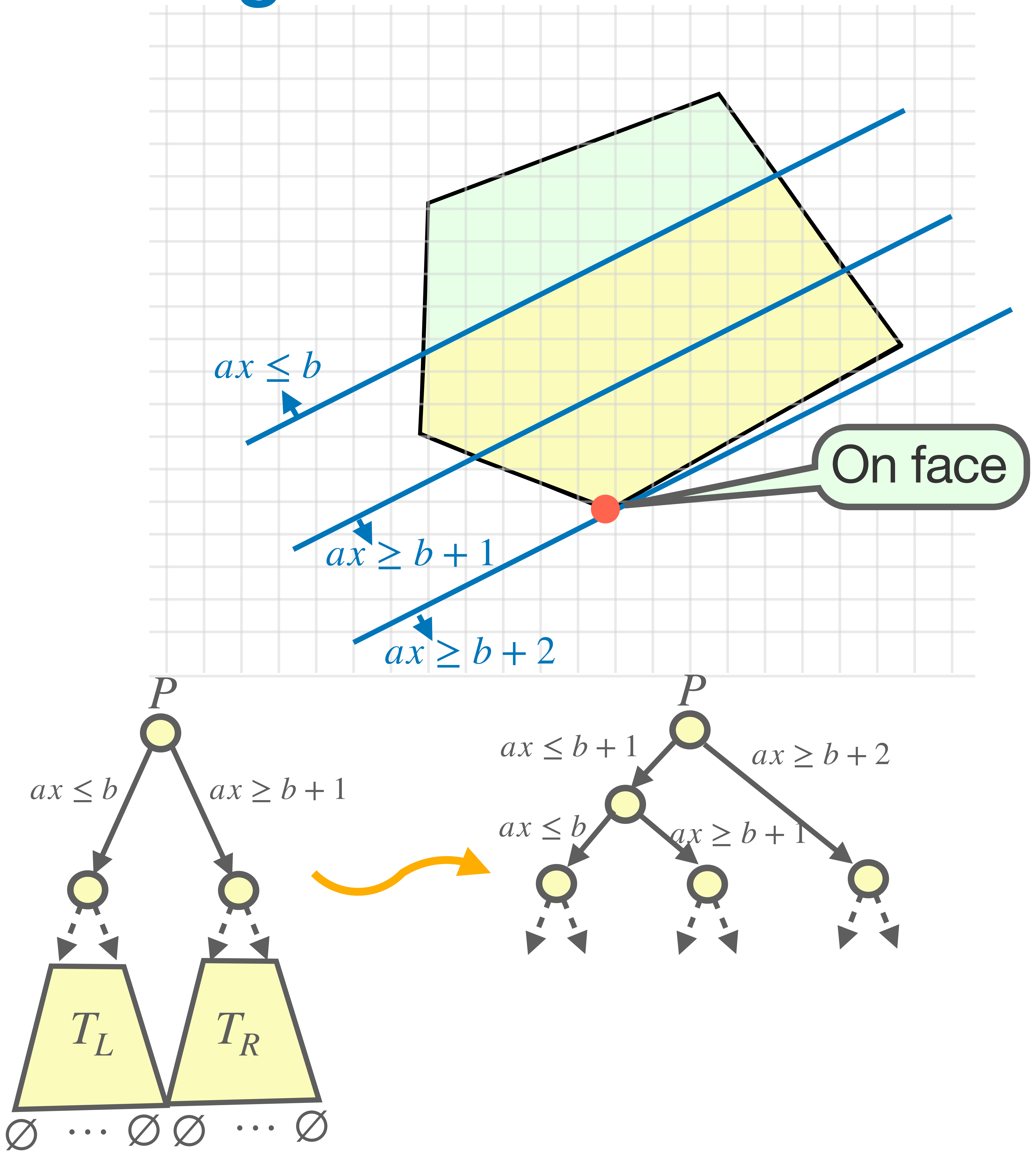
Step 2. Stabbing Planes* = Facelike Stabbing Planes

Proof sketch.

- Consider a bounded-coefficient Stabbing Planes proof.
- Look at the first query: $ax \leq b,\ ax \geq b + 1.$

Goal: convert to a facelike query.

- Add translates of the slab until we lie on the face!



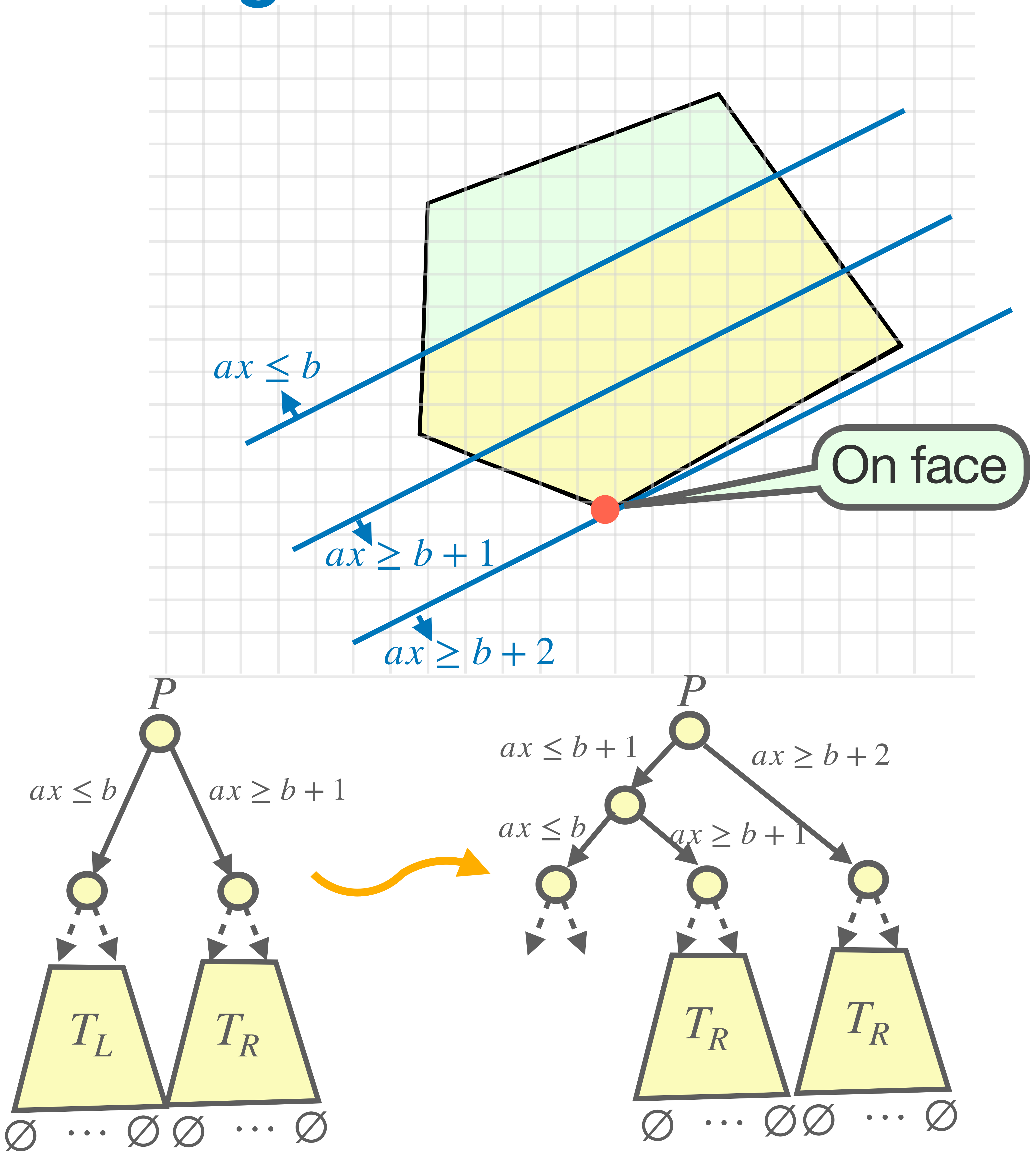
Step 2. Stabbing Planes* = Facelike Stabbing Planes

Proof sketch.

- Consider a bounded-coefficient Stabbing Planes proof.
- Look at the first query: $ax \leq b,\ ax \geq b+1$.

Goal: convert to a facelike query.

- Add translates of the slab until we lie on the face!
- Recursively refute translates using the old subtree,



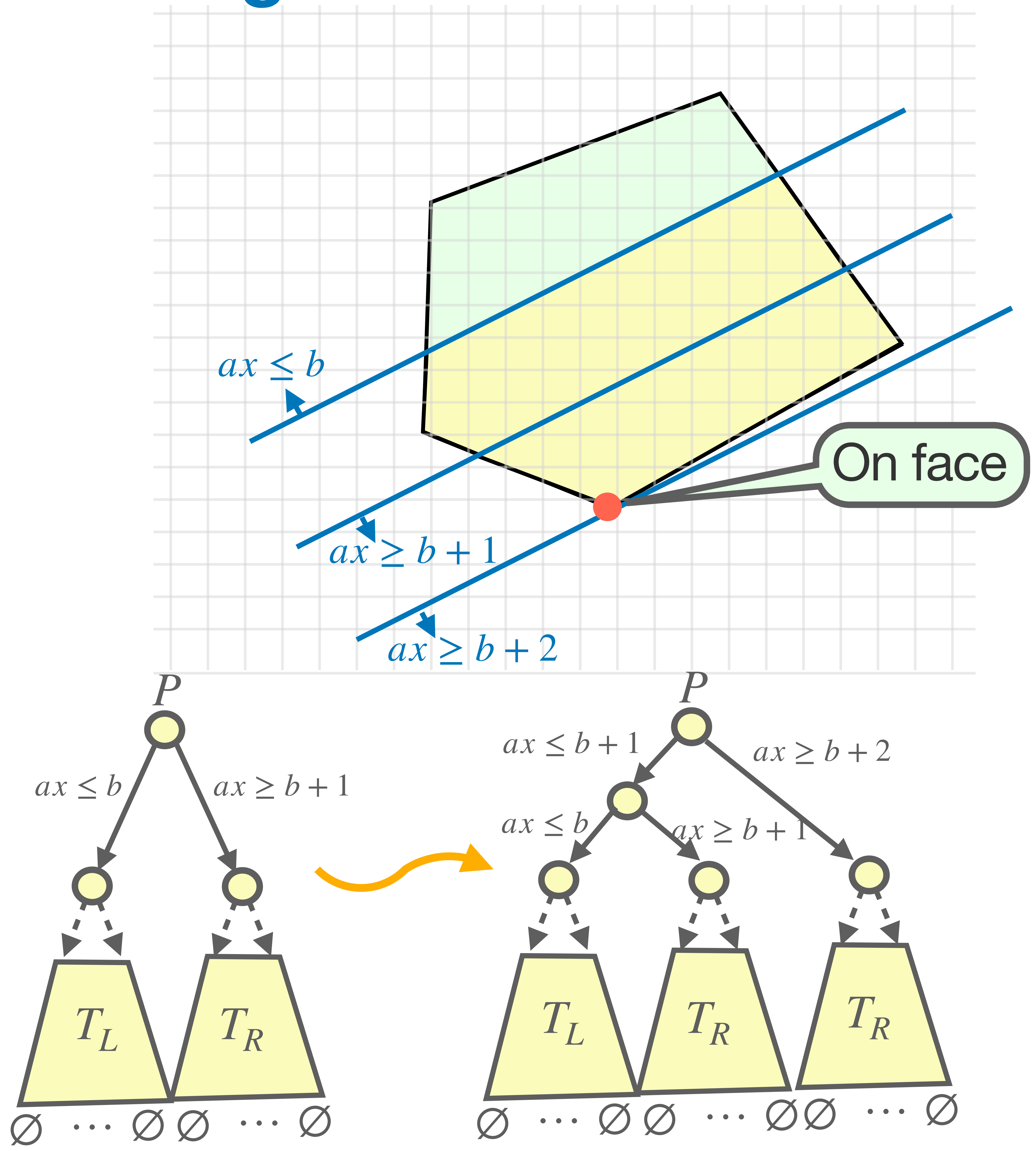
Step 2. Stabbing Planes* = Facelike Stabbing Planes

Proof sketch.

- Consider a bounded-coefficient Stabbing Planes proof.
- Look at the first query: $ax \leq b$, $ax \geq b + 1$.

Goal: convert to a facelike query.

- Add translates of the slab until we lie on the face!
- Recursively refute translates using the old subtree, recurse on the other side similarly.



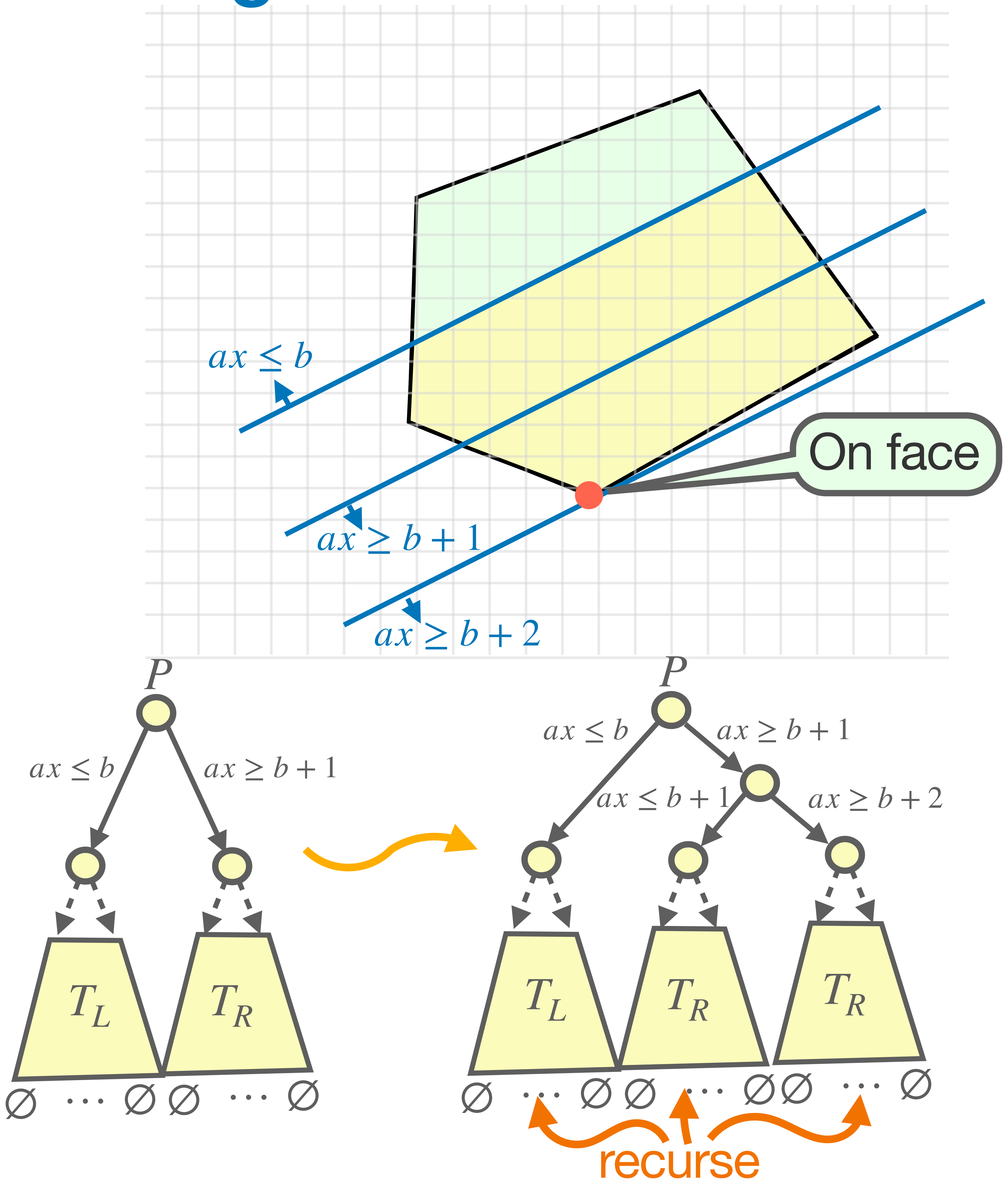
Step 2. Stabbing Planes* = Facelike Stabbing Planes

Proof sketch.

- Consider a bounded-coefficient Stabbing Planes proof.
- Look at the first query: $ax \leq b,\ ax \geq b + 1.$

Goal: convert to a facelike query.

- Add translates of the slab until we lie on the face!
- Recursively refute translates using the old subtree, recurse on the other side similarly.



Step 2. Stabbing Planes* = Facelike Stabbing Planes

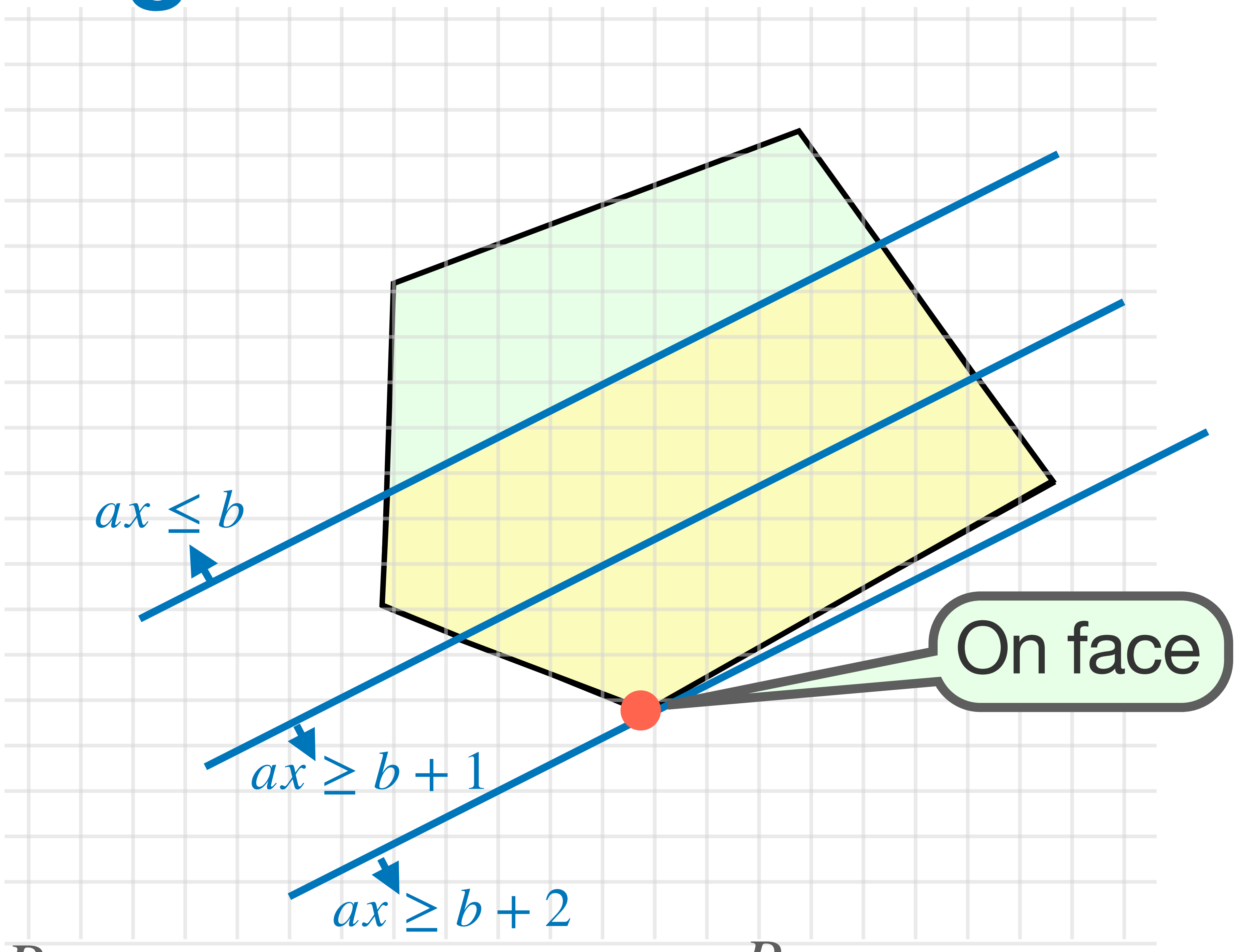
Proof sketch.

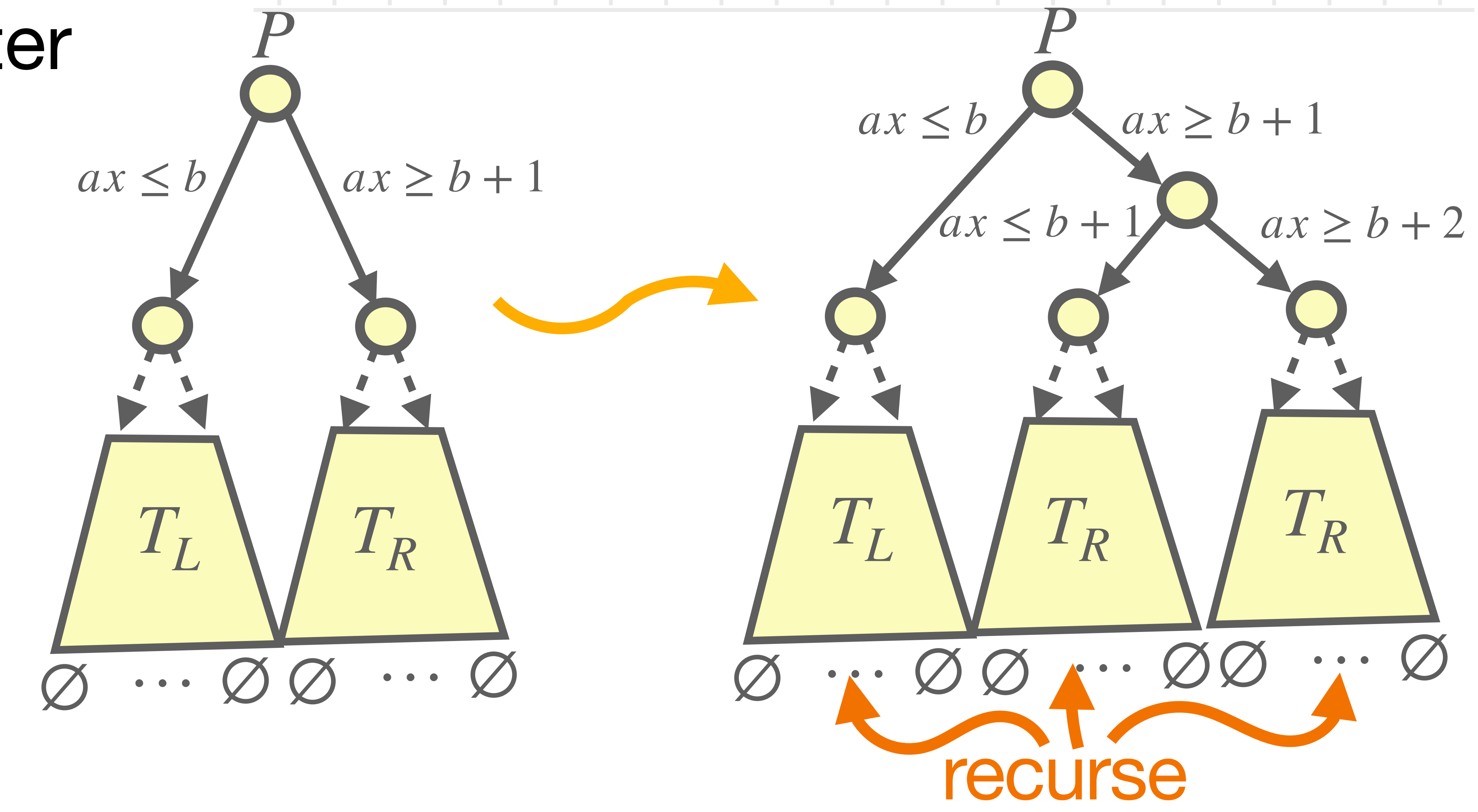
- Consider a bounded-coefficient Stabbing Planes proof.
- Look at the first query: $ax \leq b,\ ax \geq b+1.$

Goal: convert to a facelike query.

- Add translates of the slab until we lie on the face!
- Recursively refute translates using the old subtree, recurse on the other side similarly.

Recursive blowup is proportional to width of slab, diameter of polytope.





Depth Blow-Up

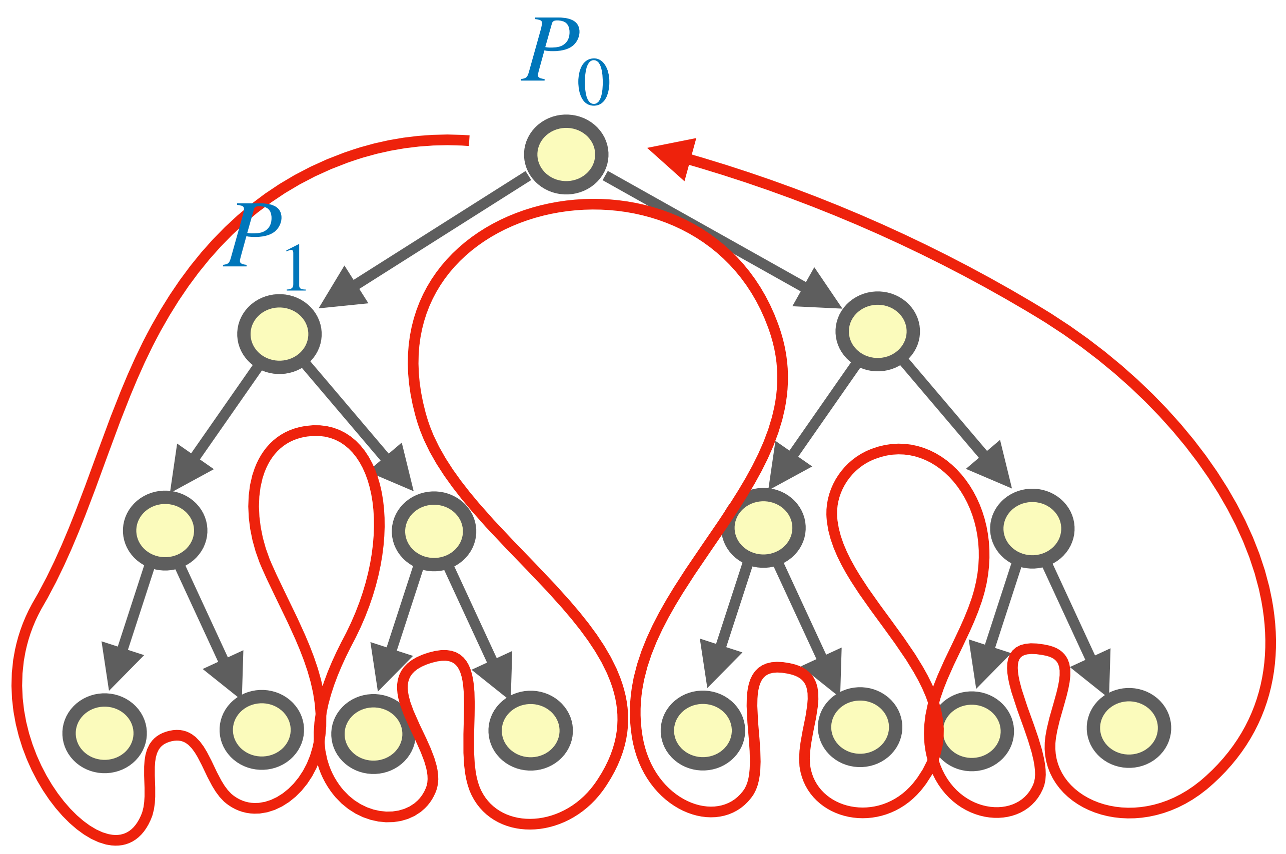
Theorem: Facelike Stabbing Planes = Cutting Planes.

- Converts shallow Stabbing Planes proofs into very deep Cutting Planes proofs.

Depth Blow-Up

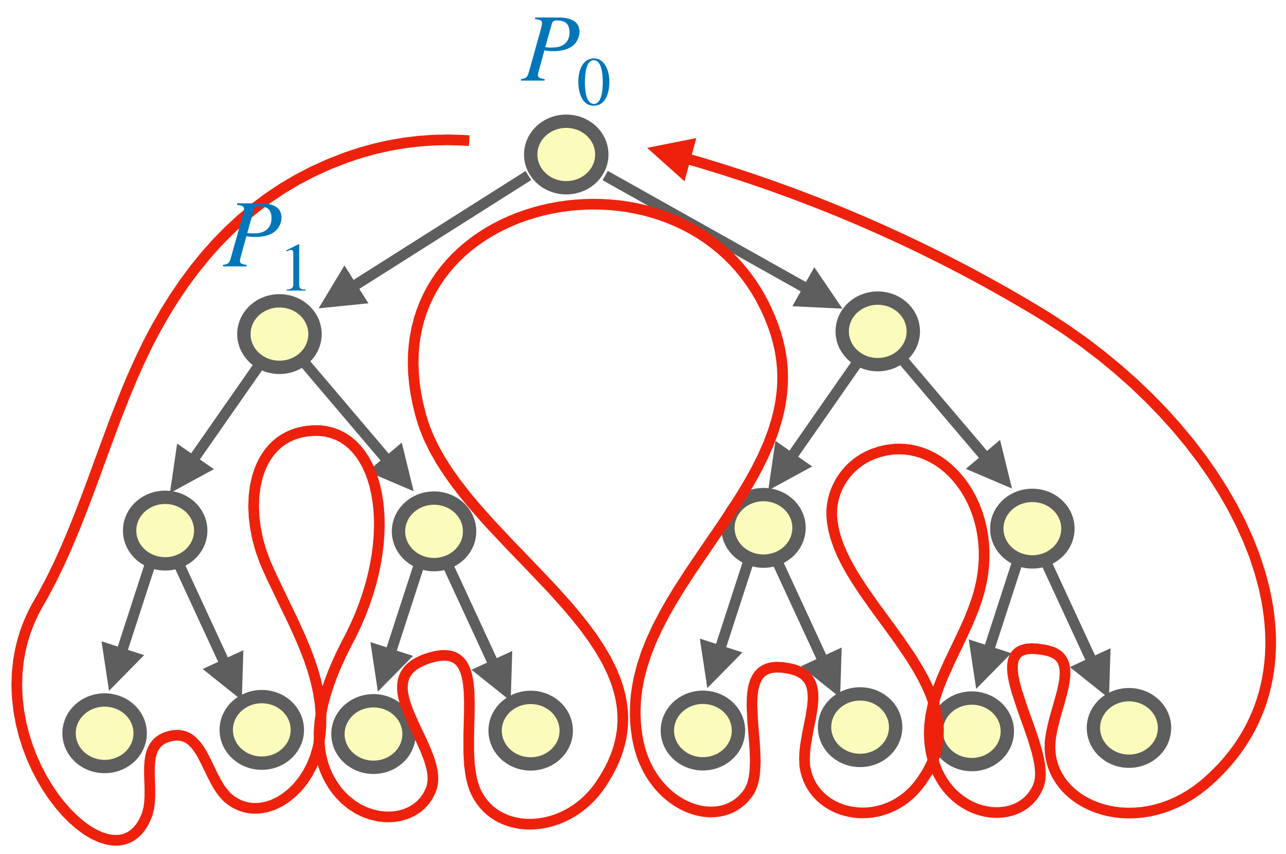
Theorem: Facelike Stabbing Planes = Cutting Planes.

- Converts shallow Stabbing Planes proofs into very deep Cutting Planes proofs.
  - Involves an in-order traversal of the SP proof.



Depth Blow-Up

Theorem: Facelike Stabbing Planes = Cutting Planes.

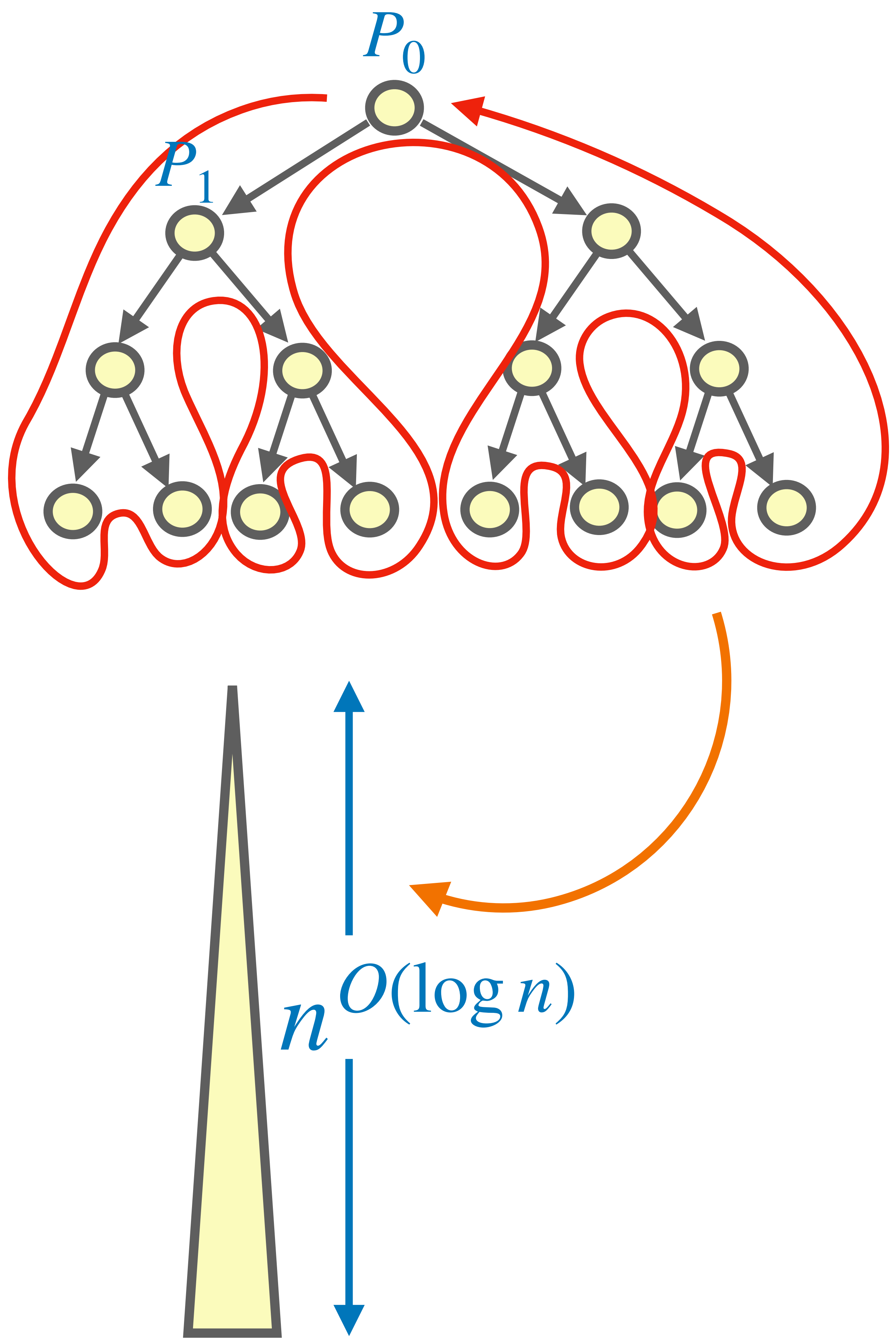


- Converts shallow Stabbing Planes proofs into very deep Cutting Planes proofs.
  - Involves an in-order traversal of the SP proof.
- The Stabbing Planes proofs of Tseitin have size $n^{O(\log n)}$ and depth $O(\log^2 n)$.

Depth Blow-Up

Theorem: Facelike Stabbing Planes = Cutting Planes.

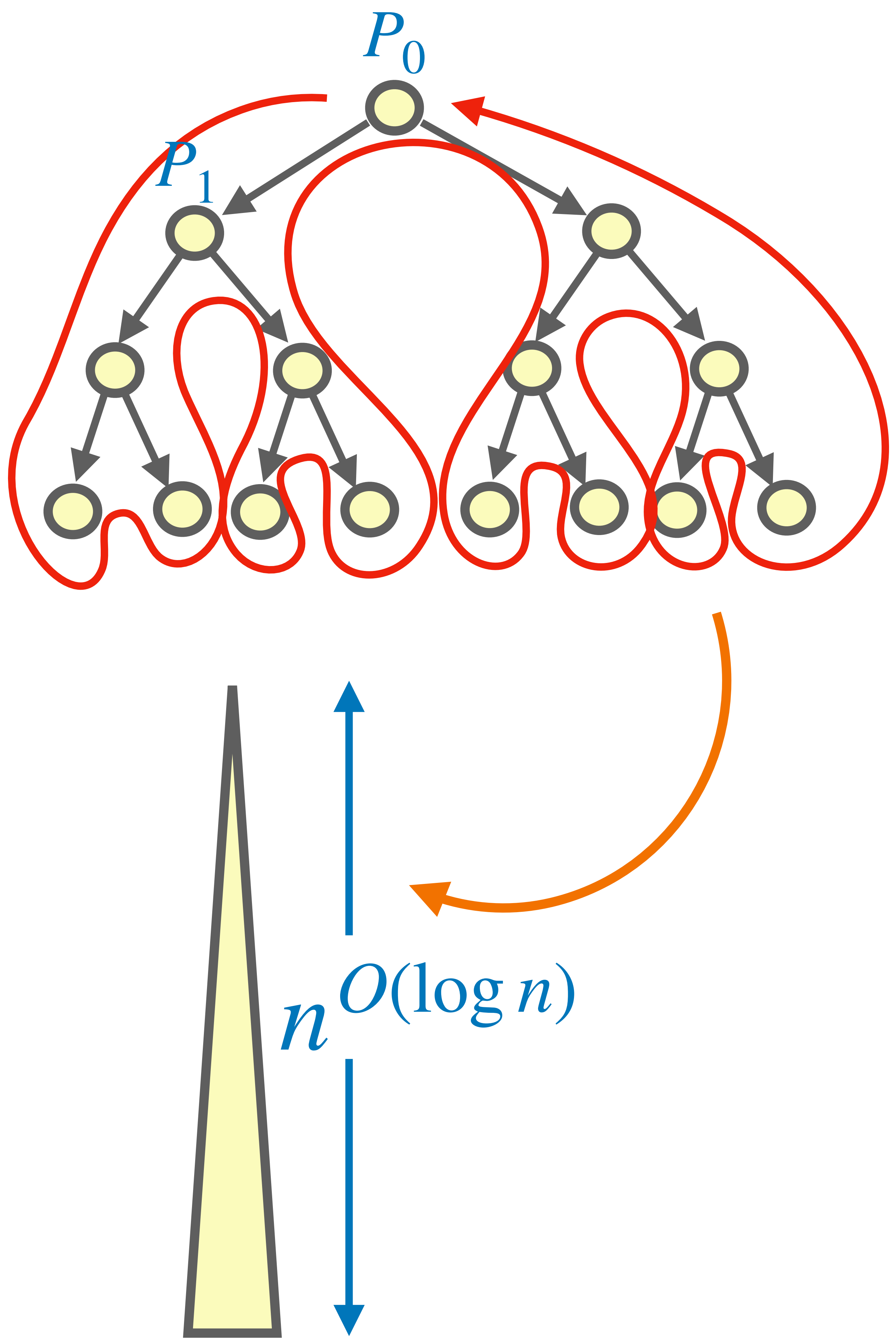
- Converts shallow Stabbing Planes proofs into very deep Cutting Planes proofs.
  - Involves an in-order traversal of the SP proof.
- The Stabbing Planes proofs of Tseitin have size $n^{O(\log n)}$ and depth $O(\log^2 n)$.
- Implies a Cutting Planes proof of **depth** and **size** $n^{O(\log n)}$.



Depth Blow-Up

Theorem: Facelike Stabbing Planes = Cutting Planes.

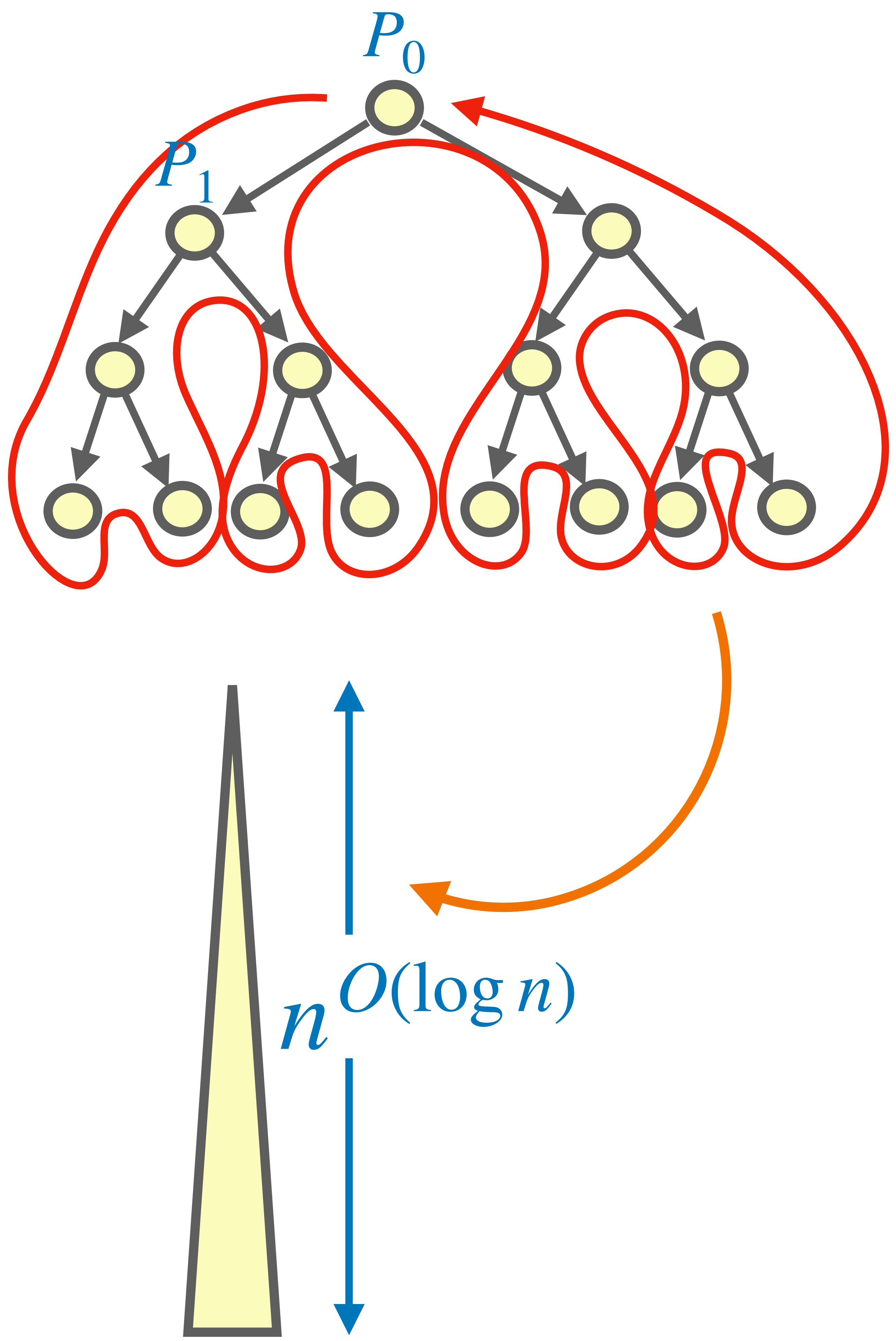
- Converts shallow Stabbing Planes proofs into very deep Cutting Planes proofs.
  - Involves an in-order traversal of the SP proof.
- The Stabbing Planes proofs of Tseitin have size $n^{O(\log n)}$ and depth $O(\log^2 n)$.
- Implies a Cutting Planes proof of **depth** and **size** $n^{O(\log n)}$.
- Tseitin has a Cutting Planes proof of depth $O(n)$ and size 2^n.



Depth Blow-Up

Theorem: Facelike Stabbing Planes = Cutting Planes.

- Converts shallow Stabbing Planes proofs into very deep Cutting Planes proofs.
  - Involves an in-order traversal of the SP proof.
- The Stabbing Planes proofs of Tseitin have size $n^{O(\log n)}$ and depth $O(\log^2 n)$.
- Implies a Cutting Planes proof of **depth** and **size** $n^{O(\log n)}$.
- Tseitin has a Cutting Planes proof of depth $O(n)$ and size 2^n.



Conjecture: Any subexponential-size Cutting Planes refutation of a Tseitin formula requires superpolynomial depth

Supercritical Size-Depth Tradeoffs

Conjecture: Any subexponential-size Cutting Planes refutation of a Tseitin formula requires superpolynomial depth

Supercritical Size-Depth Tradeoffs

Conjecture: Any subexponential-size Cutting Planes refutation of a Tseitin formula requires superpolynomial depth

- Tseitin has a Cutting Planes proof of depth $O(n)$ and size 2^n.
    - Holds for any polytope coming from an unsatisfiable CNF formula.

Supercritical Size-Depth Tradeoffs

Conjecture: Any subexponential-size Cutting Planes refutation of a Tseitin formula requires superpolynomial depth

- Tseitin has a Cutting Planes proof of depth $O(n)$ and size 2^n.
  - Holds for any polytope coming from an unsatisfiable CNF formula.

Supercritical Size/Depth Tradeoff: bounding the size increases the depth **beyond** the worst-case $O(n)$

Supercritical Size-Depth Tradeoffs

Conjecture: Any subexponential-size Cutting Planes refutation of a Tseitin formula requires superpolynomial depth

- Tseitin has a Cutting Planes proof of depth $O(n)$ and size 2^n.
  - Holds for any polytope coming from an unsatisfiable CNF formula.

Supercritical Size/Depth Tradeoff: bounding the size increases the depth **beyond** the worst-case $O(n)$

- Very few examples of supercritical tradeoffs.

Supercritical Size-Depth Tradeoffs

Conjecture: Any subexponential-size Cutting Planes refutation of a Tseitin formula requires superpolynomial depth

- Tseitin has a Cutting Planes proof of depth $O(n)$ and size 2^n.
  - Holds for any polytope coming from an unsatisfiable CNF formula.

Supercritical Size/Depth Tradeoff: bounding the size increases the depth **beyond** the worst-case $O(n)$

- Very few examples of supercritical tradeoffs.
  - [BBI12] Tradeoff between Resolution size and space.

Supercritical Size-Depth Tradeoffs

Conjecture: Any subexponential-size Cutting Planes refutation of a Tseitin formula requires superpolynomial depth

- Tseitin has a Cutting Planes proof of depth $O(n)$ and size 2^n.
  - Holds for any polytope coming from an unsatisfiable CNF formula.

Supercritical Size/Depth Tradeoff: bounding the size increases the depth **beyond** the worst-case $O(n)$

- Very few examples of supercritical tradeoffs.
  - [BBI12] Tradeoff between Resolution size and space.
  - [Razborov16] Tradeoff between tree-like Resolution size and width.

Supercritical Size-Depth Tradeoffs

Conjecture: Any subexponential-size Cutting Planes refutation of a Tseitin formula requires superpolynomial depth

- Tseitin has a Cutting Planes proof of depth $O(n)$ and size 2^n.
  - Holds for any polytope coming from an unsatisfiable CNF formula.

Supercritical Size/Depth Tradeoff: bounding the size increases the depth **beyond** the worst-case $O(n)$

- Very few examples of supercritical tradeoffs.
  - [BBI12] Tradeoff between Resolution size and space.
  - [Razborov16] Tradeoff between tree-like Resolution size and width.
  - [BNT13, BN20, Razborov18] Tradeoffs between notions of space and size for Resolution and Polynomial Calculus.

Supercritical Size-Depth Tradeoffs

Conjecture: Any subexponential-size Cutting Planes refutation of a Tseitin formula requires superpolynomial depth

- Tseitin has a Cutting Planes proof of depth $O(n)$ and size 2^n.
  - Holds for any polytope coming from an unsatisfiable CNF formula.

Supercritical Size/Depth Tradeoff: bounding the size increases the depth **beyond** the worst-case $O(n)$

- Very few examples of supercritical tradeoffs.
  - [BBI12] Tradeoff between Resolution size and space.
  - [Razborov16] Tradeoff between tree-like Resolution size and width.
  - [BNT13, BN20, Razborov18] Tradeoffs between notions of space and size for Resolution and Polynomial Calculus.

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

Supercritical Size-Depth Tradeoffs

Conjecture: Any subexponential-size Cutting Planes refutation of a Tseitin formula requires superpolynomial depth

- Tseitin has a Cutting Planes proof of depth $O(n)$ and size 2^n.
  - Holds for any polytope coming from an unsatisfiable CNF formula.

Supercritical Size/Depth Tradeoff: bounding the size increases the depth **beyond** the worst-case $O(n)$

- Very few examples of supercritical tradeoffs.
  - [BBI12] Tradeoff between Resolution size and space.
  - [Razborov16] Tradeoff between tree-like Resolution size and width.
  - [BNT13, BN20, Razborov18] Tradeoffs between notions of space and size for Resolution and Polynomial Calculus.

Builds on [Razborov16]

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Proof Idea:

- Begin with a formula F that has small size but requires large depth in Resolution

Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Proof Idea:

- Begin with a formula F that has small size but requires large depth in Resolution
  - Pebbling: $O(n)$-size and $\Omega(n/\log n)$ depth.

Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Proof Idea:

- Begin with a formula F that has small size but requires large depth in Resolution
  - Pebbling: $O(n)$-size and $\Omega(n/\log n)$ depth.
- **Compress** the number of variables from $n \to N$ while maintaining the $O(n)$ upper bound and $\Omega(n/\log n)$ lower bound.

Supercritical Size-Depth Tradeoffs

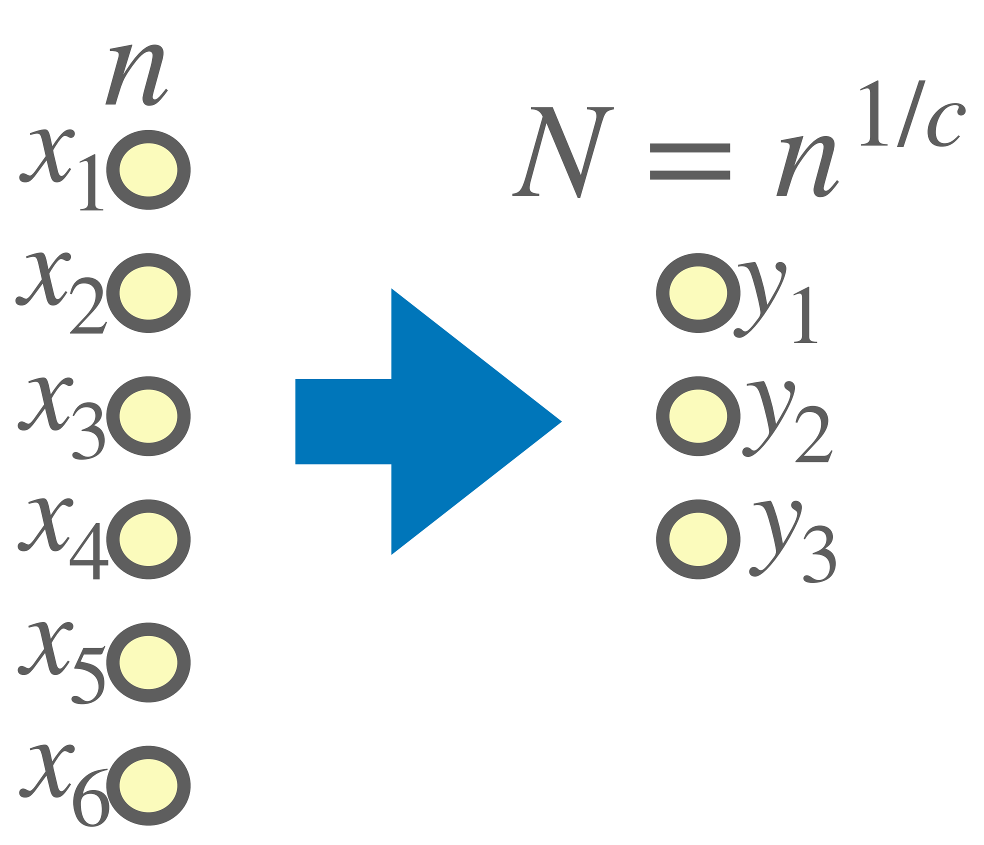
Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Proof Idea:

- Begin with a formula F that has small size but requires large depth in Resolution
  - Pebbling: $O(n)$-size and $\Omega(n/\log n)$ depth.
- **Compress** the number of variables from $n \to N$ while maintaining the $O(n)$ upper bound and $\Omega(n/\log n)$ lower bound.



Supercritical Size-Depth Tradeoffs

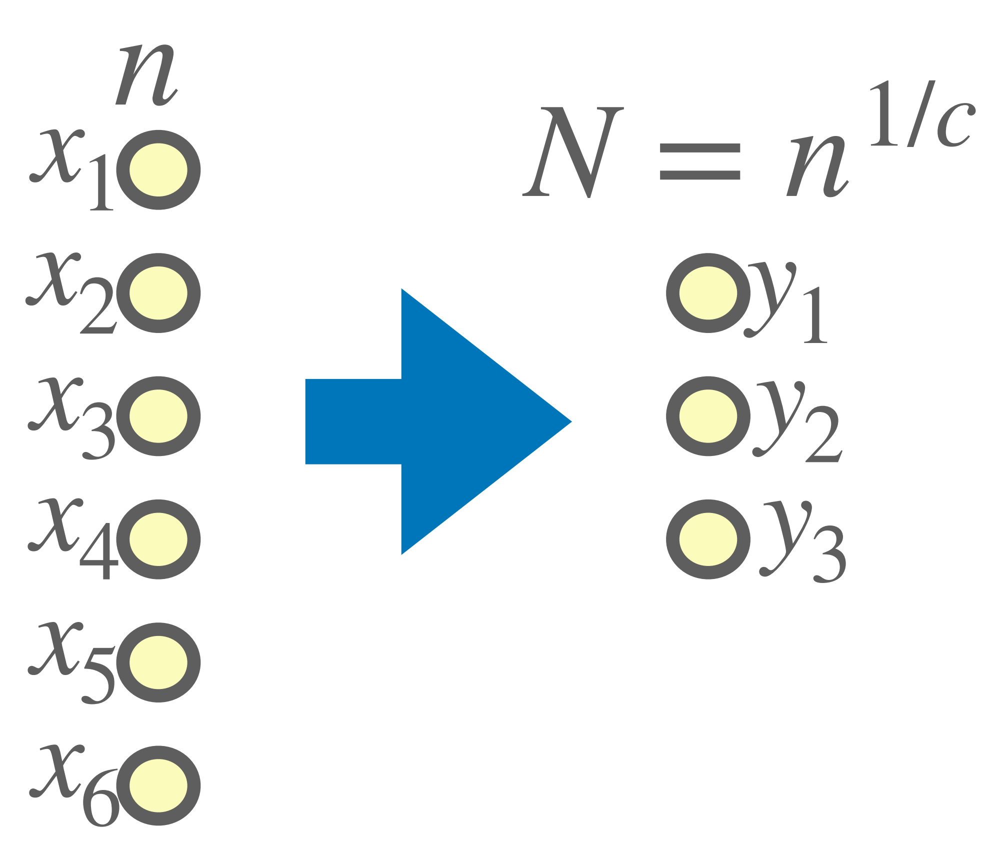
Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Proof Idea:

- Begin with a formula F that has small size but requires large depth in Resolution
  - Pebbling: $O(n)$-size and $\Omega(n/\log n)$ depth.
- **Compress** the number of variables from $n \rightarrow N$ while maintaining the $O(n)$ upper bound and $\Omega(n/\log n)$ lower bound.
  - Should be difficult for Resolution to differentiate between the original and compressed instance.



Supercritical Size-Depth Tradeoffs

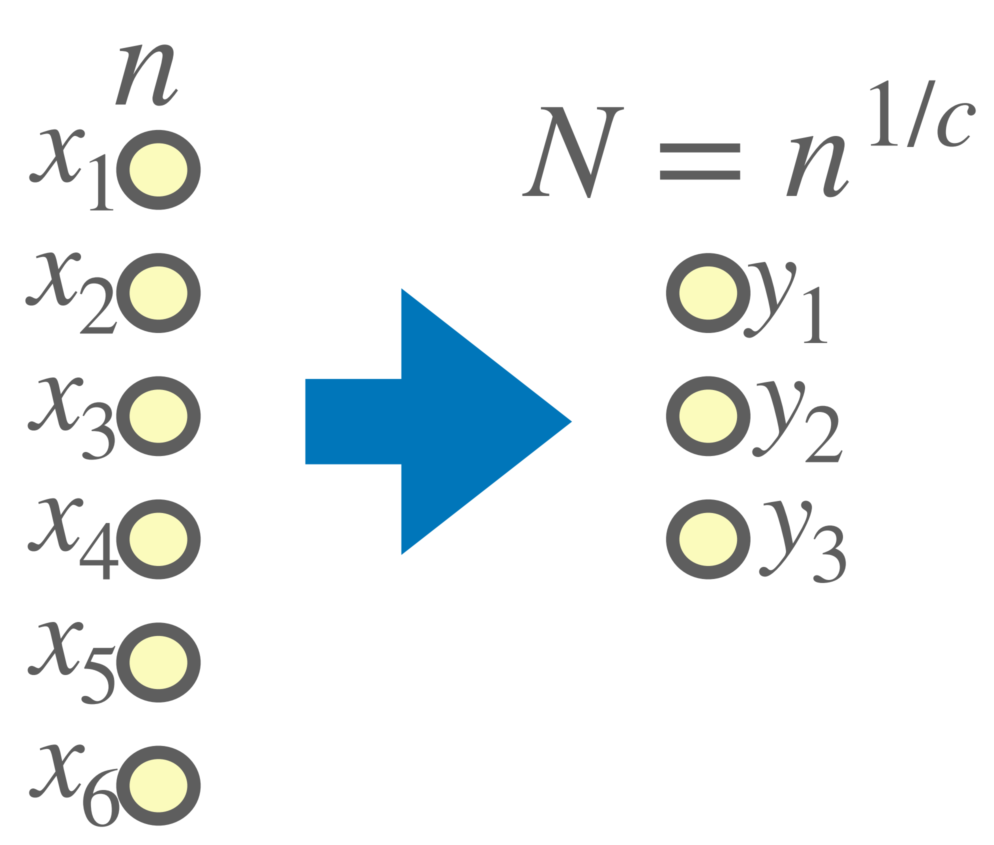
Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Proof Idea:

- Begin with a formula F that has small size but requires large depth in Resolution
  - Pebbling: $O(n)$-size and $\Omega(n/\log n)$ depth.
- **Compress** the number of variables from $n \rightarrow N$ while maintaining the $O(n)$ upper bound and $\Omega(n/\log n)$ lower bound.
  - Should be difficult for Resolution to differentiate between the original and compressed instance.
- **How?**



Supercritical Size-Depth Tradeoffs

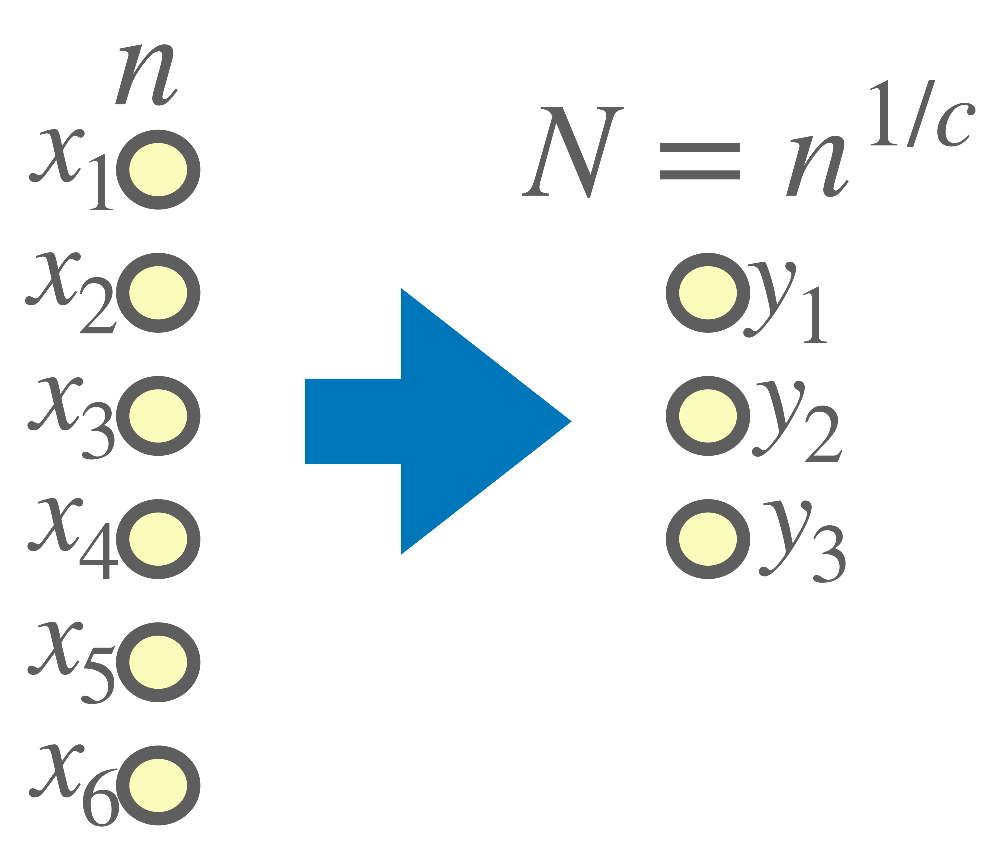
Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Proof Idea:

- Begin with a formula F that has small size but requires large depth in Resolution
  - Pebbling: $O(n)$-size and $\Omega(n/\log n)$ depth.
- **Compress** the number of variables from $n \to N$ while maintaining the $O(n)$ upper bound and $\Omega(n/\log n)$ lower bound.
  - Should be difficult for Resolution to differentiate between the original and compressed instance.
- **How?** — (roughly) compose F with Nisan-Wigderson generator



Supercritical Size-Depth Tradeoffs

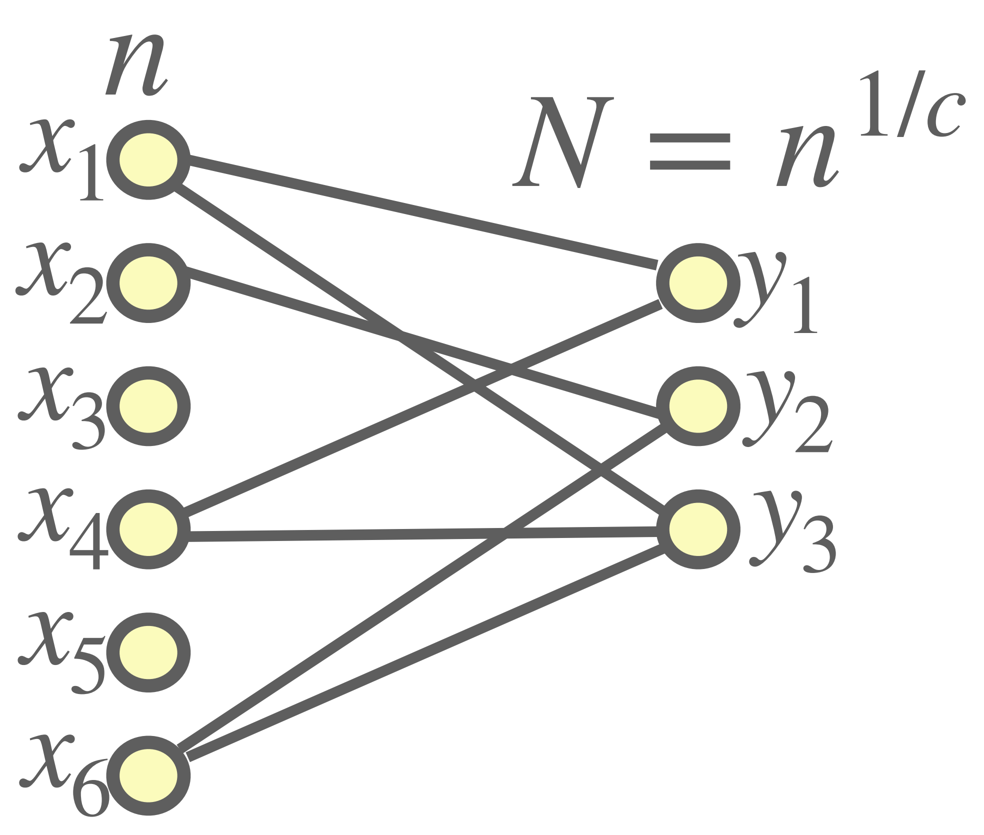
Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Proof Idea:

- Begin with a formula F that has small size but requires large depth in Resolution
    - Pebbling: $O(n)$-size and $\Omega(n/\log n)$ depth.
- **Compress** the number of variables from $n \rightarrow N$ while maintaining the $O(n)$ upper bound and $\Omega(n/\log n)$ lower bound.
    - Should be difficult for Resolution to differentiate between the original and compressed instance.
- **How?** — (roughly) compose F with Nisan-Wigderson generator
    - Replace each old variable x_i with an XOR of new variables y_j



Supercritical Size-Depth Tradeoffs

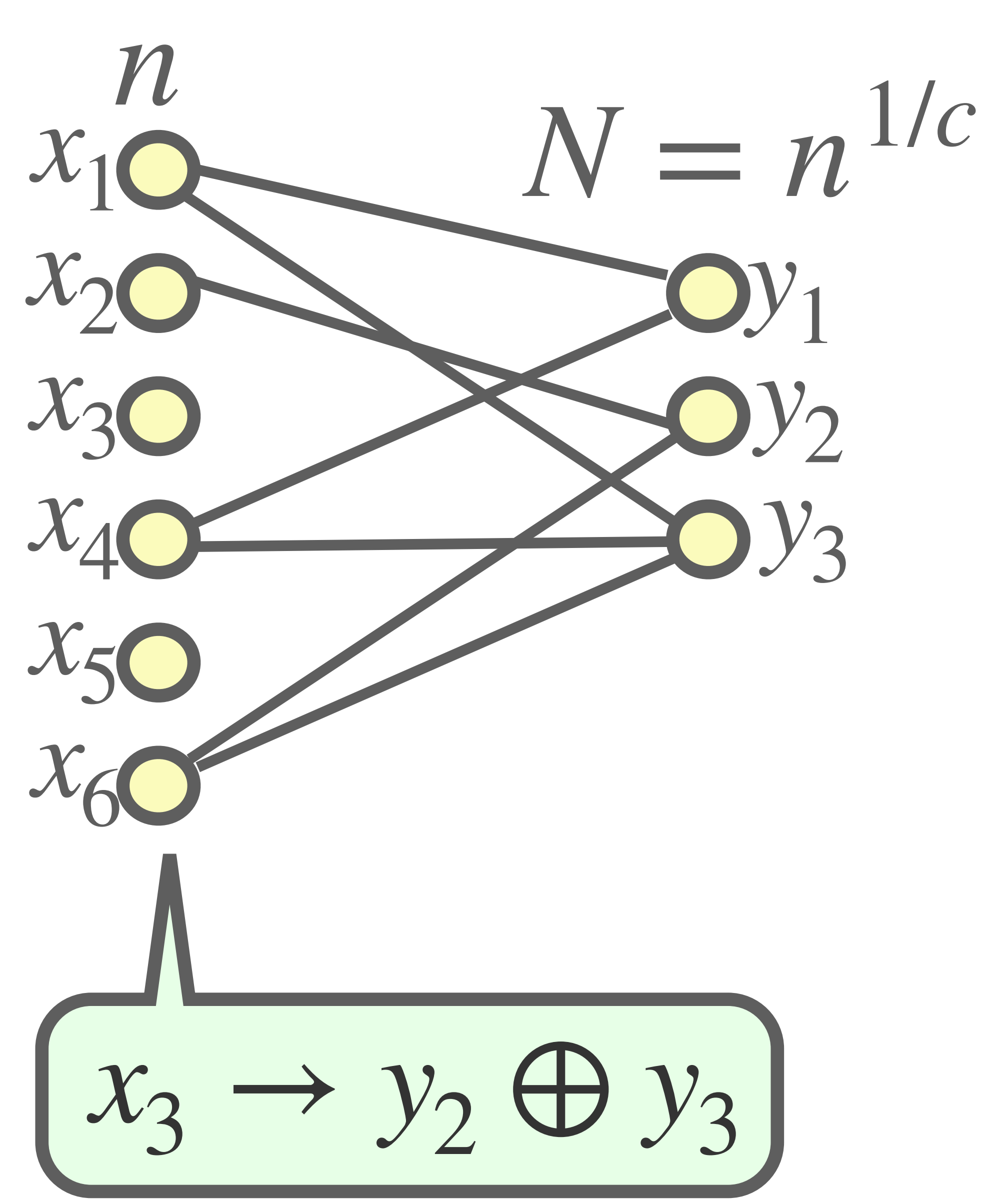
Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Proof Idea:

- Begin with a formula F that has small size but requires large depth in Resolution
  - Pebbling: $O(n)$-size and $\Omega(n/\log n)$ depth.
- **Compress** the number of variables from $n \to N$ while maintaining the $O(n)$ upper bound and $\Omega(n/\log n)$ lower bound.
  - Should be difficult for Resolution to differentiate between the original and compressed instance.
- **How?** — (roughly) compose F with Nisan-Wigderson generator
  - Replace each old variable x_i with an XOR of new variables y_j



Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Tradeoffs for Res(k), Cutting Planes follow by **lifting** the Resolution tradeoff.

Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\mathrm{depth}(\Pi) \cdot \log \mathrm{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Tradeoffs for Res(k), Cutting Planes follow by **lifting** the Resolution tradeoff.

Can we use this to obtain the tradeoff for Tseitin?

Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Tradeoffs for Res(k), Cutting Planes follow by **lifting** the Resolution tradeoff.

Can we use this to obtain the tradeoff for Tseitin?

- Not obviously — Tseitin is exponentially hard for Resolution!

Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Tradeoffs for Res(k), Cutting Planes follow by **lifting** the Resolution tradeoff.

Can we use this to obtain the tradeoff for Tseitin?

- Not obviously — Tseitin is exponentially hard for Resolution!
    - No small tradeoff to begin with.

Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Tradeoffs for Res(k), Cutting Planes follow by **lifting** the Resolution tradeoff.

Can we use this to obtain the tradeoff for Tseitin?

- Not obviously — Tseitin is exponentially hard for Resolution!
    - No small tradeoff to begin with.
- Need a more direct approach.

Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\mathrm{depth}(\Pi) \cdot \log \mathrm{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Tradeoffs for Res(k), Cutting Planes follow by **lifting** the Resolution tradeoff.

Can we use this to obtain the tradeoff for Tseitin?

- Not obviously — Tseitin is exponentially hard for Resolution!
  - No small tradeoff to begin with.
- Need a more direct approach.

Theorem [FGI+21]: Any (Semantic) Cutting Planes refutation of Tseitin requires depth $\Omega(n)$.

Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Tradeoffs for Res(k), Cutting Planes follow by **lifting** the Resolution tradeoff.

Can we use this to obtain the tradeoff for Tseitin?

- Not obviously — Tseitin is exponentially hard for Resolution!
  - No small tradeoff to begin with.
- Need a more direct approach.

Extends [BGHMP06]

Theorem [FGI+21]: Any (Semantic) Cutting Planes refutation of Tseitin requires depth $\Omega(n)$.

Supercritical Size-Depth Tradeoffs

Theorem [FPR22]: Supercritical size/depth tradeoff for Resolution, Res(k), Cutting Planes.

For $c \geq 1, \varepsilon > 0$ there is a CNF formula F on n variables such that

1. There is a Res-proof of size $n^c \cdot 2^{O(c)}$
2. Any Res proof Π with $size(\Pi) \leq \exp(o(n^{1-\varepsilon}/c))$ satisfies $\text{depth}(\Pi) \cdot \log \text{size}(\Pi) = \Omega\left(\frac{n^c}{c \log n}\right)$

Tradeoffs for Res(k), Cutting Planes follow by **lifting** the Resolution tradeoff.

Can we use this to obtain the tradeoff for Tseitin?

- Not obviously — Tseitin is exponentially hard for Resolution!
  - No small tradeoff to begin with.
- Need a more direct approach.

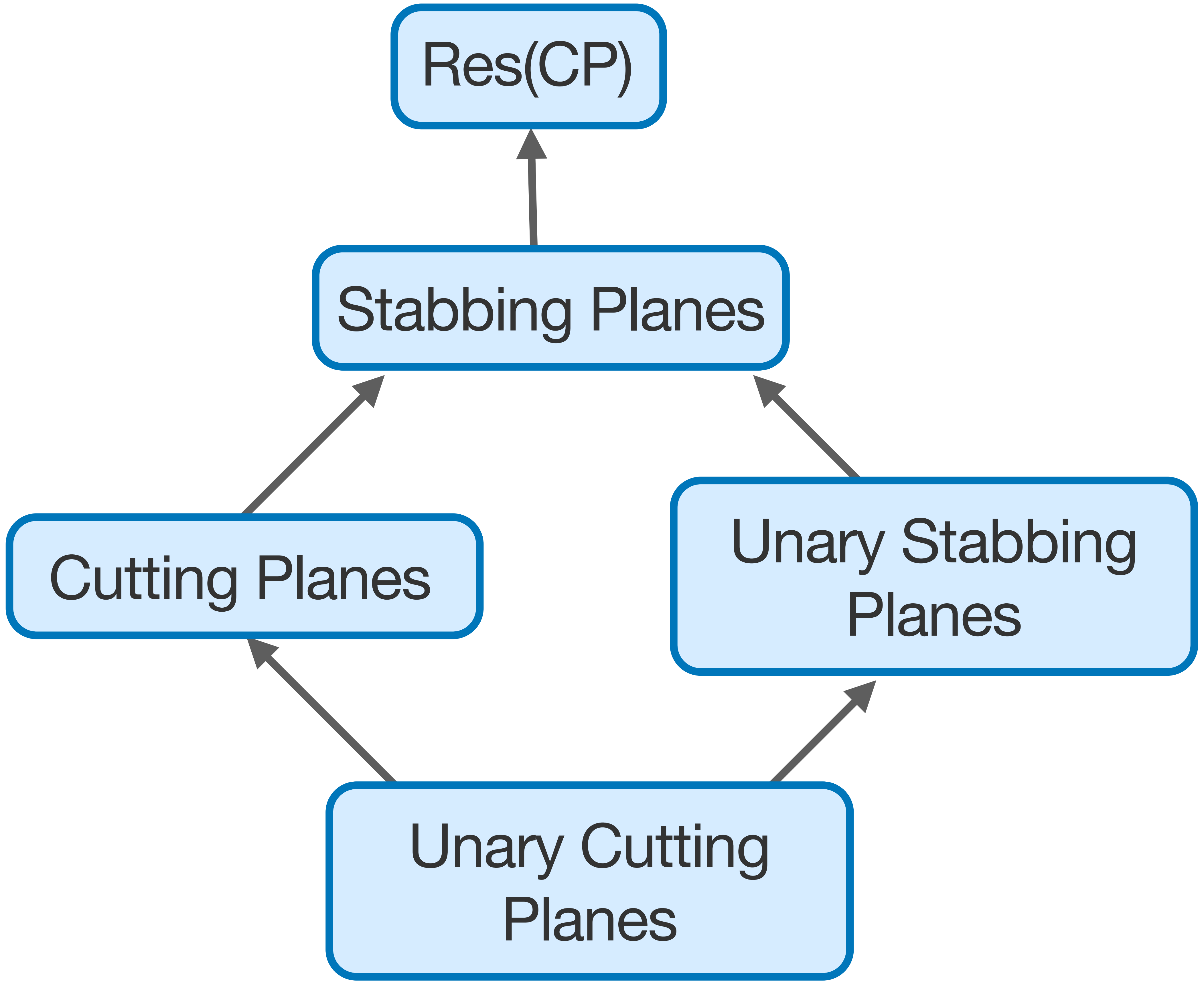
Extends [BGHMP06]

Theorem [FGI+21]: Any (Semantic) Cutting Planes refutation of Tseitin requires depth $\Omega(n)$.

- Uses a geometric argument.

Open Problems

- Prove or disprove the conjecture!
- Can we improve the simulation to **high coefficient** Stabbing Planes?
  - Or, alternatively, can we separate **high coefficient** Stabbing Planes from **low coefficient** Stabbing Planes?
- A generalization of Stabbing Planes to dag-like proofs is called Res(CP).
  - Can Stabbing Planes simulate Res(CP)?
  - [ABE02] Cutting Planes cannot simulate Res(CP)



Thanks!

Shrijver Lemma

Lemma [Schrijver80]: If there is a refutation of a face $P \cap \{ax \leq b\}$ in Cutting Planes then there is a Cutting Planes derivation of $P \cap \{ax \geq b + 1\}$ from P of the same size!

Idea: Since all points in $P \cap \{ax = b\}$ lie on the line $ax \geq b$, we can **rotate** each CG-cut so that it only depends on P and $ax \geq b$ (no longer depends on $ax \leq b$).