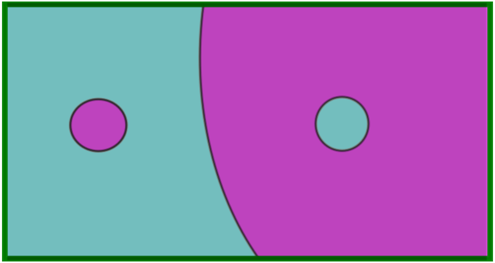
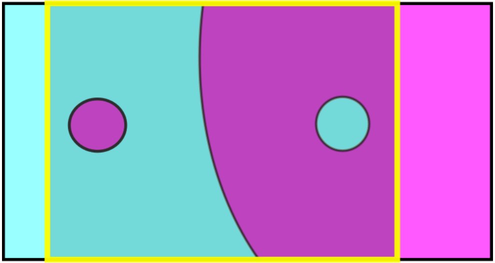
A boundary approach for set inversion

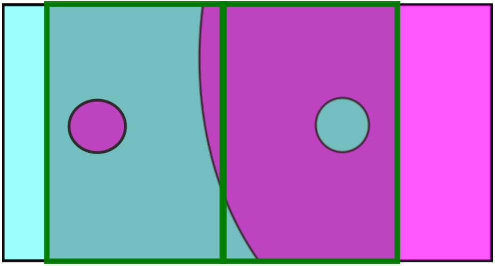
L. Jaulin

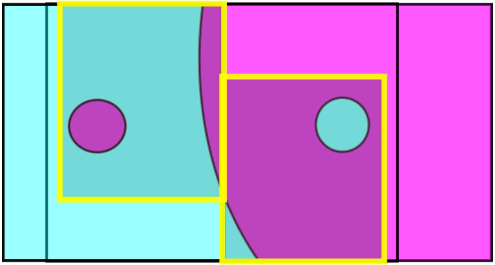
Virtual
2020, December 19

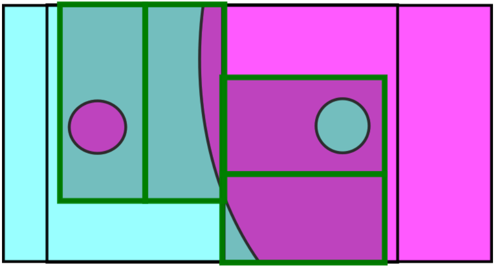
Motivation

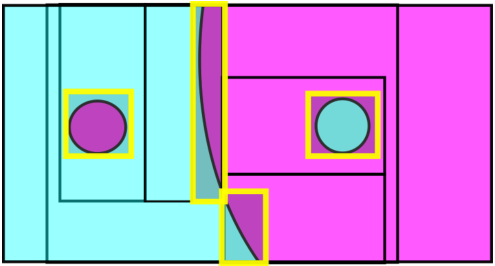


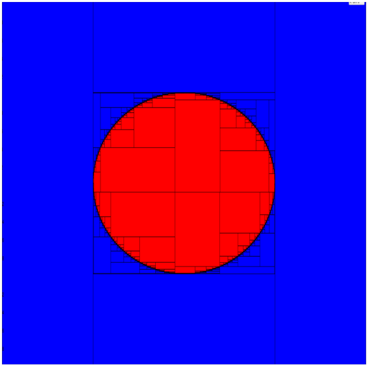
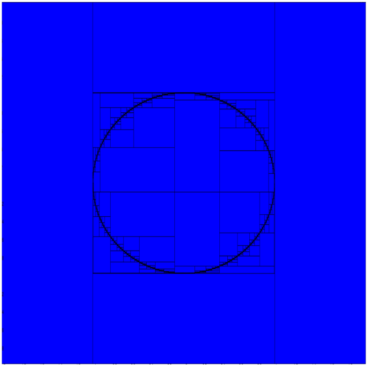




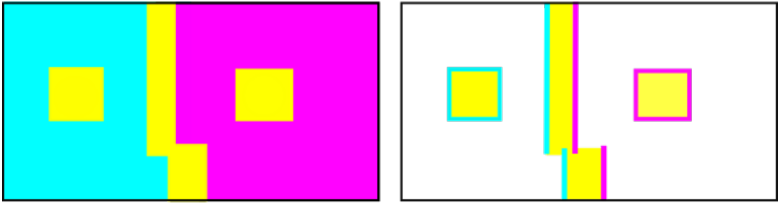






- A lot of computation are performed twice with classical approaches.
- With the boundary approach : keep the color.



What we usually compute. What we propose to compute[3]

Idea: A box is represented with colored faces

$$[\mathbf{x}] = [1, 2] \times [1, 4].$$

Forward-backward sequence

$$\mathbf{f}(\mathbf{x}) \in \mathbb{Y}, \mathbf{x} \in \mathbb{X}(0)$$
$$\mathbf{f} = \mathbf{f}_n \circ \cdots \circ \mathbf{f}_2 \circ \mathbf{f}_1$$
$$\mathbb{X} = \mathbb{X}(0) \cap \mathbf{f}^{-1}(\mathbb{Y})$$

Input: $\mathbb{X}(0)$

For $k = 1$ to n

$\quad \overrightarrow{\mathbb{X}}(k) = \mathbf{f}_k(\overrightarrow{\mathbb{X}}(k-1))$

$\overleftarrow{\mathbb{X}}(n) = \mathbb{Y} \cap \overrightarrow{\mathbb{X}}(n)$

For $k = n$ to 1

$\quad \overleftarrow{\mathbb{X}}(k-1) = \overrightarrow{\mathbb{X}}(k-1) \cap \mathbf{f}_k^{-1}(\overleftarrow{\mathbb{X}}(k))$

Return $\overleftarrow{\mathbb{X}}(0)$

$\overrightarrow{\mathbb{X}}(k)$: set of states at time k consistent with the past

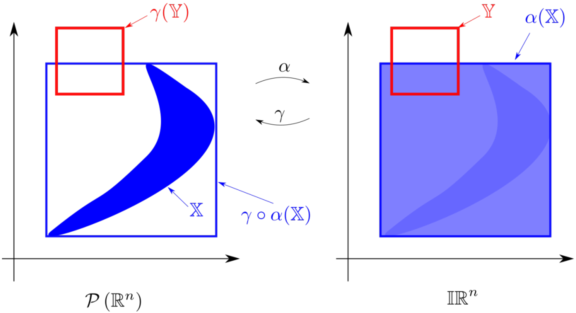
$\overleftarrow{\mathbb{X}}(k)$: set of states consistent with both past and future

Let $(\mathbb{A}, \leq)$ and $(\mathbb{B}, \leq)$ be two partially ordered sets [1].
A Galois connection consists of two monotonic functions:
$\alpha : \mathbb{A} \to \mathbb{B}$ and $\gamma : \mathbb{B} \to \mathbb{A}$, such that

$$\alpha(x) \leq y \Leftrightarrow x \leq \gamma(y).$$

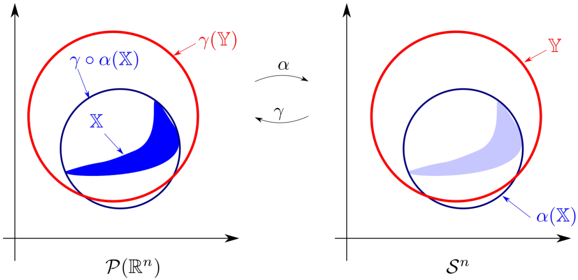
With boxes, we have

$$\alpha(\mathbb{X}) \subset \mathbb{Y} \Leftrightarrow \mathbb{X} \subset \gamma(\mathbb{Y})$$

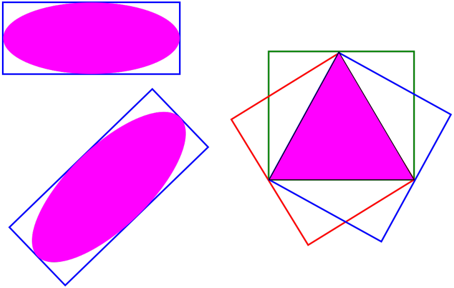


With sphere, we have

$$\alpha(\mathbb{X}) \subset \mathbb{Y} \Rightarrow \mathbb{X} \subset \gamma(\mathbb{Y})$$



Oriented boxes do not yield a Galois connection



This excludes Lohner-type algorithms

Directed contractors

A *directed contractor* $\mathscr{C}$ for the constraint $\mathbf{y}=\mathbf{f}(\mathbf{x})$ is a pair of two operators

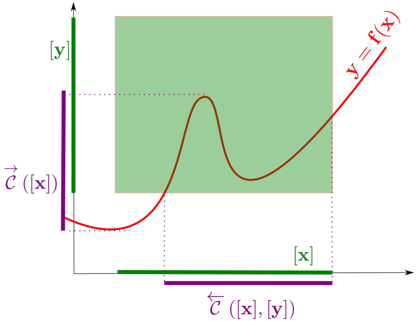
$$\mathscr{C}:([\mathbf{x}],[\mathbf{y}])\rightarrow\left(\overrightarrow{\mathscr{C}}([\mathbf{x}]),\overleftarrow{\mathscr{C}}([\mathbf{x}],[\mathbf{y}])\right)$$

with

$$\begin{array}{rcl}\mathbf{f}([\mathbf{x}]) & \subset & \overrightarrow{\mathscr{C}}([\mathbf{x}])\\ \mathbf{f}^{-1}([\mathbf{y}])\cap[\mathbf{x}] & \subset & \overleftarrow{\mathscr{C}}([\mathbf{x}],[\mathbf{y}])\subset[\mathbf{x}]\end{array}$$

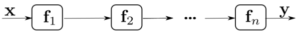
Moreover

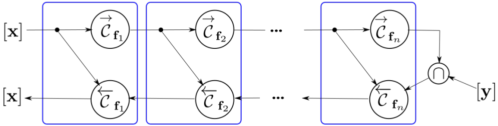
$$\left\{\begin{array}{l}[\mathbf{a}]\subset[\mathbf{x}]\\ [\mathbf{b}]\subset[\mathbf{y}]\end{array}\right.\Rightarrow\left\{\begin{array}{r}\overrightarrow{\mathscr{C}}([\mathbf{a}])\subset\overrightarrow{\mathscr{C}}([\mathbf{x}])\\ \overleftarrow{\mathscr{C}}([\mathbf{a}],[\mathbf{b}])\subset\overleftarrow{\mathscr{C}}([\mathbf{x}],[\mathbf{y}])\end{array}\right.$$


[y]
$\overrightarrow{\mathcal{C}}([\mathbf{x}])$
$\mathbf{y} = \mathbf{f}(\mathbf{x})$
[x]
$\overleftarrow{\mathcal{C}}([\mathbf{x}],[\mathbf{y}])$

Contractor chain [2]

$$\mathbf{f}(\mathbf{x}) \in [\mathbf{y}], \mathbf{x} \in \mathbb{X}(0)$$
$$\mathbf{f} = \mathbf{f}_n \circ \cdots \circ \mathbf{f}_2 \circ \mathbf{f}_1$$
$$\mathbb{X} = \mathbb{X}(0) \cap \mathbf{f}^{-1}([\mathbf{y}])$$





Composition

$$\mathscr{C}_2 \circ \mathscr{C}_1 \begin{pmatrix} [\mathbf{x}] \\ [\mathbf{y}] \end{pmatrix} = \left(\overrightarrow{\mathscr{C}}_2 \circ \overrightarrow{\mathscr{C}}_1([\mathbf{x}]), \overleftarrow{\mathscr{C}}_1 \begin{pmatrix} [\mathbf{x}] \\ \overleftarrow{\mathscr{C}}_2 \begin{pmatrix} \overrightarrow{\mathscr{C}}_1([\mathbf{x}]) \\ \overrightarrow{\mathscr{C}}_2 \circ \overrightarrow{\mathscr{C}}_1([\mathbf{x}]) \cap [\mathbf{y}] \end{pmatrix} \end{pmatrix} \right)$$

The minimal directed contractor for the constraint $y = f(\mathbf{x}) = x_1 + x_2$ is:

$$\begin{array}{ccc} \overrightarrow{\mathscr{C}}([\mathbf{x}]) & = & [x_1] + [x_2] \\ \overleftarrow{\mathscr{C}}([\mathbf{x}],[y]) & = & \begin{pmatrix} [x_1] \cap ([y] - [x_2]) \\ [x_2] \cap ([y] - [x_1]) \end{pmatrix} \end{array}$$

A function $\mathbf{f}$ for which a minimal directed contractor is available is said to be *contractible*.

Since the minimal contractor for

$$\mathbf{y} = \mathbf{A} \cdot \mathbf{x}$$

is

$$\mathscr{C}\begin{pmatrix} [\mathbf{x}] \\ [\mathbf{y}] \end{pmatrix} = \begin{pmatrix} \overrightarrow{\mathscr{C}}([\mathbf{x}]) \\ \overleftarrow{\mathscr{C}}([\mathbf{x}],[\mathbf{y}]) \end{pmatrix} = \begin{pmatrix} \mathbf{A} \cdot [\mathbf{x}] \\ \mathbf{A}^{-1} \cdot [\mathbf{y}] \cap [\mathbf{x}] \end{pmatrix}$$

the function $\mathbf{f}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$ is contractible.

Contractible decomposition

A *contractible decomposition* of a function $\mathbf{f}$ has the form

$$\mathbf{f} = \mathbf{f}_n \circ \cdots \circ \mathbf{f}_2 \circ \mathbf{f}_1 = \mathbf{f}_{1:n}$$

where each $\mathbf{f}_i$ is contractible.
Counterexample. The Fresnel integral

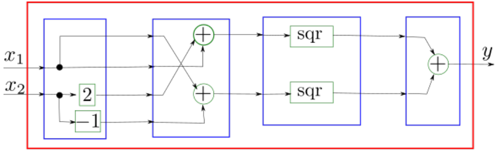
$$f(x) = \int_0^x \sin \tau^2 \cdot d\tau$$

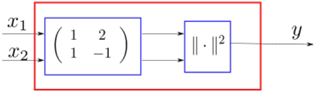
has no contractible decomposition.

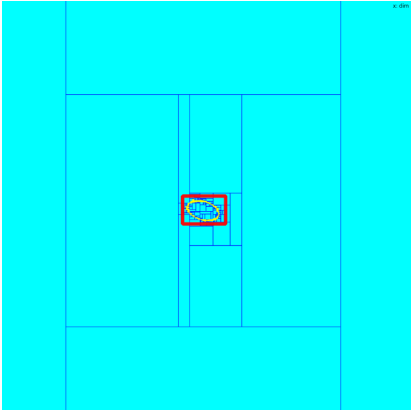
The function

$$f(x_1, x_2) = (x_1 + 2x_2)^2 + (x_1 - x_2)^2$$

has scalar and vector decompositions:





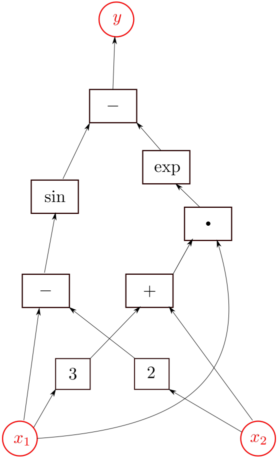


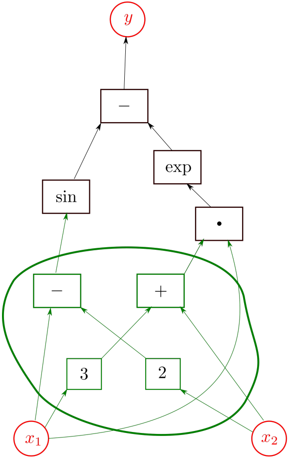
$$(x_1 + 2x_2)^2 + (x_1 - x_2)^2 = 1$$

Forward and cut algorithm

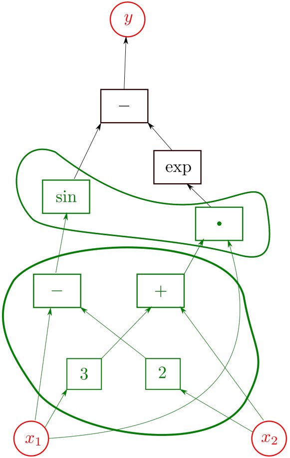
We want the contractible decomposition of

$$y = \sin(x_1 - 2x_2) - \exp(x_1 \cdot (3x_1 + x_2))$$


y
−
exp
sin
·
−
+
3
2
x1
x2

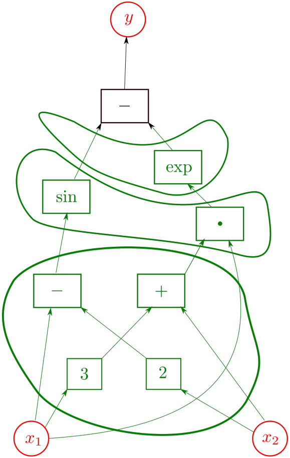


$$\begin{pmatrix} 1 & -2 \\ 1 & 0 \\ 3 & 1 \end{pmatrix}$$



$$\begin{pmatrix} \sin \\ \cdot \end{pmatrix}$$

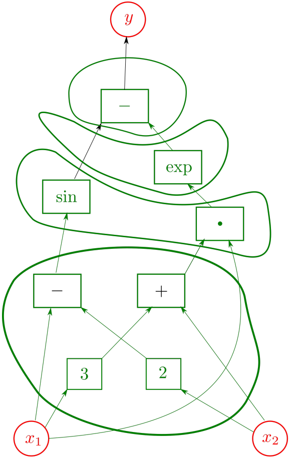
$$\begin{pmatrix} 1 & -2 \\ 1 & 0 \\ 3 & 1 \end{pmatrix}$$



$$\begin{pmatrix} 1 \\ \exp \end{pmatrix}$$

$$\begin{pmatrix} \sin \\ \cdot \end{pmatrix}$$

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



$$\begin{pmatrix} 1 & -1 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ \exp \end{pmatrix}$$

$$\begin{pmatrix} \sin \\ \cdot \end{pmatrix}$$

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

$$\begin{pmatrix} 1 & -2 \\ 1 & 0 \\ 3 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} \sin \\ . \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ \exp \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -1 \end{pmatrix}$$

For

$$\mathbf{f}_n \circ \cdots \circ \mathbf{f}_2 \circ \mathbf{f}_1(\mathbf{x}) \in \mathbb{Y}, \mathbf{x} \in [\mathbf{x}](0)$$

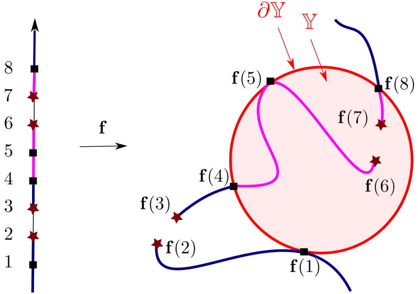
the algorithm returns a box $[\mathbf{a}](0) \supset \{\mathbf{x} \in [\mathbf{x}](0) \,|\, \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\}$.

Input: $[\mathbf{x}](0)$
For $k = 1$ to n
$\quad [\mathbf{x}](k) = \overrightarrow{\mathscr{C}}_k([\mathbf{x}](k-1))$
$[\mathbf{a}](n) = [\mathbb{Y} \cap [\mathbf{x}](n)]$
For $k = n$ to 1
$\quad [\mathbf{a}](k-1) = \overleftarrow{\mathscr{C}}_k([\mathbf{x}](k-1), [\mathbf{a}](k))$
Return $[\mathbf{a}](0)$

Boundary approach

Consider a continuous function $\mathbf{f}:\mathbb{R}^n \to \mathbb{R}^p$ defined everywhere. If $\mathbb{X}=\mathbf{f}^{-1}(\mathbb{Y})$, we have

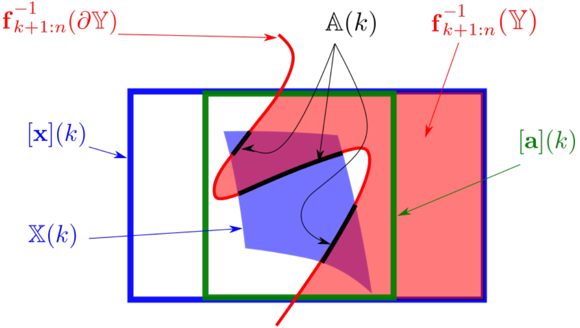
$$\partial\mathbb{X} \subset \mathbf{f}^{-1}(\partial\mathbb{Y}).$$

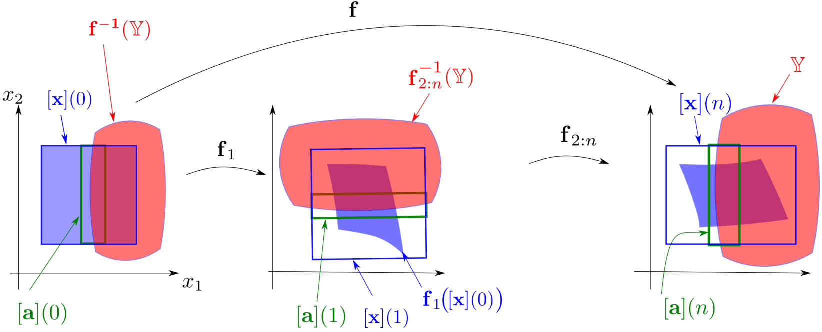


$$\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y}) = \{1\} \cup [4,6] \cup [7,8]$$
$$Dom(\mathbf{f}) =]-\infty,2] \cup [3,6] \cup [7,\infty]$$
$$\partial\mathbb{X} = \{1,4,\mathbf{6},\mathbf{7},8\}, \mathbf{f}^{-1}(\partial\mathbb{Y}) = \{1,4,\mathbf{5},8\}$$

Input: $\mathbb{X}(0)$
For $k = 1$ to n
    $\mathbb{X}(k) = \mathbf{f}_k(\mathbb{X}(k-1))$
$\mathbb{A}(n) = \partial\mathbb{Y} \cap \mathbb{X}(n)$
For $k = n$ to 1
    $\mathbb{A}(k-1) = \mathbb{X}(k-1) \cap \mathbf{f}_k^{-1}(\mathbb{A}(k))$
Return $\mathbb{A}(k-1)$

Boundary forward-backward sequence


$\mathbf{f}_{k+1:n}^{-1}(\partial\mathbb{Y})$
$\mathbb{A}(k)$
$\mathbf{f}_{k+1:n}^{-1}(\mathbb{Y})$
$[\mathbf{x}](k)$
$[\mathbf{a}](k)$
$\mathbb{X}(k)$


$\mathbf{f}$
$\mathbf{f}^{-1}(\mathbb{Y})$
x_2
$[\mathbf{x}](0)$
x_1
$[\mathbf{a}](0)$
$\mathbf{f}_1$
$\mathbf{f}_{2:n}^{-1}(\mathbb{Y})$
$[\mathbf{a}](1)$
$[\mathbf{x}](1)$
$\mathbf{f}_1([\mathbf{x}](0))$
$\mathbf{f}_{2:n}$
$\mathbb{Y}$
$[\mathbf{x}](n)$
$[\mathbf{a}](n)$

Cardinal directions

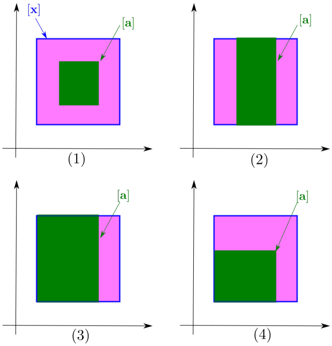
For $\lambda = (2, -)$ and

$$[\mathbf{x}] = [1,2] \times [3,4] \times [5,6].$$

We have $x^\lambda = 3$, the associated face is $[1,2] \times [3,3] \times [5,6]$ and $\mathscr{H}_\lambda([\mathbf{x}]) = \{\mathbf{x} | x_2 < 3\}$.

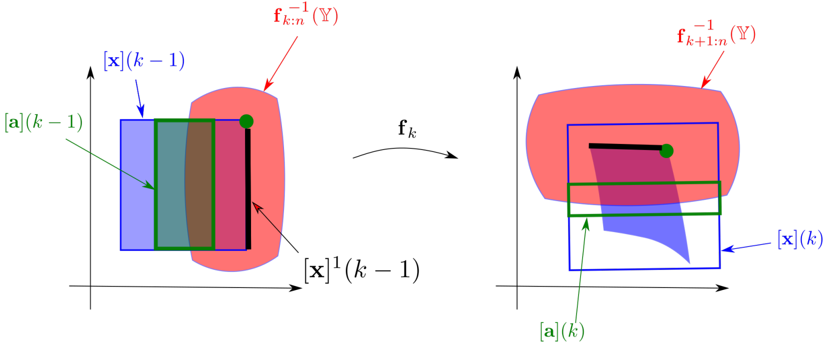
Win boxes. Take $[\mathbf{a}] \subset [\mathbf{x}]$ and $\lambda \in \mathscr{D}$, the λth win box, is

$$[\mathbf{x}] \backslash [\mathbf{a}]|_\lambda = [\mathbf{x}] \cap \mathscr{H}_\lambda([\mathbf{a}])$$



Given the pair $[\mathbf{x}](k), [\mathbf{a}](k)$ in the algorithm. We associate to each bound $a^\lambda(k)$ of $[\mathbf{a}](k)$, the quantity $c(a^\lambda(k)) \in \{0, 1, ?\}$ such that

$$\begin{array}{lcl} c(a^\lambda(k)) = 1 & \Rightarrow & \mathbf{f}_{k+1:n}([\mathbf{x}]^\lambda) \subset \mathbb{Y} \\ c(a^\lambda(k)) = 0 & \Rightarrow & \mathbf{f}_{k+1:n}([\mathbf{x}]^\lambda) \cap \mathbb{Y} = \emptyset \end{array}$$



Backward propagation of the bound colors

Sum. $f(\mathbf{x}) = x_1 + x_2$.
Forward step

$$[y] = [x_1] + [x_2]$$

Backward step, $\overleftarrow{\mathscr{C}}([\mathbf{x}],[y])$

$$[\mathbf{a}] = [\mathbf{x}] \cap \begin{pmatrix} [y]-[x_2] \\ [y]-[x_1] \end{pmatrix}$$

and

$$\begin{array}{l} \text{if } x_1^- < a_1^-, \text{ then } c(a_1^-) = c(y^-) \\ \text{if } a_1^+ < x_1^+, \text{ then } c(a_1^+) = c(y^+) \\ \text{if } x_2^- < a_2^-, \text{ then } c(a_2^-) = c(y^-) \\ \text{if } a_2^+ < x_2^+, \text{ then } c(a_2^+) = c(y^+) \end{array}$$

Square. $f(x) = x^2$

Forward step

$$[y] = [x]^2$$

Backward step

$$[a] = \left[\left\{x \in [x], x^2 \in [y]\right\}\right]$$

if $x^- < a^-$,

if $x^{-2} < y^-$, then $c(a^-) = c(y^-)$

else $c(a^-) = c(y^+)$

if $a^+ < x^+$,

if $x^{+2} < y^-$, then $c(a^+) = c(y^-)$

else $c(a^+) = c(y^+)$

Test-cases

Test-case 1. Consider the set inversion problem

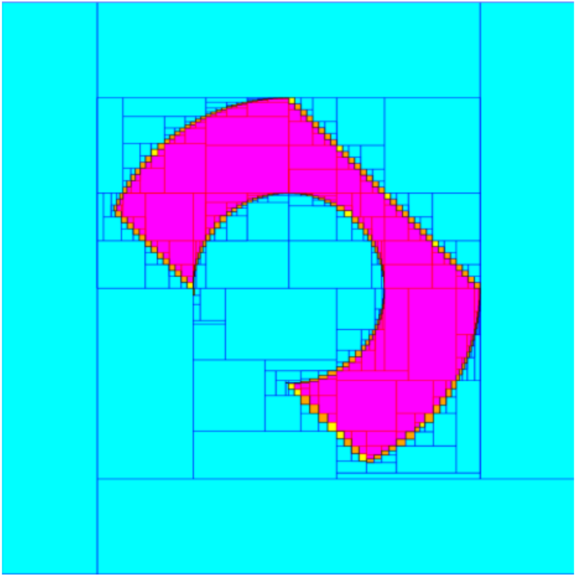
$$\mathbb{X} = \begin{pmatrix} [-3,3] \\ [-3,3] \end{pmatrix} \cap \mathbf{f}^{-1} \begin{pmatrix} [-1,2] \\ [1,4] \end{pmatrix}$$

where

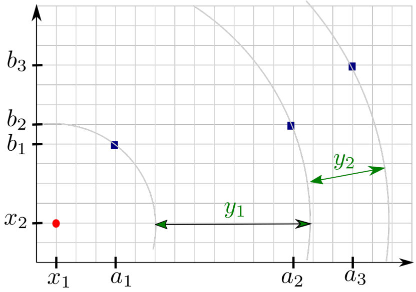
$$\mathbf{f}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ (x_1+x_2)^2 \end{pmatrix}.$$

The function $\mathbf{f}$ has the following contractible decomposition:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \left(s = x_1 + x_2 \right) \rightarrow \begin{pmatrix} s \\ s^2 \end{pmatrix}$$



Test-case 2. Pseudorange multilateration



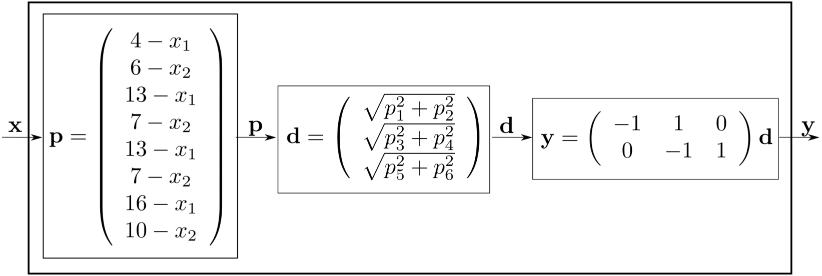
The robot measures the pseudo distances $y_1 = 8, y_2 = 4$ to the stations

We assume that the accuracy of the pseudo distance measurements is $\varepsilon = 0.001$. The set $\mathbb{X}$ of all feasible location vectors is defined by

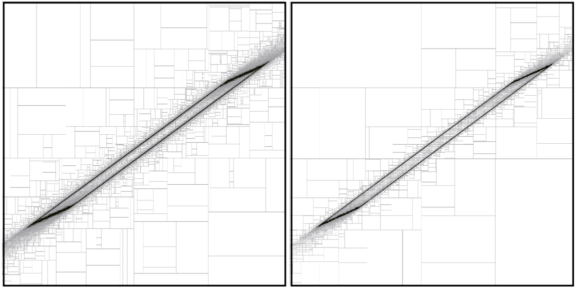
$$\mathbf{f}(\mathbf{x}) \in [8-\varepsilon, 8+\varepsilon] \times [4-\varepsilon, 4+\varepsilon]$$

where

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \sqrt{(13-x_1)^2+(7-x_2)^2} - \sqrt{(4-x_1)^2+(6-x_2)^2} \\ \sqrt{(16-x_1)^2+(10-x_2)^2} - \sqrt{(13-x_1)^2+(7-x_2)^2} \end{pmatrix}$$



Decomposition into contractible functions



Classical forward backward contractor generates 90841 boxes.

Our boundary based contractor with the contractible decomposition generated 35586 boxes.

Perspectives

Directed non-monotonic contractor for the constraint $\mathbf{y} = \mathbf{f}(\mathbf{x})$ is a pair of two operators

$$\mathscr{C} : ([\mathbf{x}],[\mathbf{y}]) \to \left(\overrightarrow{\mathscr{C}}([\mathbf{x}]), \overleftarrow{\mathscr{C}}([\mathbf{x}],[\mathbf{y}]) \right)$$

such that

$$\begin{array}{ccc} \mathbf{f}([\mathbf{x}]) & \subset & \overrightarrow{\mathscr{C}}([\mathbf{x}]) \\ \mathbf{f}^{-1}([\mathbf{y}]) \cap [\mathbf{x}] & \subset & \overleftarrow{\mathscr{C}}([\mathbf{x}],[\mathbf{y}]) \end{array}$$

Where $[\mathbf{x}],[\mathbf{y}]$ could be oriented boxes, ellipsoids, etc.
Lohner type contractors could thus be used.

P. Cousot and R. Cousot.
Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints.
In *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages*, pages 238–252, Los Angeles, California, 1977.

B. Desrochers and L. Jaulin.
Chain of set inversion problems; application to reachability analysis.
In *IFAC'2017, Toulouse*, 2017.

L. Jaulin.
A boundary approach for set inversion.
Engineering Applications of Artificial Intelligence (submitted), 2021.