Automatic Rectangular Refinement of Affine Hybrid Automata

Laurent Doyen

ULB

Jean-François Raskin

ULB

Tom Henzinger

EPFL

FORMATS 2005 – Sep 27th - Uppsala

ULB UNIVERSITÉ LIBRE DE BRUXELLES

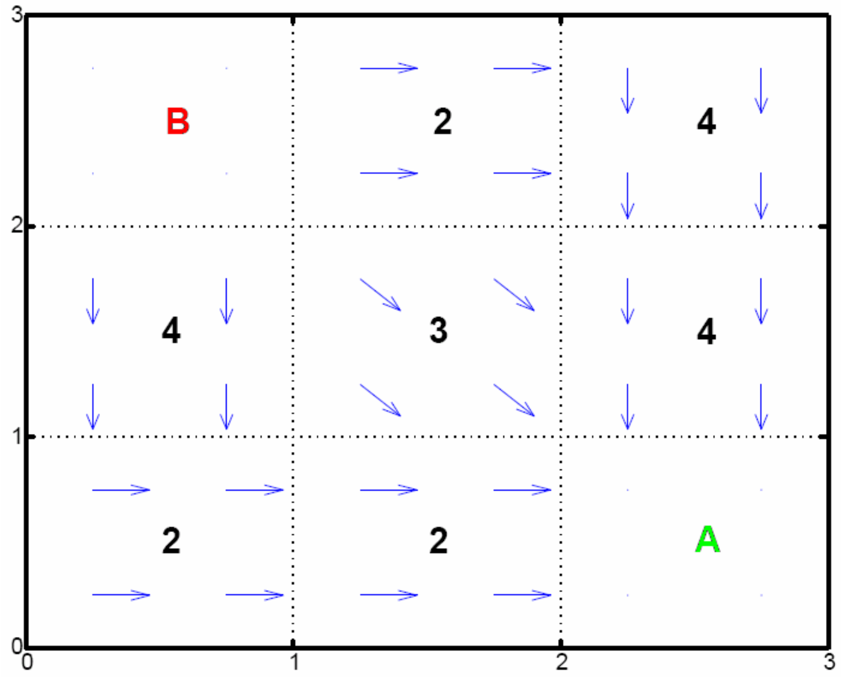
EPFL ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Overview

- Automatic analysis of affine hybrid systems

Overview

- Automatic analysis of affine hybrid systems
- Example:

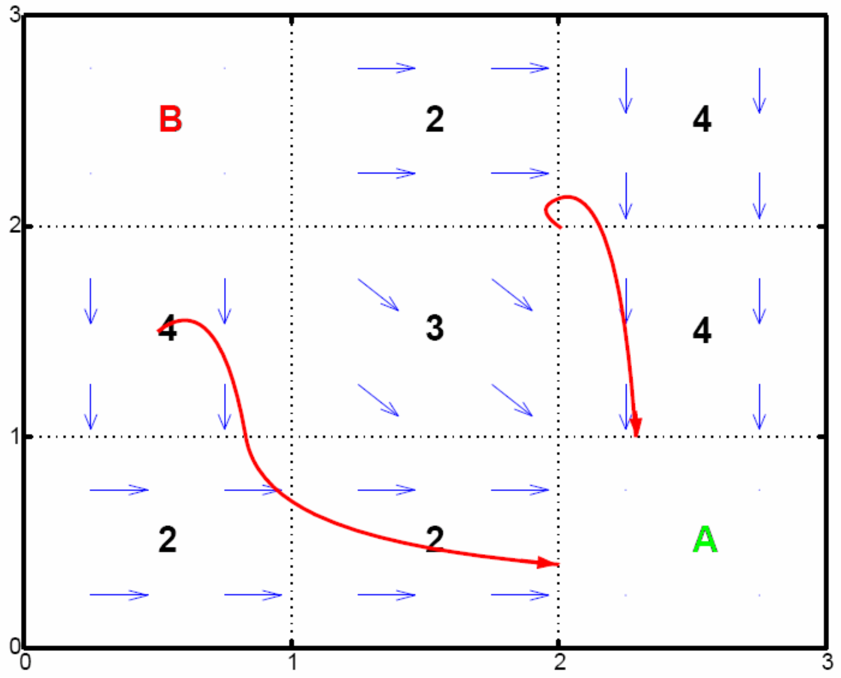


Navigation Benchmark

$$\begin{cases} \dot{x} = v \\ \dot{v} = A(v - v_d) \end{cases}$$

Overview

- Automatic analysis of **affine hybrid** systems
- Example:

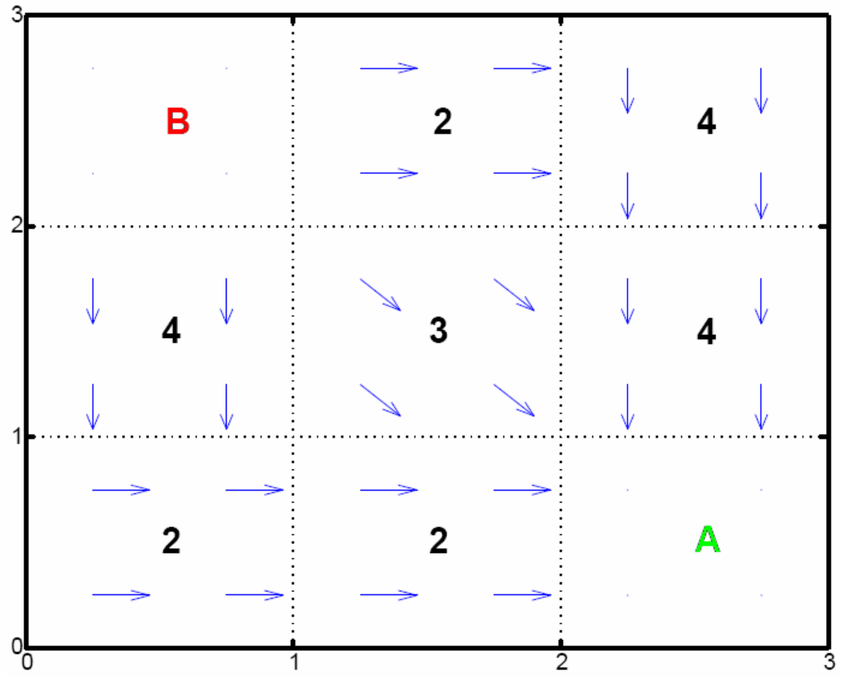


Two trajectories

$$\begin{cases} \dot{x} = v \\ \dot{v} = A(v - v_d) \end{cases}$$

Overview

- Automatic analysis of **affine** hybrid systems
- Example:



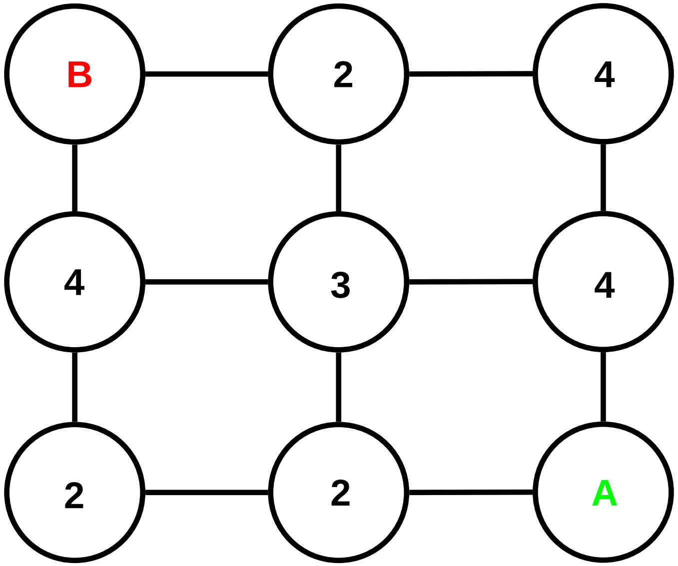
Navigation Benchmark

$$\begin{cases} \dot{x} = v \\ \dot{v} = A(v - v_d) \end{cases}$$

Affine dynamics

Overview

- Automatic analysis of affine **hybrid** systems
- Example:



$$\begin{cases} \dot{x} = v \\ \dot{v} = A(v - v_d) \end{cases}$$

Discrete states + **Affine dynamics**

Reminder

- Some classes of hybrid automata:
  - Timed automata ($\dot{x} = 1$)
  - Rectangular automata ($\dot{x} \in [a, b]$)
  - Linear automata ($\sum a_i \dot{x}_i \sim b$)

Reminder

- Some classes of hybrid automata:
  - Timed automata ($\dot{x} = 1$)
  - Rectangular automata ($\dot{x} \in [a, b]$)
  - Linear automata ($\sum a_i \dot{x}_i \sim b$)

Limit for decidability of Language Emptiness

Reminder

- Some classes of hybrid automata:
  - Timed automata ($\dot{x} = 1$)
  - Rectangular automata ($\dot{x} \in [a, b]$)
  - Linear automata ($\sum a_i \dot{x}_i \sim b$)
  - Affine automata ($\sum a_i \dot{x}_i + b_i x_i \sim c$)
  - Polynomial automata ($p(\dot{x}_i, x_i) \sim c$)
  - etc.

Limit for decidability of Language Emptiness

Reminder

- Some classes of hybrid automata:
  - Timed automata ($\dot{x} = 1$)
  - Rectangular automata ($\dot{x} \in [a,b]$)
  - Linear automata ($\sum a_i \dot{x}_i \sim b$)
  - Affine automata ($\sum a_i \dot{x}_i + b_i x_i \sim c$)
  - Polynomial automata ($p(\dot{x}_i, x_i) \sim c$)
  - etc.

Limit for symbolic computation of Post with HyTech

Limit for decidability of Language Emptiness

Methodology

- Affine automaton A and set of states Bad
- Check that Reach(A) $\cap$ Bad = Ø

Methodology

- Affine automaton A and set of states Bad
- Check that Reach(A) $\cap$ Bad = Ø
- Affine dynamics is too complex ?

→ Abstract it !

Methodology

- Affine automaton A and set of states Bad
- Check that Reach(A) $\cap$ Bad = Ø
- Affine dynamics is too complex ?

➡ Abstract it !

- Abstraction is too coarse ?

➡ Refine it !

HOW ?

Methodology

- 1. Abstraction: over-approximation

Affine dynamics → Rectangular dynamics

$$\dot{x} = 2 - x$$

$$0 \le x \le 3$$

Methodology

- 1. Abstraction: over-approximation

Affine dynamics → Rectangular dynamics

$$\dot{x} = 2 - x \qquad 0 \le x \le 3$$

→

$$\dot{x} \in [-1,2] \qquad 0 \le x \le 3$$

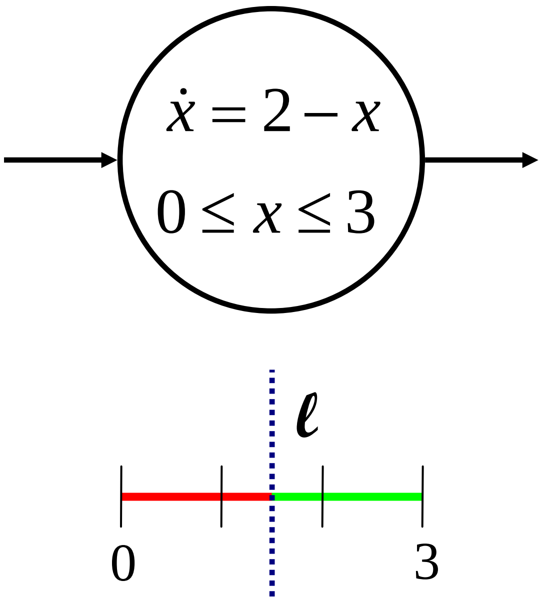
Let $\begin{cases} f(x) = 2\text{-}x \\ \mathrm{Inv} = \{0 \le x \le 3\} \end{cases}$

Then $[-1,2] = [\min_{x \in \mathrm{Inv}} f(x), \max_{x \in \mathrm{Inv}} f(x)]$

Methodology

- 2. Refinement: split locations by a line cut

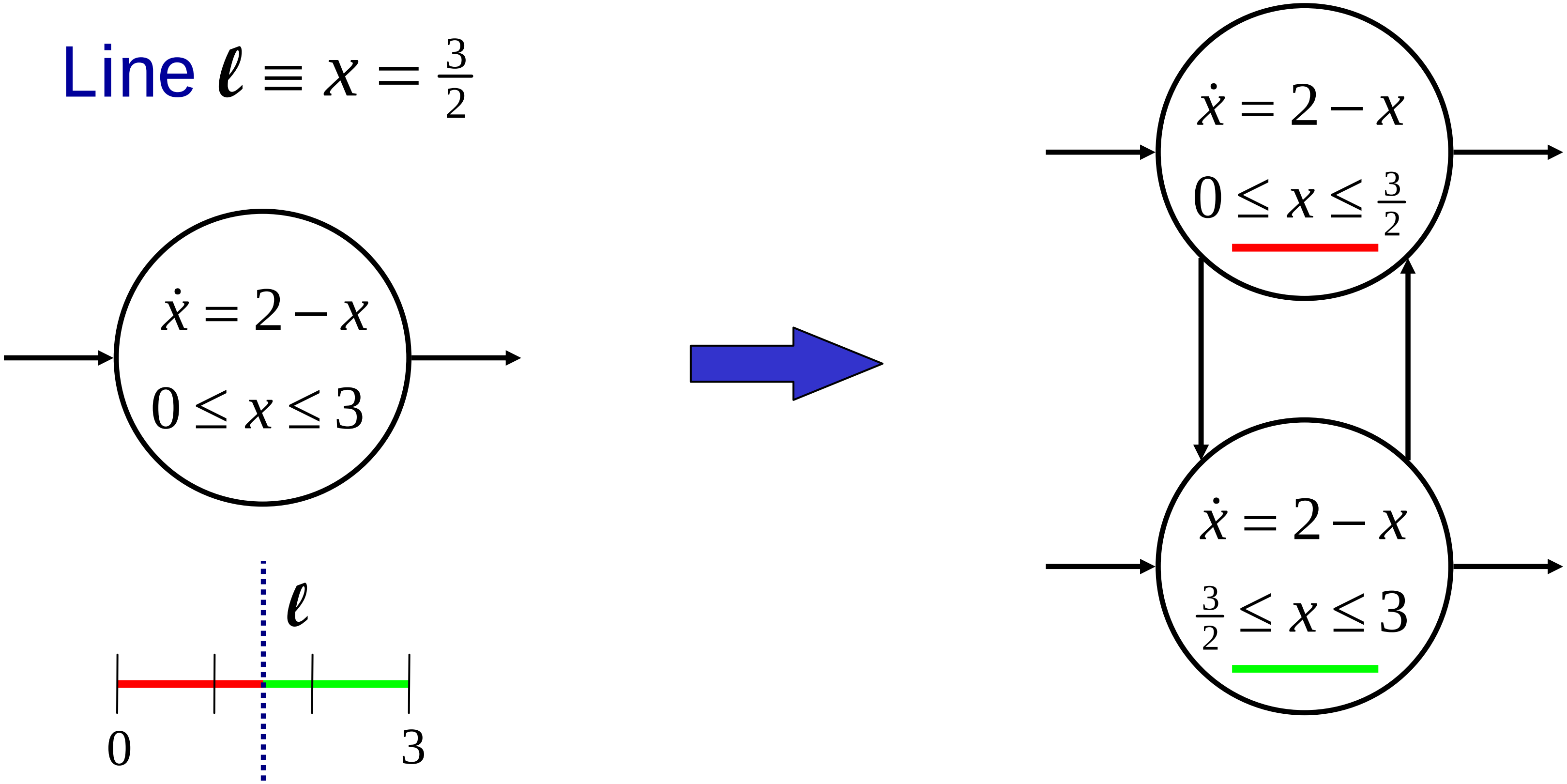
Line $\boldsymbol{\ell} \equiv x = \frac{3}{2}$



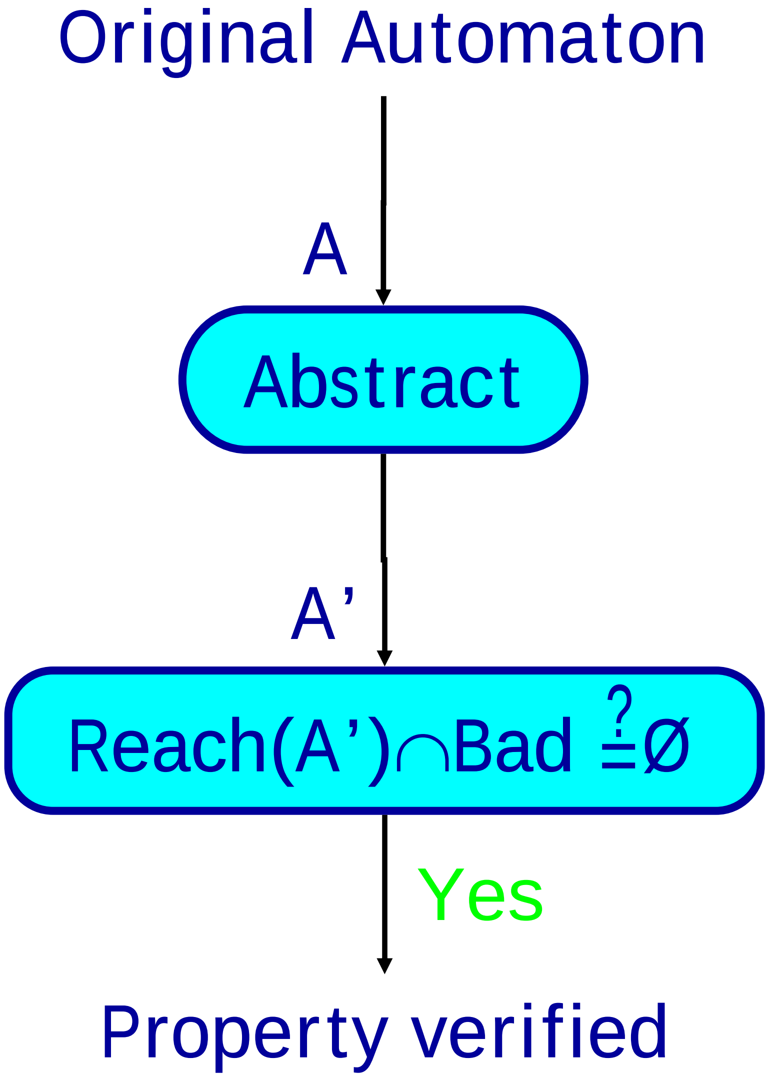
Methodology

- 2. Refinement: split locations by a line cut

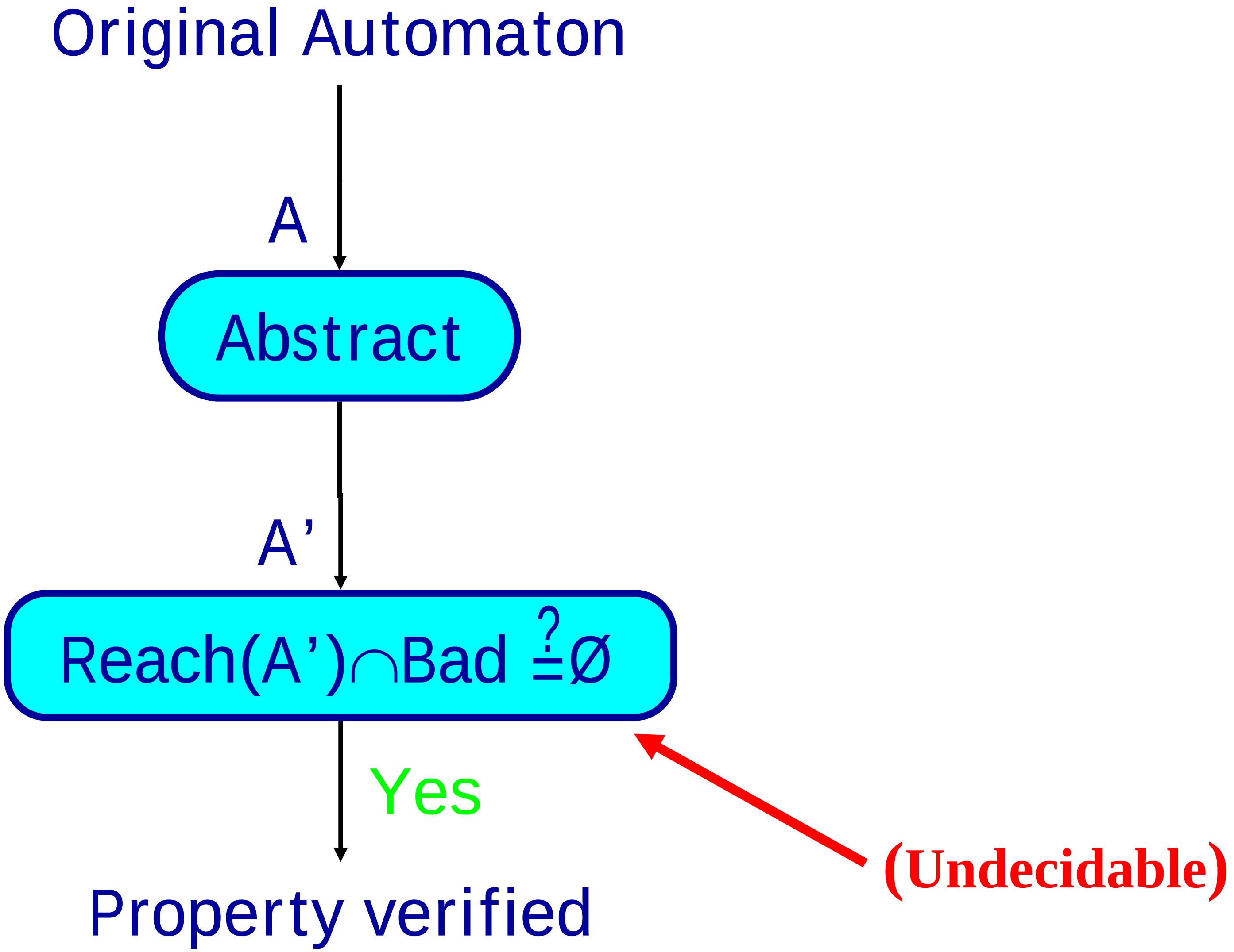
Line $\ell \equiv x = \frac{3}{2}$



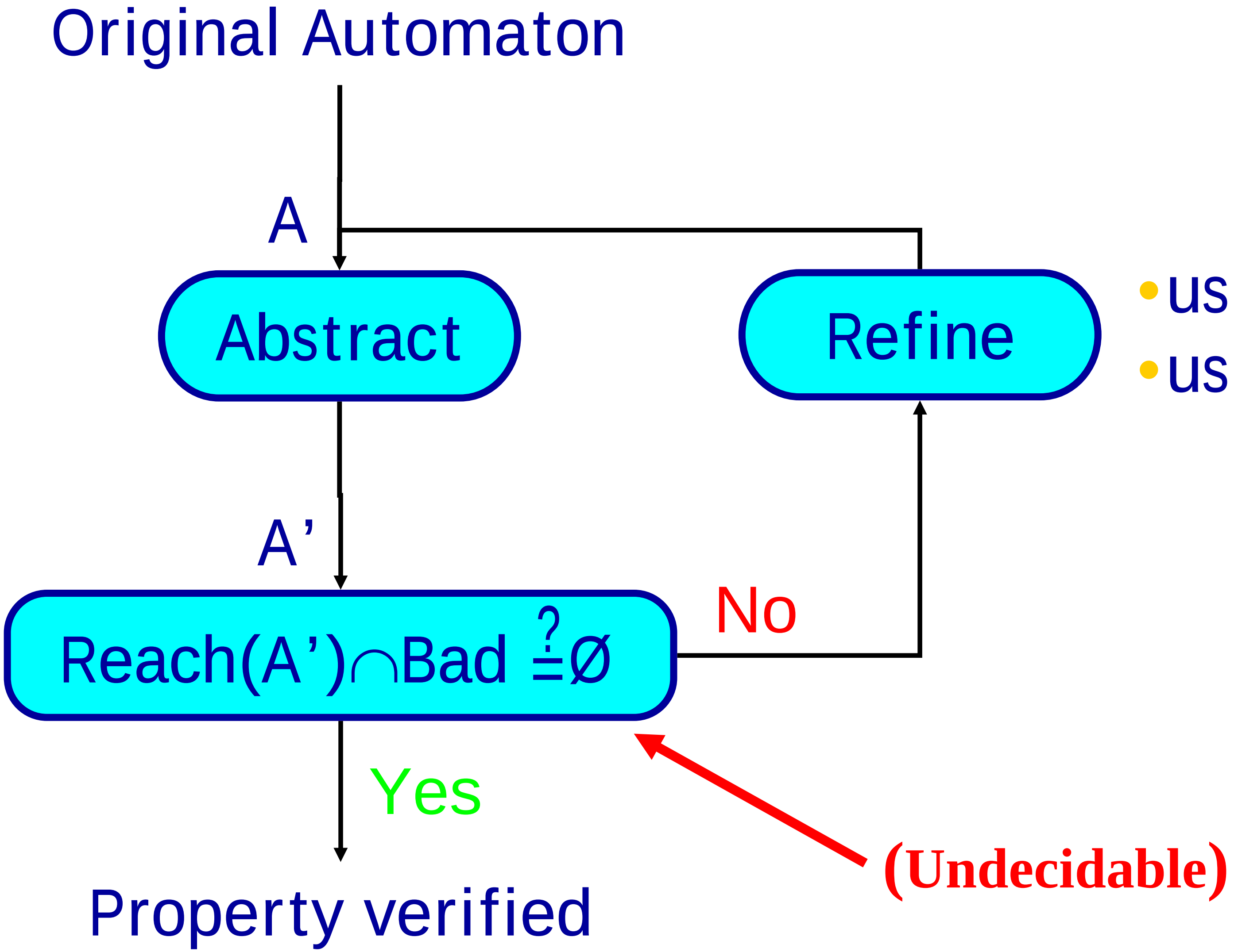
Methodology



Methodology



Methodology



- using Reach(A')
- using Pre*(Bad)

Refinement

- 2. Refinement: split locations by a line cut
- Which location(s) ?
  - Loc_1 = Locations reachable in the last step
  - Loc_2 = Reachable locations that can reach Bad
  - Better: replace the state space by Loc_2

Refinement

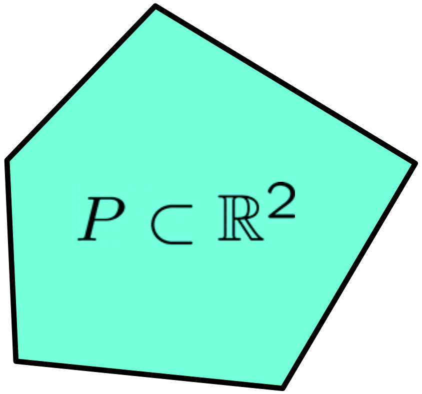
- 2. Refinement: split locations by a line cut
- Which location(s) ?
  - Loc_1 = Locations reachable in the last step
  - Loc_2 = Reachable locations that can reach Bad
  - Better: replace the state space by Loc_2
- Which line cut ?
  - The best cut for some *criterion* characterizing the *goodness* of the resulting approximation.

Notations

$$\dot{x} = f(x, y)$$

$$\dot{y} = g(x, y)$$

$$(x, y) \in P$$

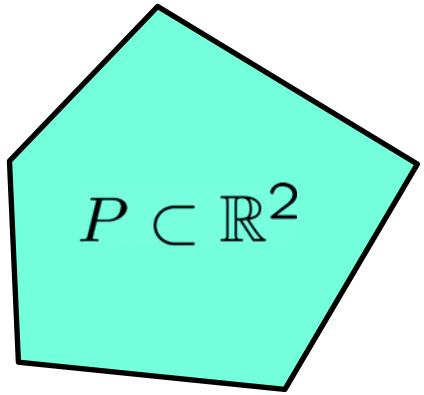


Notations

$$\dot{x} = f(x, y)$$

$$\dot{y} = g(x, y)$$

$$(x, y) \in P$$



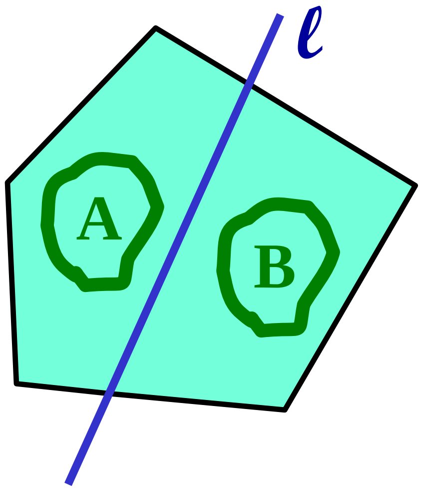
$$[\dot{x}_{\text{min}}, \dot{x}_{\text{max}}] = f(P) \qquad r_x = \dot{x}_{\text{max}} - \dot{x}_{\text{min}}$$

$$[\dot{y}_{\text{min}}, \dot{y}_{\text{max}}] = g(P) \qquad r_y = \dot{y}_{\text{max}} - \dot{y}_{\text{min}}$$

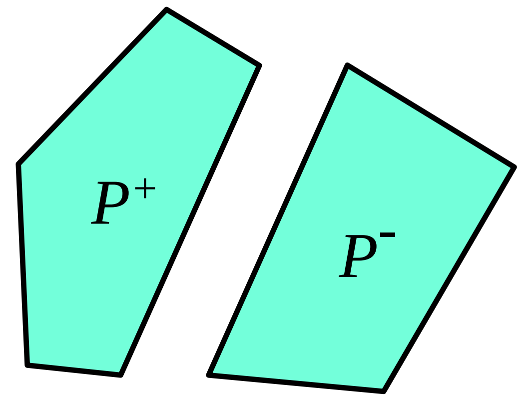
Notations

$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$

$$(x, y) \in P$$



$$P/\ell = \langle P^+, P^- \rangle$$



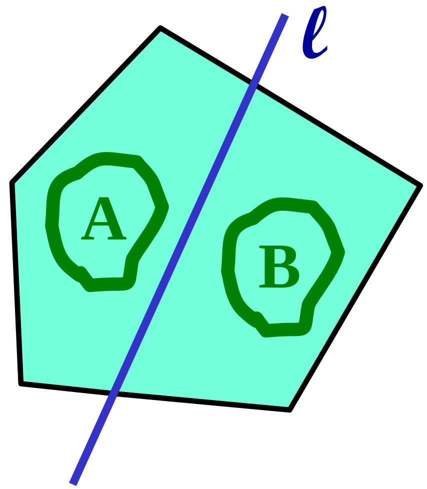
Definition Let $A \subseteq P$ and $B \subseteq P$.
We say that ℓ *separates* A and B
if $A \subseteq P^+$ and $B \subseteq P^-$.

Notations

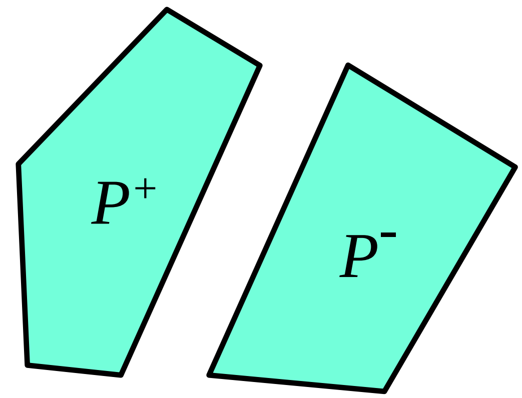
$$\dot{x} = f(x, y)$$

$$\dot{y} = g(x, y)$$

$$(x, y) \in P$$



$$P/\ell = \langle P^+, P^- \rangle$$



Definition Let $A \subseteq P$ and $B \subseteq P$. We say that ℓ *separates* A and B if $A \subseteq P^+$ and $B \subseteq P^-$.

$[a^+, b^+] = f(P^+)$ $\quad$ $[a^-, b^-] = f(P^-)$

$[c^+, d^+] = g(P^+)$ $\quad$ $[c^-, d^-] = g(P^-)$

$$\text{sizeRange}_x^\sim(P/\ell) = b^\sim - a^\sim$$

$$\text{sizeRange}_y^\sim(P/\ell) = d^\sim - c^\sim$$

$$\sim \in \{+, -\}$$

Goodness of a cut

- A good cut should minimize
  - $\max_{x \in \mathsf{Var}, \sim \in \{+,-\}} \mathsf{sizeRange}_x^{\sim}(P/\ell)$?

Goodness of a cut

- A good cut should minimize
  - $$\max_{x \in \mathsf{Var}, \sim \in \{+,-\}} \mathsf{sizeRange}_x^{\sim}(P/\ell)$$?
  - $$\sum_{x \in \mathsf{Var}, \sim \in \{+,-\}} \mathsf{sizeRange}_x^{\sim}(P/\ell)$$?

Goodness of a cut

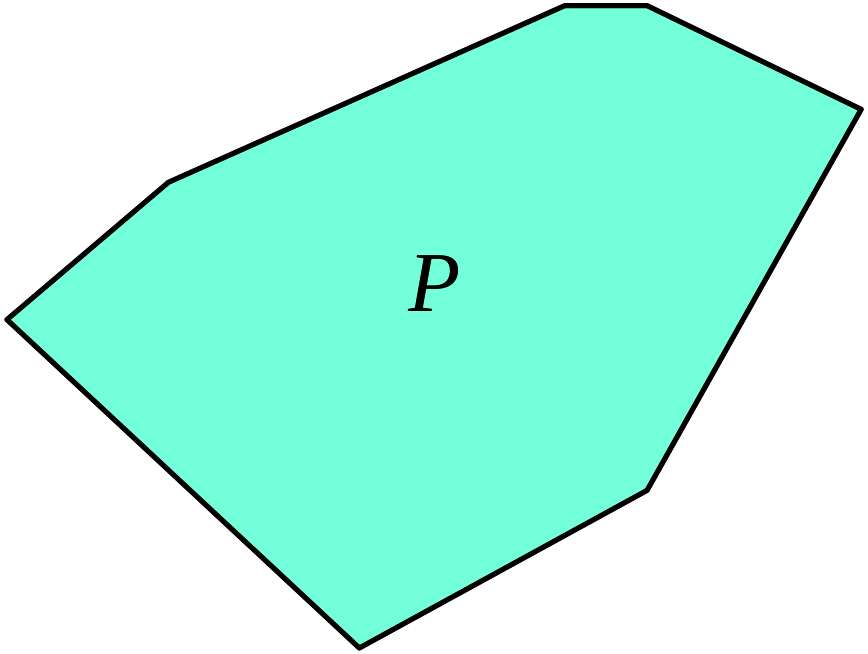
- A good cut should minimize
  - $\max_{x \in \mathsf{Var}, \sim \in \{+,-\}} \mathsf{sizeRange}_x^{\sim}(P/\ell)$?
  - $\sum_{x \in \mathsf{Var}, \sim \in \{+,-\}} \mathsf{sizeRange}_x^{\sim}(P/\ell)$?
  - $\sum_{x \in \mathsf{Var}, \sim \in \{+,-\}} \left(\mathsf{sizeRange}_x^{\sim}(P/\ell)\right)^2$?
  - ...

Goodness of a cut

- A good cut should minimize
  - $\max_{x \in \mathsf{Var}, \sim \in \{+,-\}} \mathsf{sizeRange}_x^{\sim}(P/\ell)$?
  - $\sum_{x \in \mathsf{Var}, \sim \in \{+,-\}} \mathsf{sizeRange}_x^{\sim}(P/\ell)$?
  - $\sum_{x \in \mathsf{Var}, \sim \in \{+,-\}} \left(\mathsf{sizeRange}_x^{\sim}(P/\ell)\right)^2$?
  - ...

Our choice

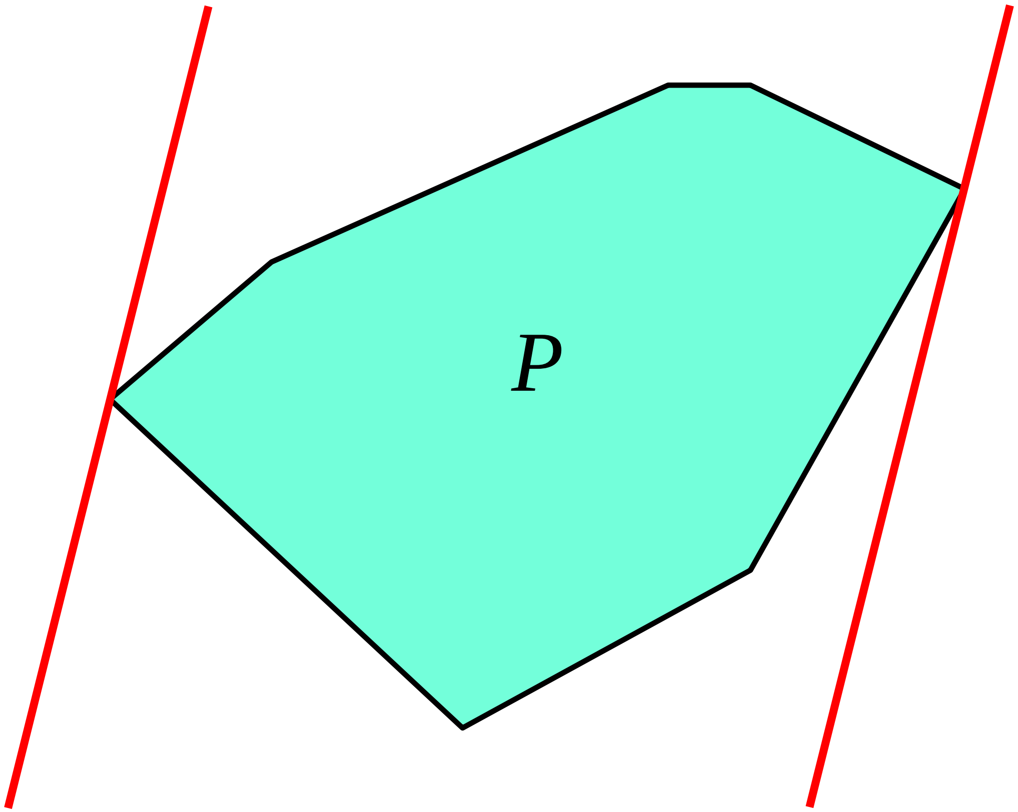
Finding the optimal cut



$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$

$$(x, y) \in P$$

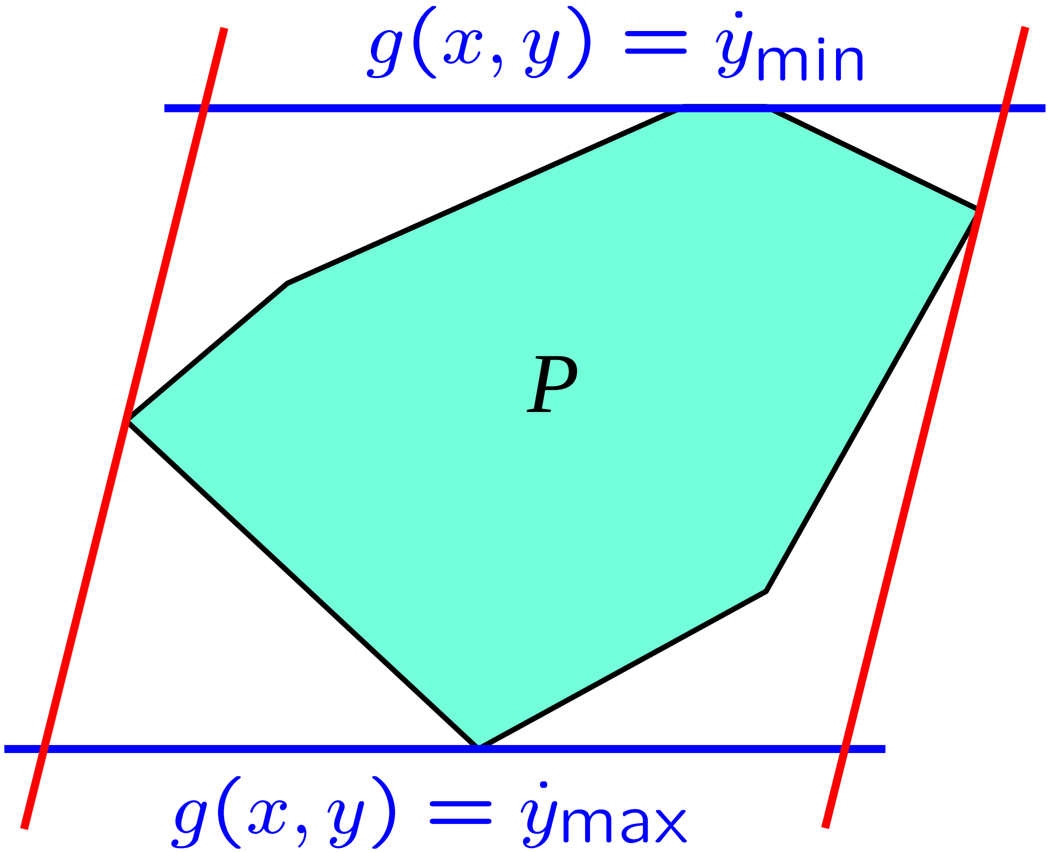
Extremal level sets of f(x,y)



$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$
$$(x, y) \in P$$

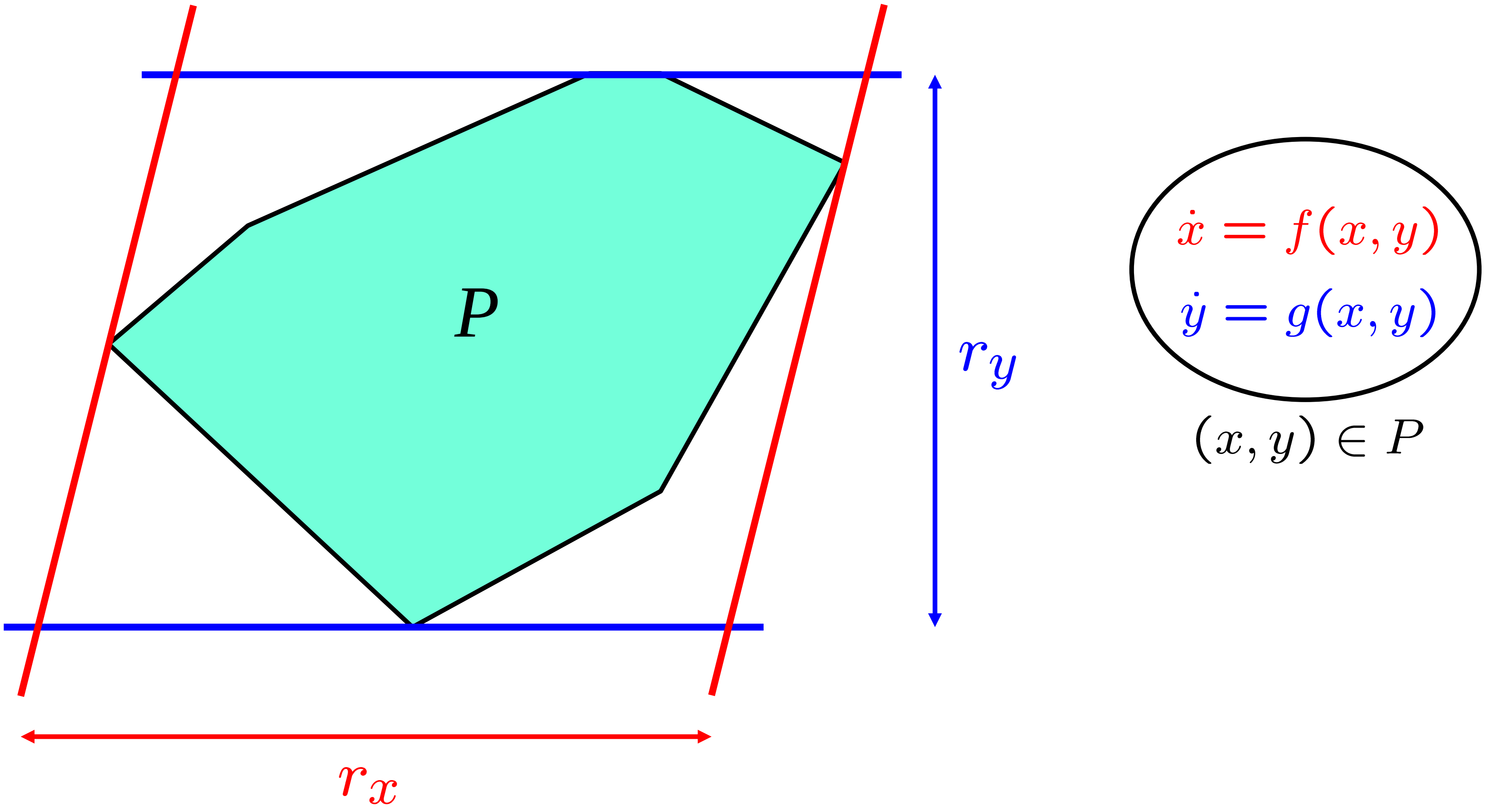
$$f(x, y) = \dot{x}_{\mathsf{min}} \qquad f(x, y) = \dot{x}_{\mathsf{max}}$$

Extremal level sets of g(x,y)



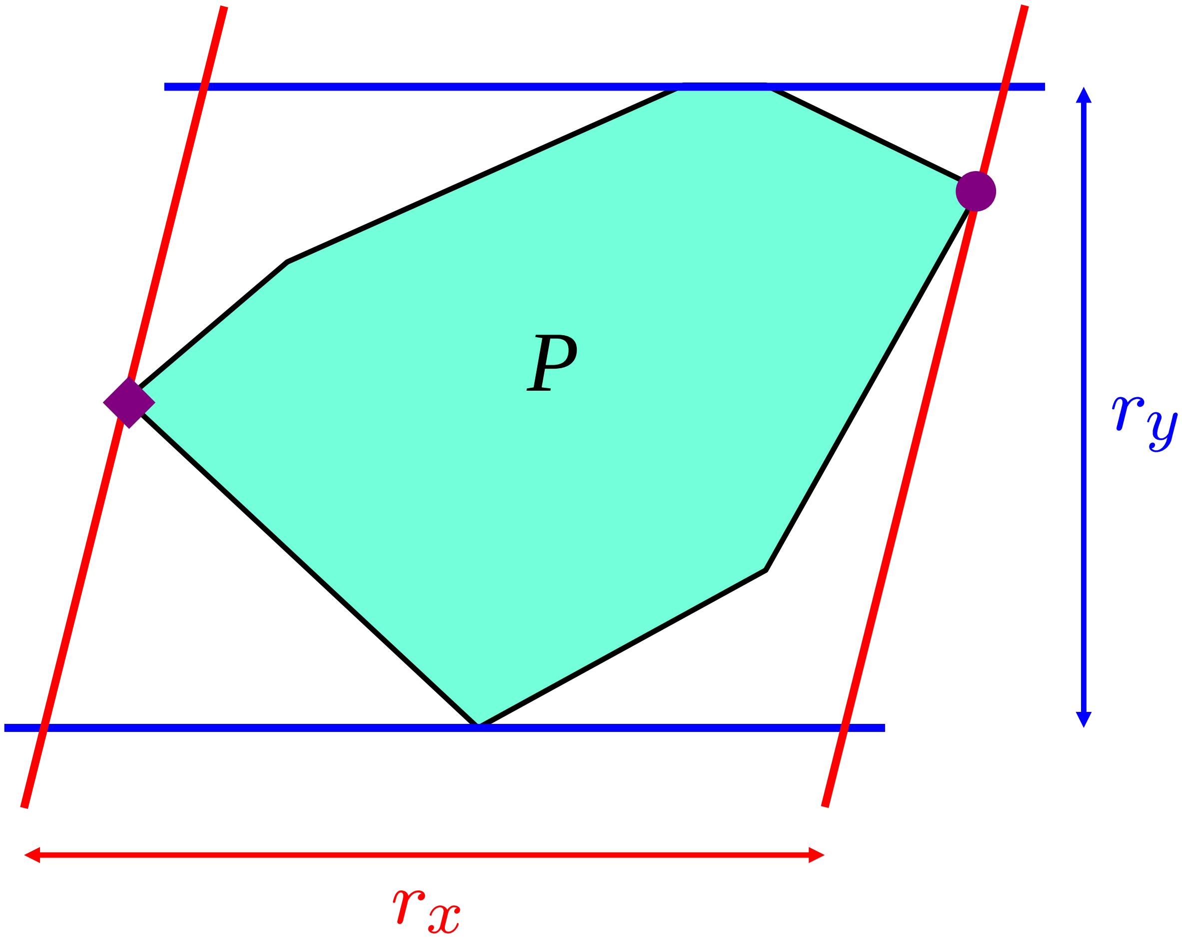
$$\dot{x} = f(x,y)$$
$$\dot{y} = g(x,y)$$
$$(x,y) \in P$$

Example



Assume $r_x > r_y$

Example



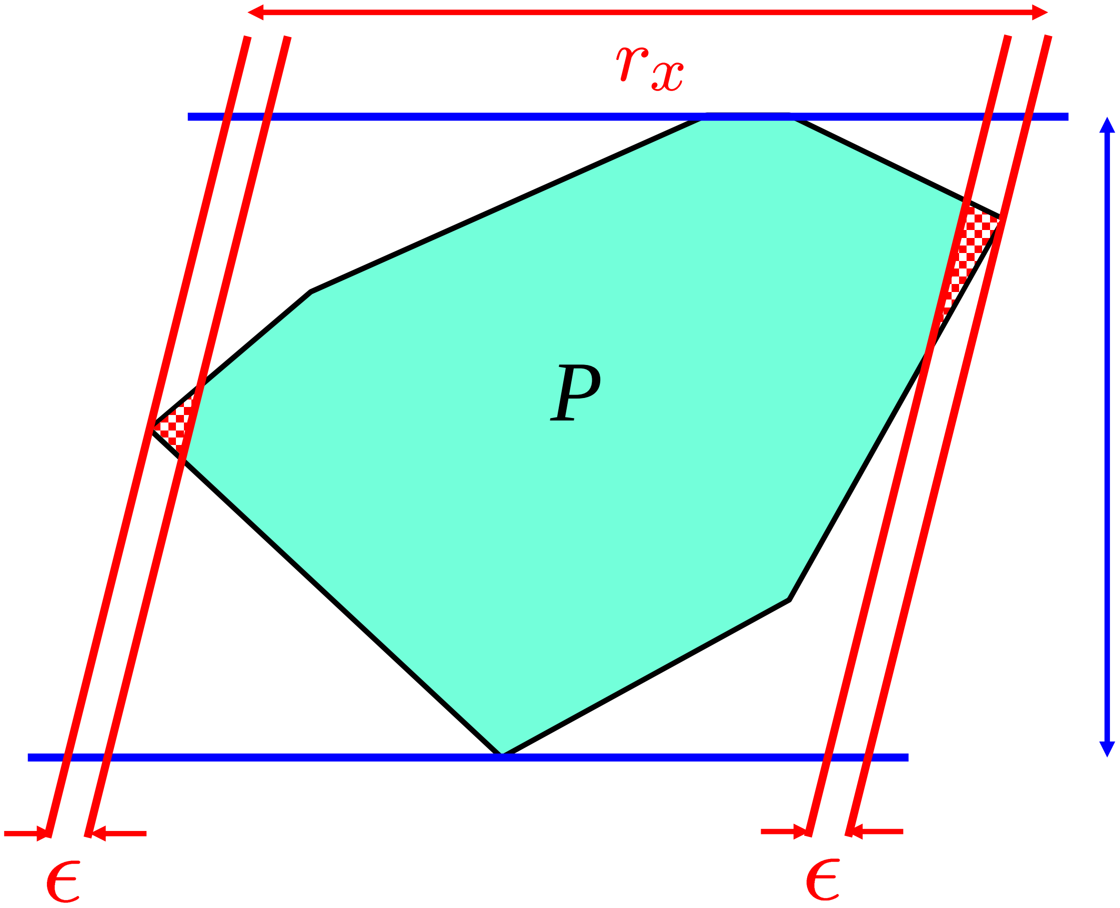
$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$
$$(x, y) \in P$$

Assume $r_x > r_y$

Then any line separating {●} and {◆}

is <u>better</u> than any other line.

Example



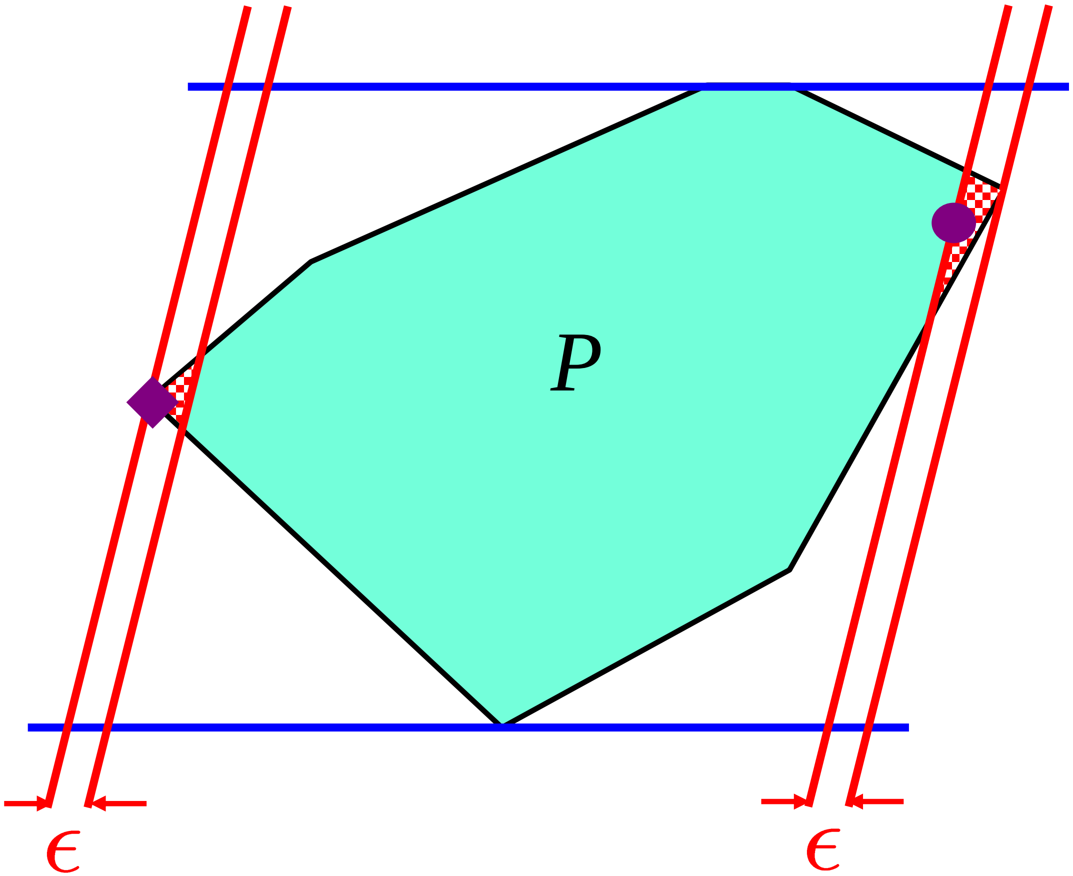
$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$

$(x, y) \in P$

Let $\epsilon > 0$ s.t.

$$r_x - \epsilon > r_y$$

Example



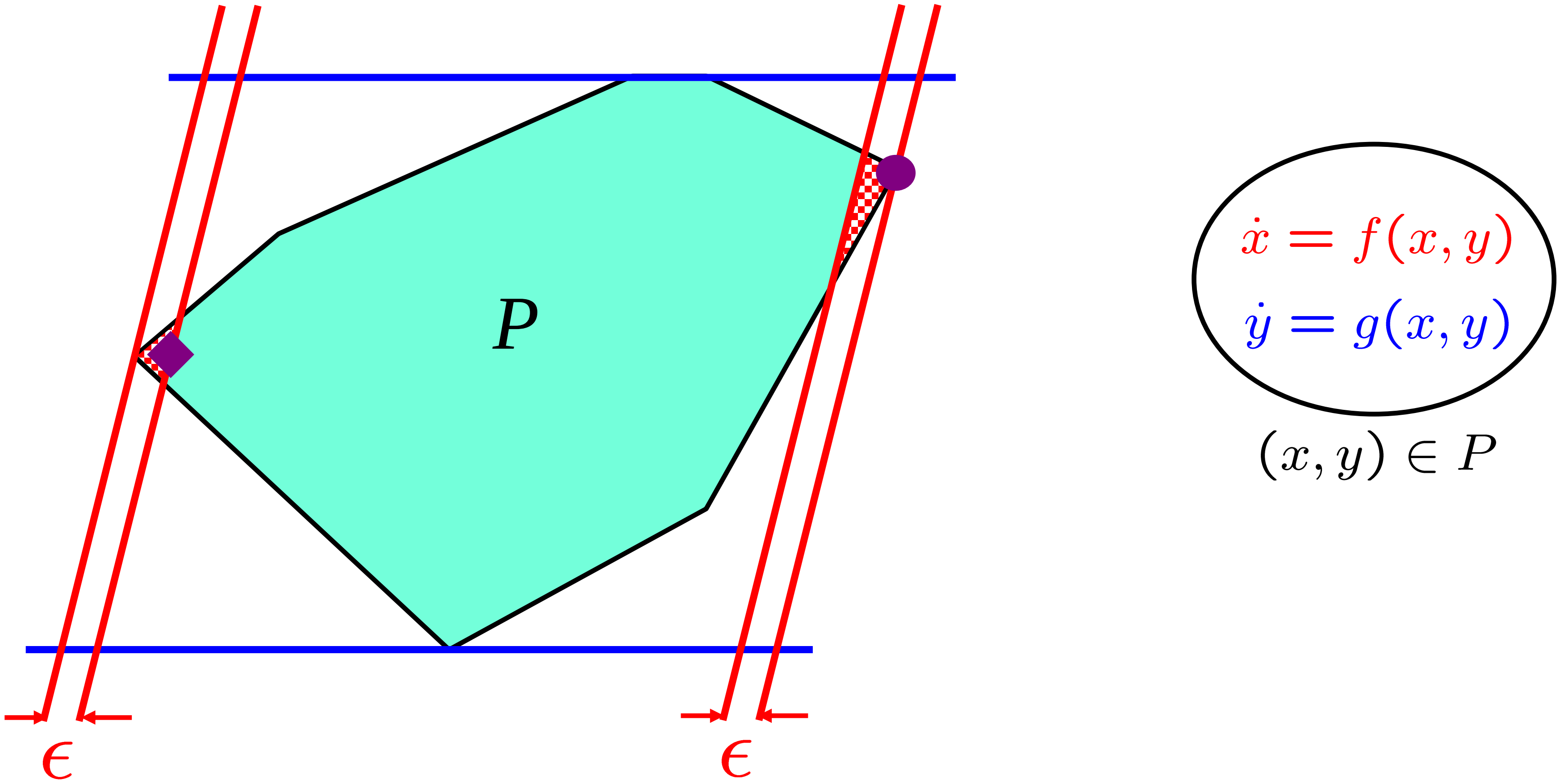
$$\dot{x} = f(x,y)$$
$$\dot{y} = g(x,y)$$
$$(x,y) \in P$$

Let $\epsilon > 0$ s.t.

$$r_x - \epsilon > r_y$$

Any line separating {●} and {◆} is <u>better</u> than any other line.

Example

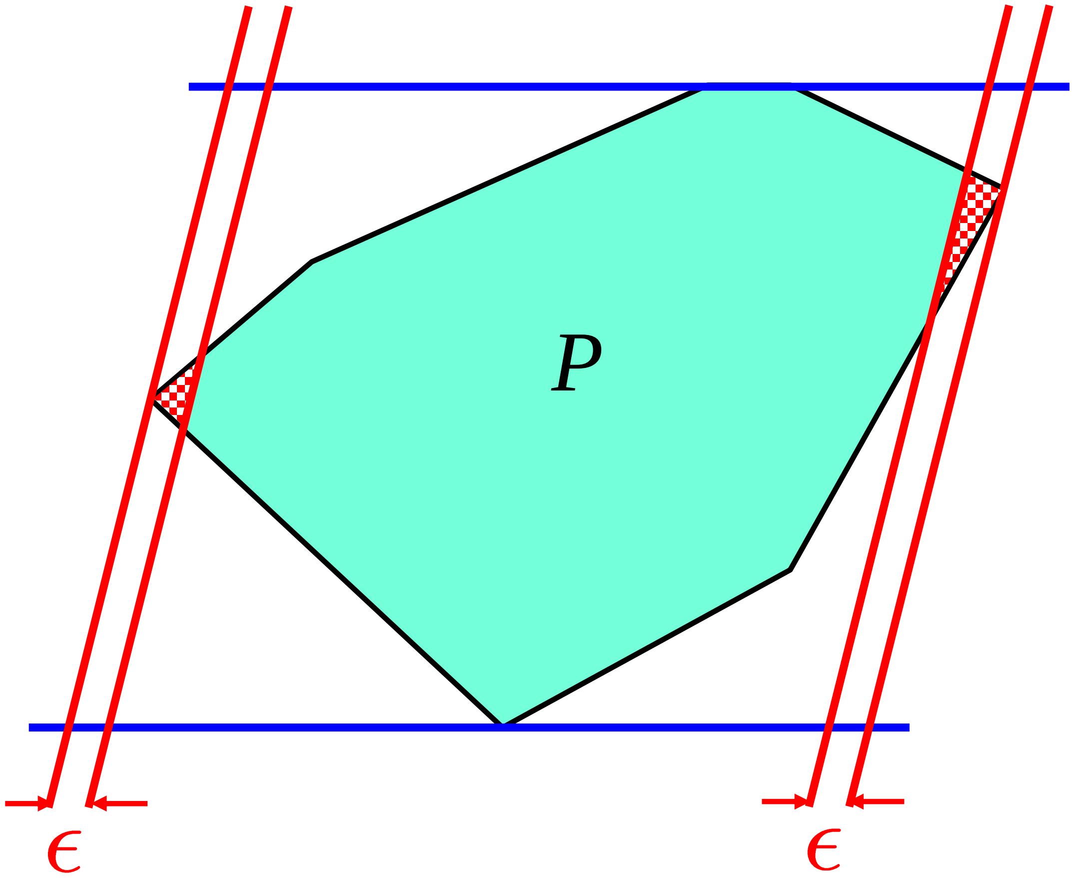


Let $\epsilon > 0$ s.t.

$r_x - \epsilon > r_y$

Any line separating {●} and {◆} is <u>better</u> than any other line.

Example

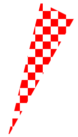


$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$
$$(x, y) \in P$$

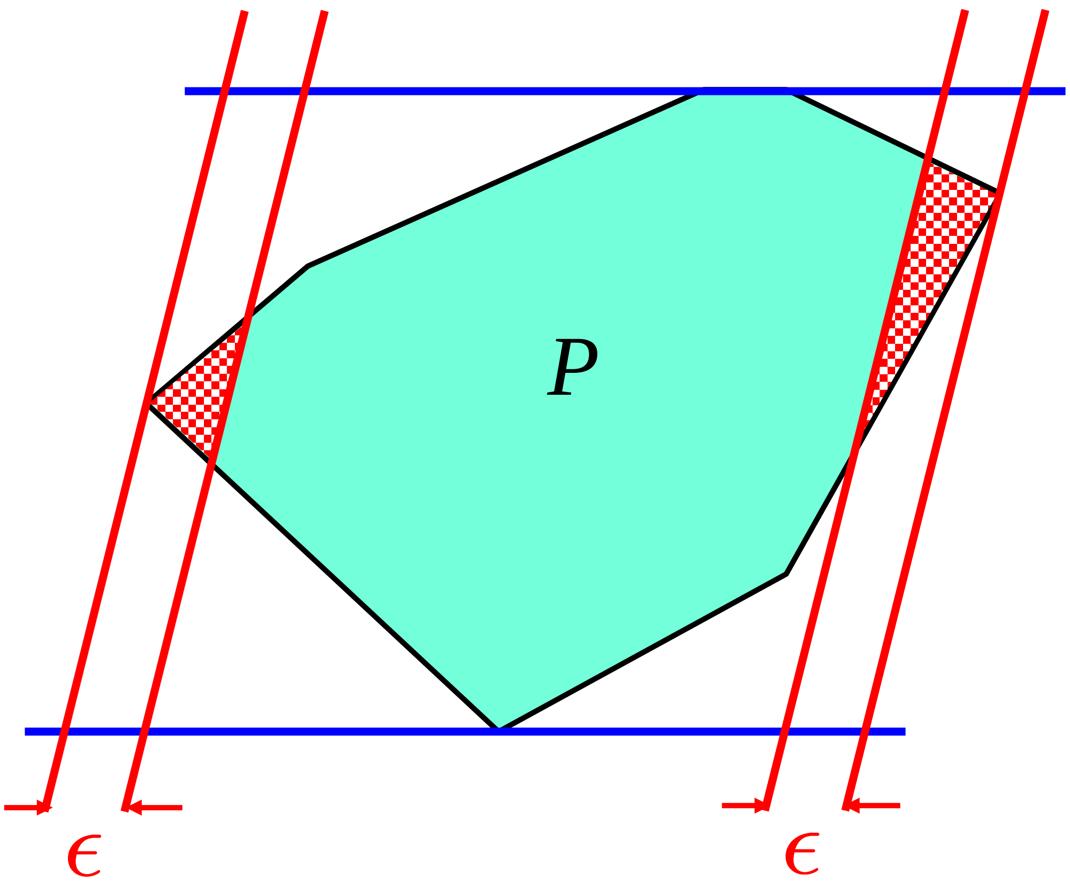
Let $\epsilon > 0$ s.t.

$r_x - \epsilon > r_y$

Thus, for every $\epsilon < r_x - r_y$

the best line separates  and 

Example

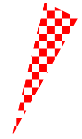


$$\dot{x} = f(x, y)$$

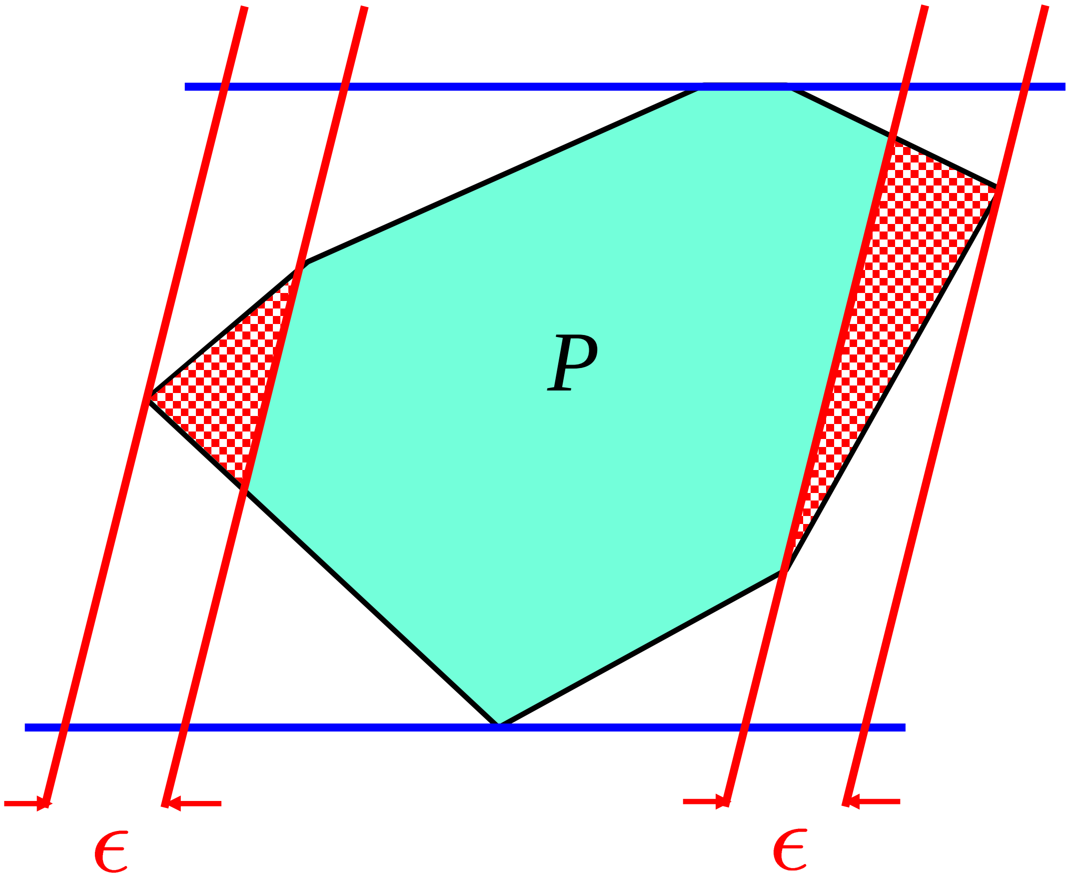
$$\dot{y} = g(x, y)$$

$$(x, y) \in P$$

Thus, for every $\epsilon < r_x - r_y$

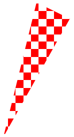
the best line separates  and 

Example

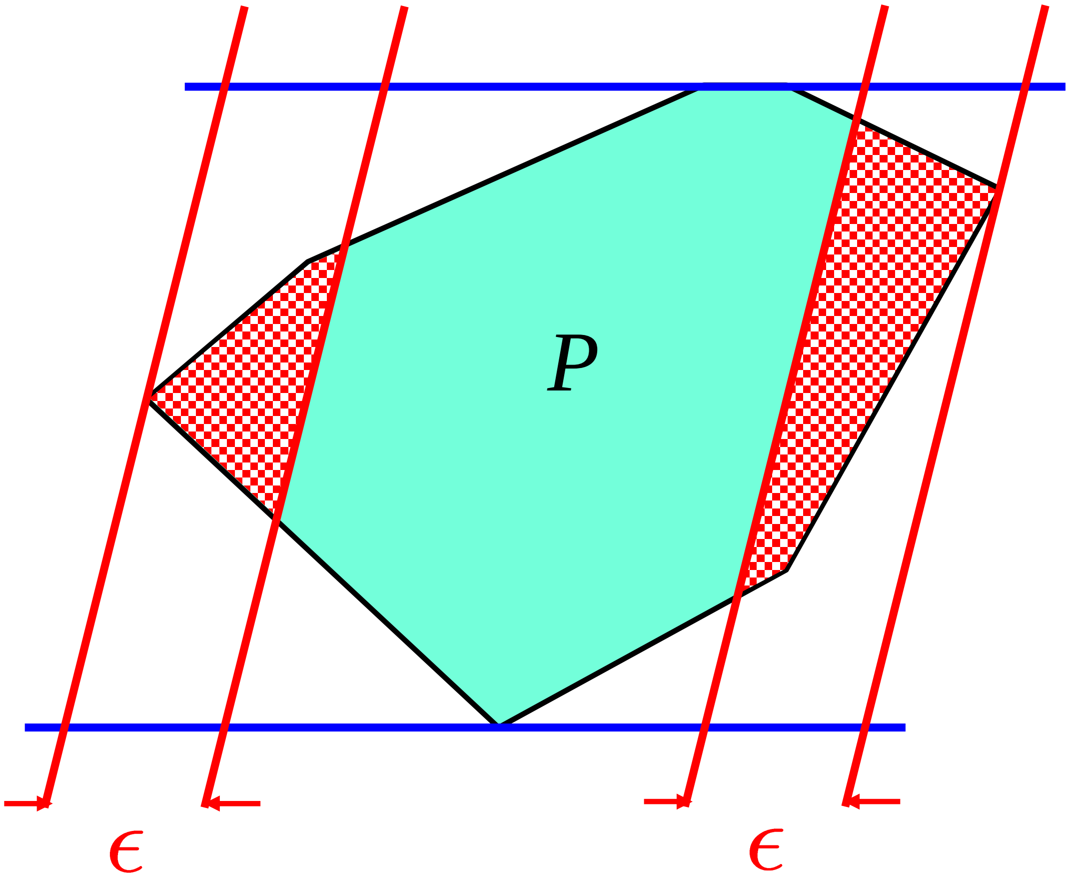


$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$
$$(x, y) \in P$$

Thus, for every $\epsilon < r_x - r_y$

the best line separates  and 

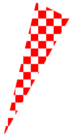
Example



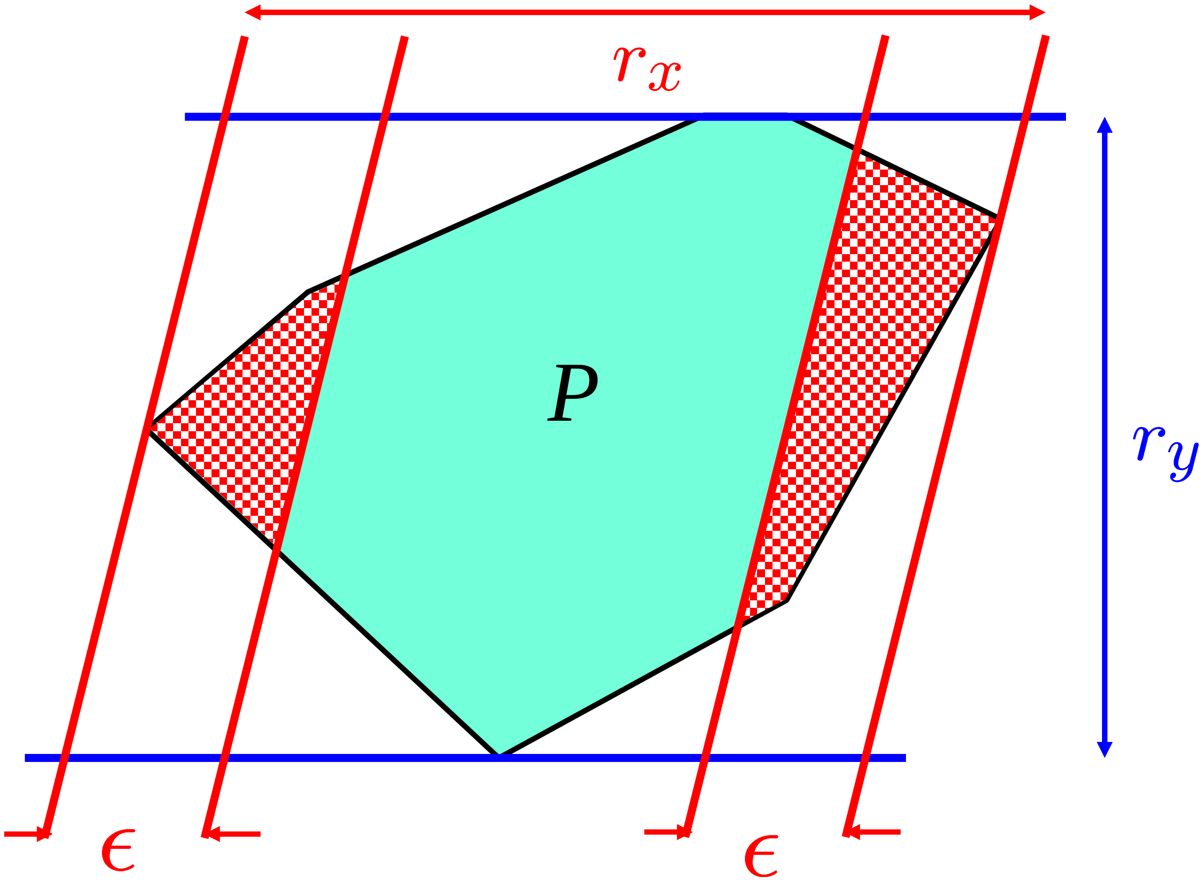
$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$

$(x, y) \in P$

Thus, for every $\epsilon < r_x - r_y$

the best line separates  and 

Example



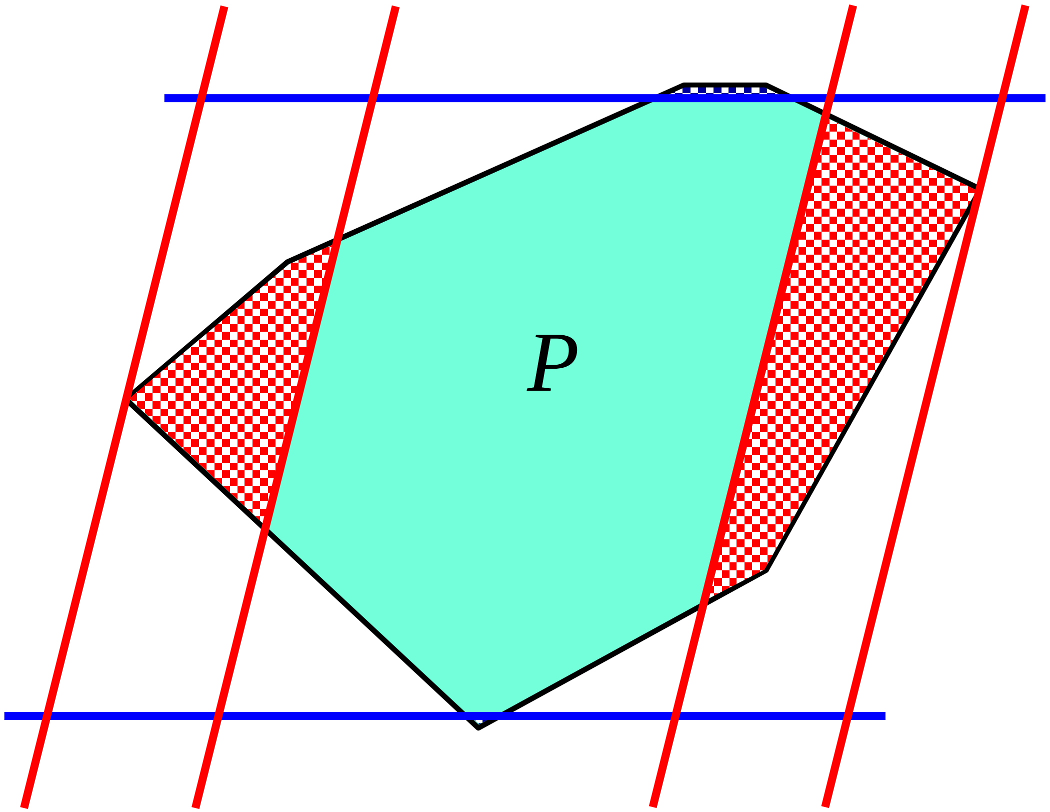
$\dot{x} = f(x, y)$

$\dot{y} = g(x, y)$

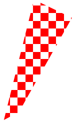
$(x, y) \in P$

When $\epsilon = r_x - r_y$

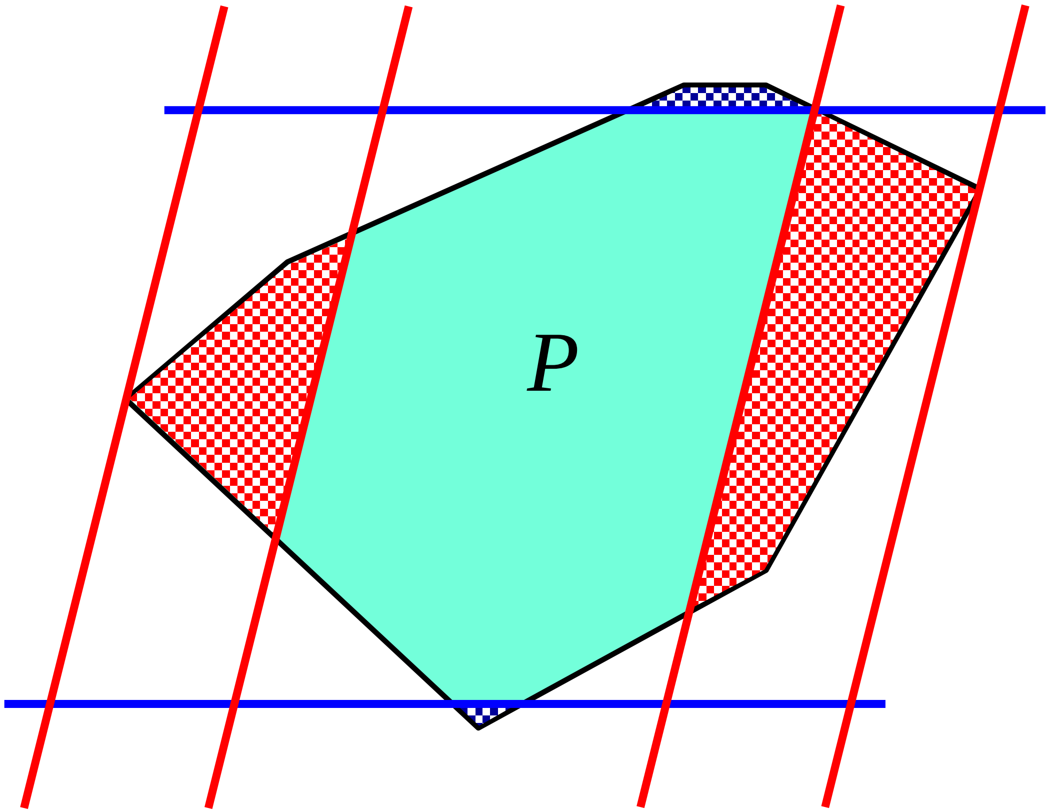
Example



$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$
$$(x, y) \in P$$

When $\epsilon = r_x - r_y$ the best line cut must separate both  from  and  from 

Example

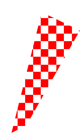


$$\dot{x} = f(x, y)$$

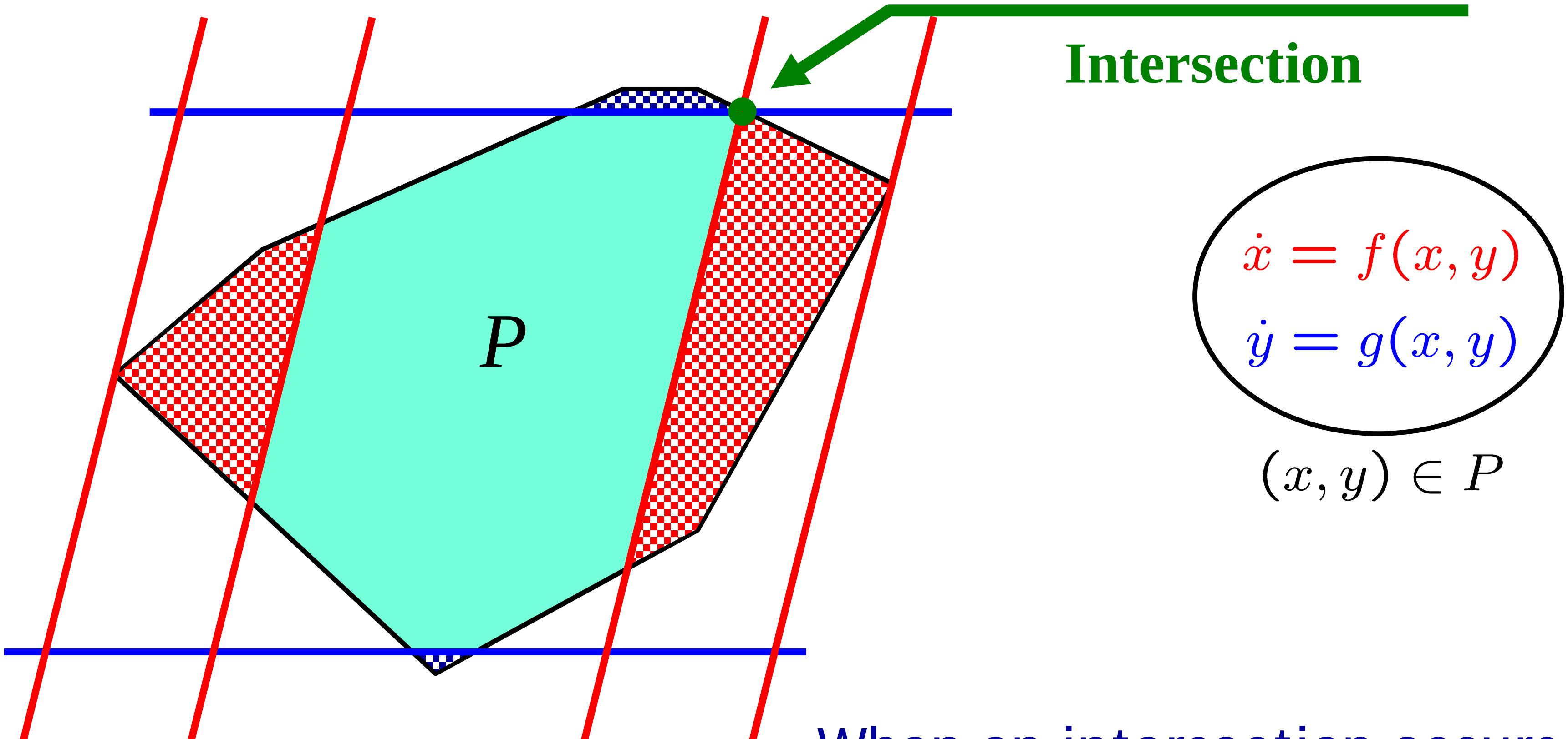
$$\dot{y} = g(x, y)$$

$$(x, y) \in P$$

The best line cut must separate both

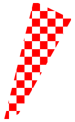
 from  and  from 

Example

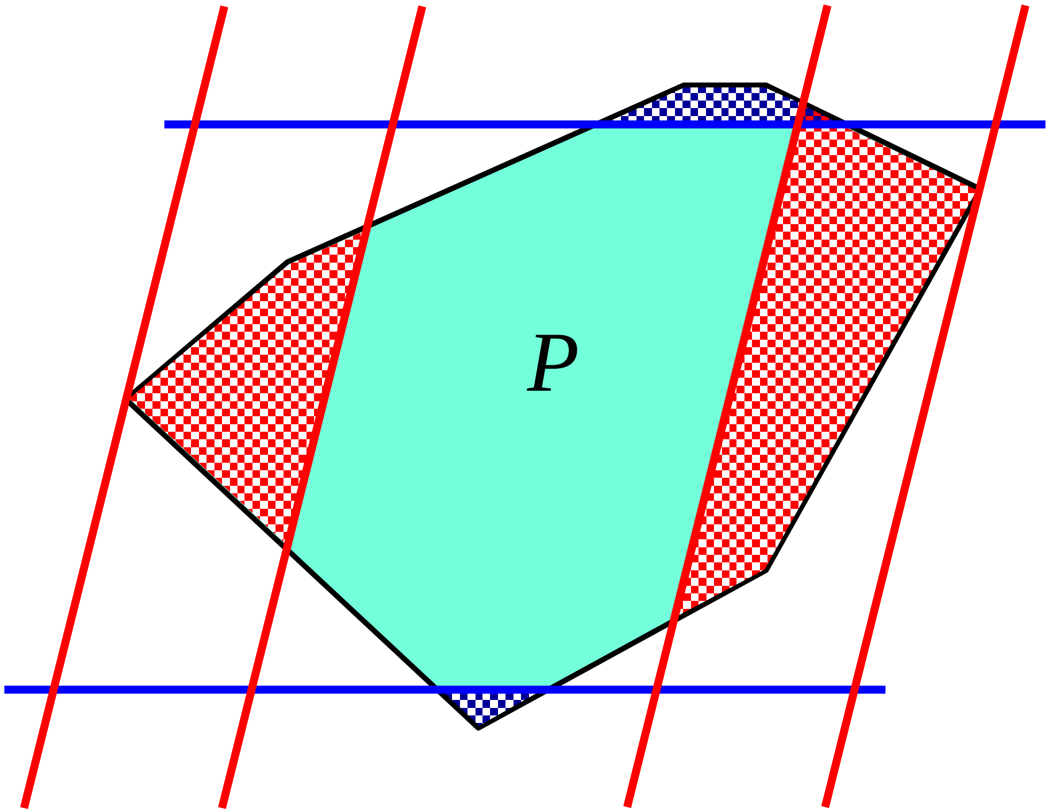


When an intersection occurs...

The process continues because it is still possible to separate

both  from  and  from 

Example

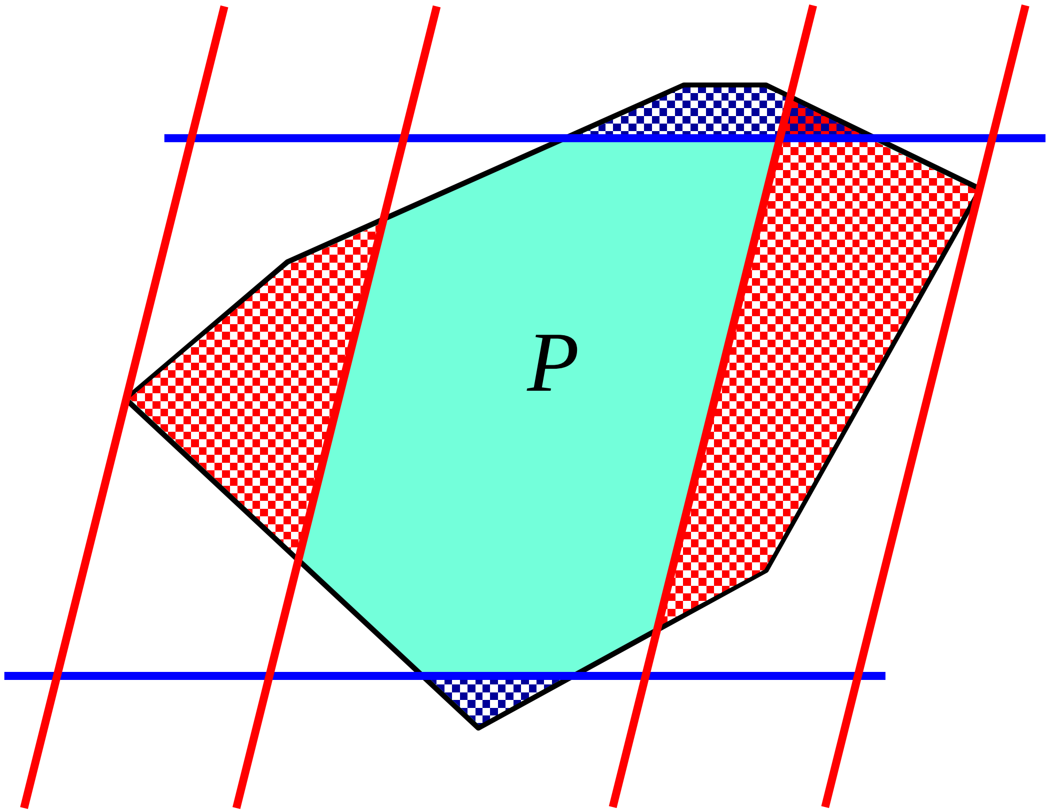


$$\dot{x} = f(x, y)$$

$$\dot{y} = g(x, y)$$

$$(x, y) \in P$$

Example

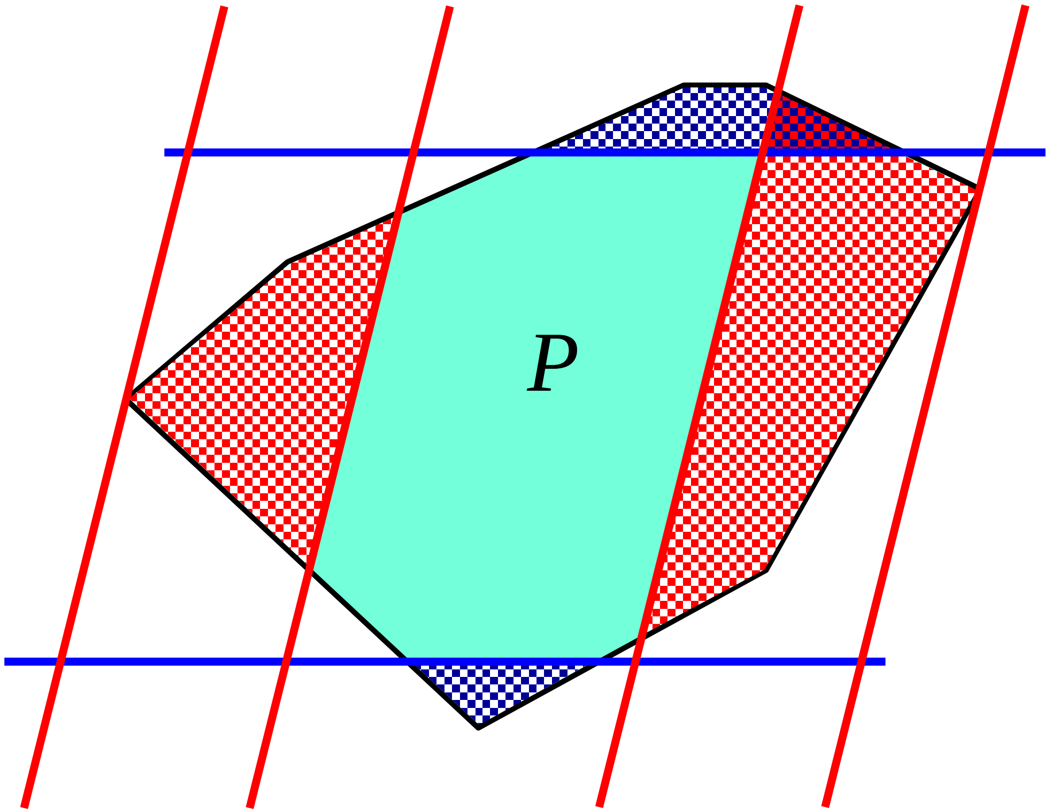


$$\dot{x} = f(x, y)$$

$$\dot{y} = g(x, y)$$

$$(x, y) \in P$$

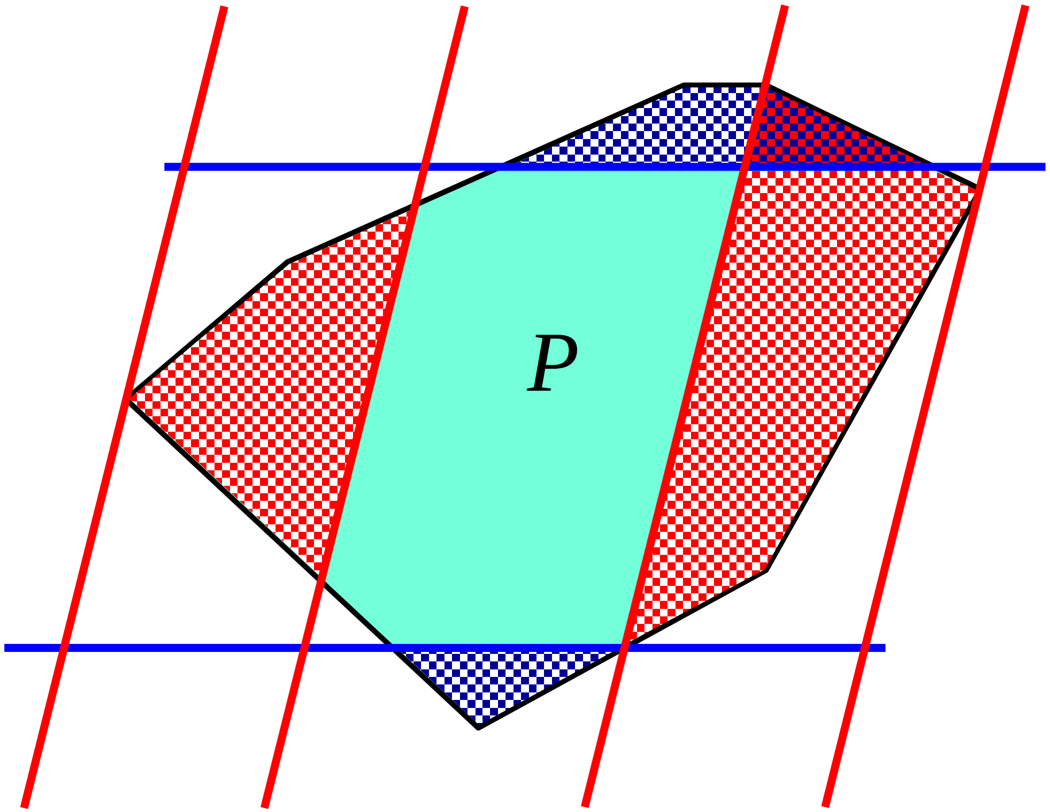
Example



$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$

$(x, y) \in P$

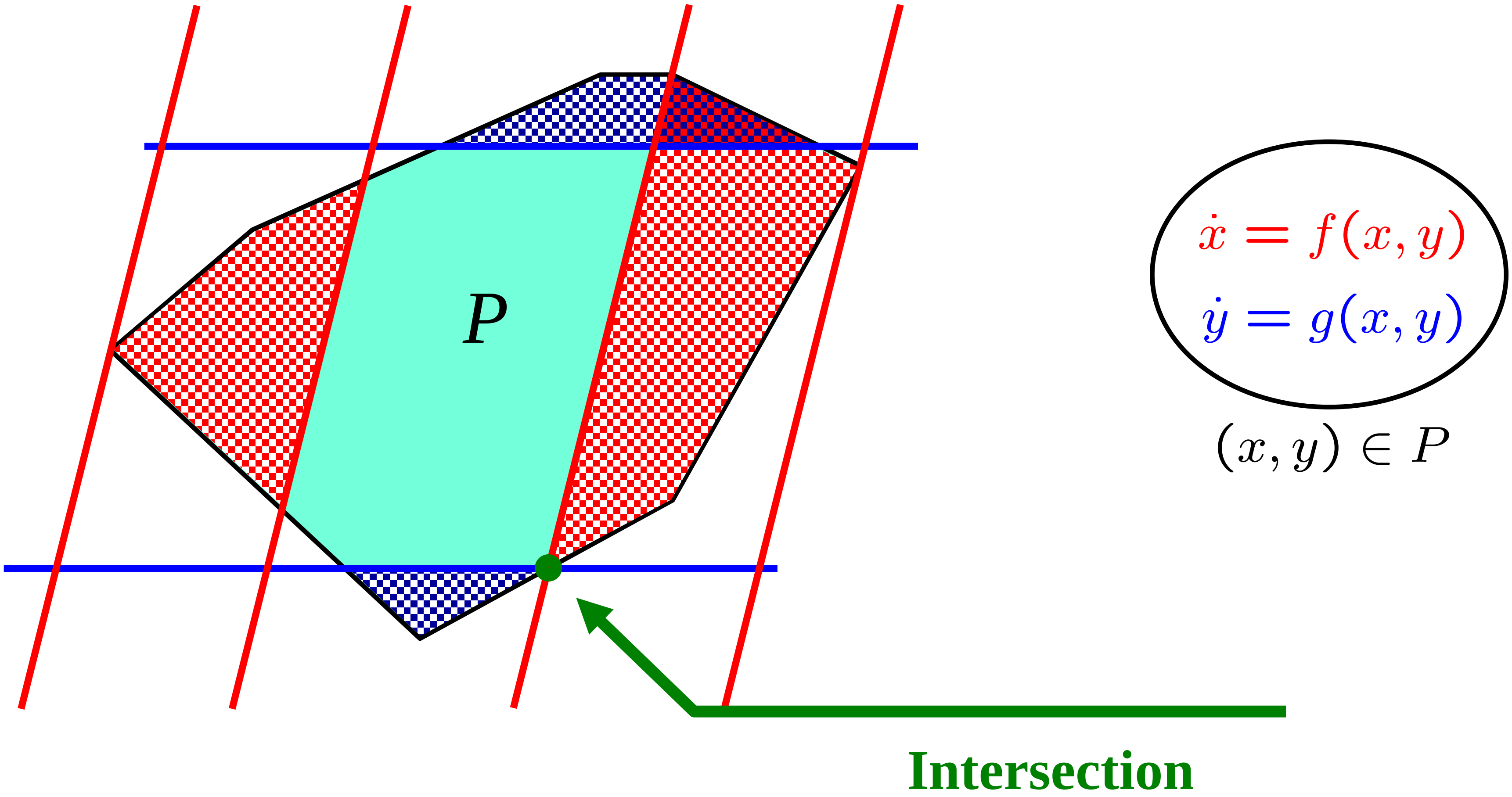
Example



$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$

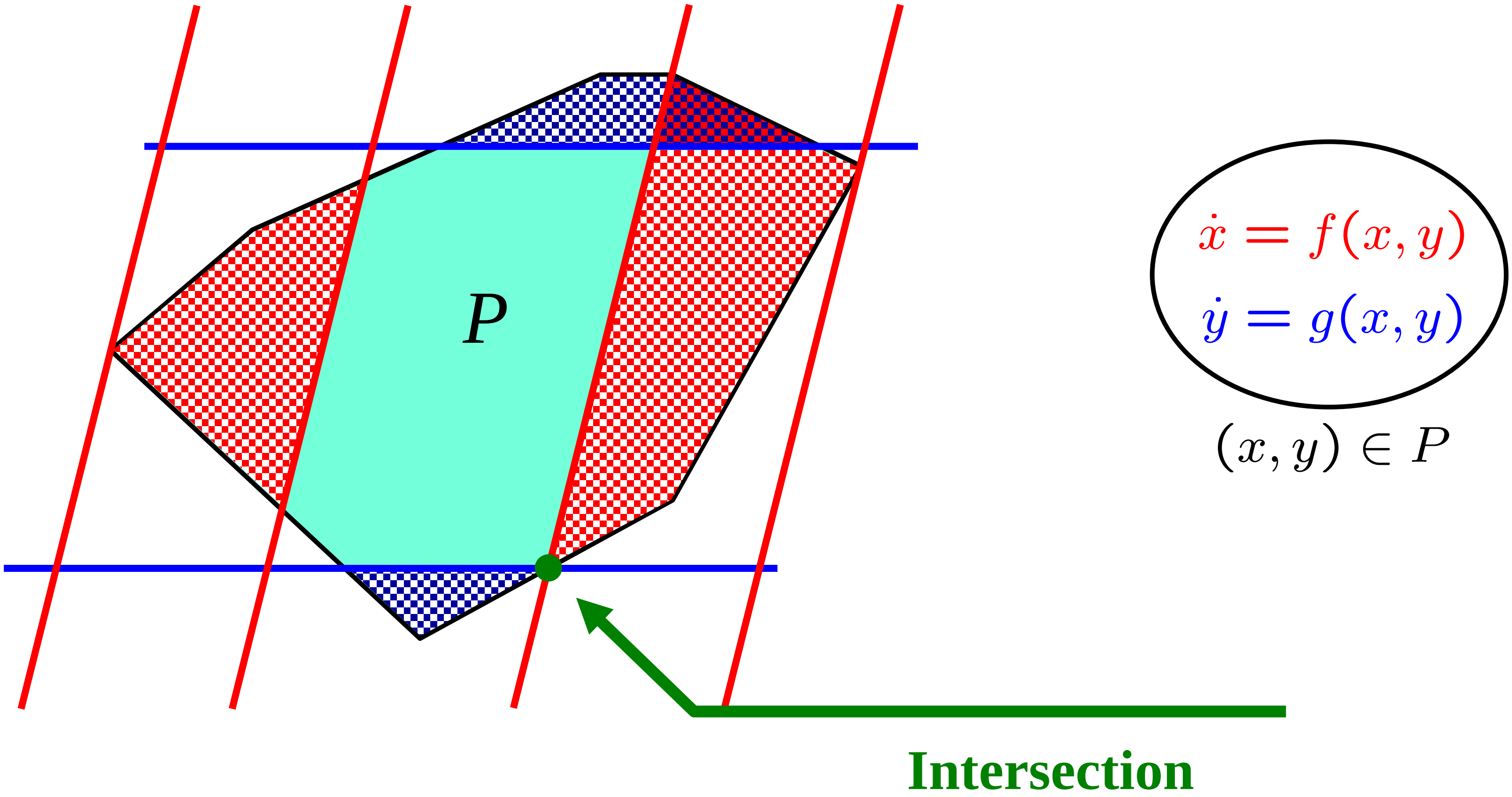
$$(x, y) \in P$$

Example



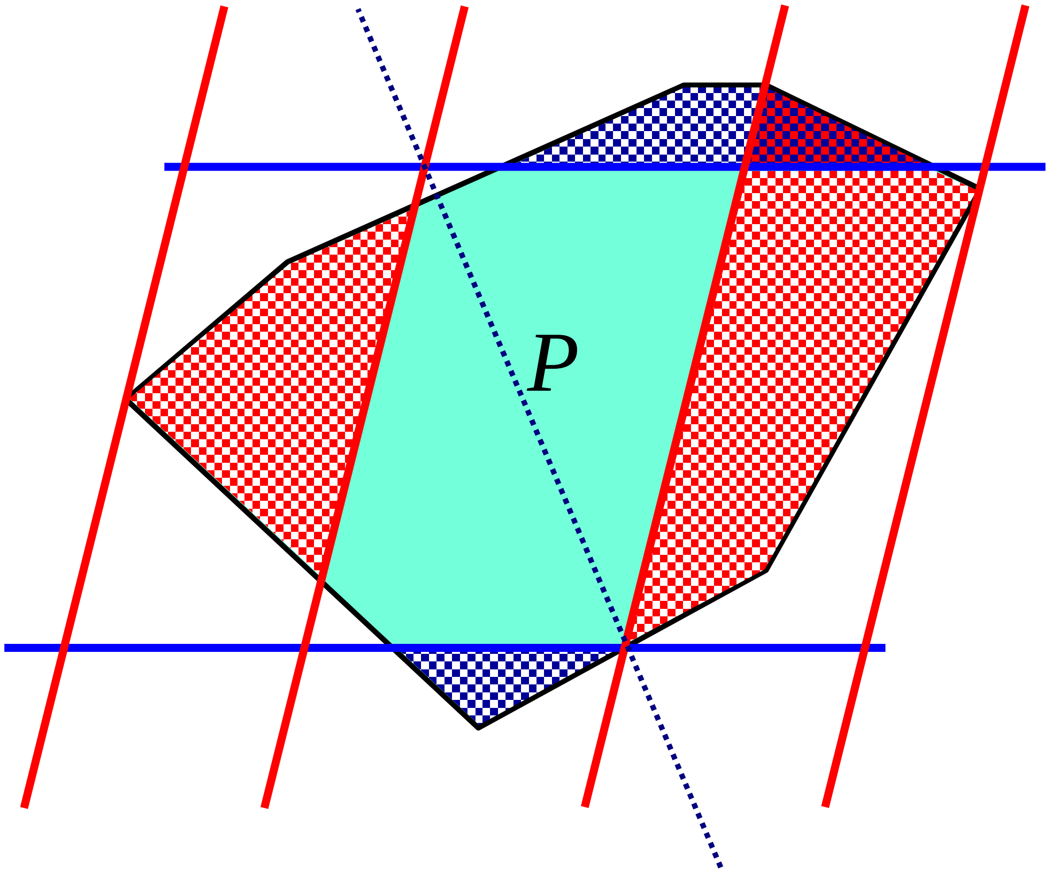
When a second intersection occurs...

Example



In this case, we have reached the "*limit of separability*"

Example

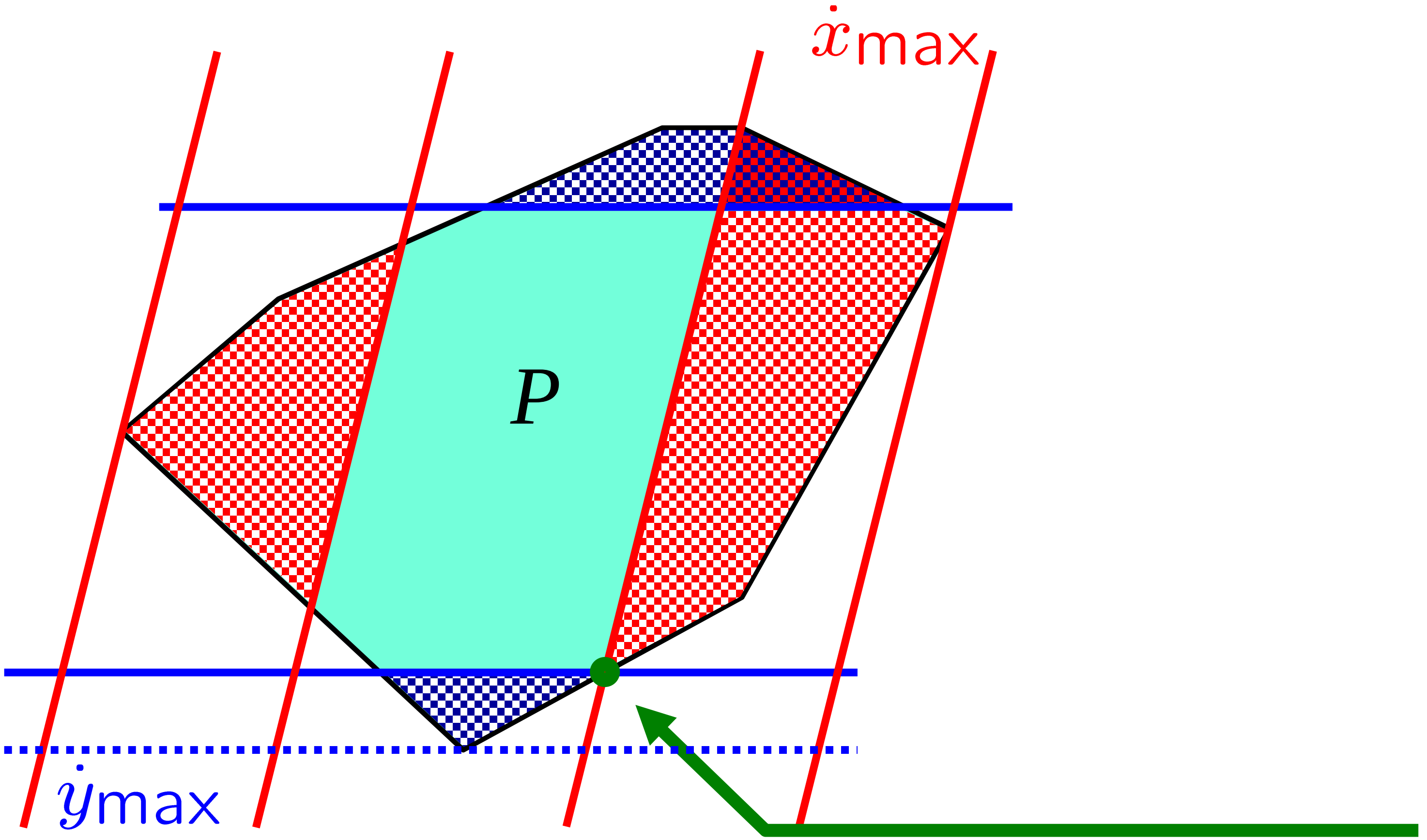


$$\dot{x} = f(x, y)$$
$$\dot{y} = g(x, y)$$

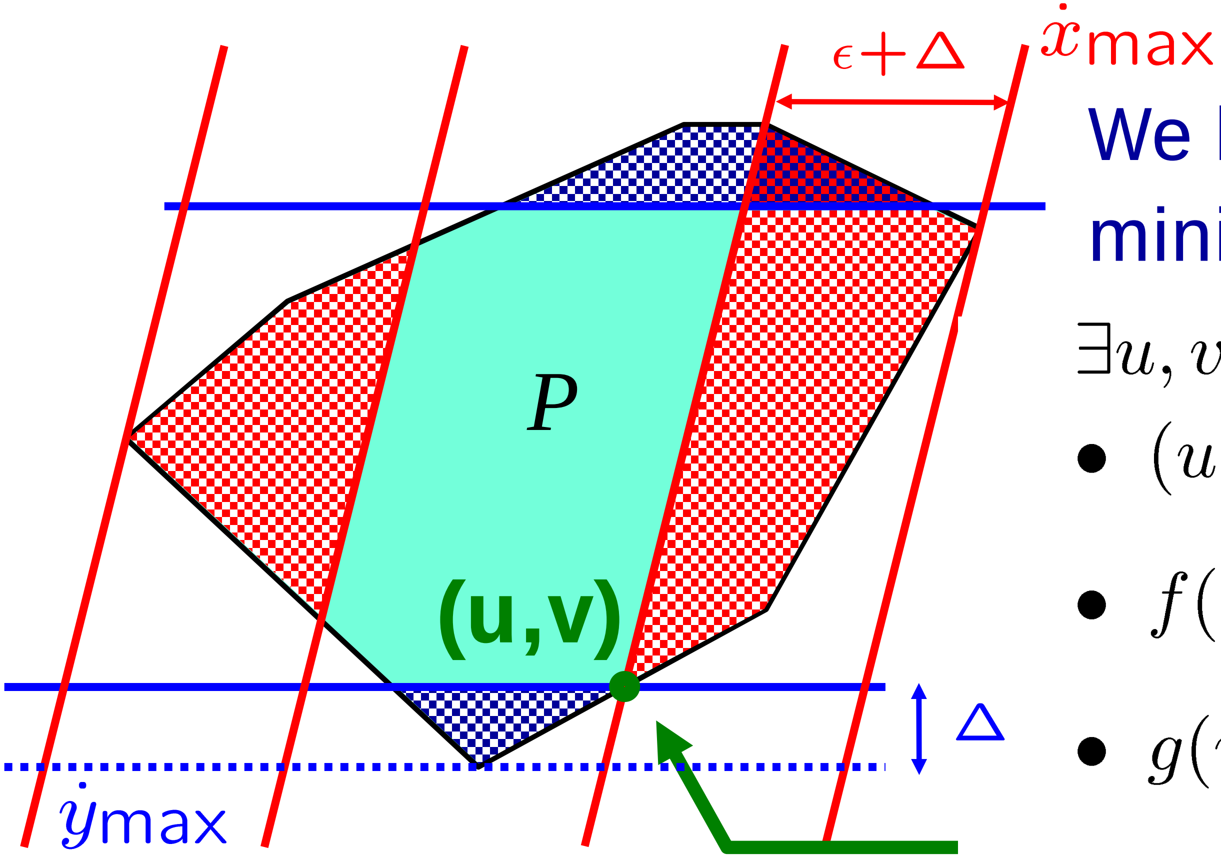
$$(x, y) \in P$$

An optimal cut

How to compute the intersection ?



How to compute the intersection ?



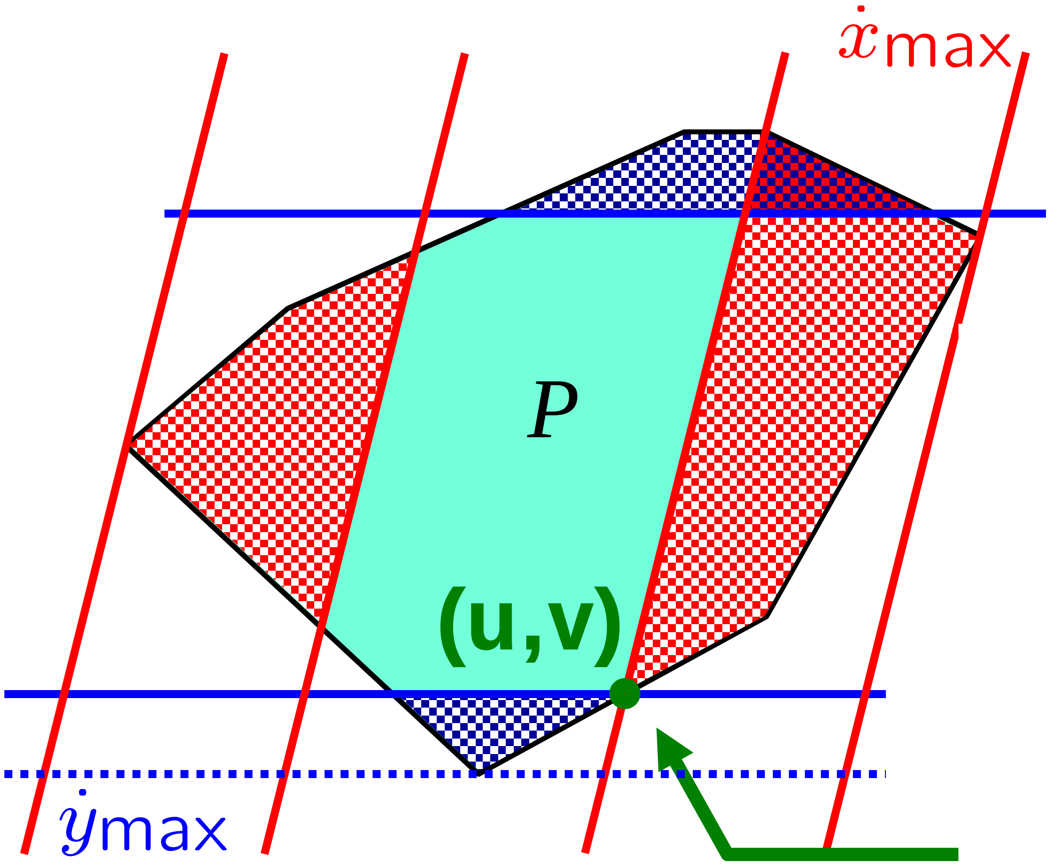
We have to find the minimal Δ such that:

$\exists u, v \in \mathbb{R}$:

- $(u, v) \in P$
- $f(u, v) = \dot{x}_{\max} - \epsilon - \Delta$
- $g(u, v) = \dot{y}_{\max} - \Delta$

where $\epsilon = r_x - r_y$

How to compute the intersection ?



We have to find the minimal Δ such that:

$\exists u, v \in \mathbb{R}$:

- $(u, v) \in P$
- $f(u, v) = \dot{x}_{\max} - \epsilon - \Delta$
- $g(u, v) = \dot{y}_{\max} - \Delta$

where $\epsilon = r_x - r_y$

➡ This is a linear program !

The algorithm

- Applies in the plane (2D)
  - Several particular cases

The algorithm

- Applies in the plane (2D)
  - Several particular cases
- What for higher dimension ?
  - An option: discretize the problem using a grid
  - Apply a (more) discrete algorithm
  - The exact solution can be arbitrarily closely approximated

The algorithm

- Applies in the plane (2D)
  - Several particular cases
- What for higher dimension ?
  - An option: discretize the problem using a grid
  - Apply a (more) discrete algorithm
  - The exact solution can be arbitrarily closely approximated

N.B.: it is possible to define a general algorithm in nD, but it requires to solve difficult geometrical problems (parametric convex hulls).

The algorithm

- Applies in the plane (2D)
  - Several particular cases
- What for higher dimension ?
  - An option: discretize the problem using a grid
  - Apply a (more) discrete algorithm
  - The exact solution can be arbitrarily closely approximated

N.B.: it is possible to define a general algorithm in nD, but it requires to solve difficult geometrical problems (parametric convex hulls).

Algorithm 1: Algorithm OptimalCut for computing the optimal cut in the 2D.

Data : An instance $S = \langle P, F \rangle$ of the optimal cut problem such that $P \subset \mathbb{R}^2$ and $F = \{f_1, f_2\}$.

Result : A line that solves the optimal cut problem for S.

begin

- $[\dot{x}_{\min}, \dot{x}_{\max}] \leftarrow f_1(P)$; $[\dot{y}_{\min}, \dot{y}_{\max}] \leftarrow f_2(P)$;
- $r_x \leftarrow \dot{x}_{\max} - \dot{x}_{\min}$; $r_y \leftarrow \dot{y}_{\max} - \dot{y}_{\min}$;
- Assume wlog. that $r_y \geq r_x$;
- **if** $r_y \geq 2\, r_x$ **then return** $\ell \equiv f_2(x,y) = \dot{y}_{\min} + \frac{r_y}{2}$;
- $\Delta_0 \leftarrow r_y - r_x$;
- Let Δ be a symbolic parameter ;
- $a_\Delta \leftarrow f_2^{-1}(\dot{y}_{\min} + \Delta_0 + \Delta) \cap f_1^{-1}(\dot{x}_{\min} + \Delta)$;
- $b_\Delta \leftarrow f_1^{-1}(\dot{x}_{\min} + \Delta) \cap f_2^{-1}(\dot{y}_{\max} - \Delta_0 - \Delta)$;
- $c_\Delta \leftarrow f_2^{-1}(\dot{y}_{\max} - \Delta_0 - \Delta) \cap f_1^{-1}(\dot{x}_{\max} - \Delta)$;
- $d_\Delta \leftarrow f_1^{-1}(\dot{x}_{\max} - \Delta) \cap f_2^{-1}(\dot{y}_{\min} + \Delta_0 + \Delta)$;
- **for** $z = a$ ***to*** d **do** $\Delta_z \leftarrow \min\{\Delta \mid z_\Delta \in P\}$;
- $\Delta_1 \leftarrow \min(\Delta_a, \Delta_c)$; $\Delta_2 \leftarrow \min(\Delta_b, \Delta_d)$;
- **if** $\Delta_1 \geq \frac{r_y}{2} - \Delta_0 \vee \Delta_2 \geq \frac{r_y}{2} - \Delta_0$ **then return** $\ell \equiv f_2(x,y) = \dot{y}_{\min} + \frac{r_y}{2}$;
- $Q_{\min} \leftarrow P \cap f_1^{-1}(\dot{x}_{\min})$; $Q_{\max} \leftarrow P \cap f_1^{-1}(\dot{x}_{\max})$;
- **if** $f_2(Q_{\min}) \cap [\dot{y}_{\min}, \dot{y}_{\min} + \Delta_0] \neq \varnothing \wedge f_2(Q_{\min}) \cap [\dot{y}_{\max} - \Delta_0, \dot{y}_{\max}] \neq \varnothing$ **then**
  - **return** $\ell \equiv f_2(x,y) = \dot{y}_{\min} + \frac{r_y}{2}$;
- **else if** $f_2(Q_{\min}) \cap [\dot{y}_{\min}, \dot{y}_{\min} + \Delta_0] \neq \varnothing \neq \varnothing$ **then**
  - **if** $f_2(Q_{\max}) \cap [\dot{y}_{\min}, \dot{y}_{\min} + \Delta_0] \neq \varnothing$ **then**
    - **return** $\ell \equiv f_2(x,y) = \dot{y}_{\min} + \frac{r_y}{2}$;
  - **else**
    - **return** $\ell ine(b_{\Delta_2}, d_{\Delta_2})$;
- **else if** $f_2(Q_{\min}) \cap [\dot{y}_{\max} - \Delta_0, \dot{y}_{\max}] \neq \varnothing$ **then**
  - **if** $f_2(Q_{\max}) \cap [\dot{y}_{\max} - \Delta_0, \dot{y}_{\max}] \neq \varnothing$ **then**
    - **return** $\ell \equiv f_2(x,y) = \dot{y}_{\min} + \frac{r_y}{2}$;
  - **else**
    - **return** $\ell ine(a_{\Delta_1}, c_{\Delta_1})$;
- **else if** $f_2(Q_{\max}) \cap [\dot{y}_{\max} - \Delta_0, \dot{y}_{\max}] \neq \varnothing \wedge f_2(Q_{\max}) \cap [\dot{y}_{\min}, \dot{y}_{\min} + \Delta_0] \neq \varnothing$ **then**
  - **return** $\ell \equiv f_2(x,y) = \dot{y}_{\min} + \frac{r_y}{2}$;
- **else if** $f_2(Q_{\max}) \cap [\dot{y}_{\max} - \Delta_0, \dot{y}_{\max}] \neq \varnothing$ **then**
  - **return** $\ell ine(b_{\Delta_2}, d_{\Delta_2})$
- **else if** $f_2(Q_{\max}) \cap [\dot{y}_{\min}, \dot{y}_{\min} + \Delta_0] \neq \varnothing$ **then**
  - **return** $\ell ine(a_{\Delta_1}, c_{\Delta_1})$;
- **else if** $\Delta_1 > \Delta_2$ **then**
  - **return** $\ell ine(a_{\Delta_1}, c_{\Delta_1})$;
- **else**
  - **return** $\ell inc(b_{\Delta_2}, d_{\Delta_2})$;

end

Navigation benchmark

In each location, the dynamics has the form:

$$\begin{cases} \dot{x} = v \\ \dot{v} = \underbrace{A\,(v - v_d)}_{x_1 \text{ and } x_2 \text{ do not appear...}} \end{cases}$$

→ We cut in the plane v_1-v_2

Navigation benchmark

In each location, the dynamics has the form:

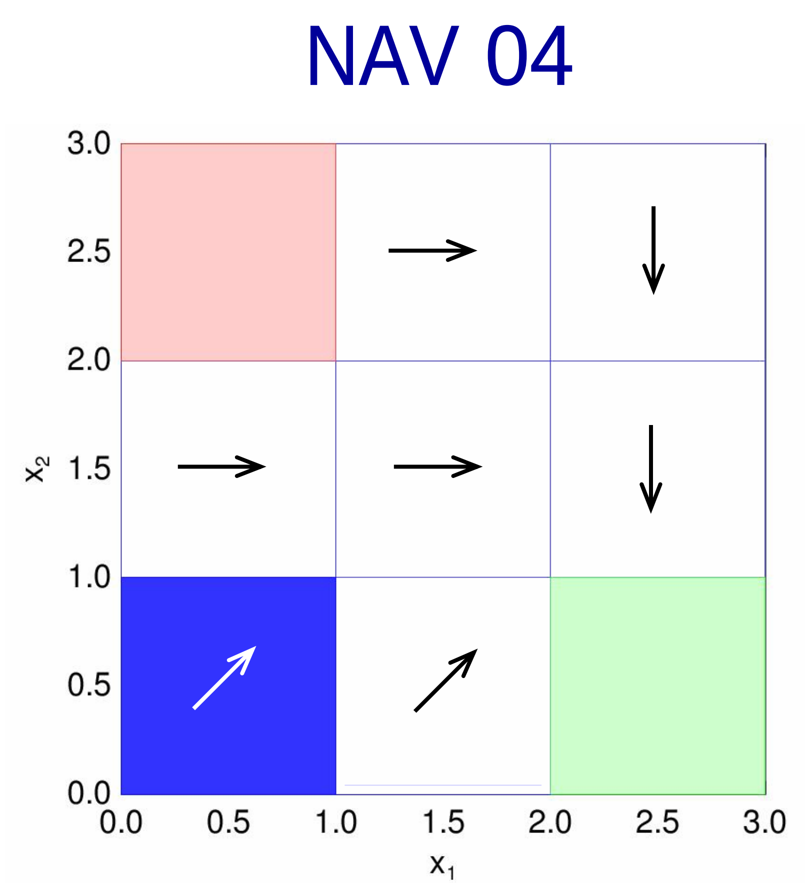
$$\begin{cases} \dot{x} = v \\ \dot{v} = \underbrace{A\,(v - v_d)}_{x_1 \text{ and } x_2 \text{ do not appear...}} \end{cases}$$

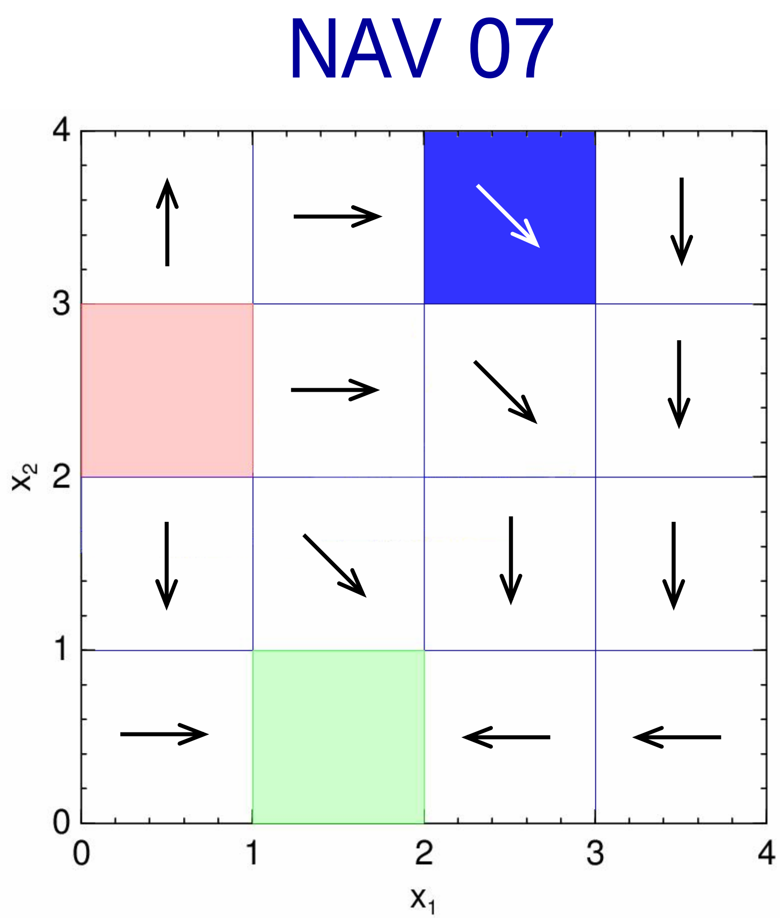
→ We cut in the plane v_1-v_2

Instance	Grid	Time	(PT)
NAV01	3×3	5s	(35s)
NAV02	3×3	10s	(62s)
NAV03	3×3	10s	(62s)
NAV04	3×3	75s	(225s[i])
NAV07	4×4	11mn	

[i] obtained with a heuristic.

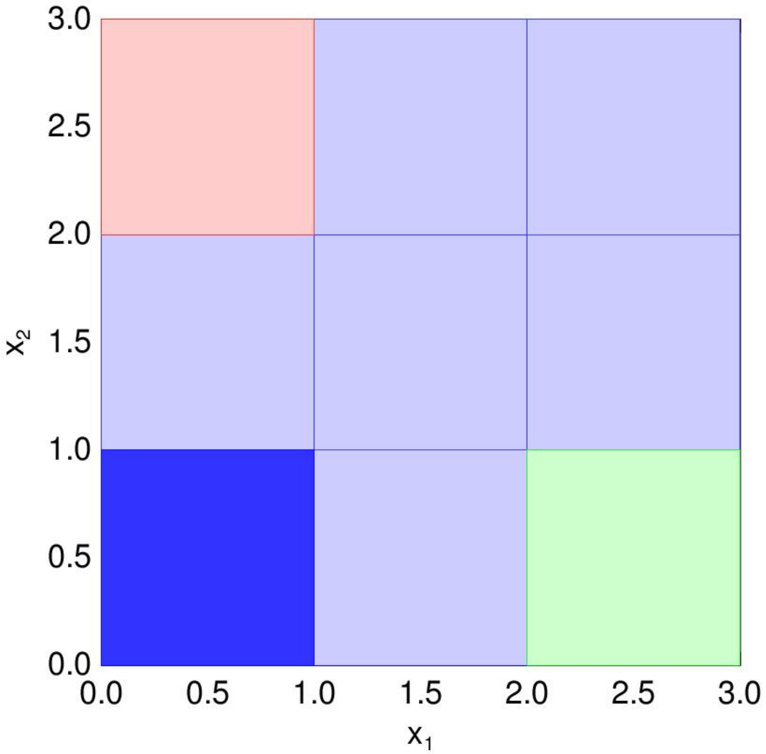
Results



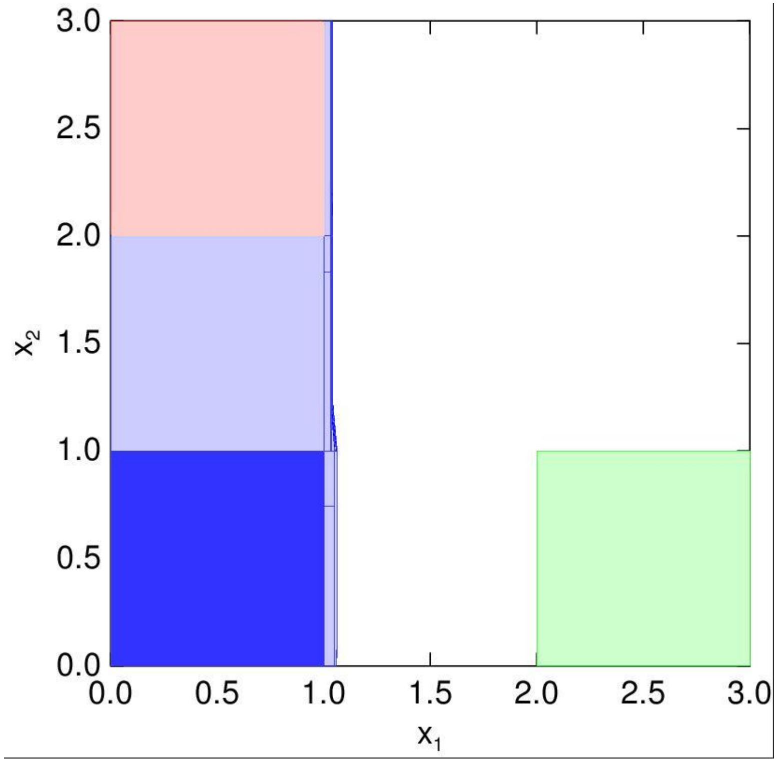


Initial states   Bad states   Good states

Results: NAV 04

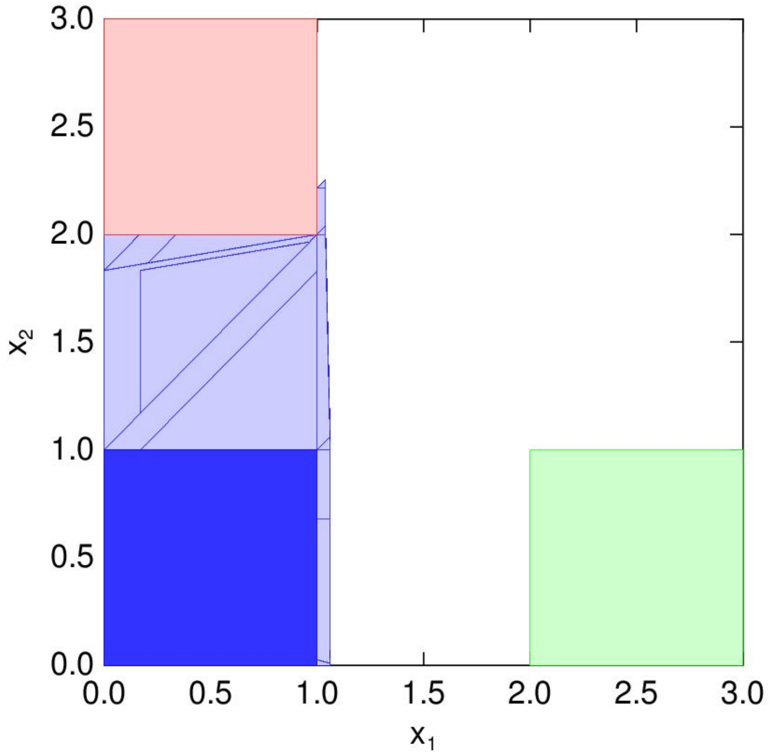


Forward

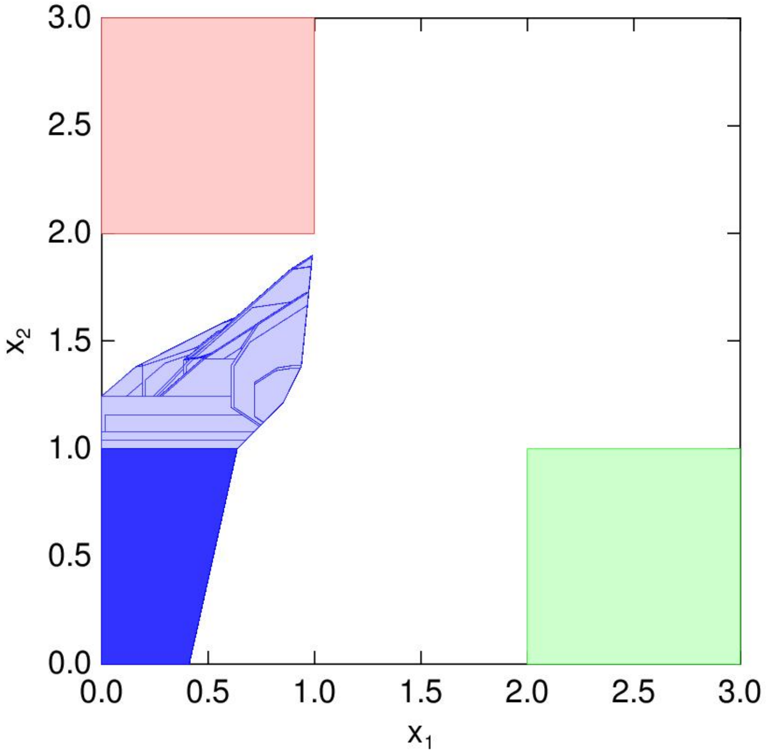


Backward

Results: NAV 04

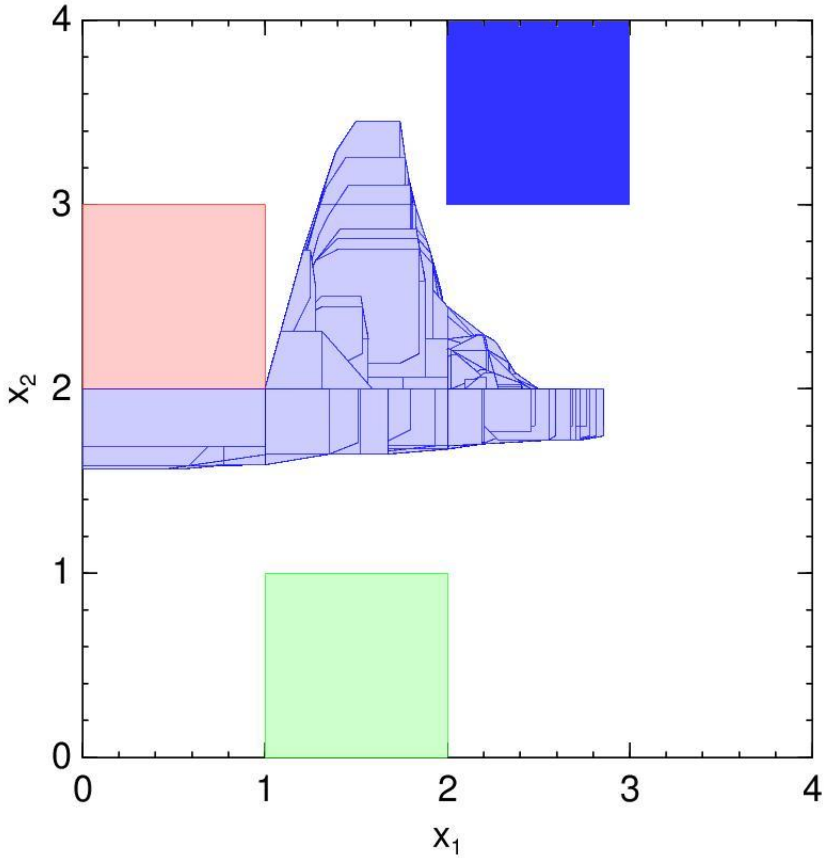


Forward



Forward

Results: NAV 07



Backward

Conclusion

- Approximations
  - Rectangular
  - Over-approximations

Conclusion

- Approximations
  - Rectangular
  - <u>Over</u>-approximations
- Refinements
  - Automatic
  - Optimal split for some criterion (at least in 2D)

Conclusion

- Approximations
  - Rectangular
  - Over-approximations
- Refinements
  - Automatic
  - Optimal split for some criterion (at least in 2D)
- Possible future work
  - Under-approximations
  - Optimal split for some other criterion
  - Combine with other approaches (barrier certificates, ellipsoïds, ...)

References

- [FI04] A. Fehnker and F. Ivancic. *Benchmarks for hybrid systems verification*. In HSCC 2004, LNCS 2993, pp 326-341.
- [Fre05] G. Frehse. *Phaver: Algorithmic verification of hybrid systems past hytech*. In HSCC 2005, LNCS 3414, pp 258-273.