Static Analysis of Numerical Programs
Constrained Affine Sets-Abstract Domain

Khalil Ghorbal
CMU

Joint work with Éric Goubault and Sylvie Putôt

NASA AMES
March 12th, 2013
Abstract Interpretation: Intuitions
Abstract Interpretation: Intuitions

\[ S \]

\[ i \]
Abstract Interpretation: Intuitions

\[ S \]

\[ \rightarrow \]

\[ i \]
What about the missed bugs? are they severe?
Abstract Interpretation: Intuitions

\[ S \]

\[ i \]
Abstract Interpretation: Intuitions

$S$

$\forall i$
Abstract Interpretation: Intuitions
Abstract Interpretation: Intuitions

- Over-approximation may lead to **false alarms**.
Accurate over-approximation gives a safety proof.
Famous bugs

Examples

- 1982, The Vancouver stock exchange: after 22 months the index had fallen to 524,811 instead of 1098,811
- 1985, Therac 25 (radiation therapy machine): 5 patients killed (overdoses of radiation)
- 1991, The Patriot Missile: 28 soldiers killed
- 1996, Ariane 5: more than 1 billion $ gone in 40 seconds

E. Dijkstra (1972)

*Program testing can be used to show the presence of bugs, but never to show their absence!*
ESA Project - Automated Transfer Vehicle (ATV)

Jules Verne (Key dates)
9th Mars 2008 launching
3th April 2008 docking to ISS
11th September 2008, undocking
29th September 2008, end of mission

✔ Publication in *DAta Systems In Aerospace (DASIA) 2009*

All existing abstract domains fail to handle precisely **normalized** quaternions
begin
x = [0,10]; ①
y = x*x - x ②
if (y >= 0) ③ then
y = x / 10; ④
else ⑤
y = x*x + 2; ⑥
done; ⑦
end

\[ y = x^2 - x \] ②
\[ y \geq 0 \] ③
\[ y = \frac{x}{10} \] ④
\[ y < 0 \] ⑤
\[ y = x^2 + 2 \] ⑥

K. Ghorbal (CMU)  
NASA Ames  
Invited Talk
Forward Propagation

\[ x = [0, 10] \]

\[ y = x^2 - x \]

\[ y \geq 0 \quad \text{and} \quad y < 0 \]

\[ y = \frac{x}{10} \]

\[ y = x^2 + 2 \]

\[ y < 0 \]

\[ y = x^2 - x \]

\[ y \geq 0 \]

\[ y < 0 \]
Forward Propagation

\[ x = [0, 10] \]

\[ y = x^2 - x \]

\[ x = [0, 10] \]
\[ y = [-10, 100] \]

\[ y \geq 0 \quad y < 0 \]

\[ y = \frac{x}{10} \]

\[ y = x^2 + 2 \]

\[ y < 0 \]

\[ y = x^2 + 2 \]

\[ y = x^2 - x \]
Forward Propagation

\[ x = [0, 10] \]
\[ y = x^2 - x \]

\[ x = [0, 10] \]
\[ y = [-10, 100] \]

\[ y \geq 0 \quad y < 0 \]

\[ x = [0, 10] \]
\[ y = [0, 100] \]
\[ y = \frac{x}{10} \]

\[ y = x^2 + 2 \]

\[ y = x^2 - x \]
\[ y = [-10, 100] \]

\[ y \geq 0 \quad y < 0 \]

\[ x = [0, 10] \]
\[ y = [0, 100] \]
\[ y = \frac{x}{10} \]

\[ y = x^2 + 2 \]
Forward Propagation

\[ x = [0, 10] \]

\[ y = x^2 - x \]

\[ x = [0, 10] \]
\[ y = [-10, 100] \]

\[ y \geq 0 \]
\[ y < 0 \]

\[ x = [0, 10] \]
\[ y = [0, 100] \]

\[ y = \frac{x}{10} \]

\[ x = [0, 10] \]
\[ y = [-10, 0] \]

\[ y = x^2 + 2 \]

\[ y < 0 \]

\[ \bigcup \]

\[ 7 \]
Forward Propagation

\[ x = [0, 10] \]
\[ y = x^2 - x \]
\[ x = [0, 10] \]
\[ y = [-10, 100] \]
\[ y \geq 0 \quad y < 0 \]

\[ x = [0, 10] \]
\[ y = [0, 100] \]
\[ y = \frac{x}{10} \]
\[ x = [0, 10] \]
\[ y = [-10, 0] \]
\[ y = x^2 + 2 \]

\[ x = [0, 10] \]
\[ y = [0, 1] \]
\[ y = [0, 1] \]
\[ y = [-10, 0] \]
Forward Propagation

\[ x = [0, 10] \]

\[ y = x^2 - x \]

\[ x = [0, 10] \]
\[ y = [-10, 100] \]

\[ y \geq 0 \]
\[ y < 0 \]

\[ x = [0, 10] \]
\[ y = [0, 100] \]

\[ y = \frac{x}{10} \]

\[ x = [0, 10] \]
\[ y = [0, 1] \]

\[ y = x^2 + 2 \]

\[ x = [0, 10] \]
\[ y = [2, 102] \]

\[ y = y^2 + y \geq 0 \]
\[ y < 0 \]
Forward Propagation

\[ x = [0, 10] \]

\[ y = x^2 - x \]

\[ y \geq 0 \]
\[ y < 0 \]

\[ y \geq 0 \]
\[ y < 0 \]

\[ y = \frac{x}{10} \]
\[ y = x^2 + 2 \]

\[ x = [0, 10] \]
\[ y = [0, 100] \]
\[ x = [0, 10] \]
\[ y = [-10, 0] \]
\[ x = [0, 10] \]
\[ y = [0, 1] \]
\[ x = [0, 10] \]
\[ y = [2, 102] \]

\[ \bigcup \]

\[ \text{invariant in } 7 \]
Forward Propagation

\[ x = [0, 10] \]
\[ y = x^2 - x \]

\[ y \geq 0 \]
\[ y < 0 \]

\[ y = \frac{x}{10} \]
\[ y = x^2 + 2 \]

\[ x = [0, 10] \]
\[ y = [0, 100] \]
\[ x = [0, 10] \]
\[ y = [-10, 0] \]
\[ x = [0, 10] \]
\[ y = [0, 1] \]
\[ x = [0, 10] \]
\[ y = [2, 102] \]

\[ y \in [-10, 100] \]

\[ y = x^2 - x \]

\[ y \geq 0 \]
\[ y < 0 \]

\[ y = \frac{x}{10} \]
\[ y = x^2 + 2 \]

\[ y \in [0, 100] \]

\[ y = [0, 1] \]
\[ y = [2, 102] \]

\[ y \in \mathbb{R} \]

\[ y \in [0, 100] \]

\[ y = x^2 - x \]

\[ y \geq 0 \]
\[ y < 0 \]

\[ y = \frac{x}{10} \]
\[ y = x^2 + 2 \]

\[ y \in [0, 100] \]

\[ y = [0, 1] \]
\[ y = [2, 102] \]

\[ \text{invariant in } 7 \]

\[ y \]
\[ x \]

\[ 1 \]
\[ 10 \]

\[ 0 \]
\[ 102 \]

\[ 3 \]
Forward Propagation

\[ x = [0, 10] \]

\[ y = x^2 - x \]

\[ y \geq 0 \]

\[ y < 0 \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ x = [0, 10] \quad y = [-0.25, 90] \]

\[ y = \frac{x}{10} \]

\[ x = [0, 10] \quad y = [0, 90] \]

\[ x = [0, 10] \quad y = [-0.25, 0] \]

\[ y = x^2 + 2 \]

\[ x = [0, 1] \quad y = [0, 1] \]

\[ x = [0, 1] \quad y = [2, 3] \]

\[ y = x^2 + 2 \]

\[ x = [0, 1] \quad y = [-0.25, 0] \]

\[ \cup \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ y = x^2 - x \]

\[ y \geq 0 \]

\[ y < 0 \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ x = [0, 10] \quad y = [-0.25, 90] \]

\[ y = \frac{x}{10} \]

\[ x = [0, 10] \quad y = [0, 90] \]

\[ x = [0, 10] \quad y = [-0.25, 0] \]

\[ y = x^2 + 2 \]

\[ x = [0, 1] \quad y = [0, 1] \]

\[ x = [0, 1] \quad y = [2, 3] \]

\[ \cup \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ y = x^2 - x \]

\[ y \geq 0 \]

\[ y < 0 \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ x = [0, 10] \quad y = [-0.25, 90] \]

\[ y = \frac{x}{10} \]

\[ x = [0, 10] \quad y = [0, 90] \]

\[ x = [0, 10] \quad y = [-0.25, 0] \]

\[ y = x^2 + 2 \]

\[ x = [0, 1] \quad y = [0, 1] \]

\[ x = [0, 1] \quad y = [2, 3] \]

\[ \cup \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ y = x^2 - x \]

\[ y \geq 0 \]

\[ y < 0 \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ x = [0, 10] \quad y = [-0.25, 90] \]

\[ y = \frac{x}{10} \]

\[ x = [0, 10] \quad y = [0, 90] \]

\[ x = [0, 10] \quad y = [-0.25, 0] \]

\[ y = x^2 + 2 \]

\[ x = [0, 1] \quad y = [0, 1] \]

\[ x = [0, 1] \quad y = [2, 3] \]

\[ \cup \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ y = x^2 - x \]

\[ y \geq 0 \]

\[ y < 0 \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ x = [0, 10] \quad y = [-0.25, 90] \]

\[ y = \frac{x}{10} \]

\[ x = [0, 10] \quad y = [0, 90] \]

\[ x = [0, 10] \quad y = [-0.25, 0] \]

\[ y = x^2 + 2 \]

\[ x = [0, 1] \quad y = [0, 1] \]

\[ x = [0, 1] \quad y = [2, 3] \]

\[ \cup \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ y = x^2 - x \]

\[ y \geq 0 \]

\[ y < 0 \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ x = [0, 10] \quad y = [-0.25, 90] \]

\[ y = \frac{x}{10} \]

\[ x = [0, 10] \quad y = [0, 90] \]

\[ x = [0, 10] \quad y = [-0.25, 0] \]

\[ y = x^2 + 2 \]

\[ x = [0, 1] \quad y = [0, 1] \]

\[ x = [0, 1] \quad y = [2, 3] \]

\[ \cup \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ y = x^2 - x \]

\[ y \geq 0 \]

\[ y < 0 \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ x = [0, 10] \quad y = [-0.25, 90] \]

\[ y = \frac{x}{10} \]

\[ x = [0, 10] \quad y = [0, 90] \]

\[ x = [0, 10] \quad y = [-0.25, 0] \]

\[ y = x^2 + 2 \]

\[ x = [0, 1] \quad y = [0, 1] \]

\[ x = [0, 1] \quad y = [2, 3] \]

\[ \cup \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ y = x^2 - x \]

\[ y \geq 0 \]

\[ y < 0 \]

\[ x = [0, 10] \quad y = [0, 10] \]

\[ x = [0, 10] \quad y = [-0.25, 90] \]

\[ y = \frac{x}{10} \]

\[ x = [0, 10] \quad y = [0, 90] \]

\[ x = [0, 10] \quad y = [-0.25, 0] \]

\[ y = x^2 + 2 \]

\[ x = [0, 1] \quad y = [0, 1] \]

\[ x = [0, 1] \quad y = [2, 3] \]

\[ \cup \]
Precision Cost Trade-off

Cost

Precision

box

octagon
Precision Cost Trade-off

Cost

Precision

box

octagon

template
Precision Cost Trade-off

Cost

Precision

box

octagon

template

polyhedron
Precision Cost Trade-off

Cost

Precision

box

octagon

template

polyhedron
Precision Cost Trade-off

Cost

Precision

box

octagon

template

zonotope

polyhedron
Precision Cost Trade-off

![Diagram showing the trade-off between precision and cost with different geometric shapes representing various constraints and approximations.](image-url)
Outlines

1. Static Analysis-based Abstract Interpretation
2. Affine Sets Abstract Domain
3. Constrained Affine Sets Abstract Domain
4. Experiments, Taylor1+
5. Appendix
Formal Verification Approaches

- Hoare 1969: wrap the code of interest with preconditions and postconditions, then prove that postconditions are met
- Clarke, Emerson et Sifakis 1974: model checking
- Cousot(s) 1977: Abstract Interpretation

Properties of Interest

- run time errors: overflow, division by zero, square root of negatives, etc.
- robustness and stability of algorithms: linear and non linear recursive schemes, filters, etc.
Formal Verification Approaches

- Hoare 1969: wrap the code of interest with preconditions and postconditions, then prove that postconditions are met
- Clarke, Emerson et Sifakis 1974: model checking
- Cousot(s) 1977: Abstract Interpretation

Properties of Interest

- run time errors: overflow, division by zero, square root of negatives, etc.
- robustness and stability of algorithms: linear and non linear recursive schemes, filters, etc.
Formal Verification Approaches

- Hoare 1969: wrap the code of interest with preconditions and postconditions, then prove that postconditions are met
- Clarke, Emerson et Sifakis 1974: model checking
- Cousot(s) 1977: Abstract Interpretation

Properties of Interest

- run time errors: overflow, division by zero, square root of negatives, etc.
- robustness and stability of algorithms: linear and non linear recursive schemes, filters, etc.
Formal Verification Approaches

- Hoare 1969: wrap the code of interest with preconditions and postconditions, then prove that postconditions are met
- Clarke, Emerson et Sifakis 1974: model checking
- Cousot(s) 1977: Abstract Interpretation

Properties of Interest

- run time errors: overflow, division by zero, square root of negatives, etc.
- robustness and stability of algorithms: linear and non linear recursive schemes, filters, etc.
Abstract Interpretation, an overview

- Program semantics formalized as a fixpoint of a monotonic operator in a complete partially ordered set (exemplified later),
- Fully automated,
- Industrial tools exist: Polyspace Verifier (MathWorks), Astrée (ENS/ABSINT), Fluctuat (CEA), aIT (ABSINT), F-Soft (Nec Labs) ...

**Challenge**

find the *suitable* abstract domain for the properties of interest.
Equations System (collecting semantic)

\[ y = x^2 - x \]

\[ y \geq 0 \]
\[ y < 0 \]

\[ y = \frac{x}{10} \]
\[ y = x^2 + 2 \]

\[ x_1 = [\mathcal{V} \rightarrow I]^b \]
\[ x_2 = [y \leftarrow x^2 - x]^b(x_1) \]
\[ x_3 = [y \geq 0]^b(x_2) \]
\[ x_4 = [y \leftarrow \frac{x}{10}]^b(x_3) \]
\[ x_5 = [y < 0]^b(x_2) \]
\[ x_6 = [y \leftarrow x^2 + 2]^b(x_5) \]
\[ x_7 = x_6 \cup x_4 \]
Solving the equations system

- $D = (\wp(V \rightarrow I), \subseteq, \cup, \cap, \emptyset, (V \rightarrow I))$ is a **complete lattice**
- each operator $X \mapsto \mathcal{F}(X)$ is monotonic
- **Tarski Theorem** ensures the existence of a least fixpoint for $\mathcal{F}$
- **Kleene Iteration Technique** reaches the least fixpoint

### Issues

- $\wp(V \rightarrow I)$ is non representable in finite memory,
- $[\cdot]^b$ are non computable,
- Iterations over the lattice may be transfinite.
Concretisation-Based Abstract Interpretation

\[ \gamma(X_1^\#) \subseteq \gamma(X_2^\#) \]

\[ [y \leftarrow x^2 - x]^b \]

\[ x_1 \rightarrow x_2 \]

\[ \alpha \]

\[ \gamma \]

\[ \subseteq \]

concrete domain

abstract domain

over approximation

K. Ghorbal (CMU)  NASA Ames  13

Invited Talk  13 / 74
Building an abstract domain

- **lattice-like structure:**
  - abstract objects
  - order relation (preorder over abstract objects)
  - monotonic concretisation function ($\gamma$)

- **Transfer Functions**
  - evaluation of arithmetic expressions ($[x^2 - x]_\#$)
  - assignment ($x_2 = [y \leftarrow x^2 - x]_\#(x_1)$)
  - upper bound (join) ($x_7 = x_6 \cup x_4$)
  - over-approximation of lower bounds (meet) ($x_3 = [y \geq 0]_\# x_2 =\ x_3 = x_2 \cap [y \geq 0]_\# \top_\#$)

- **Convergence acceleration (widening)**
Outlines

1. Static Analysis-based Abstract Interpretation
2. Affine Sets Abstract Domain
3. Constrained Affine Sets Abstract Domain
4. Experiments, Taylor1+
5. Appendix
Affine Sets

introduction

\[ \hat{X} = \begin{cases} 
\hat{x}_1 = \alpha_0^{x_1} + \sum_{i=1}^{n} \alpha_i^{x_1} \epsilon_i \\
\hat{x}_2 = \alpha_0^{x_2} + \sum_{i=1}^{n} \alpha_i^{x_2} \epsilon_i , \quad (\epsilon_1, \ldots, \epsilon_n) \in [-1, 1]^n \\
\hat{x}_3 = \ldots 
\end{cases} \]

\[ \hat{X} = \begin{pmatrix} \alpha_0^x & \cdots & \alpha_n^x \\
\vdots & \ddots & \vdots \\
\alpha_p^x & \cdots & \alpha_p^x \\
\end{pmatrix} \times \begin{pmatrix} \epsilon_0 \\
\epsilon_1 \\
\vdots \\
\epsilon_n \\
\end{pmatrix} , \quad \epsilon \in \mathbf{1} \times [-1, 1]^n \]
Geometric Concretisation: Zonotope

Minkowski sum of a set of segments

\[
\begin{align*}
\hat{x} &= 10 - 4\epsilon_1 + 2\epsilon_3 + 3\epsilon_4 \\
\hat{y} &= 5 - 2\epsilon_1 + 1\epsilon_2 - 1\epsilon_4
\end{align*}
\]
Geometric Concretisation: Zonotope

Minkowski sum of a set of segments

\[
\hat{x} = 10 - 4\epsilon_1 + 2\epsilon_3 + 3\epsilon_4 \\
\hat{y} = 5 - 2\epsilon_1 + 1\epsilon_2 - 1\epsilon_4
\]

\[x \in [1, 19] \quad \text{and} \quad y \in [1, 9]\]
Geometric Concretisation: Zonotope

Minkowski sum of a set of segments

\[ \hat{x} = 10 - 4\epsilon_1 + 2\epsilon_3 + 3\epsilon_4 \quad \text{for} \quad x \in [1, 19] \]

\[ \hat{y} = 5 - 2\epsilon_1 + 1\epsilon_2 - 1\epsilon_4 \quad \text{for} \quad y \in [1, 9] \]
Geometric Concretisation: Zonotope

Minkowski sum of a set of segments

\[ \hat{x} = 10 - 4\epsilon_1 + 2\epsilon_3 + 3\epsilon_4 \quad x \in [1, 19] \]
\[ \hat{y} = 5 - 2\epsilon_1 + 1\epsilon_2 - 1\epsilon_4 \quad y \in [1, 9] \]
Geometric Concretisation: Zonotope

Minkowski sum of a set of segments

\[
\begin{align*}
\hat{x} &= 10 - 4\epsilon_1 + 2\epsilon_3 + 3\epsilon_4 & x \in [1, 19] \\
\hat{y} &= 5 - 2\epsilon_1 + \epsilon_2 - \epsilon_4 & y \in [1, 9]
\end{align*}
\]
Geometric Concretisation: Zonotope

Minkowski sum of a set of segments

\[
\hat{x} = 10 - 4\epsilon_1 + 2\epsilon_3 + 3\epsilon_4 \quad x \in [1, 19]
\]

\[
\hat{y} = 5 - 2\epsilon_1 + \epsilon_2 - 1\epsilon_4 \quad y \in [1, 9]
\]
Perturbed Affine Sets

The Affine Sets are extended with \textbf{Perturbation} terms to handle the \textbf{non linear} operations: multiplication, join, etc.

\[
\hat{X} = \begin{cases} 
\hat{x} = \alpha_0^x + \sum_{i=1}^{n} \alpha_i^x \epsilon_i + \sum_{j=1}^{m} \beta_j^x \eta_j^x \\
\hat{y} = \alpha_0^y + \sum_{i=1}^{n} \alpha_i^y \epsilon_i + \sum_{j=1}^{m} \beta_j^y \eta_j^x \\
\hat{z} = \ldots
\end{cases}
\]

\[
\hat{X} = C^X \varepsilon + P^X \eta^X,
\]

\[
\varepsilon = (\epsilon_0, \ldots, \epsilon_n) \in 1 \times [-1, 1]^n
\]

\[
\eta^X = (\eta_1^X, \ldots, \eta_m^X) \in [-1, 1]^m
\]

- \textbf{p} numerical variables
- \textbf{n} \textit{input} noise symbols
- \textbf{m} \textit{perturbation} noise symbols
Intuition

Geometrical inclusion of the concretisation of the vector $\hat{X}$ augmented by the input noise symbols $\varepsilon$.

$$\hat{X} \leq_1 \hat{Y}$$

$$\Delta \iff$$

$$\{ \gamma(\hat{X}, \varepsilon) \mid \hat{X} = C^X \varepsilon + P^X \eta^X \} \subseteq \{ \gamma(\hat{Y}, \varepsilon) \mid \hat{Y} = C^Y \varepsilon + P^Y \eta^Y \}$$
Functional Order Relation, Equivalent Formulations

Inclusion of sets of functions

∀ε ∈ 1 × [−1, 1]ⁿ, ∀ηₓ ∈ [−1, 1]ᵐ, ∃ηᵧ ∈ [−1, 1]ᵐ:

\[ Cₓε + Pₓηₓ = Cᵧε + Pᵧηᵧ. \]

Sets Inclusion

\[(Cˣ - Cʸ)Φₓ + PₓΦₓηₓ ⊆ PʸΦᵧηᵧ, \left\{ \begin{array}{l}
Φₓ = 1 × [−1, 1]ⁿ \\
Φₓηₓ = Φᵧηᵧ = [−1, 1]ᵐ
\end{array} \right. \]

Support Function Inequality

∀t ∈ ℝᵖ, sup_{ε ∈ Φₓ} |⟨(Cˣ - Cʸ)ε, t⟩| ≤ sup_{ηᵧ ∈ Φᵧ} |⟨Pᵧηᵧ, t⟩| − sup_{ηₓ ∈ Φₓ} |⟨Pₓηₓ, t⟩|
The support function

Let $C$ be a non empty convex set of $\mathbb{R}^p$, then

$$\delta(t \mid C) \overset{\text{def}}{=} \sup\{ \langle t, x \rangle \mid x \in C \},$$

where $\langle \cdot, \cdot \rangle$ denotes the usual scalar product over $\mathbb{R}^p$.

Proposition

Let, $C_1$ and $C_2$ be non empty convex sets, then

$$C_1 \subseteq C_2 \iff \forall t \in \mathbb{R}^p, \delta(t \mid C_1) \leq \delta(t \mid C_2).$$
Support Function Inequality: Perturbed Affine Sets

\[ \forall t \in \mathbb{R}^p, \sup_{\varepsilon \in \Phi_\varepsilon} |\langle (C^X - C^Y)\varepsilon, t \rangle| \leq \sup_{\eta^Y \in \Phi^Y_\eta} |\langle P^Y \eta^Y, t \rangle| - \sup_{\eta^X \in \Phi^X_\eta} |\langle P^X \eta^X, t \rangle| \]

Norm \text{ } L_1 \text{ } Formulation

\[ \forall t \in \mathbb{R}^p, \| (C^X - C^Y)^* t \|_1 \leq \| P^Y^* t \|_1 - \| P^X^* t \|_1 \]

where \[ x \in \mathbb{R}^n, \| x \|_1 \overset{\text{def}}{=} \sum_{i=1}^n |x_i| . \]
Support Function Inequality: Perturbed Affine Sets

Perturbed Affine Sets

\[ \forall t \in \mathbb{R}^p, \sup_{\varepsilon \in \Phi_\varepsilon} |\langle (C^X - C^Y)\varepsilon, t \rangle| \leq \sup_{\eta^Y \in \Phi^Y} |\langle P^Y \eta^Y, t \rangle| - \sup_{\eta^X \in \Phi^X} |\langle P^X \eta^X, t \rangle| \]

Norm $L_1$ Formulation

\[ \forall t \in \mathbb{R}^p, \| (C^X - C^Y)^* t \|_1 \leq \| P^Y^* t \|_1 - \| P^X^* t \|_1 \]

where \( x \in \mathbb{R}^n, \| x \|_1 \overset{\text{def}}{=} \sum_{i=1}^n |x_i| \).
Arithmetic Operations on Perturbed Affine Forms

Linear Operations

Closed under affine transformations

\[ \hat{x} \pm \hat{y} \overset{\text{def}}{=} \sum_{i=0}^{n} (\alpha_i^x \pm \alpha_i^y) \epsilon_i + \sum_{j=1}^{m} (\beta_j^x \pm \beta_j^y) \eta_j \]

\[ \lambda.\hat{x} \overset{\text{def}}{=} \sum_{i=0}^{n} (\lambda \alpha_i^x) \epsilon_i + \sum_{j=1}^{m} (\lambda \beta_j^x) \eta_j \]

Proposition

The assignment of linear expression is monotonic.
Arithmetic Operations on Perturbed Affine Forms
Non Linear Unary Operations

For non linear (unary) operations (3 steps)
1. linearize using first order Taylor development
2. bound the non linear term
3. rewrite the interval as an affine form using a fresh noise symbol

Square root example
- $x \in [3, 5] : \hat{x} = 4 + \epsilon_1$
- $\hat{y} = \sqrt{\hat{x}} = 2 + 0.25\epsilon_1 + 0.024\epsilon_f$
- $\gamma(\hat{y}) = [1.726, 2.274] \supseteq [1.732, 2.236]$
Arithmetic Operations on Perturbed Affine Forms

Multiplication

Multiplication operation

\[
\hat{x} \times \hat{y} = \alpha_0^x \alpha_0^y + \sum_{i=1}^{n} (\alpha_i^x \alpha_0^y + \alpha_i^y \alpha_0^x) \epsilon_i + \sum_{j=1}^{m} (\beta_j^x \alpha_0^y + \beta_j^y \alpha_0^x) \eta_j + \ell(\hat{x}, \hat{y}) \eta_f.
\]

Two methods to bound the quadratic non linear term

1. **Straightforward method: interval arithmetic**
   - rough approximation but efficient computation

2. **SemiDefinite Programming Technique**
   - more accurate but more expensive on time
Comparative example

#My Simple Program
x = [0,2];
y = x + [0,2];
z = x*y;
t = z - 2*x - y;

Abstract computations
1. $\hat{x} = 1 + \epsilon_1$
2. $\hat{y} = 2 + \epsilon_1 + \epsilon_2$
3. $\hat{z} = 2.875 + 3\epsilon_1 + \epsilon_2 + 1.125\eta_1$
4. $\hat{t} = -1.125 + 1.125\eta_1$
Comparative example

#My Simple Program

x = [0,2];
y = x + [0,2];
z = x*y;
t = z - 2*x - y;

Abstract computations

1. $\hat{x} = 1 + \epsilon_1$
2. $\hat{y} = 2 + \epsilon_1 + \epsilon_2$
3. $\hat{z} = 2.875 + 3\epsilon_1 + \epsilon_2 + 1.125\eta_1$
4. $\hat{t} = -1.125 + 1.125\eta_1$
Comparative example

#My Simple Program

\[
x = [0, 2]; \quad 1
\]
\[
y = x + [0, 2]; \quad 2
\]
\[
z = x*y;
\]
\[
t = z - 2*x - y;
\]

Abstract computations

1. \[\hat{x} = 1 + \epsilon_1\]
2. \[\hat{y} = 1 + \epsilon_1 + (1 + \epsilon_2) = 2 + \epsilon_1 + \epsilon_2\]
3. \[\hat{z} = 2.875 + 3\epsilon_1 + \epsilon_2 + 1.125\eta_1\]
4. \[\hat{t} = -1.125 + 1.125\eta_1\]
Comparative example

My Simple Program

```plaintext
#My Simple Program
x = [0,2];  
y = x + [0,2];
z = x*y;
t = z - 2*x - y;
```

Abstract computations

1. \( \hat{x} = 1 + \epsilon_1 \)
2. \( \hat{y} = 2 + \epsilon_1 + \epsilon_2 \)
3. \( \hat{z} = 2.875 + 3\epsilon_1 + \epsilon_2 + 1.125\eta_1 \)
4. \( \hat{t} = -1.125 + 1.125\eta_1 \)
Comparative example

My Simple Program

```plaintext
x = [0,2];  
y = x + [0,2];  
z = x*y;  
t = z - 2*x - y;
```

Abstract computations

1. \( \hat{x} = 1 + \epsilon_1 \)
2. \( \hat{y} = 2 + \epsilon_1 + \epsilon_2 \)
3. \( \hat{z} = 2.875 + 3\epsilon_1 + \epsilon_2 + 1.125\eta_1 \)
4. \( \hat{t} = -1.125 + 1.125\eta_1 \)
Comparative example

#My Simple Program

1. \( x = [0,2] \)
2. \( y = x + [0,2] \)
3. \( z = x*y \)
4. \( t = z - 2*x - y \)

Abstract computations

1. \( \hat{x} = 1 + \epsilon_1 \)
2. \( \hat{y} = 2 + \epsilon_1 + \epsilon_2 \)
3. \( \hat{z} = 2.875 + 3\epsilon_1 + \epsilon_2 + 1.125\eta_1 \)
4. \( \hat{t} = -1.125 + 1.125\eta_1 \)

- \((x,t)\) invariant in control point

Polyhedra

\[
\begin{align*}
t + x & \leq 4 \\
-t - 3x & \leq 2 \\
x & \leq 2 \\
-x & \leq 0
\end{align*}
\]

Octagons

\[
\begin{align*}
t - x & \leq 8 \\
t + x & \leq 8 \\
-t - x & \leq 6 \\
-t + x & \leq 10 \\
t & \leq 8 \\
-t & \leq 8 \\
x & \leq 2 \\
-x & \leq 0
\end{align*}
\]
Comparative example

Concretisation of abstract set (4) projected onto (t,x) plane

Interval concretisation of variable $t$

- **Octagons**  \( t \in [-8, 8] \)
- **Polyhedra**  \( t \in [-8, 4] \)
- **PAS**  \( t \in [-2.25, 0] \)  \( \hat{t} = -1.125 + 1.125\eta_1 \)
Comparative example

Concretisation of abstract set (4) projected onto (t,x) plane

Interval concretisation of variable $t$

- Octagons \( t \in [-8, 8] \)
- Polyhedra \( t \in [-8, 4] \)
- PAS \( t \in [-2.25, 0] \quad (\hat{t} = -1.125 + 1.125\eta_1) \)
Comparative example
Concretisation of abstract set (4) projected onto (t,x) plane

Interval concretisation of variable $t$
- Octagons \( t \in [-8, 8] \)
- Polyhedra \( t \in [-8, 4] \)
- PAS \( t \in [-2.25, 0] \) \( \hat{t} = -1.125 + 1.125\eta_1 \)
Comparative example

Concretisation of abstract set (4) projected onto (t,x) plane

Interval concretisation of variable t

- Octagons  \( t \in [-8, 8] \)
- Polyhedra  \( t \in [-8, 4] \)
- PAS  \( t \in [-2.25, 0] \quad (\hat{t} = -1.125 + 1.125\eta_1) \)
Comparative example

Concretisation of abstract set (4) projected onto (t,x) plane

Interval concretisation of variable $t$

- Octagons \[ t \in [-8, 8] \]
- Polyhedra \[ t \in [-8, 4] \]
- PAS \[ t \in [-2.25, 0] \] \( \hat{t} = -1.125 + 1.125\eta_1 \)
Join over Perturbed Affine Sets

- we do not have a supremum in general,
- many “minimal” upper bounds exist,
- the best (if many) minimal upper bound depends on the future evaluations
- computing a minimal enclosing zonotope of two given zonotopes is a hard problem

Minimal Upper Bound with respect to $\leq / \sim$

$Z$ is a minimal upper bound of $X$ and $Y$ if and only if

- **upper bound**: $X \leq Z$ and $Y \leq Z$, and
- **minimal**: for all $W$ upper bound of $X$ and $Y$, $Z \leq W \implies Z \sim W$
A Perturbed Affine Form is nothing but 1-dim a Perturbed Affine Sets (with one perturbation noise symbol).

\[ \hat{x} = \alpha_0^x + \sum_{i=1}^{n} \alpha_i^x \epsilon_i + \beta^x \eta_u^x \]

and therefore

\[ \hat{x} \leq_1 \hat{y} \iff \|\alpha^x - \alpha^y\|_1 \leq |\beta^y| - |\beta^x| \]
Proposition

The following \( \hat{z} \) is a minimal upper bound of \( \hat{x} \) and \( \hat{y} \) (if \( \hat{x} \) and \( \hat{y} \) are non comparable) whose **interval concretisation is the union of the interval concretizations** of \( \hat{x} \) and \( \hat{y} \):

\[
\begin{align*}
\alpha_0^z &= \text{mid}(\gamma(\hat{x}) \cup \gamma(\hat{y})) & \text{(central value of} \ \hat{z}) \\
\alpha_i^z &= \text{argmin}_{\min(\alpha_i^x, \alpha_i^y) \leq \alpha \leq \max(\alpha_i^x, \alpha_i^y)} (|\alpha|), \forall i \geq 1 & \text{(coeff. of} \ \epsilon_i) \\
\beta^z &= \text{sup}(\gamma(\hat{x}) \cup \gamma(\hat{y})) - \alpha_0^z - \sum_{i \geq 1} |\alpha_i^z| & \text{(coeff. of} \ \epsilon_U) 
\end{align*}
\]

where:

- \( \gamma(\hat{x}) = [\alpha_0^x - \sum_{i=1}^n |\alpha_i^x|, \alpha_0^x + \sum_{i=1}^n |\alpha_i^x|] \),
- and \( \text{mid}([a, b]) := \frac{1}{2} (a + b) \),
- and \( \text{argmin}(|x|) := \{ x \in [a, b], |x| \text{ is minimal} \} \).
Componentwise Relaxation

\[
\hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix} \leq_1 \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix} \implies \forall i, 1 \leq i \leq p, \hat{x}_i \leq_1 \hat{y}_i
\]

Moreover, when the Perturbation Matrices $P^Y$ and $P^X$ are diagonal, the equivalence holds.

We have a **linear algorithm** to compute a **minimal upper bound** of two Perturbed Affine Forms.

Idea: use componentwise minimal upper bounds computation to define an upper bound of $\hat{X}$ and $\hat{Y}$ in the general case.
Componentwise Relaxation

\[
\hat{X} = \begin{pmatrix}
\hat{x}_1 \\
\vdots \\
\hat{x}_p
\end{pmatrix}
\leq_1 \hat{Y} = \begin{pmatrix}
\hat{y}_1 \\
\vdots \\
\hat{y}_p
\end{pmatrix}
\Rightarrow \forall i, 1 \leq i \leq p, \hat{x}_i \leq_1 \hat{y}_i
\]

- Moreover, when the Perturbation Matrices \( P^Y \) and \( P^X \) are diagonal, the equivalence holds.

- We have a linear algorithm to compute a minimal upper bound of two Perturbed Affine Forms.

- Idea: use componentwise minimal upper bounds computation to define an upper bound of \( \hat{X} \) and \( \hat{Y} \) in the general case.
Componentwise Relaxation

\[ \hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix} \leq_1 \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix} \Rightarrow \forall i, 1 \leq i \leq p, \hat{x}_i \leq_1 \hat{y}_i \]

Moreover, when the Perturbation Matrices \( P^Y \) and \( P^X \) are diagonal, the equivalence holds.

We have a linear algorithm to compute a minimal upper bound of two Perturbed Affine Forms.

Idea: use componentwise minimal upper bounds computation to define an upper bound of \( \hat{X} \) and \( \hat{Y} \) in the general case.
Componentwise Relaxation

\[ \hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix} \leq_1 \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix} \Rightarrow \forall i, 1 \leq i \leq p, \hat{x}_i \leq_1 \hat{y}_i \]

- Moreover, when the Perturbation Matrices \( P^Y \) and \( P^X \) are diagonal, the equivalence holds.
- We have a **linear algorithm** to compute a minimal upper bound of two Perturbed Affine Forms.
- Idea: use componentwise minimal upper bounds computation to define an upper bound of \( \hat{X} \) and \( \hat{Y} \) in the general case.
Componentwise Relaxation

\[
\hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix} \leq_1 \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix} \quad \Rightarrow \quad \forall i, 1 \leq i \leq p, \hat{x}_i \leq_1 \hat{y}_i
\]

- Moreover, when the Perturbation Matrices \( P^Y \) and \( P^X \) are diagonal, the equivalence holds.
- We have a linear algorithm to compute a minimal upper bound of two Perturbed Affine Forms.
- Idea: use componentwise minimal upper bounds computation to define an upper bound of \( \hat{X} \) and \( \hat{Y} \) in the general case.
Join : Relaxing the Problem

We Over-approximate \( \hat{X} \) and \( \hat{Y} \)

- \( \hat{X} \leq_1 \hat{X}' = C^X \varepsilon + P^{X'} \eta^X \)
  - \( P^{X'} \) is a **diagonal** matrix,
  - \( P^{X'}_{i,i} = \delta(P^X e_i \mid \Phi^X_{\eta})(= \| (\beta^{x_i}_1, \ldots, \beta^{x_i}_m) \|_1) \).
- The over-approximation of \( \hat{Y} \) by \( \hat{Y}' \) is similar.

- We compute componentwisely a Minimal Upper Bound of \( \hat{X}' \) and \( \hat{Y}' \).
  - We get an upper bound of \( \hat{X} \) and \( \hat{Y} \).
Example: Join over Perturbed Affine Sets

\[
\begin{align*}
\hat{x}_1 &= 3 + \epsilon_1 + 2\epsilon_2 \\
\hat{x}_2 &= 0 + \epsilon_1 + \epsilon_2 \\
\hat{y}_1 &= 1 - 2\epsilon_1 + \epsilon_2 \\
\hat{y}_2 &= 0 + \epsilon_1 + \epsilon_2
\end{align*}
\cup
\begin{align*}
\hat{z}_1 &= 2 + \epsilon_2 + 3\eta_u \\
\hat{z}_2 &= \epsilon_1 + \epsilon_2
\end{align*}
\]

Properties
- \(\gamma(\hat{z}_i) = \gamma(\hat{x}_i) \cup \gamma(\hat{y}_i)\)
- Complexity: \(\mathcal{O}(pn)\)
Example: Join over Perturbed Affine Sets

\[
\begin{align*}
\hat{x}_1 &= 3 + \epsilon_1 + 2\epsilon_2 \\
\hat{x}_2 &= 0 + \epsilon_1 + \epsilon_2
\end{align*}
\cup
\begin{align*}
\hat{y}_1 &= 1 - 2\epsilon_1 + \epsilon_2 \\
\hat{y}_2 &= 0 + \epsilon_1 + \epsilon_2
\end{align*}
\cup
\begin{align*}
\hat{z}_1 &= 2 + \epsilon_2 + 3\eta^Z_u \\
\hat{z}_2 &= \epsilon_1 + \epsilon_2
\end{align*}
\]

Properties
- \( \gamma(\hat{z}_i) = \gamma(\hat{x}_i) \cup \gamma(\hat{y}_i) \)
- Complexity: \( \mathcal{O}(pn) \)
Meet Operation

Issues

- Unlike the join, choosing a maximal lower bound is not sound,
- The set of Affine Forms is not a Riez space, that is \(-((−\hat{x}) \cup (−\hat{y}))\) does not give a maximal lower bound in general,
- The intersection of a hyperplane and a zonotope is not a zonotope in general,
- The meet of two non equal non perturbed affine forms \((\beta^x = \beta^y = 0)\) is the bottom element,

Tests are mainly ignored in Perturbed Affine Sets Abstract Domain, which is sound but too pessimistic. Reduced product with intervals is used to improve the precision.
Interpretation of Tests

Intuitions

\[ \forall i, \epsilon_i \in [-1, 1] \]

\[ \hat{x} = \epsilon_1 - \epsilon_2 \]

\[ \hat{y} = 2\epsilon_1 \]

\[ 0 \leq \hat{x} \]

- The intersection is a general polytope
- Propagate the constraint on noise symbols
Interpretation of Tests

Intuitions

∀i, ϵ_i ∈ [−1, 1]

\[ \hat{x} = \epsilon_1 - \epsilon_2 \]
\[ \hat{y} = 2\epsilon_1 \]
\[ 0 \leq \hat{x} \]

- The intersection is a general polytope
- Propagate the constraint on noise symbols
Interpretation of Tests

Intuitions

\[ \forall i, \epsilon_i \in [-1, 1] \]

\[ \hat{x} = \epsilon_1 - \epsilon_2 \]

\[ \hat{y} = 2\epsilon_1 \]

\[ 0 \leq \hat{x} \]

- The intersection is a **general polytope**
- Propagate the constraint on noise symbols
Interpretation of Tests

Intuitions

∀\(i\), \(\epsilon_i \in [-1, 1]\)

\(\epsilon_2 \leq \epsilon_1\)

\(\hat{x} = \epsilon_1 - \epsilon_2\)

\(\hat{y} = 2\epsilon_1\)

0 \(\leq \hat{x}\)

- The intersection is a general polytope
- Propagate the constraint on noise symbols
Interpretation of Tests

Intuitions

\( \forall i, \epsilon_i \in [-1, 1] \)
\( \epsilon_2 \leq \epsilon_1 \)

\( \hat{x} = \epsilon_1 - \epsilon_2 \)
\( \hat{y} = 2\epsilon_1 \)
\( 0 \leq \hat{x} \)

- The intersection is a **general polytope**
- **Propagate** the constraint on noise symbols
Outlines

1. Static Analysis-based Abstract Interpretation
2. Affine Sets Abstract Domain
3. Constrained Affine Sets Abstract Domain
4. Experiments, Taylor1+
5. Appendix

K. Ghorbal (CMU)  NASA Ames  36  Invited Talk  36 / 74
Constrained Affine Sets

\[ \hat{X} = \begin{cases} 
\hat{x} = \alpha_0^x + \sum_{i=1}^{n} \alpha_i^x \epsilon_i + \sum_{j=1}^{m} \beta_j^x \eta_j^x \\
\hat{y} = \alpha_0^y + \sum_{i=1}^{n} \alpha_i^y \epsilon_i + \sum_{j=1}^{m} \beta_j^y \eta_j^x \\
\hat{z} = \ldots 
\end{cases} \]

\[ \hat{X} = C^X \varepsilon + P^X \eta^X, (\varepsilon, \eta^X) \in \gamma_2(\Phi^X) \]

\( \Phi^X \) is an element of another abstract domain \( \mathcal{A}_2 \) (boxes, octagons, polyhedra etc.) .
Geometric Concretisation
not a zonotope in general

\[ \hat{X} = \left\{ \begin{array}{l}
\hat{x}_1 = 1 + \epsilon_1 + \epsilon_2 + \eta_{u}^{x_1} \\
\hat{x}_2 = -1 + 2\epsilon_2 + \eta_{u}^{x_2}
\end{array} \right., \Phi \]

\[ \eta_{u}^{x_1}, \eta_{u}^{x_2} \in [-1, 1] \]
Order Relation over Constrained Affine Sets

\[ \hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix}, \Phi^X \leq_{1\times2} \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix}, \Phi^Y \]

\[ \Delta \Leftrightarrow \Phi^X \leq_{2} \Phi^Y \]

\[ \{ \gamma(\hat{X}, \varepsilon) \mid \hat{X} = C^X\varepsilon + P^X\eta^X, (\varepsilon, \eta^X) \in \gamma_2(\Phi^X) \} \subseteq \{ \gamma(\hat{Y}, \varepsilon) \mid \hat{X} = C^Y\varepsilon + P^Y\eta^Y, (\varepsilon, \eta^Y) \in \gamma_2(\Phi^Y) \} \]
Upper Bound of Two Constrained Affine Sets

- \((\hat{X}, \Phi^X) \leq_{1 \times 2} (\hat{X}, \Box \Phi^X)\),
- \(\Box \Phi^X\) defined as the smallest box that contains \(\gamma_2(\Phi^X)\),
- Similarly for \((\hat{Y}, \Phi^Y)\) is over-approximated by \((\hat{Y}, \Box \Phi^Y)\).
- We then compute an upper bound of \((\hat{X}, \Box \Phi^X)\) and \((\hat{Y}, \Box \Phi^Y)\): using “Diagonal” Perturbation Sets (as seen in the non-constrained case).

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.
Upper Bound of Two Constrained Affine Sets

- \((\hat{X}, \Phi^X) \leq_{1 \times 2} (\hat{X}, \Box \Phi^X)\),
- \(\Box \Phi^X\) defined as the smallest box that contains \(\gamma_2(\Phi^X)\),
- Similarly for \((\hat{Y}, \Phi^Y)\) is over-approximated by \((\hat{Y}, \Box \Phi^Y)\),
- We then compute an upper bound of \((\hat{X}, \Box \Phi^X)\) and \((\hat{Y}, \Box \Phi^Y)\):
- using “Diagonal” Perturbation Sets (as seen in the non constrained case)

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.
Upper Bound of Two Constrained Affine Sets

- \((\hat{X}, \Phi^X) \leq_{1 \times 2} (\hat{X}, \Box \Phi^X)\),
- \(\Box \Phi^X\) defined as the smallest box that contains \(\gamma_2(\Phi^X)\),
- Similarly for \((\hat{Y}, \Phi^Y)\) is over-approximated by \((\hat{Y}, \Box \Phi^Y)\)
- We then compute an upper bound of \((\hat{X}, \Box \Phi^X)\) and \((\hat{Y}, \Box \Phi^Y)\):
- using “Diagonal” Perturbation Sets (as seen in the non constrained case)

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.
Upper Bound of Two Constrained Affine Sets

- \((\hat{X}, \Phi^X) \leq_{1\times2} (\hat{X}, \Box\Phi^X)\),
- \(\Box\Phi^X\) defined as the smallest box that contains \(\gamma_2(\Phi^X)\),
- Similarly for \((\hat{Y}, \Phi^Y)\) is over-approximated by \((\hat{Y}, \Box\Phi^Y)\)
- We then compute an upper bound of \((\hat{X}, \Box\Phi^X)\) and \((\hat{Y}, \Box\Phi^Y)\):
  - using “Diagonal” Perturbation Sets (as seen in the non constrained case)

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.
Upper Bound of Two Constrained Affine Sets

- $(\hat{X}, \Phi^X) \leq_{1 \times 2} (\hat{X}, \square \Phi^X)$,
- $\square \Phi^X$ defined as the smallest box that contains $\gamma_2(\Phi^X)$,
- Similarly for $(\hat{Y}, \Phi^Y)$ is over-approximated by $(\hat{Y}, \square \Phi^Y)$
- We then compute an upper bound of $(\hat{X}, \square \Phi^X)$ and $(\hat{Y}, \square \Phi^Y)$:
  - using “Diagonal” Perturbation Sets (as seen in the non constrained case)

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.
Upper Bound of Two Constrained Affine Sets

- \((\hat{X}, \Phi^X) \leq_{1 \times 2} (\hat{X}, \square \Phi^X)\),
- \(\square \Phi^X\) defined as the smallest box that contains \(\gamma_2(\Phi^X)\),
- Similarly for \((\hat{Y}, \Phi^Y)\) is over-approximated by \((\hat{Y}, \square \Phi^Y)\)
- We then compute an upper bound of \((\hat{X}, \square \Phi^X)\) and \((\hat{Y}, \square \Phi^Y)\):
- using “Diagonal” Perturbation Sets (as seen in the non constrained case)

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.
Upper Bound of Two Constrained Affine Sets

- \((\hat{X}, \Phi^X) \leq_{1 \times 2} (\hat{X}, \Box \Phi^X)\),
- \(\Box \Phi^X\) defined as the smallest box that contains \(\gamma_2(\Phi^X)\),
- Similarly for \((\hat{Y}, \Phi^Y)\) is over-approximated by \((\hat{Y}, \Box \Phi^Y)\)
- We then compute an upper bound of \((\hat{X}, \Box \Phi^X)\) and \((\hat{Y}, \Box \Phi^Y)\):
- using “Diagonal” Perturbation Sets (as seen in the non constrained case)

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.
Order Relation over Constrained Affine Forms

\[
\hat{x} = \alpha_0^x + \sum_{i=1}^{n} \alpha_i^x \epsilon_i + \beta_x^x \eta_u^x
\]

\[
\hat{y} = \alpha_0^y + \sum_{i=1}^{n} \alpha_i^y \epsilon_i + \beta_y^y \eta_u^y
\]

\[
\Delta \iff \sup_{\epsilon \in \Box} |x(\epsilon) - y(\epsilon)| \leq |\beta_y^y| - |\beta_x^x|
\]

\[
\eta_u^x \in [-1, 1]
\]

\[
\eta_u^y \in [-1, 1]
\]
Computing \( \sup_{\epsilon \in \square(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)| \)

Computing the supremum over \( \square(\Phi_\epsilon) \)

\[
\sup_{\epsilon \in \square(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)| = \delta(\alpha^x - \alpha^y \mid \text{convex}(1 \times \square(\Phi_\epsilon), -1 \times -\square(\Phi_\epsilon)))
\]

where \( \alpha^x = (\alpha^x_0, \ldots, \alpha^x_n) \), and \( \alpha^y = (\alpha^y_0, \ldots, \alpha^y_n) \).

A Particular Case: Non Constrained Case

- \( \square(\Phi_\epsilon) = [-1, 1]^n \)
- \( \text{convex}(1 \times \square(\Phi_\epsilon), -1 \times -\square(\Phi_\epsilon)) = M^X \ast B^{n+1} \)
- \( \text{convex}(1 \times \square(\Phi_\epsilon), -1 \times -\square(\Phi_\epsilon)) = [-1, 1]^{n+1} \)
- \( \delta(\alpha^x - \alpha^y \mid [-1, 1]^{n+1}) = \|\alpha^x - \alpha^y\|_1 \)

Indeed we have already seen in the non constrained case that

\[
\sup_{\epsilon} |x(\epsilon) - y(\epsilon)| = \|\alpha^x - \alpha^y\|_1 .
\]
Computing \( \sup_{\epsilon \in \square(\Phi_{\epsilon})} \| x(\epsilon) - y(\epsilon) \| \)

Computing the supremum over \( \square(\Phi_{\epsilon}^X) \)

\[
\sup_{\epsilon \in \square(\Phi_{\epsilon})} \| x(\epsilon) - y(\epsilon) \| = \delta (\alpha^x - \alpha^y \mid \text{convex}(1 \times \square(\Phi_{\epsilon}^X), -1 \times -\square(\Phi_{\epsilon}^X)))
\]

where \( \alpha^x = (\alpha_0^x, \ldots, \alpha_n^x) \), and \( \alpha^y = (\alpha_0^y, \ldots, \alpha_n^y) \).

A Particular Case: Non Constrained Case

- \( \square(\Phi_{\epsilon}^X) = [-1, 1]^n \)
- \( \text{convex}(1 \times \square(\Phi_{\epsilon}^X), -1 \times -\square(\Phi_{\epsilon}^X)) = M^X \mathcal{B}^{n+1} \)
- \( \text{convex}(1 \times \square(\Phi_{\epsilon}^X), -1 \times -\square(\Phi_{\epsilon}^X)) = [-1, 1]^{n+1} \)
- \( \delta (\alpha^x - \alpha^y \mid [-1, 1]^{n+1}) = \|\alpha^x - \alpha^y\|_1 \)

Indeed we have already seen in the non constrained case that

\[
\sup_{\epsilon} \| x(\epsilon) - y(\epsilon) \| = \|\alpha^x - \alpha^y\|_1 .
\]
Computing $\sup_{\epsilon \in \square(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)|$

Computing the supremum over $\square(\Phi_\epsilon)$

$$
\sup_{\epsilon \in \square(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)| = \delta(\alpha^x - \alpha^y \mid \text{convex}(1 \times \square(\Phi_\epsilon), -1 \times -\square(\Phi_\epsilon)))
$$

where $\alpha^x = (\alpha^x_0, \ldots, \alpha^x_n)$, and $\alpha^y = (\alpha^y_0, \ldots, \alpha^y_n)$.

A Particular Case: Non Constrained Case

- $\square(\Phi_\epsilon) = [-1, 1]^n$
- $\text{convex}(1 \times \square(\Phi_\epsilon), -1 \times -\square(\Phi_\epsilon)) = M^{X*}B^{n+1}$
- $\text{convex}(1 \times \square(\Phi_\epsilon), -1 \times -\square(\Phi_\epsilon)) = [-1, 1]^{n+1}$
- $\delta(\alpha^x - \alpha^y \mid [-1, 1]^{n+1}) = \|\alpha^x - \alpha^y\|_1$

Indeed we have already seen in the non constrained case that

$$
\sup_{\epsilon} |x(\epsilon) - y(\epsilon)| = \|\alpha^x - \alpha^y\|_1
$$
Computing $\sup_{\epsilon \in \square(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)|$

Computing the supremum over $\square(\Phi_\epsilon^X)$

$$\sup_{\epsilon \in \square(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)| = \delta(\alpha^x - \alpha^y \mid \text{convex}(1 \times \square(\Phi_\epsilon^X), -1 \times -\square(\Phi_\epsilon^X)))$$

where $\alpha^x = (\alpha^x_0, \ldots, \alpha^x_n)$, and $\alpha^y = (\alpha^y_0, \ldots, \alpha^y_n)$.

A Particular Case: Non Constrained Case

- $\square(\Phi_\epsilon^X) = [-1, 1]^n$
- $\text{convex}(1 \times \square(\Phi_\epsilon^X), -1 \times -\square(\Phi_\epsilon^X)) = M^{X*}B^{n+1}$
- $\text{convex}(1 \times \square(\Phi_\epsilon^X), -1 \times -\square(\Phi_\epsilon^X)) = [-1, 1]^{n+1}$
- $\delta(\alpha^x - \alpha^y \mid [-1, 1]^{n+1}) = \|\alpha^x - \alpha^y\|_1$

Indeed we have already seen in the non constrained case that

$$\sup_{\epsilon} |x(\epsilon) - y(\epsilon)| = \|\alpha^x - \alpha^y\|_1.$$
Computing $\sup_{\epsilon \in \Box(\Phi_{\epsilon})} |x(\epsilon) - y(\epsilon)|$

Computing the supremum over $\Box(\Phi_{\epsilon}^X)$

$$\sup_{\epsilon \in \Box(\Phi_{\epsilon})} |x(\epsilon) - y(\epsilon)| = \delta(\alpha^x - \alpha^y \ | \ \text{convex}(1 \times \Box(\Phi_{\epsilon}^X), -1 \times -\Box(\Phi_{\epsilon}^X)))$$

where $\alpha^x = (\alpha^x_0, \ldots, \alpha^x_n)$, and $\alpha^y = (\alpha^y_0, \ldots, \alpha^y_n)$.

A Particular Case: Non Constrained Case

- $\Box(\Phi_{\epsilon}^X) = [-1,1]^n$
- $\text{convex}(1 \times \Box(\Phi_{\epsilon}^X), -1 \times -\Box(\Phi_{\epsilon}^X)) = M^{X^*}B^{n+1}$
- $\text{convex}(1 \times \Box(\Phi_{\epsilon}^X), -1 \times -\Box(\Phi_{\epsilon}^X)) = [-1,1]^{n+1}$
- $\delta(\alpha^x - \alpha^y \ | \ [-1,1]^{n+1}) = \|\alpha^x - \alpha^y\|_1$

Indeed we have already seen in the non constrained case that

$$\sup_{\epsilon} |x(\epsilon) - y(\epsilon)| = \|\alpha^x - \alpha^y\|_1 .$$
Computing \( \sup_{\epsilon \in \Box(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)| \)

Computing the supremum over \( \Box(\Phi_\epsilon) \)

\[
\sup_{\epsilon \in \Box(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)| = \delta(\alpha^x - \alpha^y \mid \text{convex}(1 \times \Box(\Phi_\epsilon^X), -1 \times -\Box(\Phi_\epsilon^X)))
\]

where \( \alpha^x = (\alpha_0^x, \ldots, \alpha_n^x) \), and \( \alpha^y = (\alpha_0^y, \ldots, \alpha_n^y) \).

A Particular Case: Non Constrained Case

- \( \Box(\Phi_\epsilon^X) = [-1, 1]^n \)
- \( \text{convex}(1 \times \Box(\Phi_\epsilon^X), -1 \times -\Box(\Phi_\epsilon^X)) = M^{X^*} B^{n+1} \)
- \( \text{convex}(1 \times \Box(\Phi_\epsilon^X), -1 \times -\Box(\Phi_\epsilon^X)) = [-1, 1]^{n+1} \)
- \( \delta(\alpha^x - \alpha^y \mid [-1, 1]^{n+1}) = \|\alpha^x - \alpha^y\|_1 \)

Indeed we have already seen in the **non constrained** case that

\[
\sup_{\epsilon} |x(\epsilon) - y(\epsilon)| = \|\alpha^x - \alpha^y\|_1
\]
Join Operation

Two (Distinct in general) Sufficient Conditions for Minimality

1. $|\beta^z|$ is minimal over all upper bounds
2. $|\beta^z|$ is minimal over all upper bounds which minimize the interval concretisation of $\hat{z}$

give in general different minimal upper bound

$\Phi^a = [-1, 0] \times [0, 0.5]$
$\hat{a} = 1 - \epsilon_1 + 2\epsilon_2$
$\gamma(\hat{a}) = [1, 3]$

$\Phi^b = [-0.5, 0.5] \times [0, 1]$
$\hat{b} = 2 + \epsilon_1 + \epsilon_2$
$\gamma(\hat{b}) = [1.5, 3.5]$

Two (non trivial) Minimal Upper Bounds:

$\Phi^c = [-1, 0.5] \times [0, 1]$
$\hat{c} = 1.75 + \epsilon_2 + 0.75\eta^c_u$
$\gamma(\hat{c}) = [1, 3.5] = [1, 3] \cup [1.5, 3.5]$

$\Phi^d = [-1, 0.5] \times [0, 1]$
$\hat{d} = 1.7 + 0.2\epsilon_1 + 1.6\epsilon_2 + 0.7\eta^d_u$
$\gamma(\hat{d}) = [0.8, 4.1] \supseteq [1, 3.5]$
Which $\alpha^z$ minimizes $|\beta^z|$?

$\hat{z}$ is an upper bound:

- $\hat{x} \leq_{1\times 2} \hat{z} \iff \delta(M^X(\alpha^z - \alpha^x) | B^{n+1}) \leq |\beta^z| - |\beta^x|$
- $\hat{y} \leq_{1\times 2} \hat{z} \iff \delta(M^Y(\alpha^z - \alpha^y) | B^{n+1}) \leq |\beta^z| - |\beta^y|$

$$|\beta^z| \leq \max\{\delta(M^X(\alpha^z - \alpha^x) | B^{n+1}) + |\beta^x|, \delta(M^Y(\alpha^z - \alpha^y) | B^{n+1}) + |\beta^y|\}$$
Which $\alpha^z$ minimizes $|\beta^z|$?

$\hat{z}$ is an upper bound:

- $\hat{x} \leq_1 \hat{z} \iff \delta(M^X(\alpha^z - \alpha^x) | B^{n+1}) \leq |\beta^z| - |\beta^x|$
- $\hat{y} \leq_1 \hat{z} \iff \delta(M^Y(\alpha^z - \alpha^y) | B^{n+1}) \leq |\beta^z| - |\beta^y|

$\Rightarrow |\beta^z| \leq \max\{\delta(M^X(\alpha^z - \alpha^x) | B^{n+1}) + |\beta^x|, \delta(M^Y(\alpha^z - \alpha^y) | B^{n+1}) + |\beta^y|\}$
Which $\alpha^z$ minimizes $|\beta^z|$?

- $\hat{z}$ is an upper bound:
  - $|\beta^z| \leq \max\{\delta(M^x(\alpha^z - \alpha^x) \mid B^{n+1}) + |\beta^x|, \delta(M^y(\alpha^z - \alpha^y) \mid B^{n+1}) + |\beta^y|\}$

- $\hat{z}$ is a minimal upper bound: we minimize this maximum

**Characterization of $\alpha^z$**

- $|\beta^z|$ is a **saddle-value** of $L(\alpha, \lambda)$
- $\alpha^z$ is a **saddle-point** of $L(\alpha, \lambda)$

**$L(\alpha, \lambda)$**

$$L(\alpha, \lambda) = \lambda(\delta(M^x(\alpha - \alpha^x) \mid B^{n+1}) + |\beta^x|)$$

$$+ (1 - \lambda)(\delta(M^y(\alpha - \alpha^y) \mid B^{n+1}) + |\beta^y|),$$

where $\alpha \in \mathbb{R}^{n+1}$, and $\lambda \in [0, 1]$.  

use the subdifferential theory and the Fenchel duality.
Which $\alpha^z$ minimizes $|\beta^z|$?

- $\hat{z}$ is an upper bound:
  
  $|\beta^z| \leq \max\{\delta(M^X(\alpha^z - \alpha^x) | B^{n+1}) + |\beta^x|, \delta(M^Y(\alpha^z - \alpha^y) | B^{n+1}) + |\beta^y|\}$

- $\hat{z}$ is a minimal upper bound: we minimize this maximum

Characterization of $\alpha^z$

- $|\beta^z|$ is a saddle-value of $L(\alpha, \lambda)$
- $\alpha^z$ is a saddle-point of $L(\alpha, \lambda)$

$L(\alpha, \lambda)$

\[
L(\alpha, \lambda) = \lambda(\delta(M^X(\alpha - \alpha^x) | B^{n+1}) + |\beta^x|) \\
+ (1 - \lambda)(\delta(M^Y(\alpha - \alpha^y) | B^{n+1}) + |\beta^y|),
\]

where $\alpha \in \mathbb{R}^{n+1}$, and $\lambda \in [0, 1]$.

use the subdifferential theory and the Fenchel duality.
Saddle-Point

\[ f(x, y) = x^2 - y^2 \]
Characterization of the set of saddle-points of $L(\alpha, \lambda)$

Theorem

When $\hat{x}$ and $\hat{y}$ are non comparable, we have

- $\bar{\lambda} \in ]0, 1[$,
- $\delta(M^X(\bar{\alpha} - \alpha^x) | \mathcal{B}^{n+1}) + |\beta^x| = \delta(M^Y(\bar{\alpha} - \alpha^y) | \mathcal{B}^{n+1}) + |\beta^y|,$
- $\bar{\lambda}\delta(M^X(\bar{\alpha} - \alpha^x) | \mathcal{B}^{n+1}) + (1 - \bar{\lambda})\delta(M^Y(\bar{\alpha} - \alpha^y) | \mathcal{B}^{n+1}) = \delta(\alpha^x - \alpha^y | \bar{\lambda}M^X* \mathcal{B}^{n+1} \cap (1 - \bar{\lambda})M^Y* \mathcal{B}^{n+1}).$

where $(\bar{\alpha}, \bar{\lambda})$ denotes a saddle-point of $L.$
Complexity of Computations

\[ \hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix}, \Phi^X \cup_{1 \times 2} \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix}, \Phi^Y \]

\[ \hat{Z} = \begin{pmatrix} \hat{z}_1 \\ \vdots \\ \hat{z}_p \end{pmatrix}, \Phi^X \cup_2 \Phi^Y \]

**Complexity for each \( \hat{z}_i \):**

- \( O(n^3) \) in the worst case to compute \( |\beta^{z_i}| \),
- \( \alpha^{z_i} \) is deduced as a solution of a LP of dimension \( n + 1 \) with \( 2n + 3 \) constraints,
- Polynomial algorithm to compute a **minimal upper bound** with the least perturbation.
Other Join Variants

Two Sufficient Conditions for Minimality

1. \(|\beta^z|\) is minimal over all upper bounds (\(\bigcup_{1\times2}\))
2. \(|\beta^z|\) is minimal over all upper bounds which minimize the interval concretisation of \(\hat{z}\) (\(\bigcup_{1\times2}\))

\(\bigcup_{1\times2}\)
- Computes a minimal upper bound
- Linear Complexity
- May lose “many” noise symbols

\(\bigcup_{1\times2}\) (weaker version of \(\bigcup_{1\times2}\))
- Linear Complexity
- Lose less noise symbols
- Computes an upper bound in general (but may give the minimal upper bound returned by \(\bigcup_{1\times2}\))
Other Join Variants

Two Sufficient Conditions for Minimality

1. $|\beta^z|$ is minimal over all upper bounds ($\bigcup_{1 \times 2}$)
2. $|\beta^z|$ is minimal over all upper bounds which minimize the interval concretisation of $\hat{z}$ ($\bigcup_{1 \times 2}$)

$\bigcup_{1 \times 2}$

+ Computes a **minimal upper bound**
+ Linear Complexity
- May lose “many” noise symbols

$\bigcup_{1 \times 2}$ (weaker version of $\bigcup_{1 \times 2}$)

+ Linear Complexity
+ Lose less noise symbols
± Computes an upper bound in general (but may give the minimal upper bound returned by $\bigcup_{1 \times 2}$)
Other Join Variants

Two Sufficient Conditions for Minimality

1. $|\beta^z|$ is minimal over all upper bounds ($\bigcup_{1 \times 2}$)
2. $|\beta^z|$ is minimal over all upper bounds which minimize the interval concretisation of $\hat{z}$ ($\bigcup_{1 \times 2}$)

$\bigcup_{1 \times 2}$

+ Computes a **minimal upper bound**
+ Linear Complexity
  - May lose “many” noise symbols

$\bigcup_{1 \times 2}$ (weaker version of $\bigcup_{1 \times 2}$)

+ Linear Complexity
+ Lose less noise symbols
+ Computes an upper bound in general (but may give the minimal upper bound returned by $\bigcup_{1 \times 2}$)
Example 1

Minimal Upper Bounds

\[ \Phi^a = [-1, 0] \times [0, 0.5] \]
\[ \hat{a} = 1 - \epsilon_1 + 2\epsilon_2 \]
\[ \gamma(\hat{a}) = [1, 3] \]

\[ \Phi^b = [-0.5, 0.5] \times [0, 1] \]
\[ \hat{b} = 2 + \epsilon_1 + \epsilon_2 \]
\[ \gamma(\hat{b}) = [1.5, 3.5] \]

Two (non trivial) Minimal Upper Bounds:

\[ \bigcup_{1 \times 2} \bigcup_{1 \times 2} \]
\[ \Phi^c = [-1, 0.5] \times [0, 1] \]
\[ \hat{c} = 1.75 + \epsilon_2 + 0.75\eta^c_u \]
\[ \gamma(\hat{c}) = [1, 3.5] = [1, 3] \cup [1.5, 3.5] \]

\[ \bigcup_{1 \times 2} \]
\[ \Phi^d = [-1, 0.5] \times [0, 1] \]
\[ \hat{d} = 1.7 + 0.2\epsilon_1 + 1.6\epsilon_2 + 0.7\eta^d_u \]
\[ \gamma(\hat{d}) = [0.8, 4.1] \supseteq [1, 3.5] \]
Example 2

Minimal Upper Bounds

\[
\Phi^a = [-1, 0] \times [0, 0.5] \\
\hat{a} = -2\epsilon_1 + \epsilon_2 \\
\gamma(\hat{a}) = [0, 2.5]
\]

\[
\Phi^b = [-0.5, 0.5] \times [0, 1] \\
\hat{b} = -2\epsilon_1 + \epsilon_2 \\
\gamma(\hat{b}) = [-1, 2]
\]

Two Minimal Upper Bounds:

\[
\Phi^c = [-1, 0.5] \times [0, 1] \\
\hat{c} = 0.25 + \epsilon_2 + 1.25\eta_u^c \\
\gamma(\hat{c}) = [-1, 2.5]
\]

\[
\Phi^d = [-1, 0.5] \times [0, 1] \\
\hat{d} = -2\epsilon_1 + \epsilon_2 \\
\gamma(\hat{d}) = [-1, 3]
\]

An Upper Bound:

\[
\Phi^e = [-1, 0.5] \times [0, 1] \\
\hat{e} = 0.75 + 1.75\eta_u^c \\
\gamma(\hat{e}) = [-1, 2.5] = [0, 2.5] \cup [-1, 2]
\]
$\cup_{1 \times 2} \, \text{vs} \, \cup_{1 \times 2} \, \text{vs} \, \cup_{1 \times 2}$

\[ \hat{X} = \begin{cases} \Phi^X = [-1, 0] \times [0, 0.5] \\ \hat{x}_1 = 1 - \epsilon_1 + 2\epsilon_2 \\ \hat{x}_2 = -2\epsilon_1 + \epsilon_2 \end{cases} \]

\[ \hat{Y} = \begin{cases} \Phi^Y = [-0.5, 0.5] \times [0, 1] \\ \hat{y}_1 = 2 + \epsilon_1 + \epsilon_2 \\ \hat{y}_2 = -2\epsilon_1 + \epsilon_2 \end{cases} \]
\[ \Phi^Z = [-0.5, 1] \times [0, 1] \]

\[ \begin{align*}
\hat{z}_1 &= 1.75 + \epsilon_2 + 0.75\eta_{u}^{Z_1} \\
\hat{z}_2 &= 0.75 + 1.75\eta_{u}^{Z_2}
\end{align*} \]
\[ \Phi^Z = [-0.5, 1] \times [0, 1] \]

\[ \begin{aligned} \hat{\mathbf{z}}_1 &= 1.75 + \epsilon_2 + 0.75 \eta_{u}^{z_1} \\ \hat{\mathbf{z}}_2 &= 0.75 + 1.75 \eta_{u}^{z_2} \end{aligned} \]
\[ \mathbb{U}_{1 \times 2} \text{ vs } \mathbb{U}_{1 \times 2} \text{ vs } \mathbb{U}_{1 \times 2} \]

\[ \phi^Z = [-0.5, 1] \times [0, 1] \]

\[ \mathbb{U}_{1 \times 2} \quad \left\{ \begin{array}{l}
\hat{z}_1 = 1.75 + \epsilon_2 + 0.75 \eta_{z1} \\
\hat{z}_2 = 0.75 + 1.75 \eta_{z2}
\end{array} \right. \]

\[ \mathbb{U}_{1 \times 2} \quad \left\{ \begin{array}{l}
\hat{z}_1 = 0.75 + \epsilon_2 + 0.75 \eta_{z1} \\
\hat{z}_2 = 0.25 + \epsilon_2 + 1.25 \eta_{z2}
\end{array} \right. \]

\[ \mathbb{U}_{1 \times 2} \quad \left\{ \begin{array}{l}
\hat{z}_1 = 1.7 + 0.2 \epsilon_1 + 1.6 \epsilon_2 + 0.7 \eta_{z1} \\
\hat{z}_2 = -2 \epsilon_1 + \epsilon_2
\end{array} \right. \]
\( \cup_{1 \times 2} \quad \cup_{1 \times 2} \quad \cup_{1 \times 2} \)

\( \Phi^Z = [-0.5, 1] \times [0, 1] \)

\( \hat{z}_1 = 1.75 + \epsilon_2 + 0.75 \eta^{z_1}_u \)
\( \hat{z}_2 = 0.75 + 1.75 \eta^{z_2}_u \)

\( \hat{z}_1 = 1.75 + \epsilon_2 + 0.75 \eta^{z_1}_u \)
\( \hat{z}_2 = 0.25 + \epsilon_2 + 1.25 \eta^{z_2}_u \)

\( \hat{z}_1 = 1.7 + 0.2 \epsilon_1 + 1.6 \epsilon_2 + 0.7 \eta^{z_1}_u \)
\( \hat{z}_2 = -2 \epsilon_1 + \epsilon_2 \)

We use in addition a Reduced Product with Intervals.
Interpretation of Tests

\[ \hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix}, \Phi^X \]

\{ \hat{x}_i = \hat{x}_j, \hat{x}_i \leq 0, \hat{x}_j \times \hat{x}_i \leq \hat{x}_k \} 

\hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix}, \Phi^Y

\Delta \iff \Phi^Y = \llbracket \mathrm{cons} \rrbracket^\Pi_2 (\Phi^X)

\forall i, 1 \leq i \leq p, \ \hat{y}_i = \hat{x}_i
Interpretation of Equality Tests

Example

\[ \hat{X} = \begin{pmatrix} \hat{x}_1 = 4 + \epsilon_1 + \epsilon_2 + \epsilon_3 \\ \hat{x}_2 = -\epsilon_1 + 3\epsilon_2 \\ \hat{x}_3 = -\epsilon_1 + 2\epsilon_2 + \epsilon_3 \end{pmatrix}, \Phi^X = [-1, 1]^3 \quad \{\hat{x}_1 == \hat{x}_2\} \quad \hat{Y} = ? \]

- \(\hat{x}_1 == \hat{x}_2 \iff 4 + 2\epsilon_1 - 2\epsilon_2 + \epsilon_3 == 0\),
- \(\Phi^Y = \lceil 4 + 2\epsilon_1 - 2\epsilon_2 + \epsilon_3 == 0 \rceil_2(\Phi^X) = [-1, -0.5] \times [0.5, 1] \times [-1, 0]\)
- \(\hat{y}_i\) is extracted from \(\hat{x}_i\) such that the interval concretisation of \(\hat{y}_i\) is minimal.
Interpretation of Equality Tests

Example

\[ \hat{X} = \begin{pmatrix} \hat{x}_1 = 4 + \epsilon_1 + \epsilon_2 + \epsilon_3 \\ \hat{x}_2 = -\epsilon_1 + 3\epsilon_2 \\ \hat{x}_3 = -\epsilon_1 + 2\epsilon_2 + \epsilon_3 \end{pmatrix}, \Phi^X = [-1, 1]^3 \quad \{\hat{x}_1 == \hat{x}_2\} \quad \hat{Y} = ? \]

- \( \hat{x}_1 == \hat{x}_2 \iff 4 + 2\epsilon_1 - 2\epsilon_2 + \epsilon_3 == 0 \),
- \( \Phi^Y = \left[4 + 2\epsilon_1 - 2\epsilon_2 + \epsilon_3 == 0\right]_2(\Phi^X) = [-1, -0.5] \times [0.5, 1] \times [-1, 0] \)
- \( \hat{y}_i \) is extracted from \( \hat{x}_i \) such that the interval concretisation of \( \hat{y}_i \) is minimal
Interpretation of Equality Tests

Example

\[ \hat{X} = \begin{pmatrix} \hat{x}_1 = 4 + \epsilon_1 + \epsilon_2 + \epsilon_3 \\ \hat{x}_2 = -\epsilon_1 + 3\epsilon_2 \\ \hat{x}_3 = -\epsilon_1 + 2\epsilon_2 + \epsilon_3 \end{pmatrix}, \Phi^X = [-1, 1]^3 \{\hat{x}_1 == \hat{x}_2\} \hat{Y} = ? \]

- \( \hat{x}_1 == \hat{x}_2 \iff 4 + 2\epsilon_1 - 2\epsilon_2 + \epsilon_3 == 0, \)
- \( \Phi^Y = \llbracket 4 + 2\epsilon_1 - 2\epsilon_2 + \epsilon_3 == 0 \rrbracket_2(\Phi^X) = [-1, -0.5] \times [0.5, 1] \times [-1, 0] \)
- \( \hat{y}_i \) is extracted from \( \hat{x}_i \) such that the interval concretisation of \( \hat{y}_i \) is minimal
Interpretation of Equality Tests

Example Con't

\[ \hat{y}_1 = 2 + 2\epsilon_2 + 0.5\epsilon_3, \quad \text{bound}_2(\hat{Y}_1, \Phi^Y) = [2.5, 4] \]  
(by substituting \( \epsilon_1 \))

\[ \hat{y}_1 = 6 + 2\epsilon_1 + 1.5\epsilon_1, \quad \text{bound}_2(\hat{Y}_1, \Phi^Y) = [2.5, 5] \]  
(by substituting \( \epsilon_2 \))

\[ \hat{y}_1 = -\epsilon_1 + 3\epsilon_2, \quad \text{bound}_2(\hat{Y}_1, \Phi^Y) = [2, 4] \]  
(by substituting \( \epsilon_3 \))

\[ \Phi^Y := [-1, -0.5] \times [0.5, 1] \times [-1, 0] \]

\[ \hat{y}_1 := 2 + 2\epsilon_2 + 0.5\epsilon_3, \quad \text{bound}_2(\hat{Y}_1, \Phi^Y) = [2.5, 4] \]

\[ \hat{y}_2 := 2 + 2\epsilon_2 + 0.5\epsilon_3, \quad \text{bound}_2(\hat{Y}_2, \Phi^Y) = [2.5, 4] \]

\[ \hat{y}_3 := 2 + \epsilon_2 + 1.5\epsilon_3, \quad \text{bound}_2(\hat{Y}_3, \Phi^Y) = [1, 3] \]
Interpretation of Equality Tests

Example Con't

\[ \hat{y}_1 = 2 + 2\epsilon_2 + 0.5\epsilon_3, \quad \text{bound}_2(\hat{Y}_1, \Phi^Y) = [2.5, 4] \quad \text{(by substituting \( \epsilon_1 \))} \]

\[ \hat{y}_1 = 6 + 2\epsilon_1 + 1.5\epsilon_1, \quad \text{bound}_2(\hat{Y}_1, \Phi^Y) = [2.5, 5] \quad \text{(by substituting \( \epsilon_2 \))} \]

\[ \hat{y}_1 = -\epsilon_1 + 3\epsilon_2, \quad \text{bound}_2(\hat{Y}_1, \Phi^Y) = [2, 4] \quad \text{(by substituting \( \epsilon_3 \))} \]

\[ \Phi^Y := [-1, -0.5] \times [0.5, 1] \times [-1, 0] \]

\[ \hat{y}_1 := 2 + 2\epsilon_2 + 0.5\epsilon_3, \quad \text{bound}_2(\hat{Y}_1, \Phi^Y) = [2.5, 4] \]

\[ \hat{y}_2 := 2 + 2\epsilon_2 + 0.5\epsilon_3, \quad \text{bound}_2(\hat{Y}_2, \Phi^Y) = [2.5, 4] \]

\[ \hat{y}_3 := 2 + \epsilon_2 + 1.5\epsilon_3, \quad \text{bound}_2(\hat{Y}_3, \Phi^Y) = [1, 3] \]
Interpretation of Equality Tests

Example Con't

Properties

- for each $i$, we solve the above problem with an average complexity of $O(n \log(n))$,
- the equality constraint is \textbf{algebraically satisfied} in $\hat{Y}$: $\hat{y}_1 = \hat{y}_2$,
- the concretisation of each $\hat{y}_i$ is optimal.
Assignment - Widening

Assignment

- $\Phi$ is unchanged
- For linear expressions, we simply use affine arithmetic as in Perturbed Affine Sets ($\Phi$ is unused)
- For non-linear operations, $\Phi$ is used to improve the linearization of non-linear terms

Widening

We use the same widening as in Perturbed Affine Sets: losing the relations encoded by noise symbols.
Outlines

1. Static Analysis-based Abstract Interpretation
2. Affine Sets Abstract Domain
3. Constrained Affine Sets Abstract Domain
4. Experiments, Taylor1+
5. Appendix
Taylor1+ Features

- analyses programs with **real number semantics**, 
- APRON [B. Jeannet, A.Miné, SAS07] like abstract domain (level 0), 
- written in C and offers an OCAMl interface, 
- linked to interproc [B. Jeannet], 
- uses **double-precision floating-point numbers** in a **sound** manner for computations in abstract domain (supports also GMP and MPFR), 
- Noise symbols abstract domains may be any APRON like abstract domain.
Unrolled scheme for the 2\textsuperscript{nd} order filter

\textbf{2\textsuperscript{nd} order filter}

- \( S_n = 0.7E_n - 1.3E_{n-1} + 1.1E_{n-2} + 1.4S_{n-1} - 0.7S_{n-2} \)
- Poles are inside the unit circle (norm close to 0.84)
### Fixpoint Computation

<table>
<thead>
<tr>
<th>filter o2</th>
<th>fixpoint</th>
<th>t(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boxes</td>
<td>⊤</td>
<td>$6 \times 10^{-3}$</td>
</tr>
<tr>
<td>Octagons</td>
<td>⊤</td>
<td>0.19</td>
</tr>
<tr>
<td>Polyhedra</td>
<td>$[-1.30, 2.82]$</td>
<td>0.49</td>
</tr>
<tr>
<td>Taylor1+</td>
<td>$[-5.40, 7.07]$</td>
<td>0.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>filter o8</th>
<th>fixpoint</th>
<th>t(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boxes</td>
<td>⊤</td>
<td>0.01</td>
</tr>
<tr>
<td>Octagons</td>
<td>⊤</td>
<td>21</td>
</tr>
<tr>
<td>Polyhedra</td>
<td>abort</td>
<td>&gt;24h</td>
</tr>
<tr>
<td>Taylor1+</td>
<td>$[-3.81, 4.81]$</td>
<td>0.5</td>
</tr>
</tbody>
</table>
3\textsuperscript{rd} order Householder Iteration Scheme

Inverse of the square root

- $h_n = 1 - Ax_n^2$, $A \in [16, 20]$ and $x_0 = 2^{-4}$

$$x_{n+1} = x_n + x_n \left( \frac{1}{2} h_n + \frac{3}{8} h_n^2 \right)$$

<table>
<thead>
<tr>
<th>Unrolling (5 It.)</th>
<th>$\sqrt{A} = Ax_n$</th>
<th>t(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boxes</td>
<td>[0.51, 8.44]</td>
<td>$1 \times 10^{-4}$</td>
</tr>
<tr>
<td>Octagons</td>
<td>[0.51, 7.91]</td>
<td>0.01</td>
</tr>
<tr>
<td>Polyhedra</td>
<td>[2.22, 6.56]</td>
<td>310</td>
</tr>
<tr>
<td>T.1+</td>
<td>[3.97, 4.51]</td>
<td>$1 \times 10^{-3}$</td>
</tr>
<tr>
<td>• 10 subdivisions</td>
<td>[4.00, 4.47]</td>
<td>0.02</td>
</tr>
<tr>
<td>• SDP</td>
<td>[3.97, 4.51]</td>
<td>0.16</td>
</tr>
</tbody>
</table>
Benchmarks

- **InterQ1**: linear tests with quadratic expressions
- **Cosine**: piecewise 3rd order polynomial interpolation of the cosine function
- **SinCos**: sum of the squares of the sine and cosine functions
- **InterL2** (resp. **InterQ2**): the inverse image of 1 by a piecewise affine (resp. quadratic) function

<table>
<thead>
<tr>
<th></th>
<th>Exact</th>
<th>Octagons</th>
<th>Polyhedra</th>
<th>Taylor1+</th>
<th>Cons. Taylor1+ $(\cup_{1} \times 2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>InterQ1</td>
<td>[0, 1875]</td>
<td>[−3750, 6093]</td>
<td>[−2578, 4687]</td>
<td>[0, 2500]</td>
<td>[0, 1875]</td>
</tr>
<tr>
<td>Cosine</td>
<td>[−1, 1]</td>
<td>[−1.50, 1.0]</td>
<td>[−1.50, 1.0]</td>
<td>[−1.073, 1]</td>
<td>[−1, 1]</td>
</tr>
<tr>
<td>SinCos</td>
<td>{1}</td>
<td>[0.84, 1.15]</td>
<td>[0.91, 1.07]</td>
<td>[0.86, 1.15]</td>
<td>[0.99, 1.00]</td>
</tr>
<tr>
<td>InterL2</td>
<td>{0.1}</td>
<td>[−1, 1]</td>
<td><strong>[0.1, 0.4]</strong></td>
<td>[−1, 1]</td>
<td>[0.1, 1]</td>
</tr>
<tr>
<td>InterQ2</td>
<td>{0.36}</td>
<td>[−1, 1]</td>
<td>[−0.8, 1]</td>
<td>[−1, 1]</td>
<td>[−0.4, 1]</td>
</tr>
<tr>
<td>InterQ2b</td>
<td>[−0.1, 3]</td>
<td>[−3, 27]</td>
<td>[−3, 27]</td>
<td>[−0.1, 27]</td>
<td>[−0.1, 3.77]</td>
</tr>
</tbody>
</table>
Does the domain scale up?

\[ g(x) = \frac{\sqrt{x^2 - x + 0.5}}{\sqrt{x^2 + 0.5}} \]

\[ x = [-2, 2]; \]

/* for n subdivisions */

\[ h = \frac{4}{n}; \]

if (\(-x <= h-2\))

\[ y = g(x); z = g(y); \]

...

/* 2 <= i <= n-1 */

else if (\(-x <= i*h-2\))

\[ y = g(x); z = g(y); \]

...

else

\[ y = g(x); z = g(y); \]
Taylor1+ Scales up

CPU time (s)

width of $g(g(x))$

# subdivisions (constraints)

# subdivisions (constraints)
## Comparison of Join Variants

<table>
<thead>
<tr>
<th></th>
<th>Exact</th>
<th>Taylor1+</th>
<th>Cons. Taylor1+ $(\sqcup 1 \times 2)$</th>
<th>Cons. Taylor1+ $(\sqcup 1 \times 2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>InterQ1</td>
<td>[0, 1875]</td>
<td>[0, 2500]</td>
<td>[0, 1875]</td>
<td>[0, 1875]</td>
</tr>
<tr>
<td>Cosine</td>
<td>[−1, 1]</td>
<td>[−1.073, 1]</td>
<td>[−1, 1]</td>
<td>[−1, 1]</td>
</tr>
<tr>
<td>SinCos</td>
<td>{1}</td>
<td>[0.86, 1.15]</td>
<td>[0.99, 1.00]</td>
<td>[0.99, 1.00]</td>
</tr>
<tr>
<td>InterL2</td>
<td>{0.1}</td>
<td>[−1, 1]</td>
<td>[0.1, 1]</td>
<td>[0.066, 0.4]</td>
</tr>
<tr>
<td>InterQ2</td>
<td>{0.36}</td>
<td>[−1, 1]</td>
<td>[−0.4, 1]</td>
<td>[−0.29, 0.52]</td>
</tr>
<tr>
<td>InterQ2b</td>
<td>[−0.1, 3]</td>
<td>[−0.1, 27]</td>
<td>[−0.1, 3.77]</td>
<td>[−0.1, 3.77]</td>
</tr>
</tbody>
</table>
Scalability of Join Operators

![Graph showing CPU time vs. number of noise symbols (n + m)]

- `∪[1x2]` is represented by circles.
- `∪[1x2]` is represented by plusses.

K. Ghorbal (CMU)
Perturbation and Lost Noise Symbols

\[
\begin{align*}
\text{Perturbation} &\quad \# \text{noise symbols} (n) \\
\circ \ & [\cup_1 \times 2] \\
\oplus \ & [\cup_1 \times 2]
\end{align*}
\]

\[
\begin{align*}
\text{remaining noise symbols} &\quad \text{noise symbols} (n) \\
\circ \ & [\cup_1 \times 2] \\
\oplus \ & [\cup_1 \times 2]
\end{align*}
\]
Conclusion

Future Directions

- Using **non linear templates** abstract domain to abstract noise symbols (back to the quaternion normalization problem, we can use the abstract domain [Adjé,Gaubert,Goubault] for noise symbols).

- **Abstract the coefficients** to catch some specific non-convex (disjunctive) properties.

- better global join than the diagonal relaxation (we have already promising results).
Thanks for your attention!
Kolev Multiplication

- no overestimation if certain simple monotonicity conditions are valid [Kolev 2007].
- However, the affine form obtained is not always correct when dealing with future evaluations.

Example

- $\hat{x} = 10 + 5\epsilon_1 + 3\epsilon_2$ and $\hat{y} = 10 - 2\epsilon_1 + \epsilon_3$,
- Kolev multiplication gives $\hat{z} = 92 + 31\epsilon_1 + 21\epsilon_2 + 2\epsilon_3 + 16\epsilon_4$.
- $\gamma(z)$ is $[22, 162]$ which is the exact range of $xy$.
- for $t = -4x + 0.8z - 79$,
- $\hat{t} = -45.4 + 4.8\epsilon_1 + 4.8\epsilon_2 + 1.6\epsilon_3 + 12.8\epsilon_4$,
- and $\gamma(t) \in [-69.4, -21.4]$.
- for $\epsilon_1 = 0$ and $\epsilon_2 = 1$ and $\epsilon_3 = 1$,
- $x = 13$ and $y = 11$ and $z = 143$, then $t = -16.6 \notin \gamma(t)$. 
Lemma [Gaubert 2006]

\[
\max_{|\epsilon_i| \leq 1} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i^x \alpha_j^y \epsilon_i \epsilon_j = \max_{|\epsilon_i| \leq 1} \epsilon^t \Phi \epsilon \leq \inf_{\mu \in \mathbb{R}_+^n} \{ \text{trace}(\mu I_n) | \Phi - \mu I_n \preceq 0 \} \quad (1)
\]

where

- \((\phi_{i,j})_{1 \leq i,j \leq n} = \frac{1}{2}(\alpha_i^x \alpha_j^y + \alpha_j^x \alpha_i^y) \)

- \(M \preceq 0\) (\(M\) is negative semidefinite)

The equality holds when matrix \(\Phi\) is negative semidefinite. The right hand side of (1) is a typical SDP problem.
using SDP throw an example

Let \( \hat{x} = 10 + 5\epsilon_1 + 3\epsilon_2 \) and \( \hat{y} = 10 - 2\epsilon_1 + \epsilon_3 \), then
\[
\hat{z} = \hat{x} \times \hat{y} = 100 + 30\epsilon_1 + 30\epsilon_2 + 10\epsilon_3 + q(\epsilon),
\]
where \( q(\epsilon) = \epsilon^t Q \epsilon \) and
\[
Q = \begin{pmatrix}
-10 & -3 & 2.5 \\
-3 & 0 & 1.5 \\
2.5 & 1.5 & 0
\end{pmatrix}
\]

SDP problems to solve are:

1) \( M = \min \mu.1_n \) \( s.t \) \( \mu l_n - Q \) \begin{pmatrix} 0 \\ 0 \end{pmatrix} \succeq 0 

2) \( -m = \min \mu.1_n \) \( s.t \) \( \mu l_n - (-Q) \) \begin{pmatrix} 0 \\ 0 \end{pmatrix} \succeq 0
The final invariant of InterQ2