Precise contractors for generic abstract domain

Damien Massé
joint work with Luc Jaulin

LabSTICC
Université de Bretagne Occidentale
Brest, France

SMART 2015, Manchester

Interval analysis

Interval analysis represents an efficient approach for reliable computation.

- Manipulated elements are boxes.
- Safe bounds are computed.
- With efficient libraries to manipulate non-linear expressions.

Abstract interpretation

Abstract interpretation is a formal framework (initially) designed for static program analysis.

- Manipulated elements are in (any) abstract domains.
- Main goal is the safe approximations of fixpoints.
- Efficient libraries on numerical abstract domains, mainly for *linear* expressions.

Relationships

Boxes are a well-known numerical abstract domain (Cousot & Cousot, 1976), but is considered to be **too imprecise**.

- More precision $\rightarrow$ use more precise domains.
- Poor handling of non-linear expressions: not the main issue.

Existing work to replace intervals by octagons on constraint programming [Pelleau, Truchet, Miné]:

- relies either on interval libraries (IBEX) with “rotated” constraints ($\rightarrow$ lost of precision);
- or on abstract interpretation-based libraries (Apron) ($\rightarrow$ problem in handling non-linear expressions).

Outline

1. **Abstract interpretation**
2. Contractors
3. Construction of optimal contractors
4. Implementation and tests

Closure operators

In interval analyses, sets of $\mathbb{R}^n$ are approximated by their *interval hulls*. The approximation operation is therefore the interval hull operator $\Box$, which is a *closure operator*:

Definition

A closure operator $\rho : \wp(\mathbb{R}^n) \to \wp(\mathbb{R}^n)$ is:

1. monotonic (if $X \subseteq Y$, then $\rho(X) \subseteq \rho(Y)$)
2. extensive ($\rho(X) \supseteq X$)
3. and idempotent ($\rho(\rho(X)) = \rho(X)$)

Extensivity ensures *overapproximation*. Monotonicity and idempotence guarantees some kind of *optimality*.

Moore families

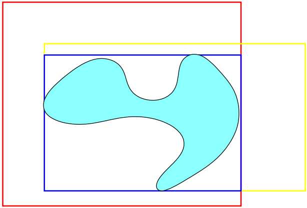
The image of a closure operator is closed by (infinite) intersection: it is a *Moore family*. Conversely, any Moore family induces a closure operator.

Theorem

Let $\mathbb{A}_n$ a Moore family. The operator ρ defined as:

$$\rho(X) = \cap\{S \in \mathbb{A}_n \mid S \supseteq X\}$$

is a closure operator, whose image is $\mathbb{A}_n$.

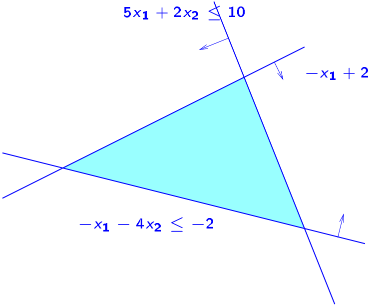


Closure operators/Moore families represent *classical abstractions*:

- each "concrete" subset S has a best abstraction;
- each abstract element $\rho(X)$ has a direct concrete value.

Generic example: linear constraints

On $\mathbb{R}^n$, linear constraints $\sum_i a_i x_i \leq c$ represents (closed) half-spaces.
Convex polyhedra are **finite** intersection of closed half-space.



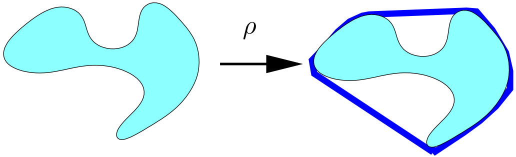
A polyhedron can be represented in matrix form, e.g.:

$$\begin{pmatrix} 5 & 2 \\ -1 & 2 \\ -1 & -4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 10 \\ 10 \\ -2 \end{pmatrix}$$

Domain closed by finite intersection, but **not** a Moore family. Hence no closure operator, no best abstraction.

Moore closure

Adding infinite intersection gives the set of **closed convex sets**.

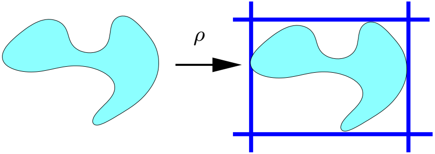


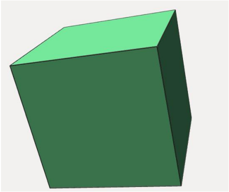
- Closure operator: closed convex hull.
- Not usable in practice: no memory representation, no manipulation.

Hence we consider subsets of polyhedra which could be Moore families.

Boxes

The set of boxes is a Moore family, where the constraints are restricted to Cartesian products of intervals.





$$\begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} \max x_1 \\ -\min x_1 \\ \max x_2 \\ -\min x_2 \end{pmatrix}$$

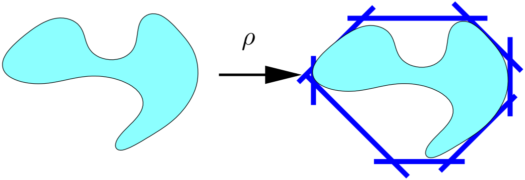
Boxes are **non-relational**: relationships between variables are forgotten.

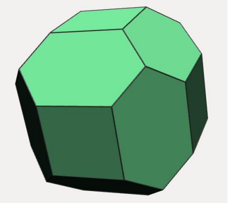
Advantages: operations (intersection, convex union, ...) are fast (mostly linear).

Drawback: non-relational operations are imprecise.

Octagons[Mine, 2001]

Octagons is the most well-known example of (weakly) relational domain, with relations of the form: $\pm x_i \pm x_j \leq c$.





$$\begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 1 & 1 \\ -1 & -1 \\ -1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} c_1 \\ c_{-1} \\ c_2 \\ c_{-2} \\ c_{1,2} \\ c_{-1,-2} \\ c_{-1,2} \\ c_{1,-2} \end{pmatrix}$$

Number of constraints for dim. n: $2n^2$.

Advantages: more precise than boxes.

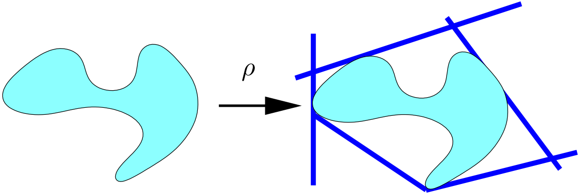
Drawbacks: maybe too many constraints. Slower than boxes.

Template polyhedral domain[Sankaranarayanan *et al.*,2005]

Domains with fixed constraint matrix are called **template polyhedral domains**.

Example with:

$$T = \begin{pmatrix} -1 & 0 \\ -1 & 3 \\ 4 & 3 \\ 1 & -4 \\ -2 & -3 \end{pmatrix}$$



Advantages: possible to customize the template.

Drawbacks: in general, linear programming must be used to compute convex union, intersection.

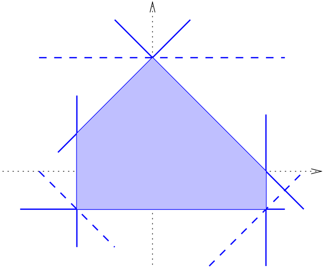
Boxes and octagons are, of course, particular cases.

Elements of polyhedral domains

In general, an abstract element has two canonical (constraint-based) representations:

1. A **minimal** representation where only useful constraints are kept.
2. A **closed** representation where all constraints are associated to their minimum (needed for emptiness checking, $\cup$).

$$
\left\{
\begin{array}{rrcl}
-x_1 & & \leq & 1 \\
 & -x_2 & \leq & 2 \\
x_1 & -x_2 & \leq & 3 \\
x_1 & +x_2 & \leq & 3 \\
x_1 & & \leq & 3 \\
\hline
-x_1 & -x_2 & \leq & 3 \\
 & x_2 & \leq & 3 \\
x_1 & -x_2 & \leq & 3
\end{array}
\right.
$$



Representation and operations

Operation	Requires closed form	Result in closed form
Emptiness test	yes	
Inclusion	partial	
Union	**yes**	yes
Intersection	no	**no**
Other operations	often	sometimes

Computing the closure of an abstract element may be costly (cubic algorithms for octagons, using LP in general). Thus it must be computed only when needed.

Outline

1. Abstract interpretation
2. Contractors
3. Construction of optimal contractors
4. Implementation and tests

Contractors

A **contractor**[Chabert & Jaulin] $\mathcal{C}$ on $\mathbb{A}_n$ is:

- *monotonic*
- and *reductive*.

To link contractors with set membership:

1. When $\mathbb{A}_n$ contains all the singletons:

$$\mathrm{set}(\mathcal{C}) = \{a \in \Sigma \mid C(\{a\}) = \{a\}\}$$

2. In the general case, set consistency $S \sim \mathcal{C}$ is:

$$S \sim \mathcal{C} \iff \forall X \in \mathbb{A}_n, S \cap X = S \cap \mathcal{C}(X)$$

Obviously, if $\mathrm{set}(\mathcal{C})$ is defined:

$$S \sim \mathcal{C} \iff S \subseteq \mathrm{set}(\mathcal{C})$$

Optimal contractors

Let's pose $I_S : X \mapsto S \cap X$.

Theorem

Let ρ a closure operator, $\mathbb{A}_n$ its image, and $\mathcal{C}$ a contractor on $\mathbb{A}_n$. then

$$S \sim \mathcal{C} \iff \forall X \in \mathbb{A}_n, \mathcal{C}(X) \supseteq \rho(S \cap X)$$

Furthermore, on $\mathbb{A}_n$

$$\rho \circ I_S = \rho \circ I_S \circ \rho$$

is a contractor, the minimal contractor consistent with S.

Best abstract transformer

Definition

Let ρ a closure operator on $\wp(\mathbb{R}^n)$, and ϕ a monotonic operator on $\wp(\mathbb{R}^n)$. Then $\phi^\star = \rho \circ \phi \circ \rho$ is the *best abstraction* of ϕ by ρ.

From now, we note, for all $S \subseteq \mathbb{R}^n$:

$$\mathcal{C}_S^\star = \rho \circ I_S \circ \rho$$

In abstract interpretation-based static analysis, this operation is used for tests.

Outline

Algebra of contractors

Operations on contractors:

1. Union: $(\mathcal{C}_1 \cup \mathcal{C}_2)(\mathbf{x}) = \rho(\mathcal{C}_1(\rho(\mathbf{x})) \cup \mathcal{C}_2(\rho(\mathbf{x})))$.
2. Intersection: $(\mathcal{C}_1 \cap \mathcal{C}_2)(\rho(\mathbf{x})) = \mathcal{C}_1(\rho(\mathbf{x})) \cap \mathcal{C}_2(\rho(\mathbf{x}))$
3. Composition: $(\mathcal{C}_1 \circ \mathcal{C}_2)(\rho(\mathbf{x})) = \mathcal{C}_1(\mathcal{C}_2(\rho(\mathbf{x})))$

If $S_1 \sim \mathcal{C}_{S_1}$ and $S_2 \sim \mathcal{C}_{S_2}$:

$$\begin{aligned} S_1 \cup S_2 &\sim \mathcal{C}_{S_1} \cup \mathcal{C}_{S_2} \\ S_1 \cap S_2 &\sim \mathcal{C}_{S_1} \cap \mathcal{C}_{S_2} \\ S_1 \cap S_2 &\sim \mathcal{C}_{S_1} \circ \mathcal{C}_{S_2} \end{aligned}$$

However, these operations may not be optimal...

Optimal abstraction

Let ϕ_1 and ϕ_2 two functions on $\wp(\mathbb{R}^n)$:

$$\begin{aligned}
\phi_1^\star \circ \phi_2^\star &= \rho \circ \phi_1 \circ \rho \circ \rho \circ \phi_2 \circ \rho \\
&= \rho \circ \phi_1 \circ \rho \circ \phi_2 \circ \rho \\
&\supseteq \rho \circ \phi_1 \circ \phi_2 \circ \rho \\
&= (\phi_1 \circ \phi_2)^\star
\end{aligned}$$

How to ensure $\phi_1^\star \circ \phi_2^\star = (\phi_1 \circ \phi_2)^\star$?

Completeness

Definition

1. ϕ is **backward-complete** w.r.t. ρ if:

$$\rho \circ \phi \circ \rho = \rho \circ \phi$$

2. ϕ is **forward-complete** w.r.t. ρ if:

$$\rho \circ \phi \circ \rho = \phi \circ \rho$$

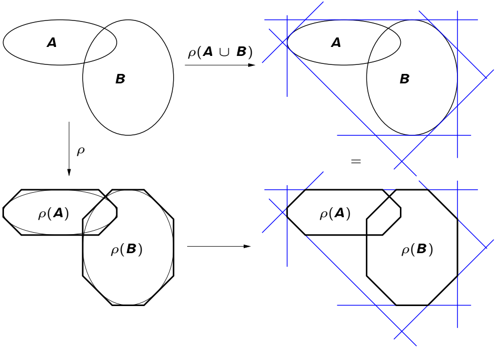
The property $\phi_1^\star \circ \phi_2^\star = (\phi_1 \circ \phi_2)^\star$ holds if:

1. ϕ_1 is forward-complete w.r.t. ρ;
2. **or** ϕ_2 is backward-complete w.r.t. ρ.

Completeness of union

Good example of backward-complete operator:

$$\rho(A \cup B) = \rho(\rho(A) \cup \rho(B))$$



As a result:

$$\phi^\star_{S_1 \cup S_2} = \rho(\phi^\star_{S_1} \cup \phi^\star_{S_2}) = \phi^\star_{S_1} \cup^\star \phi^\star_{S_2}$$

Forward completeness

A function ϕ is forward-complete w.r.t. ρ iff for all $A \in \mathbb{A}_n$, $\phi(A) \in \mathbb{A}_n$. If $\phi = I_S$, since $\mathbb{R}^n \in \mathbb{A}_n$:

$$S \cap \mathbb{R}^n = S \in \mathbb{A}_n$$

Theorem

The function $I_S : \mathbf{x} \mapsto \mathbf{x} \cap S$ is forward-complete w.r.t. ρ iff $S \in \mathbb{A}_n$.

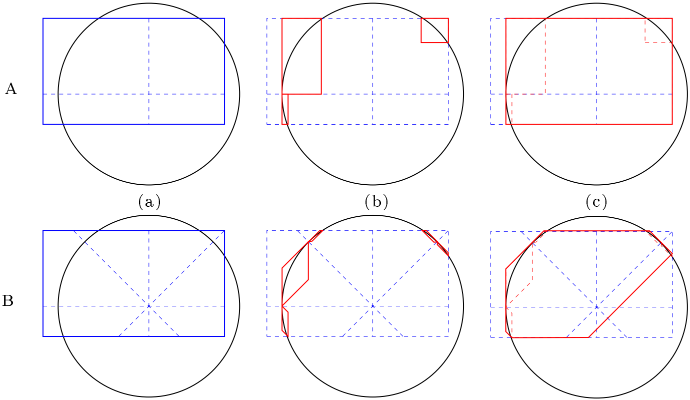
As a result, all intersections with an element of $\mathbb{A}_n$ can be applied once at the beginning of the computation:

$$S_2 \in \mathbb{A}_n \implies \phi^\star_{S_1 \cap S_2} = \phi^\star_{S_1} \circ \phi^\star_{S_2}$$

Application: covering the set

Assuming the construction of minimal contractors is easier for small sets:

1. We decompose the set S into a covering set of sets (S_i) where each $S_i = S \cap X_i$ with $X_i \in \mathbb{A}_n$.
2. Each computation of $C^{\star}_{S_i}(X)$ starts by an intersection with X_i.
3. The union of the result is optimal.



Set transformation

Let $f : \mathbb{R}^n \to \mathbb{R}^n$.
We define:

$$f^\star = \rho \circ f \circ \rho$$

and

$$(f^{-1})^\star = \rho \circ f^{-1} \circ \rho$$

Then if $S \sim \mathcal{C}_S$:

$$f(S) \sim f^\star \circ \mathcal{C}_S \circ (f^{-1})^\star \boxed{\cap \rho}$$

Note that $f^\star \circ \mathcal{C}_S \circ (f^{-1})^\star$ may not be a contractor (not reductive on $\mathbb{A}_n$).

Set transformation (2)

When is $f^\star \circ \mathcal{C}_S \circ (f^{-1})^\star$ a contractor?
Sufficient condition: when $f^\star \circ (f^{-1})^\star = \rho \circ f \circ \rho \circ f^{-1} \circ \rho$ is a contractor.
Since $f \circ f^{-1}$ is reductive:

1. when f is backward-complete;
2. *or* when f^{-1} is forward-complete.

Theorem

Both conditions are equivalent to

$$f^{-1}(X) \in \mathbb{A}_n \text{ for all } X \in \mathbb{A}_n.$$

And in this case, the transformation is optimal:

$$\mathcal{C}^\star_{f(S)} = f^\star \circ \mathcal{C}^\star_S \circ (f^{-1})^\star$$

Set transformation (3)

Furthermore, if f is bijective and $f(X) \in \mathbb{A}_n$ for all X:

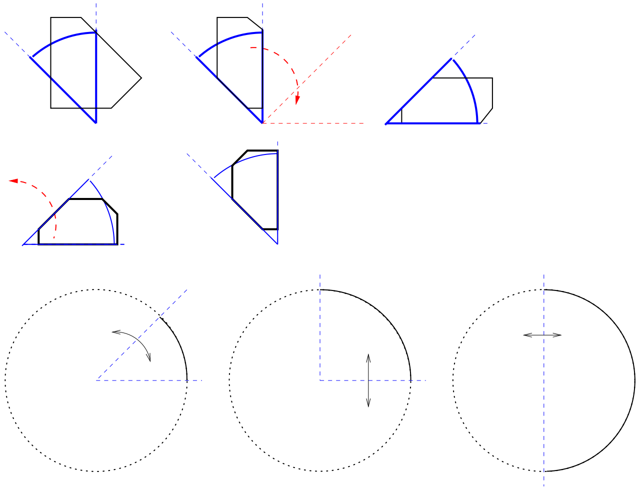
$$\mathcal{C}^\star_{f(S)} = f \circ \mathcal{C}^\star_S \circ f^{-1}$$

Applications:

1. for all template polyhedral domains, translations and positive homotheties;
2. for octagons, any permutation and “negations” of dimensions;
3. for boxes, directional scalings.

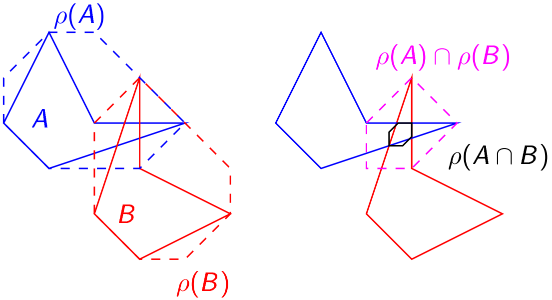
Application: optimal contractor for a disc

We can construct the optimal contractor for a disc from the contractor for $1/8^{\text{th}}$ of the disc.



Intersections

Intersection of contractors is known to be non-optimal. Composition is better but still not optimal. Repeated composition (*local iterations*) may be better but (still) not optimal.



Heuristics: push intersections at the lowest level and the union at the higher lever (use De Morgan's laws).

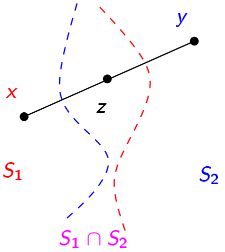
Intersections: specific result

For octagons (or other linear-based convex domains):

Theorem

If two sets S_1 and S_2 are such that for all $x \in S_1$ and $y \in S_2$, $[x, y] \cap (S_1 \cap S_2) \neq \emptyset$, then

$$C^\star_{S_1} \cap C^\star_{S_2} = C^\star_{S_1 \cap S_2}$$



for all V, we cannot have $Vx > Vz$ **and** $Vy > Vz$

Application: $S = f^{-1}([a, b])$, f is continuous, and $S_1 = f^{-1}(]-\infty, b])$, $S_2 = f^{-1}([a, +\infty])$.

Outline

1. Abstract interpretation
2. Contractors
3. Construction of optimal contractors
4. **Implementation and tests**

Bisection and size

To implement paving algorithm, we need a splitting (bisection) and a size operator:

1. The bisection operator $\mathrm{bisect} : \mathbb{A}_n \to \mathbb{A}_n \times \mathbb{A}_n$ must satisfy:

$$\forall \mathbf{x} \in \mathbb{A}_n, (\mathbf{x}_1, \mathbf{x}_2) = \mathrm{bisect}(\mathbf{x}) \to \mathbf{x}_1 \cup \mathbf{x}_2 = \mathbf{x}$$

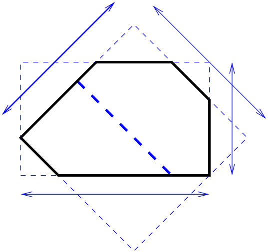
2. The size operator $\mathrm{size} : \mathbb{A}_n \to \mathbb{R}^+ \cup \{-\infty, +\infty\}$ must satisfy:
   - $\mathrm{size}(\mathbf{x}) \leq 0$ iff $\mathbf{x}$ is empty of a singleton
   - size is monotonic.
   - if X is bounded, then $\mathrm{size}(X)$ is finite.

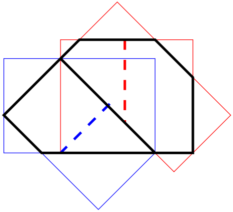
Furthermore, the termination of the algorithm can be ensured by the existence of $\varepsilon \in [0, 1[$ such that:

$$(\mathbf{x}_1, \mathbf{x}_2) = \mathrm{bisect}(\mathbf{x}) \;\Rightarrow\; \mathrm{size}(\mathbf{x}_i) < \varepsilon.\mathrm{size}(\mathbf{x})$$

Bisection operation

We consider a template polyhedral domain with “interval” templates (constraints of the form $m_j \leq A_j.x \leq M_j$).
Empirically[Pelleau, Truchet], an efficient operator (at least for octagons) “cuts” the element along the maximum dimension of the “smallest” (w.r.t. max dimension) enclosing (rotated) box.

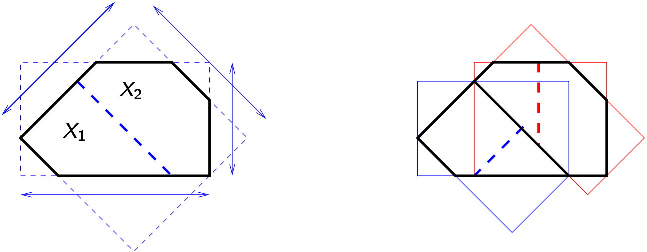




Size of abstract elements

The maximum dimension of the smallest enclosing box is a good candidate, but it may not satisfy the simple termination condition (e.g. with a square). Other choice: sum of all dimensions of all enclosing boxes. Very coarse bound (dimension n):

$$\mathrm{size}(X_i) \leq \frac{1}{2\sqrt{2}n^2}\mathrm{size}(X)$$



$\mathrm{size}(X_1) \simeq 0.75\mathrm{size}(X)$
$\mathrm{size}(X_2) \simeq 0.81\mathrm{size}(X)$

Implementation of octagons

Implementation of octagons:

- Closure not optimised, but (assumed to be) correct.
- Only continuous variables.
- Possibility of adding variables or removing variables (projection operators).
- Indifferent use of x_i and $-x_i$ (for contractors operators)
- Optimal contractors for:
  - sum: $x_i + x_j + x_k \leq 0$
  - directional scaling: $ax_i + x_j \leq 0$
  - square operator: $x_i \leq kx_j^2$ (of $\geq$)
  - bounded distance: $\sqrt{x_i^2 + x_j^2} \leq k$ (or $\geq$)
  - sinus operator: $\sin(x_i) \leq x_j$ (or $\geq$)

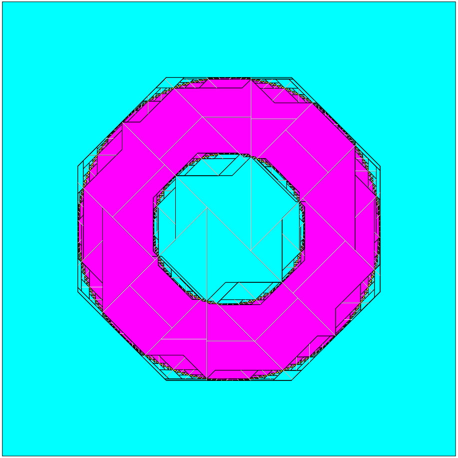
Paver

We use **separators**[Jaulin & Desrochers] for our algorithm: given a set S, our separator $\mathcal{S}$ combines a contractor for S and a contractor for $\overline{S}$.
Algorithm:

1. contract with respect to S, then to $\overline{S}$;
2. stop if abstract element is too small;
3. otherwise, bisect the abstract element and recursively execute the algorithm on the resulting elements.

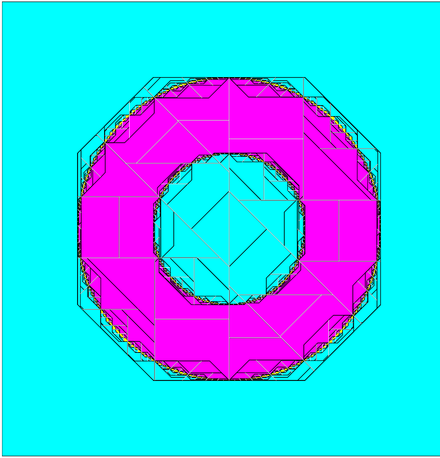
First example: ring

Initial box: $[-1.5, 1.5] \times [-1.5, 1.5]$. Representation of $0.5 \leq \sqrt{x^2 + y^2} \leq 1$. Stop when $\mathrm{size} < 0.03$.



Optimal contractor

2073 octagons



Using $\frac{1}{4} \leq x^2 + y^2 \leq 1$

2109 octagons

Why $\frac{1}{4} \leq x^2 + y^2 \leq 1$ not optimal

Let's consider $x \in [0.6, 0.8]$, $y \in [0.6, 0.8]$ and $x^2 + y^2 \leq 1$.

- Optimal result: no change except $\boxed{x + y \leq \sqrt{2}}$.
- With expression computation:
  1. Consider $x2 = x^2$. Result: $x2 \in [0.36, 0.64]$ and $x - x2 \in [0.16, 0.24]$ (optimal contraction). Same result with $y2 = y^2$.
  2. Closure with $x2 + y2 \leq 1$:
     - ⋆ $\Rightarrow x + y2 \leq 1.24$
     - ⋆ $\Rightarrow \boxed{x + y \leq 1.48}$

     Successive iterations do not change the result.

Not optimal and slower, but at least there is something, **but...**

Just an homothety

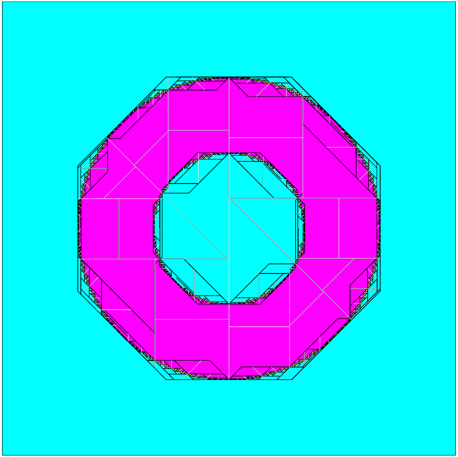
Let's consider $x \in [6,8]$, $y \in [6,8]$ and $x^2 + y^2 \leq 100$.

- Optimal result: $\boxed{x + y \leq 10\sqrt{2}}$ **ok**
- With expression computation:
  1. Consider $x2 = x^2$. Result: $x2 \in [36,64]$ but $x - x2 \in [-56,-30]$ (optimal contraction). Same result with $y2 = y^2$.
  2. Closure with $x2 + y2 \leq 100$:
     - ⋆ $\Rightarrow x + y2 \leq 70$ **redundant!**
     - ⋆ $\Rightarrow \boxed{x + y \leq 40}$ **redundant!**

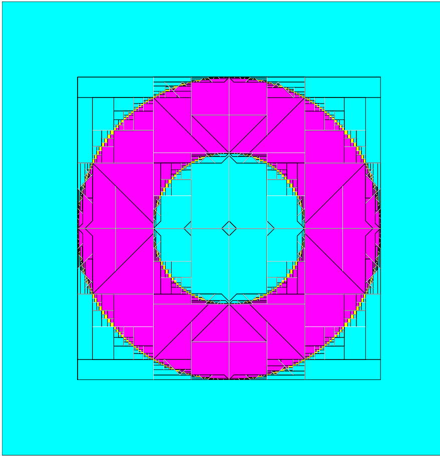
  No improvement $\rightarrow$ no successive iterations.

Octagons with intermediate ("polynomial") variables very sensible to scaling.

And so...



Optimal contractor
no change

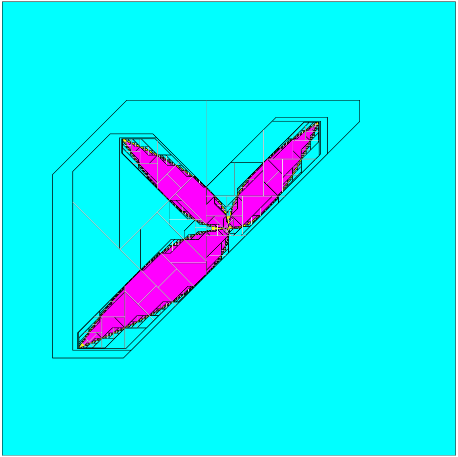


Using $25 \leq x^2 + y^2 \leq 100$
2457 octagons (+350)

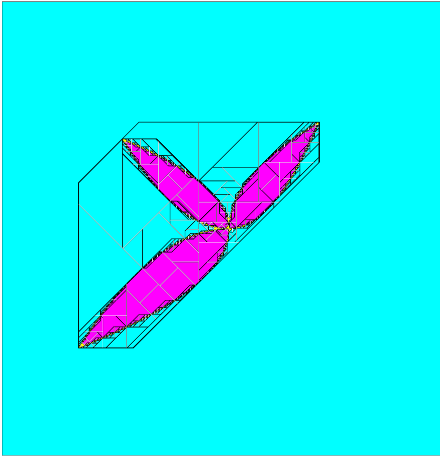
Boxes are predominant in the second case: the benefit of octagons is lost.

Intersection of parabols

This example illustrates, with the intersection of parabols,
the non-completeness of intersection (and how local iterations can reduce it).



Without local iterations



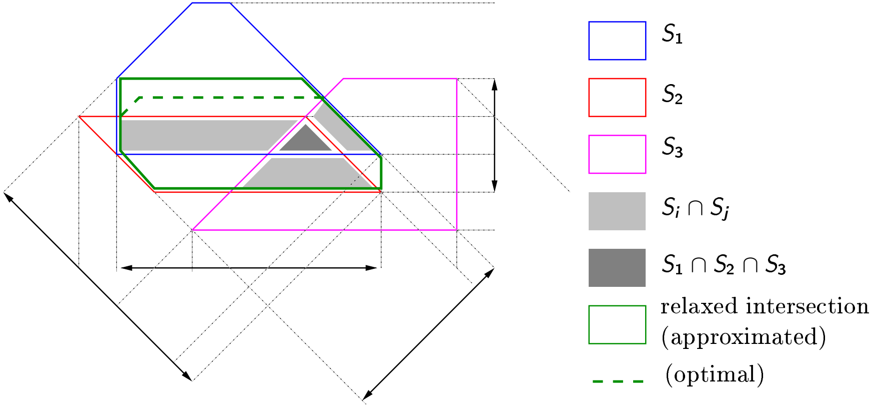
With local iterations

However, the number of octagons is the same for both representations.

Relaxed intersection

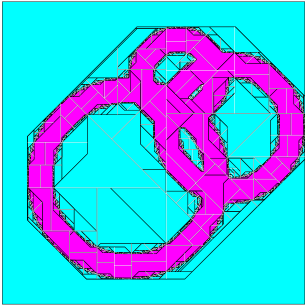
Is $S_1, \dots, S_n$ are subsets of S, the **k-relaxed intersection** of (S_i) is the set of points in (at least) $n - k$ sets S_i.

Optimal contractor hard to compute, but using projection enables to compute a fast and sound approximation.

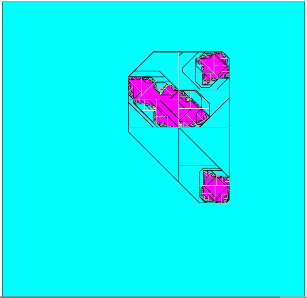


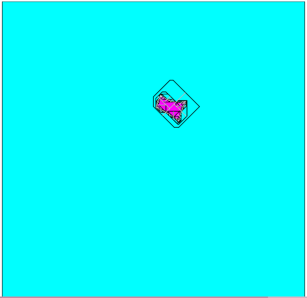
Iterating the algorithm gives better results.

Relaxed intersection: example



$c_1 = [1, 3]$, $r_1 = [1, 2]$
$c_2 = [3, 1]$, $r_2 = [2, 3]$
$c_3 = [-1, -1]$, $r_2 = [3, 4]$
Optimal for $k = 2$.





Discussion and conclusion

Interest to adapt the domain to the (local) form of the set.

1. Octagons more flexible than boxes.
2. But useful only if the contractors are precise enough.
3. Two many available linear forms?
4. Using specialised linear templates? How? At which cost?
5. Non-linear templates? Which one? Which operations?

Extend to higher dimensions?

1. More costly.
2. Splitting less usable.
3. Optimal contractors hard to design (e.g. $x.y = z$ or $x^2 + y^2 = z^2$). Still need to work on arithmetic expressions.

Thank you