New Widening Operators for Convex Polyhedra

Roberto BAGNARA, Patricia M. HILL,
Elisa RICCI, Enea ZAFFANELLA

`http://www.cs.unipr.it/ppl/`

MOTIVATIONS

- **➜** Linear Relation Analysis is a key component of many static analysis and (semi-) automatic verification tools.
- **➜** Since it has infinite chains, the domain of convex polyhedra has to be provided with widening operators.
- **➜** The standard widening (Cousot and Halbwachs, POPL'78) is the one and only champion: since then, no challanger has been proposed.
- **➜** But some applications need more precision. Solutions include:
  1. the widening delay technique (Cousot, '81);
  2. the widening 'up to' technique (Halbwachs, CAV'93);
  3. various extrapolation operators (no convergence guarantee).
- **➜** Our goal: *provide a framework for the definition of new widening operators on convex polyhedra improving upon the precision of the standard widening.*

Plan of the Talk

COMPUTING THE CONCRETE SEMANTICS

```
x := 0; b := true;

while (b) do

  x := x+2;

  read(b);

endwhile
```

Computing the Concrete Semantics

```
x := 0; b := true;

while (b) do
```

$x \in S \in \wp(\mathbb{R})$

```
  x := x+2;

  read(b);

endwhile
```

Computing the Concrete Semantics

```
x := 0; b := true;

while (b) do
```
$x \in S \in \wp(\mathbb{R})$
```
  x := x+2;

  read(b);

endwhile
```

Let $\mathcal{F} \colon \wp(\mathbb{R}) \to \wp(\mathbb{R})$ be such that

$$\mathcal{F}(X) \stackrel{\text{def}}{=} \{0\} \cup \{\, n+2 \mid n \in X \,\}$$

The concrete semantics S is computed as the least fixpoint of $\mathcal{F}$ on the complete lattice $\langle \wp(\mathbb{R}), \subseteq, \varnothing, \mathbb{R}, \cup, \cap \rangle$.

Computing the Concrete Semantics

```
x := 0; b := true;

while (b) do
```

$x \in S = 2\mathbb{N}$

```
  x := x+2;

  read(b);

endwhile
```

$$\mathcal{F}(X) \stackrel{\text{def}}{=} \{0\} \cup \{\, n+2 \mid n \in X \,\}$$

$$X_0 = \varnothing;$$

$$X_1 = \mathcal{F}(\varnothing) = \{0\};$$

$$X_2 = \mathcal{F}(\mathcal{F}(\varnothing)) = \{0, 2\};$$

$$\ldots$$

$$S = X_\omega = \mathrm{lfp}(\mathcal{F}) = 2\mathbb{N}.$$

THE DOMAIN $\mathbb{CP}_n$ OF CLOSED CONVEX POLYHEDRA

A lattice $\langle \mathbb{CP}_n, \subseteq, \varnothing, \mathbb{R}^n, \uplus, \cap \rangle$, with infinite chains.

Constraint Representation: $\mathcal{P} = \mathrm{con}(\mathcal{C})$

- ➜ $\mathcal{C}$ is a finite set of linear non-strict inequality (resp., equality) constraints.
- ➜ No redundant constraint $+$ max number of equalities $\implies$ minimal form.
- ➜ Inequalities orthogonal wrt equalities $\implies$ orthogonal form.

Generator Representation: $\mathcal{P} = \mathrm{gen}(\mathcal{G})$

- ➜ $\mathcal{G} = (L, R, P)$, where
  - ➜ P is a finite set of points of $\mathcal{P}$;
  - ➜ R is a finite set of rays (directions of infinity) of $\mathcal{P}$;
  - ➜ L is a finite set of lines (bidirectional rays) of $\mathcal{P}$.
- ➜ No redundant generator $+$ max number of lines $\implies$ minimal form.
- ➜ Points and rays orthogonal wrt lines $\implies$ orthogonal form.

APPROXIMATING THE SEMANTICS ON $\mathbb{CP}_1$

```
x := 0; b := true;

while (b) do
```

$x \in \mathcal{Q} \in \mathbb{CP}_1$

```
  x := x+2;

  read(b);

endwhile
```

Approximating the Semantics on $\mathbb{CP}_1$

Let $\mathcal{F}^\sharp \colon \mathbb{CP}_1 \to \mathbb{CP}_1$ be such that

$$\mathcal{F}^\sharp(\mathcal{P}) \stackrel{\text{def}}{=} \{0\} \uplus \{\, n+2 \mid n \in \mathcal{P} \,\}$$

```
x := 0; b := true;

while (b) do
```
$x \in \mathcal{Q} \in \mathbb{CP}_1$
```
  x := x+2;

  read(b);

endwhile
```

APPROXIMATING THE SEMANTICS ON $\mathbb{CP}_1$

```
x := 0; b := true;

while (b) do
```
$x \in \mathcal{Q} \in \mathbb{CP}_1$
```
  x := x+2;

  read(b);

endwhile
```

Let $\mathcal{F}^\sharp \colon \mathbb{CP}_1 \to \mathbb{CP}_1$ be such that

$$\mathcal{F}^\sharp(\mathcal{P}) \stackrel{\text{def}}{=} \{0\} \uplus \{\, n+2 \mid n \in \mathcal{P} \,\}$$

Correctness of $\mathcal{F}^\sharp$ wrt $\mathcal{F}$:

$$X \subseteq \mathcal{P} \implies \mathcal{F}(X) \subseteq \mathcal{F}^\sharp(\mathcal{P}).$$

APPROXIMATING THE SEMANTICS ON $\mathbb{CP}_1$

```
x := 0; b := true;

while (b) do
```
$x \in \mathcal{Q} \in \mathbb{CP}_1$
```
  x := x+2;

  read(b);

endwhile
```

Let $\mathcal{F}^\sharp \colon \mathbb{CP}_1 \to \mathbb{CP}_1$ be such that

$$\mathcal{F}^\sharp(\mathcal{P}) \stackrel{\text{def}}{=} \{0\} \uplus \{\, n+2 \mid n \in \mathcal{P} \,\}$$

Correctness of $\mathcal{F}^\sharp$ wrt $\mathcal{F}$:

$$X \subseteq \mathcal{P} \implies \mathcal{F}(X) \subseteq \mathcal{F}^\sharp(\mathcal{P}).$$

The concrete semantics $S \in \mathbb{R}$ is approximated by computing a post-fixpoint $\mathcal{Q} \in \mathbb{CP}_1$ of the abstract semantic function $\mathcal{F}^\sharp$.

Approximating the Semantics on $\mathbb{CP}_1$

$$\mathcal{F}(X) \stackrel{\text{def}}{=} \{0\} \cup \{\, n+2 \mid n \in X \,\}$$

$$\mathcal{F}^\sharp(\mathcal{P}) \stackrel{\text{def}}{=} \{0\} \uplus \{\, n+2 \mid n \in \mathcal{P} \,\}$$

$$\begin{aligned} X_0 &= \varnothing; \\ X_1 &= \mathcal{F}(\varnothing) = \{0\}; \\ X_2 &= \mathcal{F}(\mathcal{F}(\varnothing)) = \{0, 2\}; \\ &\dots \\ S &= 2\mathbb{N}. \end{aligned}$$

$$\begin{aligned} \mathcal{P}_0 &= \varnothing; \\ \mathcal{P}_1 &= \mathcal{F}^\sharp(\varnothing) = \{0\}; \\ \mathcal{P}_2 &= \mathcal{F}^\sharp(\mathcal{F}^\sharp(\varnothing)) = [0, 2]; \\ &\dots \\ \mathcal{Q} &= [0, +\infty). \end{aligned}$$

Problems in the Approximated Computation

① The "limit" of the approximated computation may not be representable in the abstract domain (e.g., a circle is not a polyhedron);

Problems in the Approximated Computation

① The “limit” of the approximated computation may not be representable in the abstract domain (e.g., a circle is not a polyhedron);

② Reaching a post-fixpoint may still require an infinite number of computation steps, as was the case in the example we have seen;

Problems in the Approximated Computation

① The "limit" of the approximated computation may not be representable in the abstract domain (e.g., a circle is not a polyhedron);

② Reaching a post-fixpoint may still require an infinite number of computation steps, as was the case in the example we have seen;

③ Even when the computation is intrinsically finite, it may be practically unfeasible if it requires too many approximated iterations; for instance,

```
x := 0;
while (x < 1000000) do
  x := x+1; y := f(x);
endwhile
```

PROBLEMS IN THE APPROXIMATED COMPUTATION

① The “limit” of the approximated computation may not be representable in the abstract domain (e.g., a circle is not a polyhedron);

② Reaching a post-fixpoint may still require an infinite number of computation steps, as was the case in the example we have seen;

③ Even when the computation is intrinsically finite, it may be practically unfeasible if it requires too many approximated iterations; for instance,

```
x := 0;
while (x < 1000000) do
  x := x+1; y := f(x);
endwhile
```

Widening operators try to solve all of these problems at once.

Definition of Widening Operator

A variant of the classical one (see Cousot and Cousot, PLILP'92):

➜ Let $\langle L, \sqsubseteq, \perp, \sqcup \rangle$ be a join-semi-lattice. Then, the operator $\nabla \colon L \times L \rightarrowtail L$ is a widening on L if

① $\forall x, y \in L : x \sqsubseteq y \implies y \sqsubseteq x \nabla y$;

② for all increasing chains $y_0 \sqsubseteq y_1 \sqsubseteq \cdots$, the chain defined by $x_0 \stackrel{\text{def}}{=} y_0, \ldots, x_{i+1} \stackrel{\text{def}}{=} x_i \nabla (x_i \sqcup y_{i+1}), \ldots$ is not strictly increasing.

Definition of Widening Operator

A variant of the classical one (see Cousot and Cousot, PLILP'92):

- ➜ Let $\langle L, \sqsubseteq, \bot, \sqcup \rangle$ be a join-semi-lattice. Then, the operator $\nabla \colon L \times L \rightarrowtail L$ is a widening on L if
  - ① $\forall x, y \in L : x \sqsubseteq y \implies y \sqsubseteq x \nabla y$;
  - ② for all increasing chains $y_0 \sqsubseteq y_1 \sqsubseteq \cdots$, the chain defined by $x_0 \stackrel{\text{def}}{=} y_0, \ldots, x_{i+1} \stackrel{\text{def}}{=} x_i \nabla (x_i \sqcup y_{i+1}), \ldots$ is not strictly increasing.
- ➜ The upward iteration sequence with widenings (starting from $x_0 = \bot$)

$$x_{i+1} = \begin{cases} x_i, & \text{if } \mathcal{F}^\sharp(x_i) \sqsubseteq x_i; \\ x_i \nabla \bigl(x_i \sqcup \mathcal{F}^\sharp(x_i)\bigr), & \text{otherwise;} \end{cases}$$

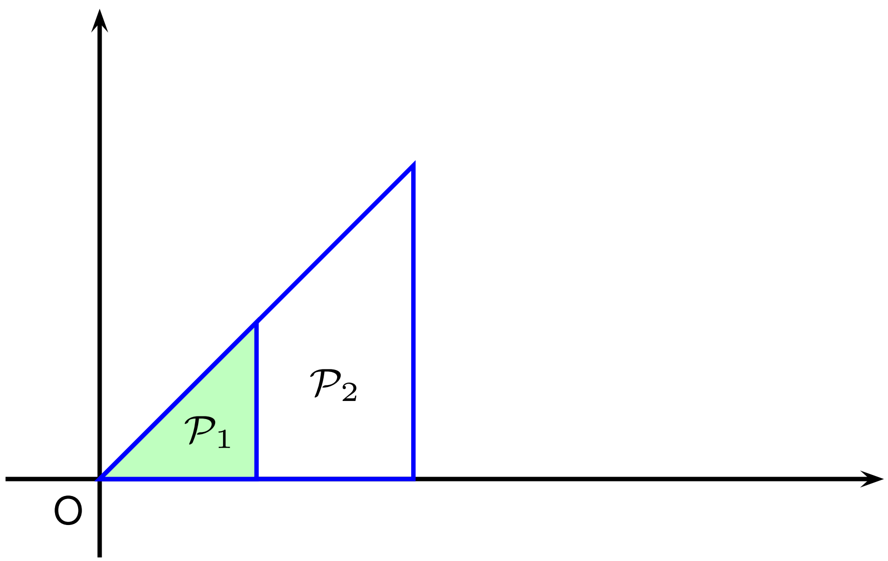
converges (to a post-fixpoint of $\mathcal{F}^\sharp$) after a finite number of iterations.

THE STANDARD WIDENING ∇_s ON $\mathbb{CP}_n$

- Initially proposed in Cousot and Halbwachs, POPL'78.
- Intuitively, $\mathcal{P}_1 \nabla_s \mathcal{P}_2$ *is defined by all the non-redundant constraints of* $\mathcal{P}_1$ *that are also satisfied by* $\mathcal{P}_2$*.*

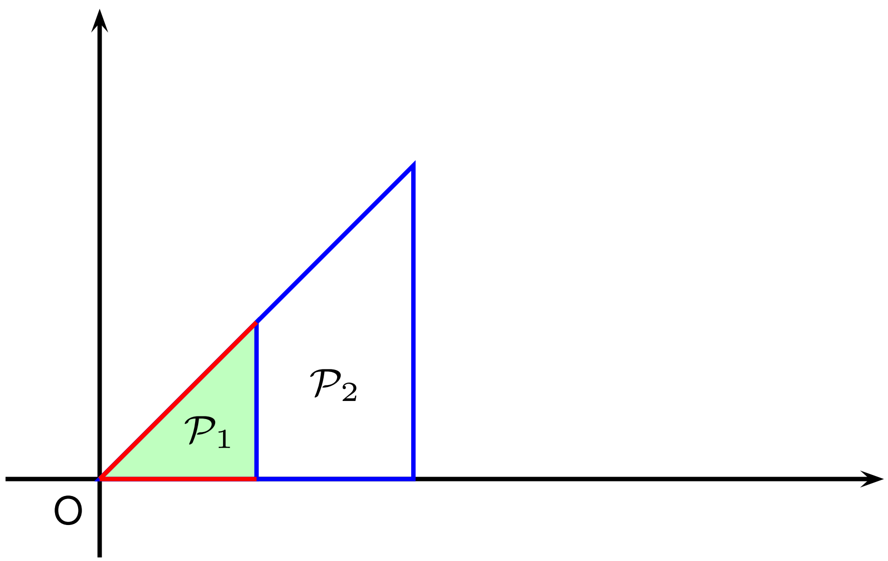
THE STANDARD WIDENING ∇_s ON $\mathbb{CP}_n$

- ➜ Initially proposed in Cousot and Halbwachs, POPL'78.
- ➜ Intuitively, *$\mathcal{P}_1 \nabla_s \mathcal{P}_2$ is defined by all the non-redundant constraints of $\mathcal{P}_1$ that are also satisfied by $\mathcal{P}_2$.*



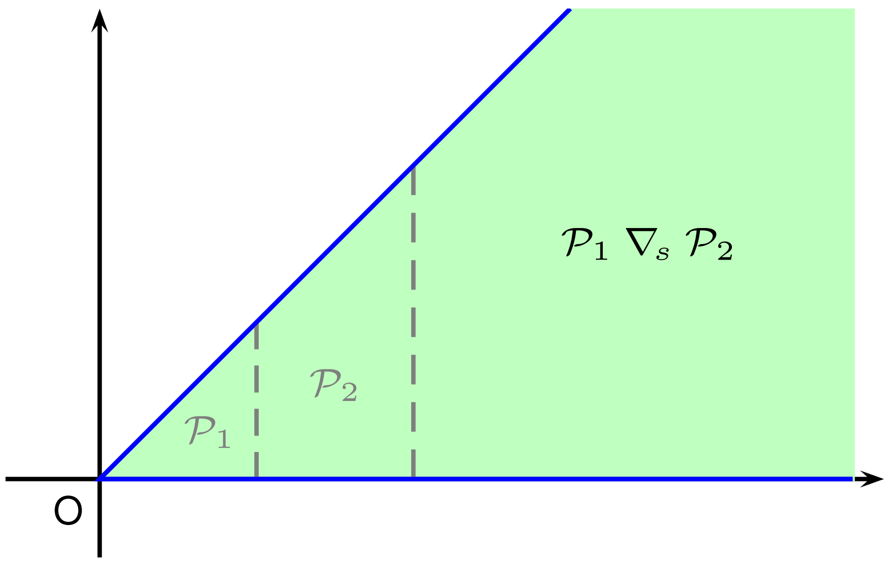
THE STANDARD WIDENING ∇_s ON $\mathbb{CP}_n$

- ➜ Initially proposed in Cousot and Halbwachs, POPL'78.
- ➜ Intuitively, *$\mathcal{P}_1 \nabla_s \mathcal{P}_2$ is defined by all the non-redundant constraints of $\mathcal{P}_1$ that are also satisfied by $\mathcal{P}_2$.*



The Standard Widening ∇_s on $\mathbb{CP}_n$

- ➜ Initially proposed in Cousot and Halbwachs, POPL'78.
- ➜ Intuitively, *$\mathcal{P}_1 \nabla_s \mathcal{P}_2$ is defined by all the non-redundant constraints of $\mathcal{P}_1$ that are also satisfied by $\mathcal{P}_2$.*

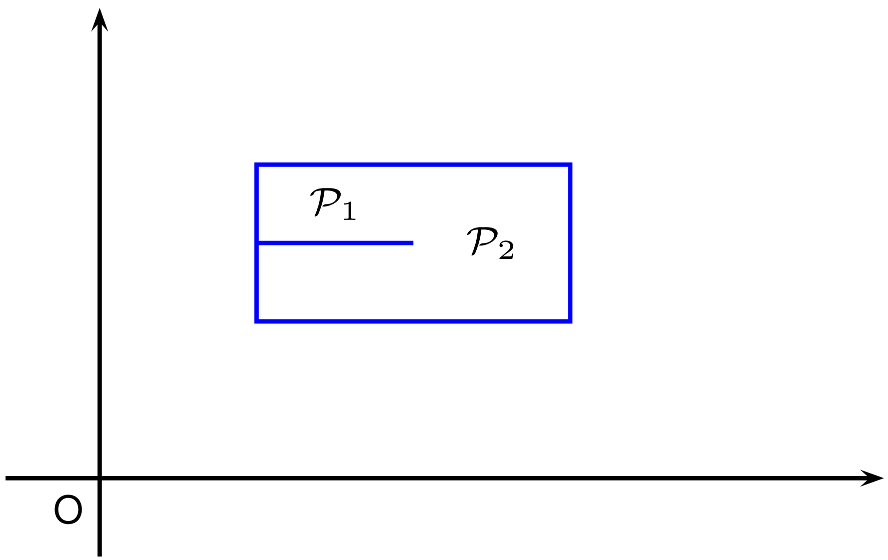


THE STANDARD WIDENING ∇_s ON $\mathbb{CP}_n$ (II)

➜ Improved in Halbwachs'79 (the PhD thesis), so that it does not depend on the chosen constraint representations.

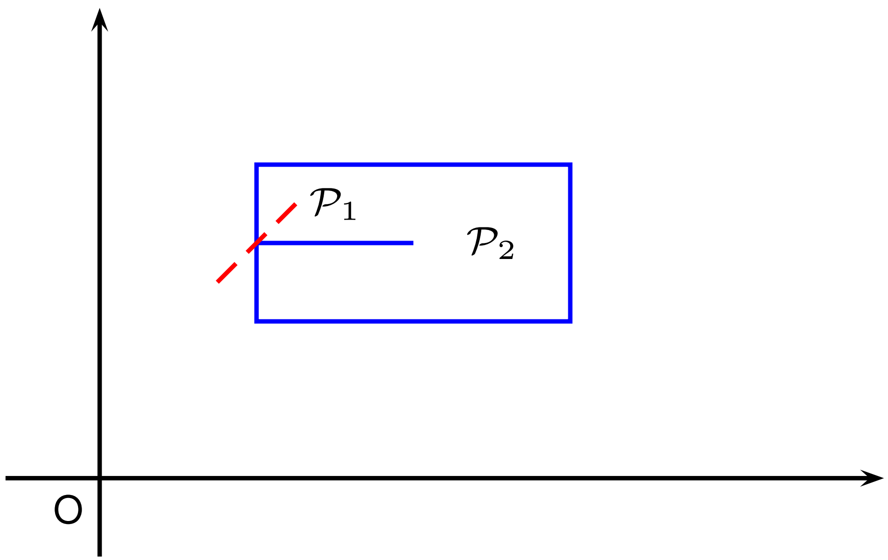
The Standard Widening ∇_s on $\mathbb{CP}_n$ (II)

➜ Improved in Halbwachs'79 (the PhD thesis), so that it does not depend on the chosen constraint representations.



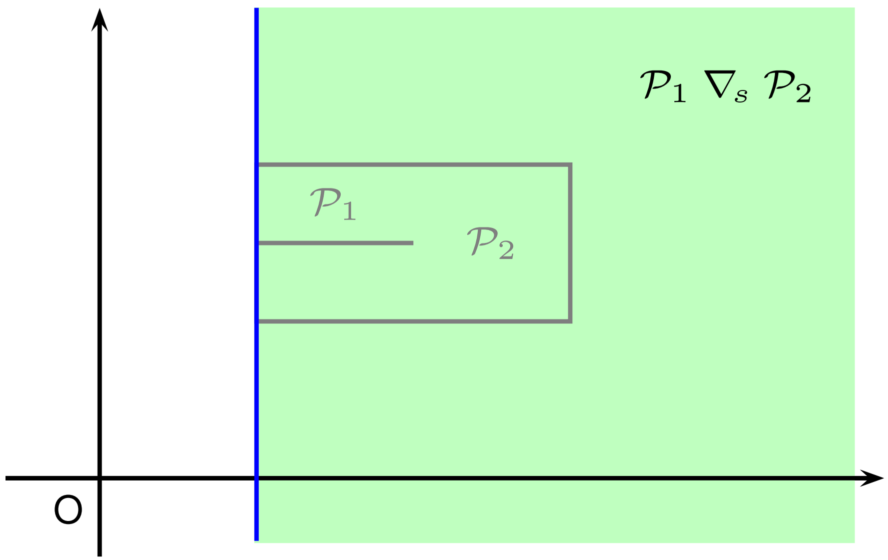
THE STANDARD WIDENING ∇_s ON $\mathbb{CP}_n$ (II)

➜ Improved in Halbwachs'79 (the PhD thesis), so that it does not depend on the chosen constraint representations.



THE STANDARD WIDENING ∇_s ON $\mathbb{CP}_n$ (II)

➜ Improved in Halbwachs'79 (the PhD thesis), so that it does not depend on the chosen constraint representations.



The Standard Widening ∇_s on $\mathbb{CP}_n$ (III)

➜ The resulting operator is both precise and efficient: this "*tentative*" definition has been the one and only available approach for 25 years.

THE STANDARD WIDENING ∇_s ON $\mathbb{CP}_n$ (III)

➜ The resulting operator is both precise and efficient: this "*tentative*" definition has been the one and only available approach for 25 years.

➜ Its precision can be improved (while keeping the convergence guarantee) by applying

① the widening delay technique: delay the application of the widening for a fixed number of iteration steps;

The Standard Widening ∇_s on $\mathbb{CP}_n$ (III)

➜ The resulting operator is both precise and efficient: this "*tentative*" definition has been the one and only available approach for 25 years.

➜ Its precision can be improved (while keeping the convergence guarantee) by applying

① the widening delay technique: delay the application of the widening for a fixed number of iteration steps;

② the widening 'up to' technique: partially recover from rough approximations that go beyond a fixed set of constraints that are known to hold for the considered application.

The Standard Widening ∇_s on $\mathbb{CP}_n$ (III)

➜ The resulting operator is both precise and efficient: this "*tentative*" definition has been the one and only available approach for 25 years.

➜ Its precision can be improved (while keeping the convergence guarantee) by applying

① the widening delay technique: delay the application of the widening for a fixed number of iteration steps;

② the widening 'up to' technique: partially recover from rough approximations that go beyond a fixed set of constraints that are known to hold for the considered application.

➜ For an increasing number of applications, this precision level is not sufficient. Can we further improve upon the precision of the standard widening? (Perhaps, trading some efficiency.)

∇-compatible Limited Growth Ordering

➜ Let $\langle L, \sqsubseteq, \perp, \sqcup \rangle$ be a join-semi-lattice.

➜ A limited growth ordering (lgo) is the strict version of a finitely computable preorder relation that satisfies the ascending chain condition on L.

① preorder: reflexive and transitive;

② ascending chain condition $\sim$ well-founded;

③ computable: we will use it in the implementation.

∇-COMPATIBLE LIMITED GROWTH ORDERING

- Let $\langle L, \sqsubseteq, \perp, \sqcup \rangle$ be a join-semi-lattice.
- A limited growth ordering (lgo) is the strict version of a finitely computable preorder relation that satisfies the ascending chain condition on L.
  ① preorder: reflexive and transitive;
  ② ascending chain condition $\sim$ well-founded;
  ③ computable: we will use it in the implementation.
- Let ∇ be a widening on L. An lgo $\curvearrowright$ is ∇-compatible if

$$\forall x, y \in L : x \sqsubset y \implies x \curvearrowright x \nabla y.$$

∇-COMPATIBLE LIMITED GROWTH ORDERING

➜ Let $\langle L, \sqsubseteq, \perp, \sqcup \rangle$ be a join-semi-lattice.

➜ A limited growth ordering (lgo) is the strict version of a finitely computable preorder relation that satisfies the ascending chain condition on L.

① preorder: reflexive and transitive;

② ascending chain condition $\sim$ well-founded;

③ computable: we will use it in the implementation.

➜ Let ∇ be a widening on L. An lgo $\curvearrowright$ is ∇-compatible if

$$\forall x, y \in L : x \sqsubset y \implies x \curvearrowright x \nabla y.$$

➜ A ∇-compatible lgo formalizes the notion of computable convergence guarantee for the widening ∇.

A Framework for Improving Upon a Fixed Widening

Suppose that

- $\nabla\colon L \times L \rightarrowtail L$ is a widening on the join-semi-lattice $\langle L, \sqsubseteq, \bot, \sqcup \rangle$;
- $\curvearrowright \subseteq L \times L$ is a ∇-compatible lgo;
- $h\colon L \times L \to L$ is an upper bound operator.

A Framework for Improving Upon a Fixed Widening

Suppose that

- $\nabla \colon L \times L \rightarrowtail L$ is a widening on the join-semi-lattice $\langle L, \sqsubseteq, \bot, \sqcup \rangle$;
- $\curvearrowright \subseteq L \times L$ is a ∇-compatible lgo;
- $h \colon L \times L \to L$ is an upper bound operator.

For all $x, y \in L$ such that $x \sqsubseteq y$, define

$$x \mathbin{\tilde{\nabla}} y \stackrel{\mathrm{def}}{=} \begin{cases} h(x, y), & \text{if } x \curvearrowright h(x, y) \sqsubset x \mathbin{\nabla} y; \\ x \mathbin{\nabla} y, & \text{otherwise.} \end{cases}$$

A Framework for Improving Upon a Fixed Widening

Suppose that

- $\nabla \colon L \times L \rightarrowtail L$ is a widening on the join-semi-lattice $\langle L, \sqsubseteq, \bot, \sqcup \rangle$;
- $\curvearrowright \subseteq L \times L$ is a ∇-compatible lgo;
- $h \colon L \times L \to L$ is an upper bound operator.

For all $x, y \in L$ such that $x \sqsubseteq y$, define

$$x \mathbin{\tilde{\nabla}} y \stackrel{\text{def}}{=} \begin{cases} h(x, y), & \text{if } x \curvearrowright h(x, y) \sqsubset x \mathbin{\nabla} y; \\ x \mathbin{\nabla} y, & \text{otherwise.} \end{cases}$$

- Then $\tilde{\nabla}$ is a widening operator at least as precise as ∇.

A FINE-GRAINED LGO ON $\mathbb{CP}_n$

➜ Variant of a well-founded preorder defined in Besson *et al.*, SAS'99. It is obtained as the lexicographic product of five lgo's.

A Fine-Grained lgo on $\mathbb{CP}_n$

- Variant of a well-founded preorder defined in Besson *et al.*, SAS'99. It is obtained as the lexicographic product of five lgo's.
- For $i = 1, 2$, let $\mathcal{P}_i = \mathrm{con}(\mathcal{C}_i) = \mathrm{gen}(\mathcal{G}_i) \neq \varnothing$, where $\mathcal{C}_i$ is in minimal form and $\mathcal{G}_i = (L_i, R_i, P_i)$ is in orthogonal form.

A Fine-Grained lgo on $\mathbb{CP}_n$

➜ Variant of a well-founded preorder defined in Besson *et al.*, SAS'99. It is obtained as the lexicographic product of five lgo's.

➜ For $i = 1, 2$, let $\mathcal{P}_i = \mathrm{con}(\mathcal{C}_i) = \mathrm{gen}(\mathcal{G}_i) \neq \varnothing$, where $\mathcal{C}_i$ is in minimal form and $\mathcal{G}_i = (L_i, R_i, P_i)$ is in orthogonal form.

① $\mathcal{P}_1 \preceq_d \mathcal{P}_2 \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \#\,\mathrm{eq}(\mathcal{C}_1) \geq \#\,\mathrm{eq}(\mathcal{C}_2)$;

② $\mathcal{P}_1 \preceq_\ell \mathcal{P}_2 \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \#\,L_1 \leq \#\,L_2$;

③ $\mathcal{P}_1 \preceq_c \mathcal{P}_2 \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \#\,\mathcal{C}_1 \geq \#\,\mathcal{C}_2$;

④ $\mathcal{P}_1 \preceq_p \mathcal{P}_2 \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \#\,P_1 \geq \#\,P_2$;

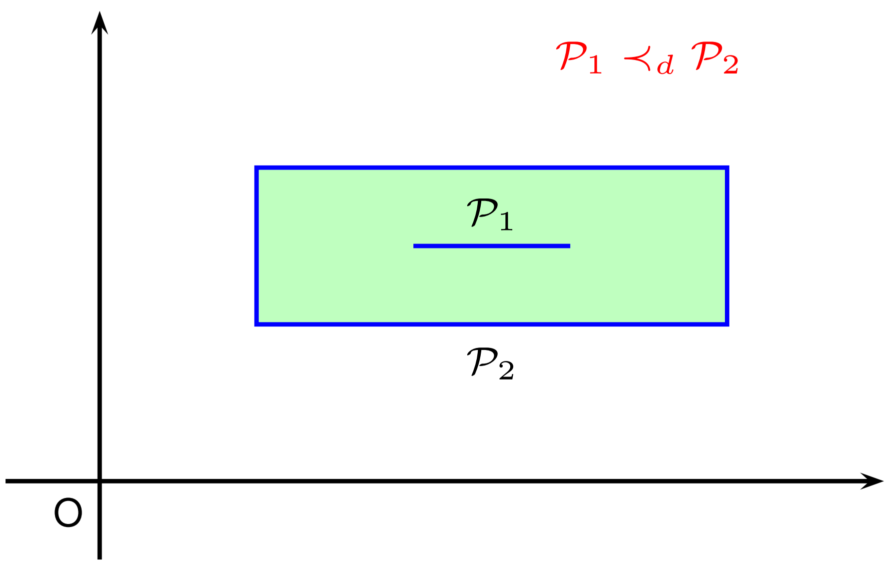
⑤ $\mathcal{P}_1 \preceq_r \mathcal{P}_2 \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \kappa(R_1) \sqsubseteq_{ms} \kappa(R_2)$.

A FINE-GRAINED LGO ON $\mathbb{CP}_n$

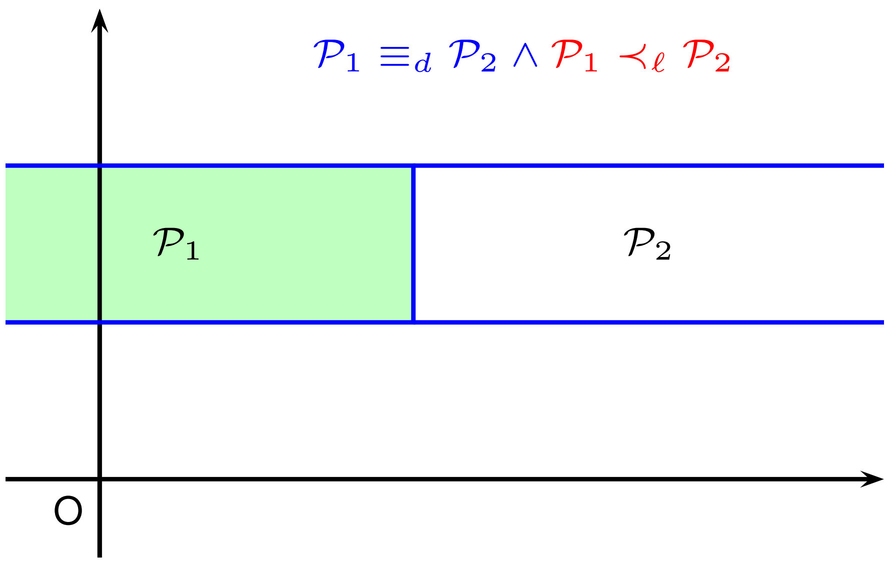
- Variant of a well-founded preorder defined in Besson *et al.*, SAS'99. It is obtained as the lexicographic product of five lgo's.
- For $i = 1, 2$, let $\mathcal{P}_i = \mathrm{con}(\mathcal{C}_i) = \mathrm{gen}(\mathcal{G}_i) \neq \varnothing$,
  where $\mathcal{C}_i$ is in minimal form and $\mathcal{G}_i = (L_i, R_i, P_i)$ is in orthogonal form.
  ① $\mathcal{P}_1 \preceq_d \mathcal{P}_2 \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \#\,\mathrm{eq}(\mathcal{C}_1) \geq \#\,\mathrm{eq}(\mathcal{C}_2)$;
  ② $\mathcal{P}_1 \preceq_\ell \mathcal{P}_2 \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \#\,L_1 \leq \#\,L_2$;
  ③ $\mathcal{P}_1 \preceq_c \mathcal{P}_2 \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \#\,\mathcal{C}_1 \geq \#\,\mathcal{C}_2$;
  ④ $\mathcal{P}_1 \preceq_p \mathcal{P}_2 \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \#\,P_1 \geq \#\,P_2$;
  ⑤ $\mathcal{P}_1 \preceq_r \mathcal{P}_2 \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \kappa(R_1) \sqsubseteq_{ms} \kappa(R_2)$.
- We denote by $\curvearrowright_n$ the strict version of the lexicographic product

$$\mathcal{P}_1 \preceq_n \mathcal{P}_2 \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \mathcal{P}_1 \preceq_{d\ell cpr} \mathcal{P}_2.$$

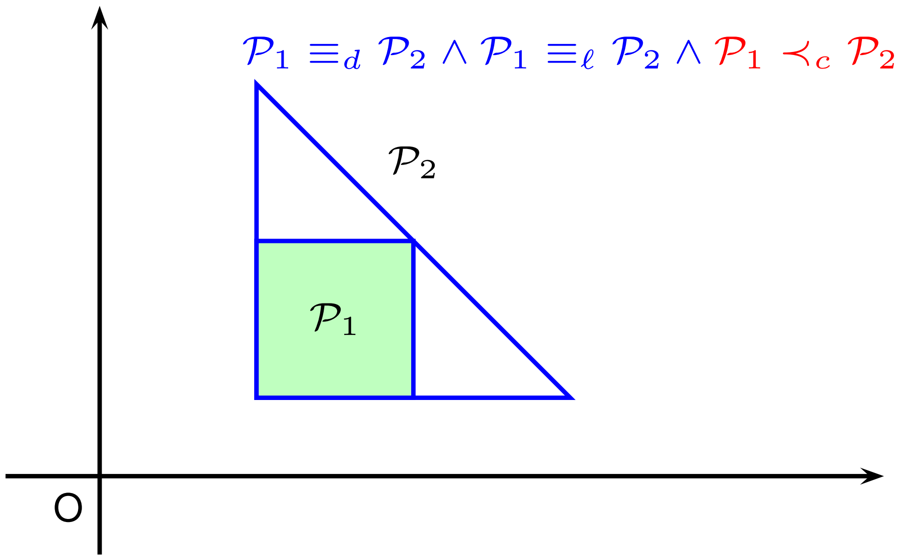
EXAMPLES FOR $\mathcal{P}_1 \curvearrowright_n \mathcal{P}_2$: CASE 1



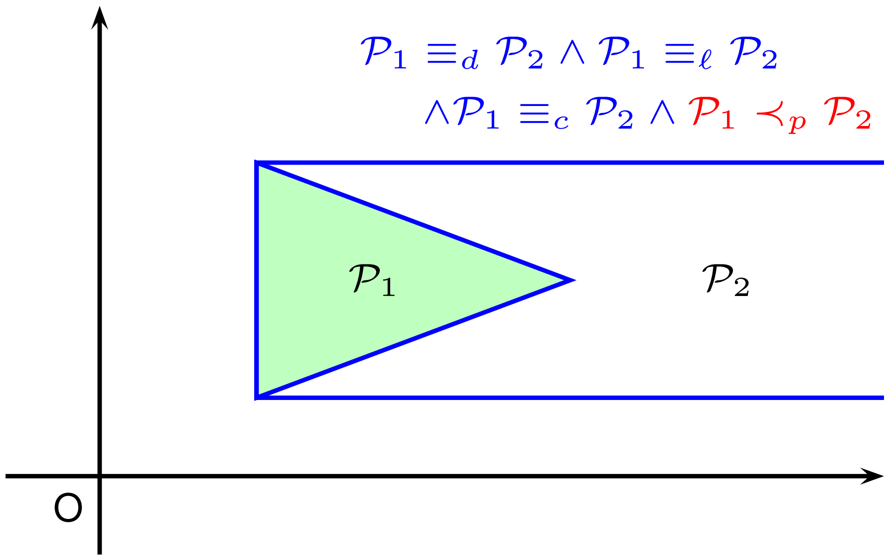
Examples for $\mathcal{P}_1 \curvearrowright_n \mathcal{P}_2$: Case 2



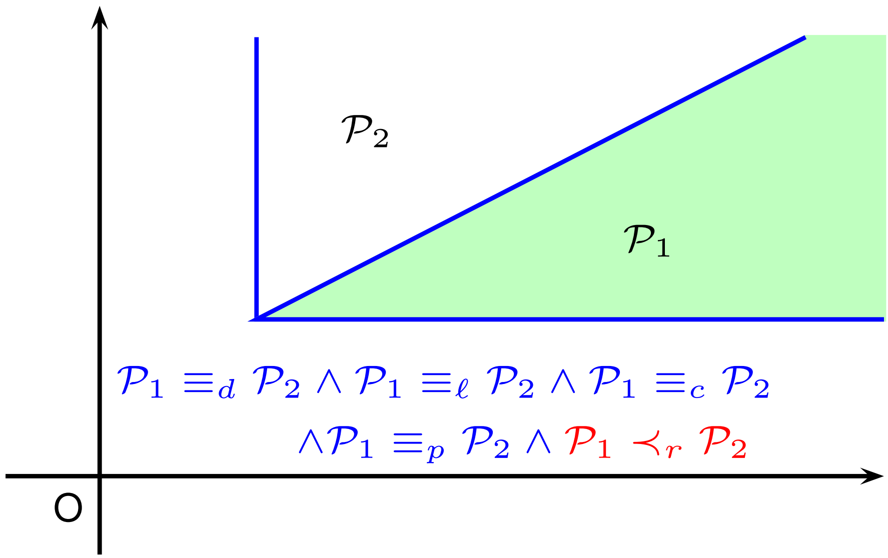
Examples for $\mathcal{P}_1 \curvearrowright_n \mathcal{P}_2$: Case 3



EXAMPLES FOR $\mathcal{P}_1 \curvearrowright_n \mathcal{P}_2$: CASE 4



EXAMPLES FOR $\mathcal{P}_1 \curvearrowright_n \mathcal{P}_2$: CASE 5



INSTANTIATING THE FRAMEWORK

The key result.

➜ $\curvearrowright_n$ is a ∇_s-compatible lgo on $\mathbb{CP}_n$.
(This is not the case for the ordering defined in Besson *et al.*, SAS'99.)

Instantiating the Framework

The key result.

- ➜ $\curvearrowright_n$ is a ∇_s-compatible lgo on $\mathbb{CP}_n$.
  (This is not the case for the ordering defined in Besson *et al.*, SAS'99.)
- ➜ For any upper bound operator $h \colon \mathbb{CP}_n \times \mathbb{CP}_n \to \mathbb{CP}_n$, the framework will return a proper widening operator on $\mathbb{CP}_n$ improving on the standard widening.

INSTANTIATING THE FRAMEWORK

The key result.

- ➜ $\curvearrowright_n$ is a ∇_s-compatible lgo on $\mathbb{CP}_n$.
  (This is not the case for the ordering defined in Besson *et al.*, SAS'99.)
- ➜ For any upper bound operator $h \colon \mathbb{CP}_n \times \mathbb{CP}_n \to \mathbb{CP}_n$, the framework will return a proper widening operator on $\mathbb{CP}_n$ improving on the standard widening.
- ➜ In our attempt to improve precision, we can consider any finite set of such heuristic techniques: our new widening will use four upper bounds.

1ST HEURISTICS: DO NOT WIDEN

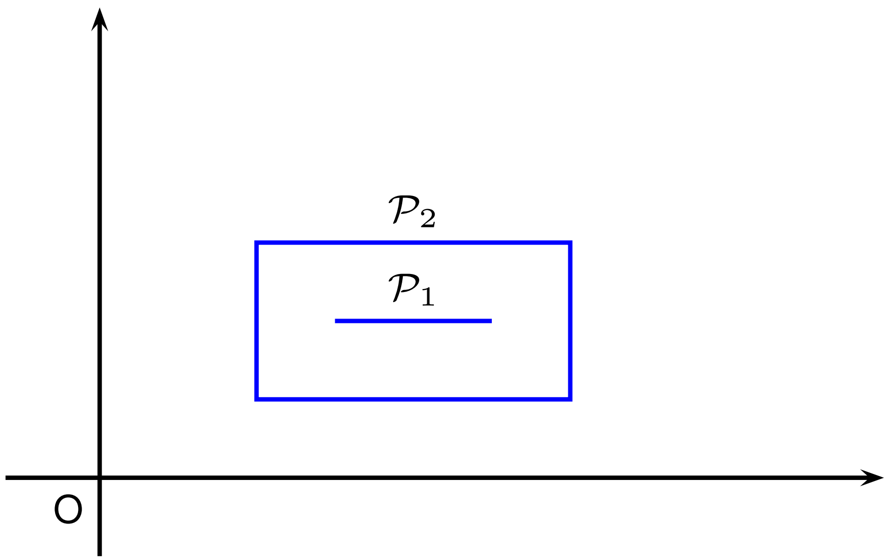
Let h be the least upper bound, so that $h(\mathcal{P}_1, \mathcal{P}_2) = \mathcal{P}_2$.

1ST HEURISTICS: DO NOT WIDEN

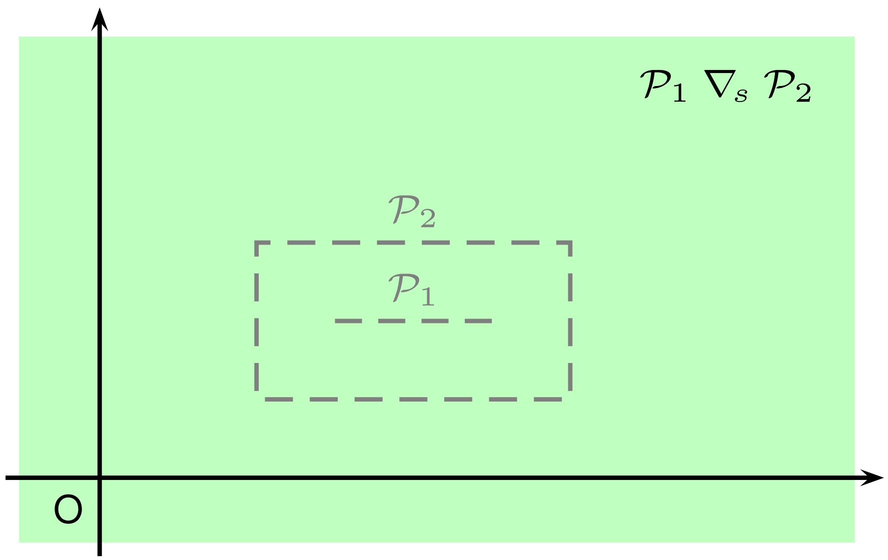
Let h be the least upper bound, so that $h(\mathcal{P}_1, \mathcal{P}_2) = \mathcal{P}_2$.

- Applicable whenever $\mathcal{P}_1 \curvearrowright \mathcal{P}_2$.
- No precision loss: to be tried before all other techniques.
- Already suggested by Cousot and Cousot, PLILP'92.

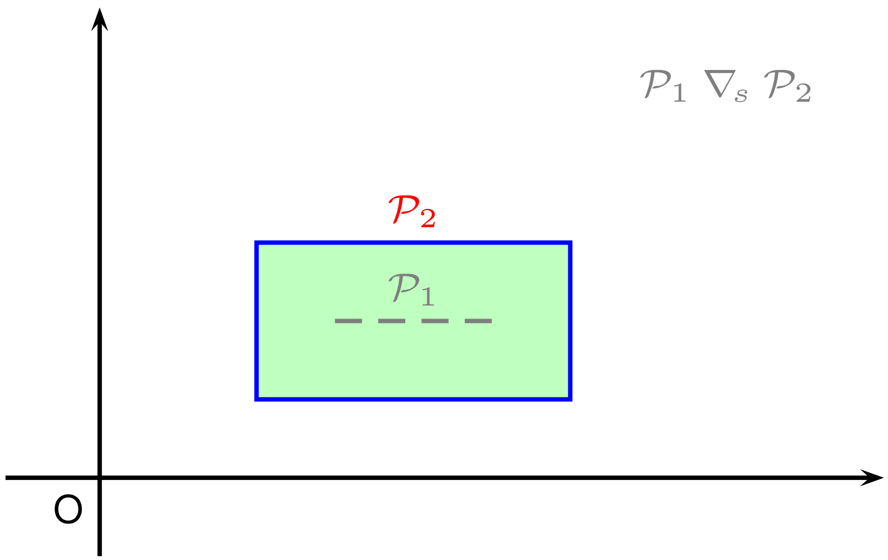
STANDARD WIDENING VS. DO NOT WIDEN (I)



Standard Widening vs. Do Not Widen (ii)



Standard Widening vs. Do Not Widen (iii)



1ST HEURISTICS: DO NOT WIDEN

Let h be the least upper bound, so that $h(\mathcal{P}_1, \mathcal{P}_2) = \mathcal{P}_2$.

- ➜ Applicable whenever $\mathcal{P}_1 \curvearrowright \mathcal{P}_2$.
- ➜ No precision loss: to be tried before all other techniques.
- ➜ Already suggested by Cousot and Cousot, PLILP'92.
- ➜ All the other techniques may safely assume $\mathcal{P}_1 \not\curvearrowright \mathcal{P}_2$.
- ➜ Since by hypothesis $\mathcal{P}_1 \subseteq \mathcal{P}_2$, we can also assume

$$\begin{aligned} \operatorname{aff.hull}(\mathcal{P}_1) &= \operatorname{aff.hull}(\mathcal{P}_2), \\ \operatorname{lin.space}(\mathcal{P}_1) &= \operatorname{lin.space}(\mathcal{P}_2). \end{aligned}$$

2ND HEURISTICS: COMBINING CONSTRAINTS

Let $h_c(\mathcal{P}_1, \mathcal{P}_2) \stackrel{\text{def}}{=} \operatorname{con}(\mathcal{C}_\oplus) \cap (\mathcal{P}_1 \nabla_s \mathcal{P}_2)$, where

- $\mathcal{C}_\nabla$ are the constraints of the standard widening;
- $\mathcal{C}_\oplus \stackrel{\text{def}}{=} \left\{ \oplus(\mathcal{C}_{\boldsymbol{p}}) \;\middle|\; \begin{array}{l} \boldsymbol{p} \in P_1, \operatorname{sat\_con}\bigl(\boldsymbol{p}, \operatorname{ineq}(\mathcal{C}_\nabla)\bigr) = \varnothing, \\ \mathcal{C}_{\boldsymbol{p}} = \operatorname{sat\_con}\bigl(\boldsymbol{p}, \operatorname{ineq}(\mathcal{C}_2)\bigr) \neq \varnothing \end{array} \right\}$.
- $\oplus$ is a (deliberately left unspecified) convex combination.

Informally, *we ensure that each non-redundant point $\boldsymbol{p} \in \mathcal{P}_1$ that was lying on a facet of $\mathcal{P}_2$ will still lie on a facet of $h_c(\mathcal{P}_1, \mathcal{P}_2)$*.

2nd Heuristics: Combining Constraints

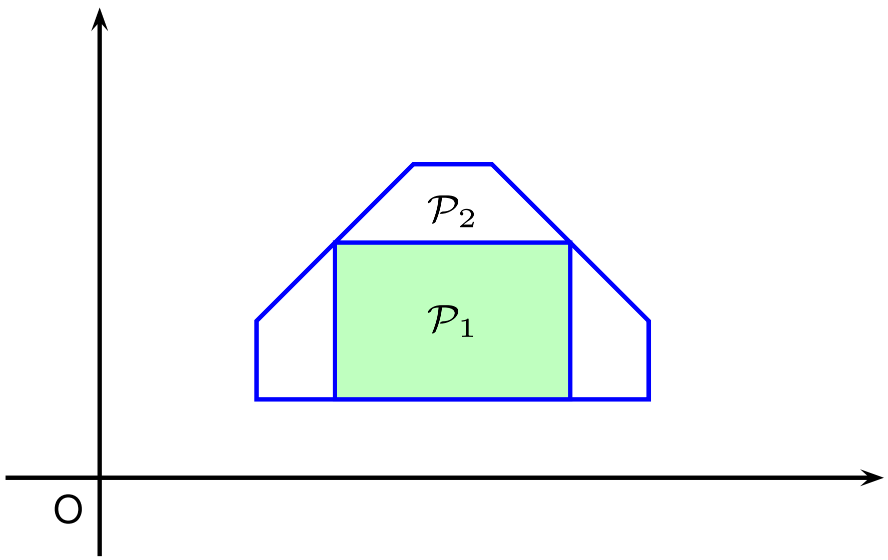
Let $h_c(\mathcal{P}_1, \mathcal{P}_2) \stackrel{\text{def}}{=} \mathrm{con}(\mathcal{C}_\oplus) \cap (\mathcal{P}_1 \nabla_s \mathcal{P}_2)$, where

- ➜ $\mathcal{C}_\nabla$ are the constraints of the standard widening;
- ➜ $\mathcal{C}_\oplus \stackrel{\text{def}}{=} \left\{ \oplus(\mathcal{C}_{\boldsymbol{p}}) \;\middle|\; \begin{array}{l} \boldsymbol{p} \in P_1, \mathrm{sat\_con}\big(\boldsymbol{p}, \mathrm{ineq}(\mathcal{C}_\nabla)\big) = \varnothing, \\ \mathcal{C}_{\boldsymbol{p}} = \mathrm{sat\_con}\big(\boldsymbol{p}, \mathrm{ineq}(\mathcal{C}_2)\big) \neq \varnothing \end{array} \right\}.$
- ➜ $\oplus$ is a (deliberately left unspecified) convex combination.

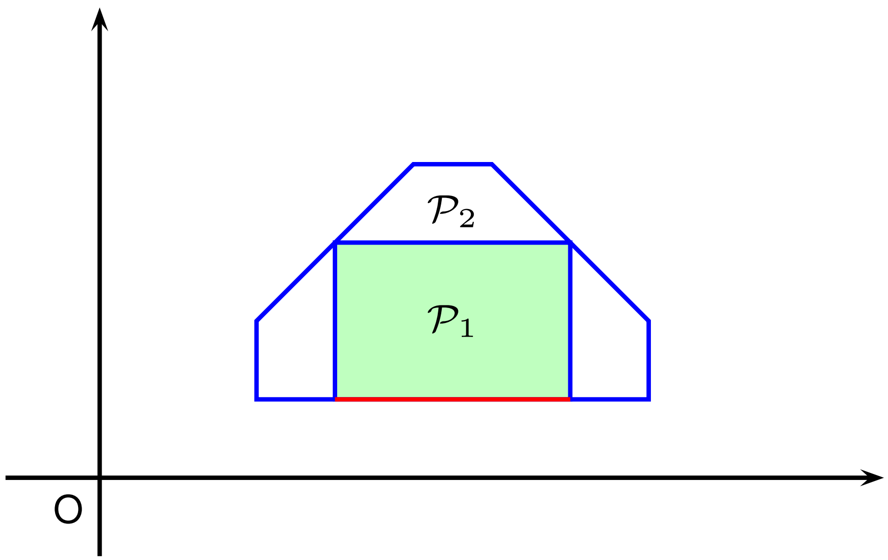
Informally, *we ensure that each non-redundant point $\boldsymbol{p} \in \mathcal{P}_1$ that was lying on a facet of $\mathcal{P}_2$ will still lie on a facet of $h_c(\mathcal{P}_1, \mathcal{P}_2)$.*

- ➜ Besson et al., SAS'99 suggest to average the constraints in $\mathcal{C}_{\boldsymbol{p}}$.
- ➜ Afterall, the choice of $\oplus$ is arbitrary: we opted for a simpler combination.
- ➜ A similar heuristics, with no convergence guarantee, was proposed by Henzinger et al., CDC'01.

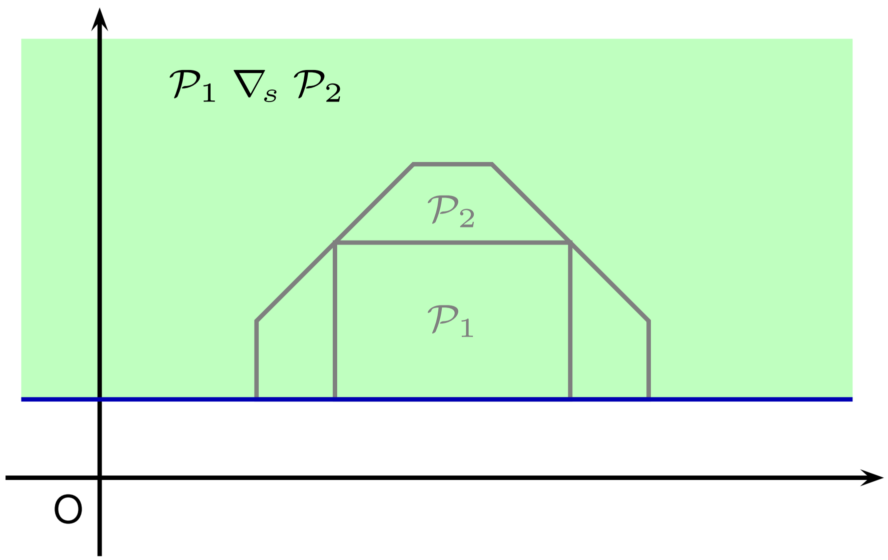
Standard Widening vs. Combining Constraints (i)



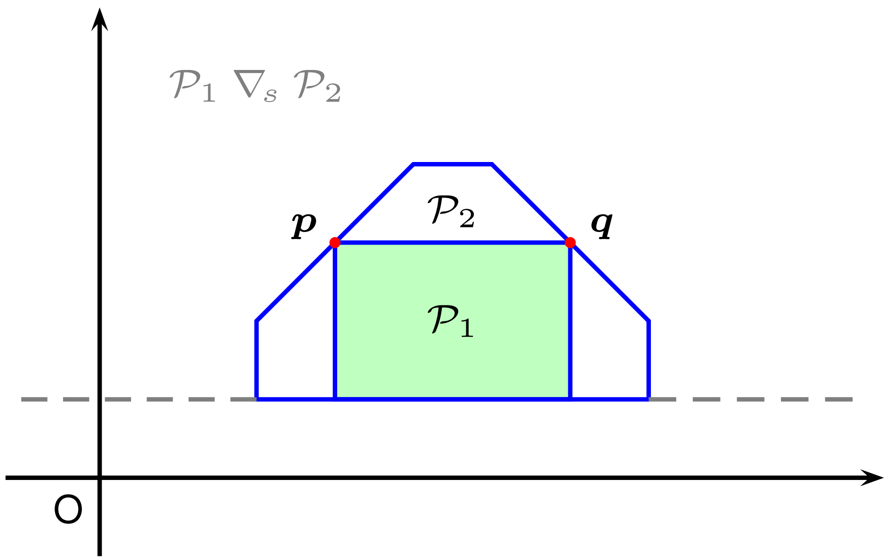
Standard Widening vs. Combining Constraints (II)



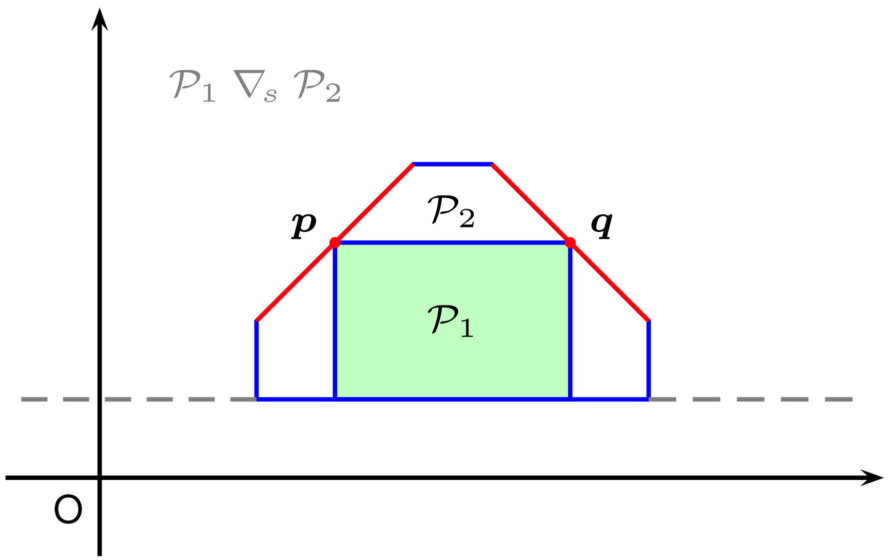
STANDARD WIDENING VS. COMBINING CONSTRAINTS (III)



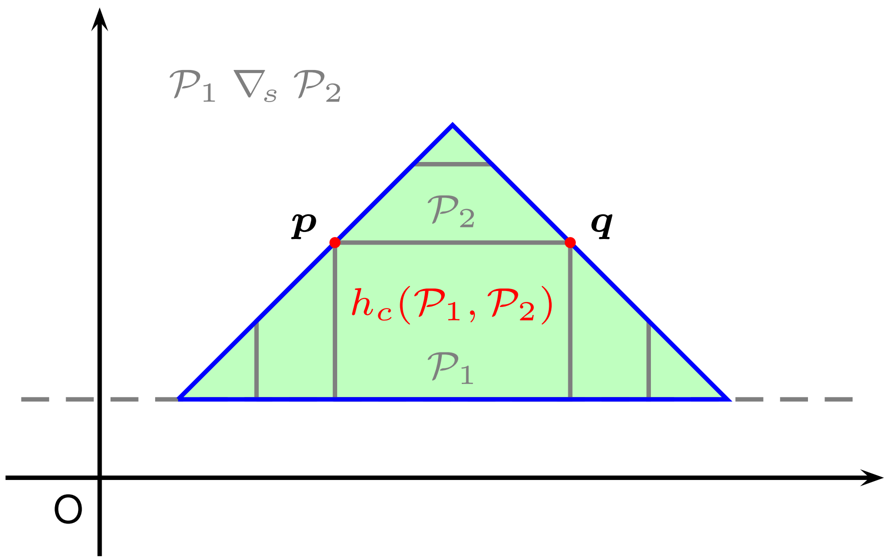
STANDARD WIDENING VS. COMBINING CONSTRAINTS (IV)



Standard Widening vs. Combining Constraints (v)



Standard Widening vs. Combining Constraints (vi)



3RD HEURISTICS: EVOLVING POINTS

- ➜ A (slightly simpler) variant of the extrapolation operator '$\propto$' defined in Henzinger and Ho, Hibrid Systems II, 95.
- ➜ Also similar to another operator sketched in Besson et al., SAS'99.

3RD HEURISTICS: EVOLVING POINTS

➜ A (slightly simpler) variant of the extrapolation operator '$\propto$' defined in Henzinger and Ho, Hibrid Systems II, 95.

➜ Also similar to another operator sketched in Besson et al., SAS'99.

➜ Consider the set of rays

$$R \stackrel{\text{def}}{=} \{\, \boldsymbol{p}_2 - \boldsymbol{p}_1 \bigm| \boldsymbol{p}_1 \in P_1, \boldsymbol{p}_2 \in P_2 \setminus P_1 \,\}.$$

➜ Informally, *each point $\boldsymbol{p}_2 \in P_2 \setminus P_1$ is seen as an evolution of point $\boldsymbol{p}_1 \in P_1$. By generating the ray $\boldsymbol{p}_2 - \boldsymbol{p}_1$, we extrapolate this evolution towards infinity.*

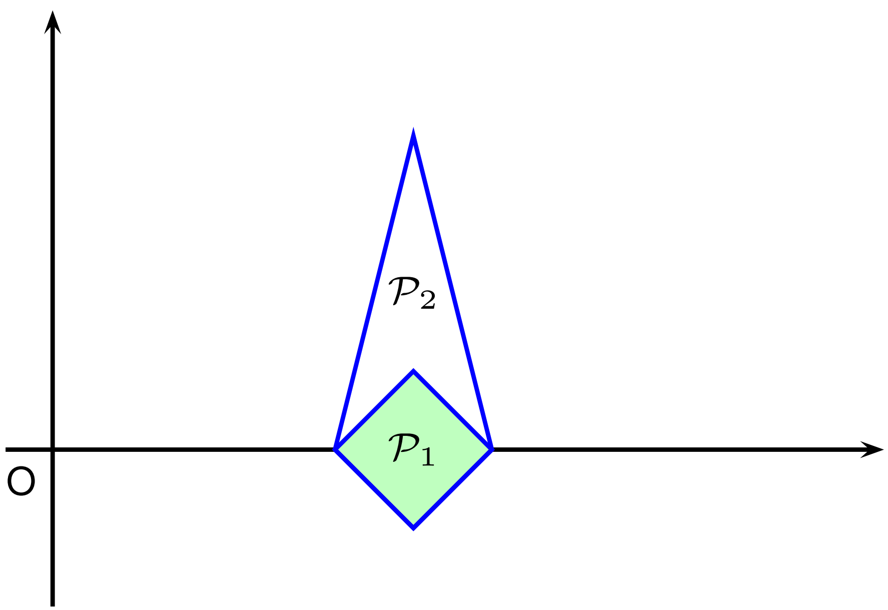
3RD HEURISTICS: EVOLVING POINTS

- A (slightly simpler) variant of the extrapolation operator '$\propto$' defined in Henzinger and Ho, Hibrid Systems II, 95.
- Also similar to another operator sketched in Besson et al., SAS'99.
- Consider the set of rays

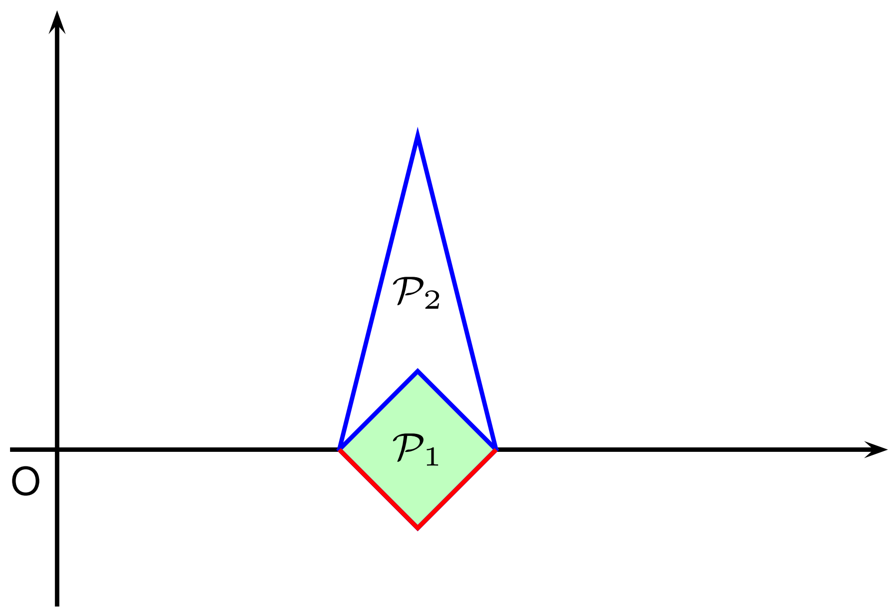
$$R \stackrel{\text{def}}{=} \left\{ \boldsymbol{p}_2 - \boldsymbol{p}_1 \;\middle|\; \boldsymbol{p}_1 \in P_1, \boldsymbol{p}_2 \in P_2 \setminus P_1 \right\}.$$

- Informally, *each point $\boldsymbol{p}_2 \in P_2 \setminus P_1$ is seen as an evolution of point $\boldsymbol{p}_1 \in P_1$. By generating the ray $\boldsymbol{p}_2 - \boldsymbol{p}_1$, we extrapolate this evolution towards infinity.*
- Thus, let $h_p(\mathcal{P}_1, \mathcal{P}_2) \stackrel{\text{def}}{=} \operatorname{gen}\big((L_2, R_2 \cup R, P_2)\big) \cap (\mathcal{P}_1 \nabla_s \mathcal{P}_2)$.

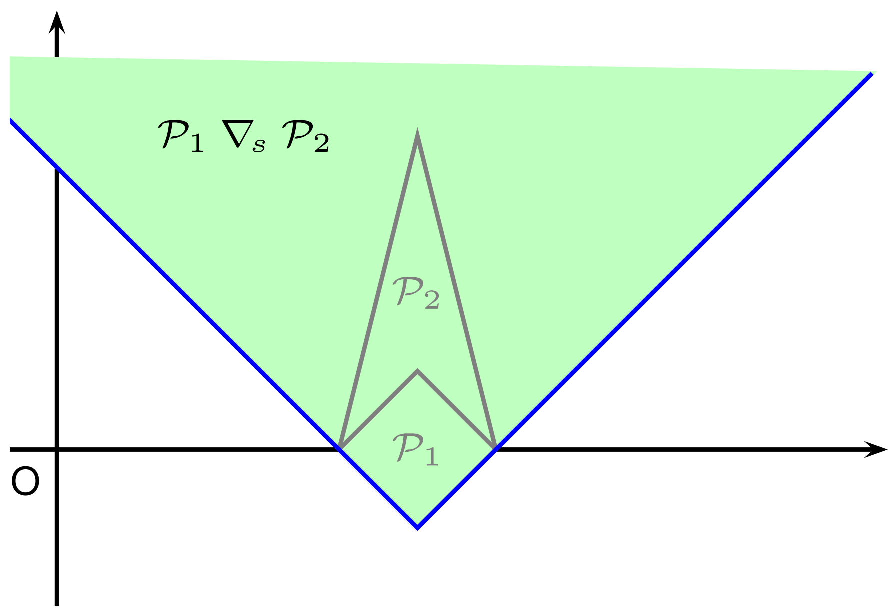
STANDARD WIDENING VS. EVOLVING POINTS (I)



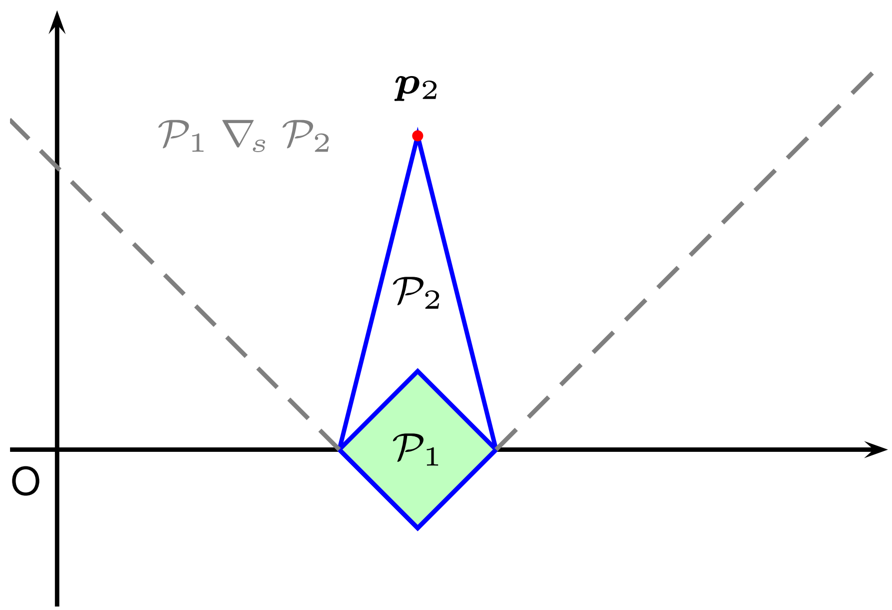
Standard Widening vs. Evolving Points (ii)



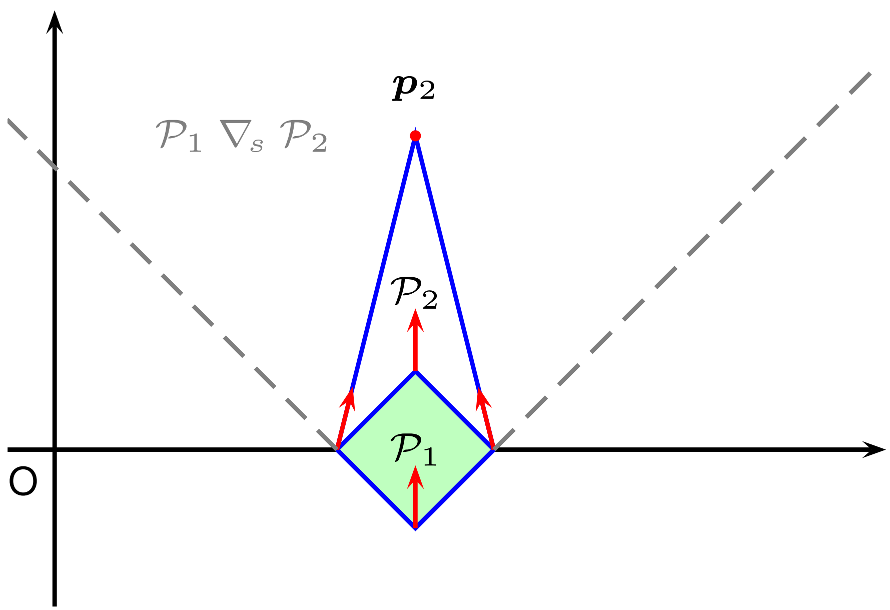
STANDARD WIDENING VS. EVOLVING POINTS (III)



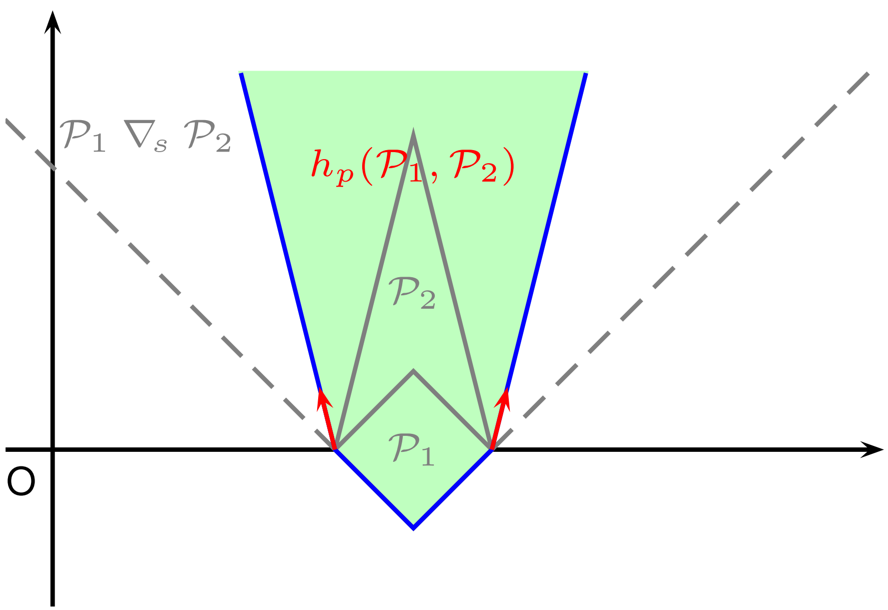
STANDARD WIDENING VS. EVOLVING POINTS (IV)



Standard Widening vs. Evolving Points (v)



Standard Widening vs. Evolving Points (vi)



4TH HEURISTICS: EVOLVING RAYS

➜ A brand new widening heuristics.

4TH HEURISTICS: EVOLVING RAYS

- ➜ A brand new widening heuristics.
- ➜ Define the set of rays

$$R \stackrel{\mathrm{def}}{=} \left\{ \mathrm{evolve}(\boldsymbol{r}_2, \boldsymbol{r}_1) \;\middle|\; \boldsymbol{r}_1 \in R_1, \boldsymbol{r}_2 \in R_2 \setminus R_1 \right\}.$$

- ➜ Informally, *each ray $\boldsymbol{r}_2 \in R_2 \setminus R_1$ is seen as an evolution of ray $\boldsymbol{r}_1 \in R_1$. We extrapolate this evolution by rotating ray $\boldsymbol{r}_2$, stopping as soon as it touches the boundary of the Cartesian orthant*.

4TH HEURISTICS: EVOLVING RAYS

- A brand new widening heuristics.
- Define the set of rays

$$R \stackrel{\text{def}}{=} \left\{ \operatorname{evolve}(\boldsymbol{r}_2, \boldsymbol{r}_1) \;\middle|\; \boldsymbol{r}_1 \in R_1, \boldsymbol{r}_2 \in R_2 \setminus R_1 \right\}.$$

- Informally, *each ray $\boldsymbol{r}_2 \in R_2 \setminus R_1$ is seen as an evolution of ray $\boldsymbol{r}_1 \in R_1$. We extrapolate this evolution by rotating ray $\boldsymbol{r}_2$, stopping as soon as it touches the boundary of the Cartesian orthant*.
- Thus, let $h_r(\mathcal{P}_1, \mathcal{P}_2) \stackrel{\text{def}}{=} \operatorname{gen}\big((L_2, R_2 \cup R, P_2)\big) \cap (\mathcal{P}_1 \mathbin{\nabla_s} \mathcal{P}_2)$.

4TH HEURISTICS: EVOLVING RAYS

➜ A brand new widening heuristics.

➜ Define the set of rays

$$R \stackrel{\text{def}}{=} \{\, \text{evolve}(\boldsymbol{r}_2, \boldsymbol{r}_1) \mid \boldsymbol{r}_1 \in R_1, \boldsymbol{r}_2 \in R_2 \setminus R_1 \,\}.$$

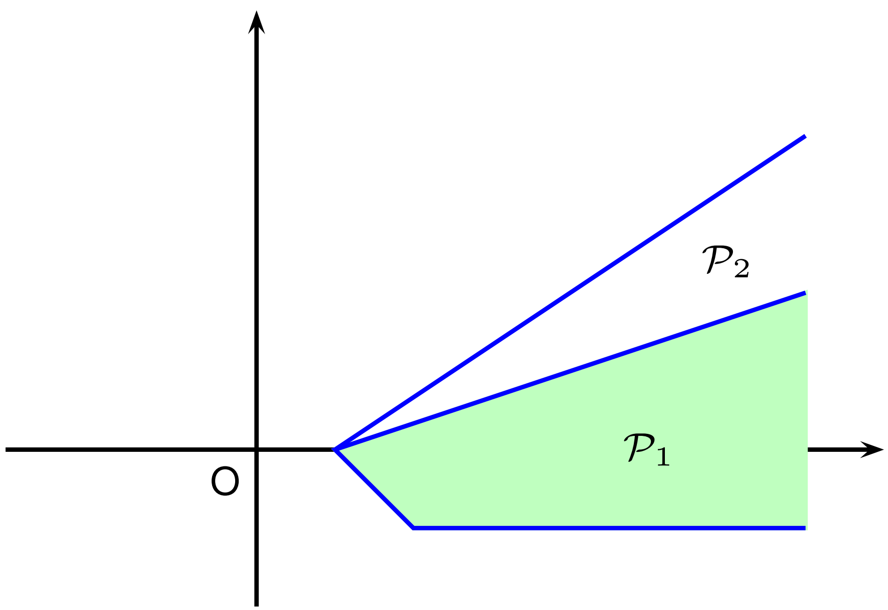
➜ Informally, *each ray $\boldsymbol{r}_2 \in R_2 \setminus R_1$ is seen as an evolution of ray $\boldsymbol{r}_1 \in R_1$. We extrapolate this evolution by rotating ray $\boldsymbol{r}_2$, stopping as soon as it touches the boundary of the Cartesian orthant*.

➜ Thus, let $h_r(\mathcal{P}_1, \mathcal{P}_2) \stackrel{\text{def}}{=} \text{gen}\big((L_2, R_2 \cup R, P_2)\big) \cap (\mathcal{P}_1 \nabla_s \mathcal{P}_2)$.

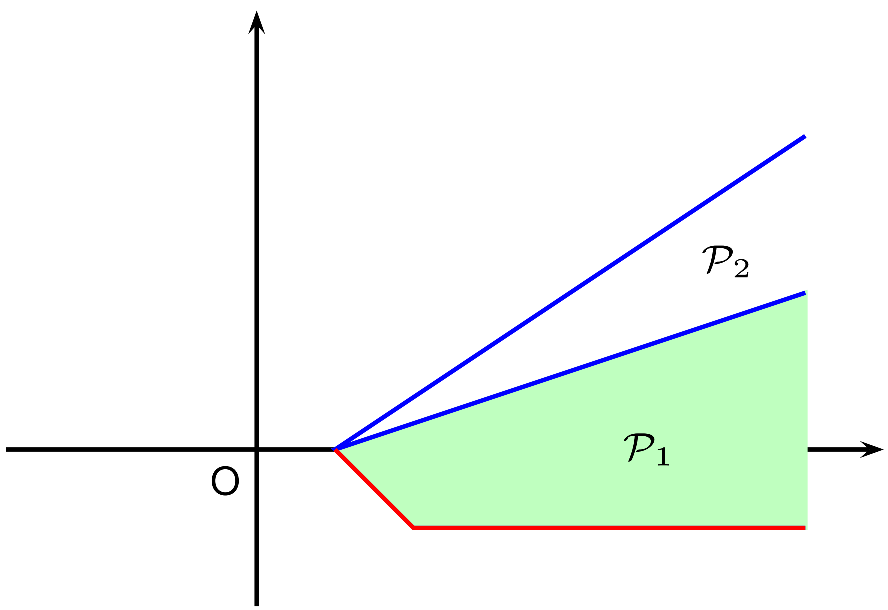
➜ The extrapolation will decrease the total number of non-zero coordinates of the ray $\implies$ hopefully satisfying the last case in the definition of the lgo $\curvearrowright_n$:

$$\mathcal{P}_1 \prec_r h_r(\mathcal{P}_1, \mathcal{P}_2).$$

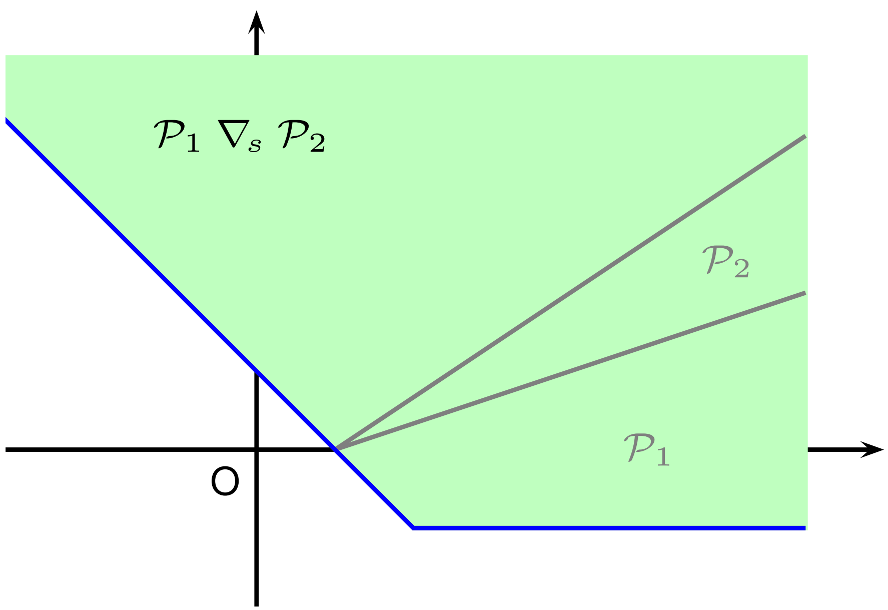
Standard Widening vs. Evolving Rays (i)



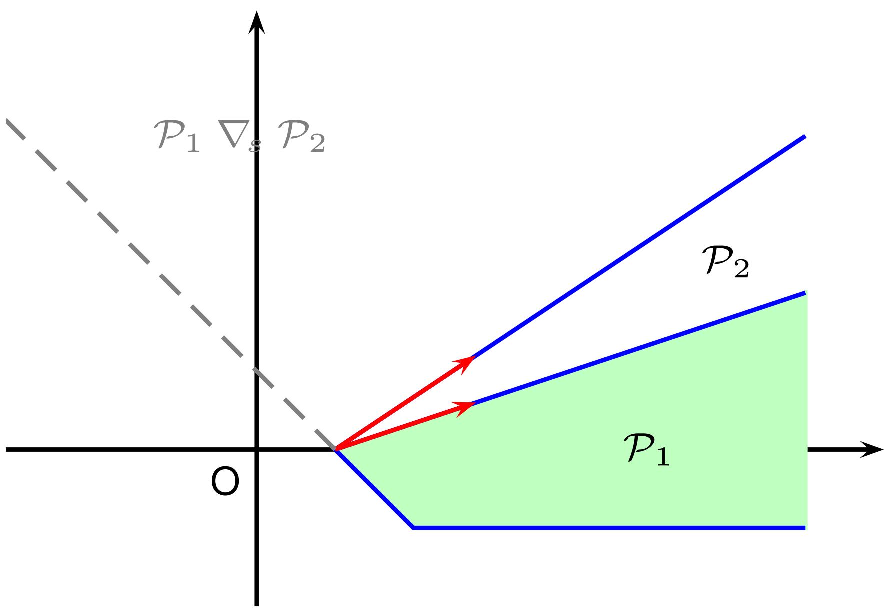
STANDARD WIDENING VS. EVOLVING RAYS (II)



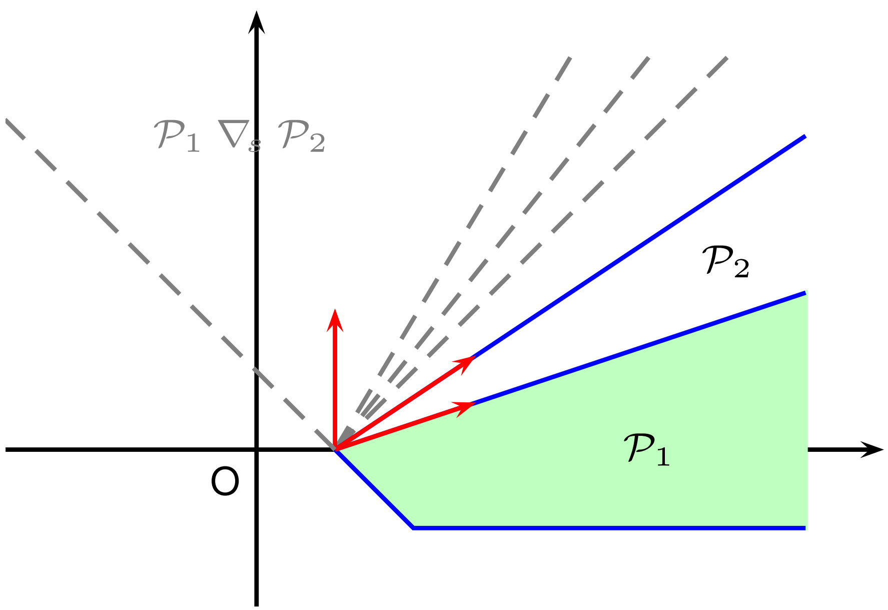
STANDARD WIDENING VS. EVOLVING RAYS (III)



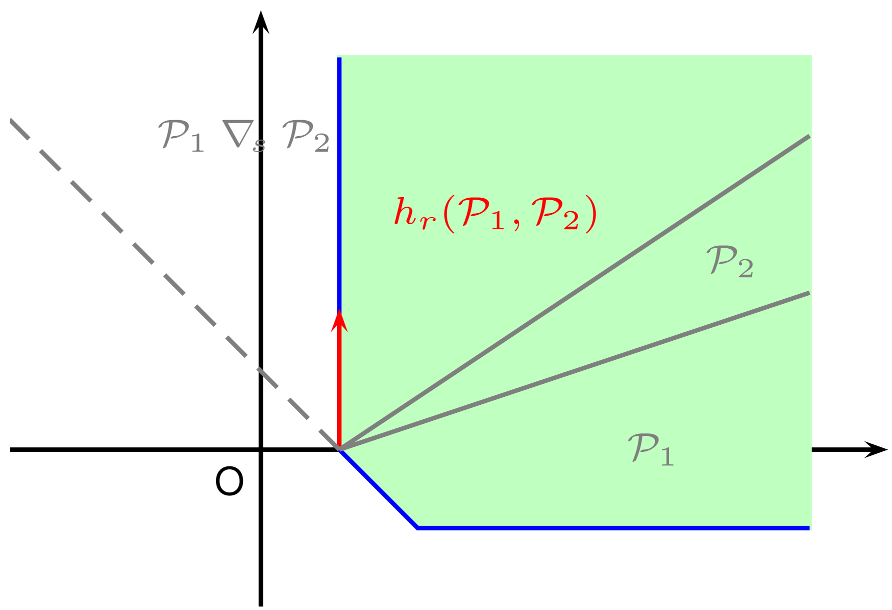
STANDARD WIDENING VS. EVOLVING RAYS (IV)



Standard Widening vs. Evolving Rays (v)



Standard Widening vs. Evolving Rays (vi)



The New Widening ∇_n

➜ An instance of the framework: try the four heuristics in the given order, eventually falling back to the standard widening.

$$\mathcal{P}_1 \nabla_n \mathcal{P}_2 \stackrel{\text{def}}{=} \begin{cases} \mathcal{P}_2, & \text{if } \mathcal{P}_1 \curvearrowright \mathcal{P}_2; \\ h_c(\mathcal{P}_1, \mathcal{P}_2), & \text{if } \mathcal{P}_1 \curvearrowright h_c(\mathcal{P}_1, \mathcal{P}_2) \subset \mathcal{P}_1 \nabla_s \mathcal{P}_2; \\ h_p(\mathcal{P}_1, \mathcal{P}_2), & \text{if } \mathcal{P}_1 \curvearrowright h_p(\mathcal{P}_1, \mathcal{P}_2) \subset \mathcal{P}_1 \nabla_s \mathcal{P}_2; \\ h_r(\mathcal{P}_1, \mathcal{P}_2), & \text{if } \mathcal{P}_1 \curvearrowright h_r(\mathcal{P}_1, \mathcal{P}_2) \subset \mathcal{P}_1 \nabla_s \mathcal{P}_2; \\ \mathcal{P}_1 \nabla_s \mathcal{P}_2, & \text{otherwise.} \end{cases}$$

The New Widening ∇_n

- An instance of the framework: try the four heuristics in the given order, eventually falling back to the standard widening.

$$\mathcal{P}_1 \nabla_n \mathcal{P}_2 \stackrel{\mathrm{def}}{=} \begin{cases} \mathcal{P}_2, & \text{if } \mathcal{P}_1 \curvearrowright \mathcal{P}_2; \\ h_c(\mathcal{P}_1, \mathcal{P}_2), & \text{if } \mathcal{P}_1 \curvearrowright h_c(\mathcal{P}_1, \mathcal{P}_2) \subset \mathcal{P}_1 \nabla_s \mathcal{P}_2; \\ h_p(\mathcal{P}_1, \mathcal{P}_2), & \text{if } \mathcal{P}_1 \curvearrowright h_p(\mathcal{P}_1, \mathcal{P}_2) \subset \mathcal{P}_1 \nabla_s \mathcal{P}_2; \\ h_r(\mathcal{P}_1, \mathcal{P}_2), & \text{if } \mathcal{P}_1 \curvearrowright h_r(\mathcal{P}_1, \mathcal{P}_2) \subset \mathcal{P}_1 \nabla_s \mathcal{P}_2; \\ \mathcal{P}_1 \nabla_s \mathcal{P}_2, & \text{otherwise.} \end{cases}$$

- Uniformly more precise than the standard widening.

THE NEW WIDENING ∇_n

- An instance of the framework: try the four heuristics in the given order, eventually falling back to the standard widening.

$$\mathcal{P}_1 \nabla_n \mathcal{P}_2 \stackrel{\text{def}}{=} \begin{cases} \mathcal{P}_2, & \text{if } \mathcal{P}_1 \curvearrowright \mathcal{P}_2; \\ h_c(\mathcal{P}_1, \mathcal{P}_2), & \text{if } \mathcal{P}_1 \curvearrowright h_c(\mathcal{P}_1, \mathcal{P}_2) \subset \mathcal{P}_1 \nabla_s \mathcal{P}_2; \\ h_p(\mathcal{P}_1, \mathcal{P}_2), & \text{if } \mathcal{P}_1 \curvearrowright h_p(\mathcal{P}_1, \mathcal{P}_2) \subset \mathcal{P}_1 \nabla_s \mathcal{P}_2; \\ h_r(\mathcal{P}_1, \mathcal{P}_2), & \text{if } \mathcal{P}_1 \curvearrowright h_r(\mathcal{P}_1, \mathcal{P}_2) \subset \mathcal{P}_1 \nabla_s \mathcal{P}_2; \\ \mathcal{P}_1 \nabla_s \mathcal{P}_2, & \text{otherwise.} \end{cases}$$

- Uniformly more precise than the standard widening.
- In general, this does not hold for the final result of upward iteration sequences, because neither the standard widening nor the new one are monotonic operators.

Precision Comparison

Argument size relations for Prolog programs using China $+$ PPL.

Note: carefully chosen widening strategy (Bourdoncle, FMPTA'93)
$+$ widening delay $+$ widening 'up to'.

PRECISION COMPARISON

Argument size relations for Prolog programs using China + PPL.

Note: carefully chosen widening strategy (Bourdoncle, FMPTA'93) + widening delay + widening 'up to'.

	# programs (361)			# predicates (23279)		
k (delay)	improve	degr	incomp	improve	degr	incomp
0	121	-	2	1340	3	2
1	34	-	-	273	-	-
2	29	-	-	222	-	-
3	28	-	-	160	-	-
4	25	-	2	126	2	-
10	25	-	-	124	-	-

Efficiency Comparison

Argument size relations for Prolog programs using China $+$ PPL.

Total analysis time

	${}_k\nabla_s$		${}_k\nabla_n$	
k (delay)	all	top 20	all	top 20
0	**1.00**	0.72	1.05	0.77
1	1.09	0.79	1.11	0.80
2	1.16	0.83	1.18	0.84
3	1.23	0.88	1.25	0.89
4	1.32	0.95	1.34	0.95
10	1.82	1.23	1.85	1.24

Conclusion

- We have defined a domain independent framework for improving upon the precision of a fixed widening operator;
- We have instantiated the framework on the domain of convex polyhedra improving on the precision of the standard widening;
- The new widening has been implemented in the PPL and a first experimental evaluation has yielded promising results.

Conclusion

- ➜ We have defined a domain independent framework for improving upon the precision of a fixed widening operator;
- ➜ We have instantiated the framework on the domain of convex polyhedra improving on the precision of the standard widening;
- ➜ The new widening has been implemented in the PPL and a first experimental evaluation has yielded promising results.

Current and Future Work

- ➜ Widening operators are the corner stone for both the feasibility and precision of static analyses adopting accurate abstract domains:
  1. We have defined (generic) widenings for disjunctive domains, such as finite sets of polyhedra (see the last planned seminar);
  2. Many interesting domains are still missing (non-trivial) widening operators (e.g., $\mathbb{Z}$-polyhedra).