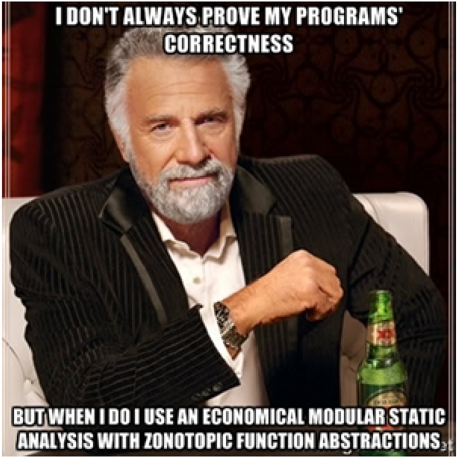
Zonotopic abstract domains for numerical program analysis

Sylvie Putot
Joint work with Eric Goubault
LIX, Ecole Polytechnique

VMCAI Winter School, January 11, 2019

ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

Programme of the talk



Zonotopes = a swiss knife of numerical program analysis ?

Outline of the talk

Zonotopes as a general purpose numerical abstract domain

- Inspired from guaranteed numerical methods (affine arithmetic, Taylor methods)
- A functional abstraction: parametrized sub-polyhedral abstraction with low complexity (allows modular analysis, test generation, etc)
- Set operations and a word on fixpoint computations

Good at expressing (and propagationg) perturbations

- Used for assessing safety and robustness of neural networks (ETH Zurich)
- Finite precision accuracy and robustness analysis (Fluctuat analyzer)

Possible extensions

- An interesting representation of ellipsoids ?
- Under-approximations
- Mixed non-deterministic and probabilistic analysis

Example: Householder scheme for square root approx

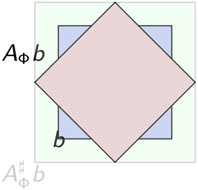
Householder

Interval (boxes) abstraction and numerical abstract domains

Consider the rotation $A_\Phi b$ of an initial box $b = [-1,1] \times [-1,1]$, with

$$A_\Phi = \begin{matrix} \cos\Phi & \sin\Phi \\ -\sin\Phi & \cos\Phi \end{matrix}$$

The initial box is b, its exact image by the rotation is $A_\Phi b$, and the best interval abstraction $A^\sharp_\Phi b$



- A typical example of the wrapping effect of the interval abstraction.
- Many abstract domains aim at good compromise between cost and precision: linear equalities, polyhedra, congruences, zones, octagons, templates, ellipsoids, gauges, parallelotopes, etc
- Often combined (reduced product)

Affine Arithmetic (Comba & Stolfi 93) for real-numbers abstraction

Affine forms

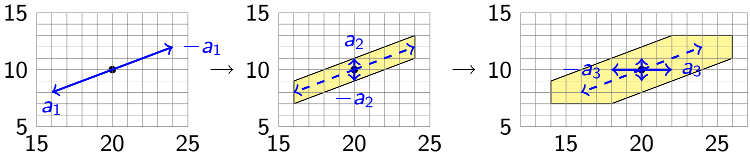
- Affine form for variable x:

$$\hat{x} = x_0 + x_1\varepsilon_1 + \ldots + x_n\varepsilon_n, \; x_i \in \mathbb{R}$$

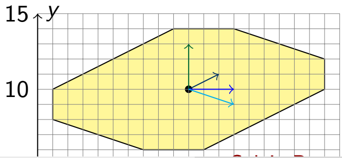
  where the ε_i are symbolic variables (*noise symbols*), with value in $[-1, 1]$.

- Sharing ε_i between variables expresses implicit dependency

Geometric concretization as zonotopes (center symmetric polytopes, huge literature)



$$\begin{aligned} \hat{x} &= 20 - 4\varepsilon_1 \qquad + 2\varepsilon_3 + 3\varepsilon_4 \\ \hat{y} &= 10 - 2\varepsilon_1 + \varepsilon_2 \qquad - \varepsilon_4 \end{aligned}$$



Affine arithmetic

- Assignment $x := [a, b]$ introduces a noise symbol:

$$\hat{x} = \frac{(a+b)}{2} + \frac{(b-a)}{2}\,\varepsilon_i.$$

- Addition/subtraction are exact:

$$\hat{x} + \hat{y} = (x_0 + y_0) + (x_1 + y_1)\varepsilon_1 + \ldots + (x_n + y_n)\varepsilon_n$$

- Non linear operations : approximate linear form, new noise term bounding the approximation error

$$\hat{x} \times \hat{y} = x_0 y_0 + \sum_{i=0}^{n} (x_0 y_i + x_i y_0)\varepsilon_i + \left(\sum_{1 \le i \ne j \le n} |\, x_i y_j \,| \right) \varepsilon_{n+1}$$

(better approximations possible)

- Close to Taylor models of low degree : low time complexity! and easy to implement on a finite-precision machine

Example (transformers are exact for affine operations)

Consider, with $a \in [-1, 1]$ and $b \in [-1, 1]$, the expressions

```
x = 1 + a + 2 * b;
y = 2 - a;
z = x + y - 2 * b;
```

- The representation as affine forms is $\hat{x} = 1 + \varepsilon_1 + 2\varepsilon_2$, $\hat{y} = 2 - \varepsilon_1$, with noise symbols $\varepsilon_1, \varepsilon_2 \in [-1, 1]$
- This implies $\hat{x} \in [-2, 4]$, $\hat{y} \in [1, 3]$ (same as Interval Arithmetic)
- It also contains implicit relations, such as $\hat{x} + \hat{y} = 3 + 2\epsilon_2 \in [1, 5]$ or $\boxed{\hat{z} = \hat{x} + \hat{y} - 2b = 3}$
- Whereas we get with intervals

$$\boxed{z = x + y - 2b \in [-3, 9]}$$

Taylor models

Very appealing model...part of a bigger picture

Taylor models approximate variables values by polynomial plus remainder:

$$f(x_1, \ldots, x_n) = f(0) + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(0) x_i + \ldots$$

Taylor models

Very appealing model...part of a bigger picture

Taylor models approximate variables values by polynomial plus remainder:

$$f(x_1, \ldots, x_n) = f(0) + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(0) x_i + \sum_{i,j=1}^{n} \frac{1}{2} \frac{\partial^2 f}{\partial x_i \partial x_j}(0) x_i x_j + \ldots$$

Taylor models

Very appealing model...part of a bigger picture

Taylor models approximate variables values by polynomial plus remainder:

$$\hat{x} = x_0 + \sum_{i=1}^{n} x_i \varepsilon_i + \sum_{i,j=1}^{n} x_{i,j} \varepsilon_i \varepsilon_j + [R]$$

(quadratic zonotopes Adje et al. 2015, etc)

Taylor models

Very appealing model...part of a bigger picture

Taylor models approximate variables values by polynomial plus remainder:

$$\hat{x} = x_0 + \sum_{i=1}^{n} x_i \varepsilon_i + \sum_{i,j=1}^{n} x_{i,j} \varepsilon_i \varepsilon_j + [R]$$

(quadratic zonotopes Adje et al. 2015, etc)

Both zonotopes and Taylor models are very successfully used in hybrid system reachability analysis

Numerical abstract domain (in short...) when not a lattice

Concretization-based analysis

- Machine-representable abstract values X (affine sets)
- A concretization function γ_f defining the set of concrete values represented by an abstract value
- A partial order on these abstract values, induced by γ_f:
  $X \sqsubseteq Y \iff \gamma_f(X) \subseteq \gamma_f(Y)$

Abstract transfer functions

- Arithmetic operations: F is a sound abstraction of f iff

$$\forall x \in \gamma_f(X),\ f(x) \in \gamma_f(F(X))$$

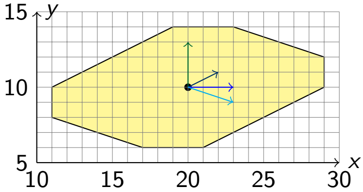
- Set operations: join ($\cup$), meet ($\cap$), widening
  - no least upper bound / greatest lower bound on affine sets
  - (minimal) upper bounds / over-approximations of the intersection ...

and ... hopefully accurate and effective to compute!!!

Concretization and order structure?

$$
\begin{aligned}
x &= 20 - 4\varepsilon_1 + 2\varepsilon_3 + 3\varepsilon_4 \\
y &= 10 - 2\varepsilon_1 + \varepsilon_2 - \varepsilon_4
\end{aligned}
$$

$$
\begin{pmatrix} x \\ y \end{pmatrix} = A^T \begin{pmatrix} \varepsilon_0 = 1 \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \end{pmatrix} \quad A = \begin{pmatrix} 20 & 10 \\ -4 & -2 \\ 0 & 1 \\ 2 & 0 \\ 3 & -1 \end{pmatrix}
$$



$$\gamma(A) = \{A^T \varepsilon \mid \|\varepsilon\|_\infty \leq 1\}$$

"Geometric" order

$$A \leq B \Leftrightarrow \gamma(A) \leq \gamma(B)$$

For centered zonotopes: $A \leq B$ iff for all $t \in \mathbb{R}^p$, $\|At\|_1 \leq \|Bt\|_1$

Functional order ?

Parameterization...is almost input-output relationship?

$$x = 20 - 4\varepsilon_1 + 2\varepsilon_3 + 3\varepsilon_4$$

Two kinds of noise symbols

- Input noise symbols (ε_i): created by uncertain inputs
- Perturbation noise symbols (η_j): created by uncertainty in analysis

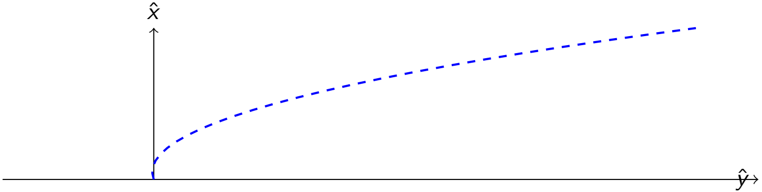
Affine sets $X = (C^X, P^X)$

$$\begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \dots \\ \hat{x}_p \end{pmatrix} = {C^X}^T \begin{pmatrix} 1 \\ \varepsilon_1 \\ \dots \\ \varepsilon_n \end{pmatrix} + {P^X}^T \begin{pmatrix} \eta_1 \\ \eta_2 \\ \dots \\ \eta_m \end{pmatrix}$$

- Central part links the current values of the program variables to the initial values of the input variables (linear functional)
- Perturbation part encodes the uncertainty in the description of values of program variables due to non-linear computations (multiplication, join etc.)
- Practical use for modular static analysis, test generation

A simple example: functional interpretation

```
real x = [0,10];
real y = x*x - x;
```

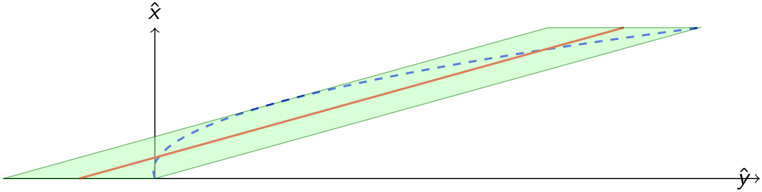


Abstraction of x: $x = 5 + 5\varepsilon_1$
Abstraction of function $x \to y = x^2 - x$ as

$$y = 32.5 + 50\varepsilon_1 + 12.5\eta_1$$

A simple example: functional interpretation

```
real x = [0,10];
real y = x*x - x;
```



Abstraction of x: $x = 5 + 5\varepsilon_1$
Abstraction of function $x \to y = x^2 - x$ as

$$\begin{aligned} y &= 32.5 + 50\varepsilon_1 + 12.5\eta_1 \\ &= -17.5 + 10x + 12.5\eta_1 \end{aligned}$$

Functional order relation

Want an order that preserves the parametrization as input-output relationships.

Concretization in terms of sets of functions from $\mathbb{R}^n$ to $\mathbb{R}^p$:

$$\gamma_f(X) = \left\{ f : \mathbb{R}^n \to \mathbb{R}^p \mid \forall \epsilon \in [-1,1]^n, \exists \eta \in [-1,1]^m, f(\varepsilon) = {C^X}^T \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix} + {P^X}^T \eta \right\}.$$

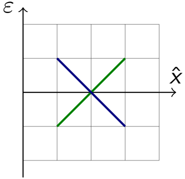
- $\gamma_f(X) \subseteq \gamma_f(Y)$ equivalent to

$$X \sqsubseteq Y \iff \forall u \in \mathbb{R}^p,\ \|(C^Y - C^X)u\|_1 \leq \|P^Y u\|_1 - \|P^X u\|_1$$

  (implies the geometric ordering $\|C^X u\|_1 + \|P^X u\|_1 \leq \|C^Y u\|_1 + \|P^Y u\|_1$)
- In the general case, deciding inclusion means solving possibly many linear programs (but can be avoided in practice)

Example

- $x_1 = 2 + \varepsilon_1$, $x_2 = 2 - \varepsilon_1$
- x_1 and x_2 are incomparable



Functional order relation

Want an order that preserves the parametrization as input-output relationships.

Concretization in terms of sets of functions from $\mathbb{R}^n$ to $\mathbb{R}^p$:

$$\gamma_f(X) = \left\{ f : \mathbb{R}^n \to \mathbb{R}^p \mid \forall \epsilon \in [-1,1]^n, \exists \eta \in [-1,1]^m, f(\varepsilon) = {C^X}^T \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix} + {P^X}^T \eta \right\}.$$

- $\gamma_f(X) \subseteq \gamma_f(Y)$ equivalent to

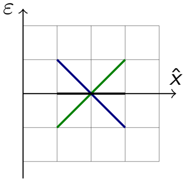
$$X \sqsubseteq Y \iff \forall u \in \mathbb{R}^p,\ \|(C^Y - C^X)u\|_1 \leq \|P^Y u\|_1 - \|P^X u\|_1$$

(implies the geometric ordering $\|C^X u\|_1 + \|P^X u\|_1 \leq \|C^Y u\|_1 + \|P^Y u\|_1$)

- In the general case, deciding inclusion means solving possibly many linear programs (but can be avoided in practice)

Example

- $x_1 = 2 + \varepsilon_1$, $x_2 = 2 - \varepsilon_1$ (geometric concretization $[1, 3]$)
- x_1 and x_2 are incomparable



Functional order relation

Want an order that preserves the parametrization as input-output relationships.

Concretization in terms of sets of functions from $\mathbb{R}^n$ to $\mathbb{R}^p$:

$$\gamma_f(X) = \left\{ f : \mathbb{R}^n \to \mathbb{R}^p \mid \forall \epsilon \in [-1,1]^n, \exists \eta \in [-1,1]^m, f(\varepsilon) = {C^X}^T \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix} + {P^X}^T \eta \right\}.$$

- $\gamma_f(X) \subseteq \gamma_f(Y)$ equivalent to

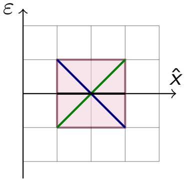
$$X \sqsubseteq Y \iff \forall u \in \mathbb{R}^p,\ \|(C^Y - C^X)u\|_1 \leq \|P^Y u\|_1 - \|P^X u\|_1$$

  (implies the geometric ordering $\|C^X u\|_1 + \|P^X u\|_1 \leq \|C^Y u\|_1 + \|P^Y u\|_1$)

- In the general case, deciding inclusion means solving possibly many linear programs (but can be avoided in practice)

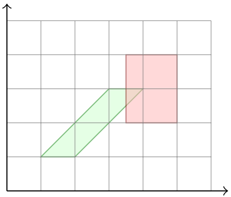
Example

- $x_1 = 2 + \varepsilon_1$, $x_2 = 2 - \varepsilon_1$, $x_3 = 2 + \eta_1$ (geometric concretization $[1, 3]$)
- x_1 and x_2 are incomparable , both are included in x_3.



Set operations on affine sets / zonotopes: meet

Intersection of zonotopes are not zonotopes!



Interpreting conditionals

- Translate the condition on noise symbols: constrained affine sets
- Abstract domain for the noise symbols: intervals, octagons, etc.
- Equality tests are interpreted by the substitution of one noise symbol of the constraint (also summary instantiation for modular analysis)
- Arithmetic operations carry over nicely to this logical/reduced product

Example

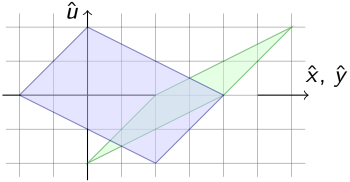
Example

```
real x = [0,10]; real y = 2*x;
if (y >= 10) y = x;
```

- Affine forms before tests: $x = 5 + 5\varepsilon_1$, $y = 10 + 10\varepsilon_1$
- In the `if` branch $\varepsilon_1 \geq 0$: condition acts on both x and y

Join operator

$$\begin{pmatrix} \hat{x} = 3 + \varepsilon_1 + 2\varepsilon_2 \\ \hat{u} = 0 + \varepsilon_1 + \varepsilon_2 \end{pmatrix} \cup \begin{pmatrix} \hat{y} = 1 - 2\varepsilon_1 + \varepsilon_2 \\ \hat{u} = 0 + \varepsilon_1 + \varepsilon_2 \end{pmatrix} = \begin{pmatrix} \hat{x} \cup \hat{y} & = & 2 + \varepsilon_2 + 3\eta_1 \\ \hat{u} & = & 0 + \varepsilon_1 + \varepsilon_2 \end{pmatrix}$$



Construction (low complexity!: $\mathcal{O}(n \times p)$)

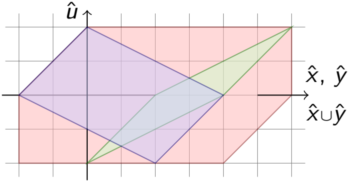
- Keep "minimal common dependencies"

$$z_i = \operatorname*{argmin}_{x_i \wedge y_i \leq r \leq x_i \vee y_i} |r|, \ \forall i \geq 1$$

- For each dimension, concretization is the interval union of the concretizations: $\gamma(\hat{x} \cup \hat{y}) = \gamma(\hat{x}) \cup \gamma(\hat{y})$
- A minimal upper bound under some conditions (several uncomparable minimal upper boundsin general)

Join operator

$$\begin{pmatrix} \hat{x} = 3 + \varepsilon_1 + 2\varepsilon_2 \\ \hat{u} = 0 + \varepsilon_1 + \varepsilon_2 \end{pmatrix} \cup \begin{pmatrix} \hat{y} = 1 - 2\varepsilon_1 + \varepsilon_2 \\ \hat{u} = 0 + \varepsilon_1 + \varepsilon_2 \end{pmatrix} = \begin{pmatrix} \hat{x} \cup \hat{y} & = & 2 + \varepsilon_2 + 3\eta_1 \\ \hat{u} & = & 0 + \varepsilon_1 + \varepsilon_2 \end{pmatrix}$$



Construction (low complexity!: $\mathcal{O}(n \times p)$)

- Keep "minimal common dependencies"

$$z_i = \operatorname*{argmin}_{x_i \wedge y_i \leq r \leq x_i \vee y_i} |r|, \; \forall i \geq 1$$

- For each dimension, concretization is the interval union of the concretizations: $\gamma(\hat{x} \cup \hat{y}) = \gamma(\hat{x}) \cup \gamma(\hat{y})$
- A minimal upper bound under some conditions (several uncomparable minimal upper boundsin general)

Convergence schemes

Fixpoint computation

Given a continuous upper-bound operator U, the U-iteration scheme for a strict, continuous functional F on affine sets (extended with a formal $\perp$ and $\top$), is as follows:

- Start with $X_0 = \perp$
- Then iterate: $X_{u+1} = X_u U F(X_u)$
  - if $X_{u+1} \leq X_u$ then stop with X_u
  - if $\gamma(X_{u+1}) \not\subseteq I^p$, then end with $\top$ (or any thresholding mechanism...)

Convergence schemes

Fixpoint computation

Given a continuous upper-bound operator U, the U-iteration scheme for a strict, continuous functional F on affine sets (extended with a formal $\perp$ and $\top$), is as follows:

- Start with $X_0 = \perp$
- Then iterate: $X_{u+1} = X_u U F(X_u)$
  - if $X_{u+1} \leq X_u$ then stop with X_u
  - if $\gamma(X_{u+1}) \not\subseteq I^p$, then end with $\top$ (or any thresholding mechanism...)

Stopping criterion

- Test $X_{u+1} \leq X_u$ guarantees that X_u is a post-fixed point of F, but is costly
- We can use simpler componentwise geometrical inclusion, and have the full test only when the simpler test is satisfied
- We can use the fact that a particular join operator is used

Convergence schemes

Fixpoint computation

Given a continuous upper-bound operator U, the U-iteration scheme for a strict, continuous functional F on affine sets (extended with a formal $\perp$ and $\top$), is as follows:

- Start with $X_0 = \perp$
- Then iterate: $X_{u+1} = X_u U F(X_u)$
  - if $X_{u+1} \leq X_u$ then stop with X_u
  - if $\gamma(X_{u+1}) \not\subseteq I^p$, then end with $\top$ (or any thresholding mechanism...)

Stopping criterion

- Test $X_{u+1} \leq X_u$ guarantees that X_u is a post-fixed point of F, but is costly
- We can use simpler componentwise geometrical inclusion, and have the full test only when the simpler test is satisfied
- We can use the fact that a particular join operator is used

In practice

- Initial unfolding - i.e. start fp solving at some $F^l(\perp)$
- Cyclic unfolding - i.e. compute $fp(F^k) \cup F(fp(F^k)) \cup \ldots F^{k-1}(fp(F^k))$

Convergence results: from concrete to abstract

General result on recursive linear filters, pervasive in embedded programs:

$$x_{k+n+1} = \sum_{i=1}^{n} a_i x_{k+i} + \sum_{j=1}^{n+1} b_j e_{k+j}, \quad e[*] = input(m, M);$$

- Suppose this concrete scheme has bounded outputs (zeros of $x^n - \sum_{i=0}^{n-1} a_{i+1} x^i$ have modules stricty lower than 1).
- Then there exists q such that the Kleene abstract scheme "unfolded modulo q" converges towards a finite over-approximation of the outputs

$$\hat{X}_i = \hat{X}_{i-1} \cup f^q(E_i, \ldots, E_{i-k}, \hat{X}_{i-1}, \ldots, \hat{X}_{i-k})$$

  in finite time, potentially with a widening partly losing dependency information
  - The abstract scheme is a perturbation (by the join operation) of the concrete scheme
  - Proof uses the stability property of our join operator: for each dimension $\gamma(\hat{x} \cup \hat{y}) = \gamma(\hat{x}) \cup \gamma(\hat{y})$ and f^q "contractive enough" for some q

Illustration: a simple order 2 filter

$$S_{n+2} = 0.7E_{n+2} - 1.3E_{n+1} + 1.1E_n + 1.4S_{n+1} - 0.7S_n$$

Step 0: initial unfolding (10)+first cyclic unfolding (80) - first join
Step 1: After first join, perturbation of the original numerical scheme!
Step 2: second cyclic unfolding, contracting back - second join and post-fixpoint

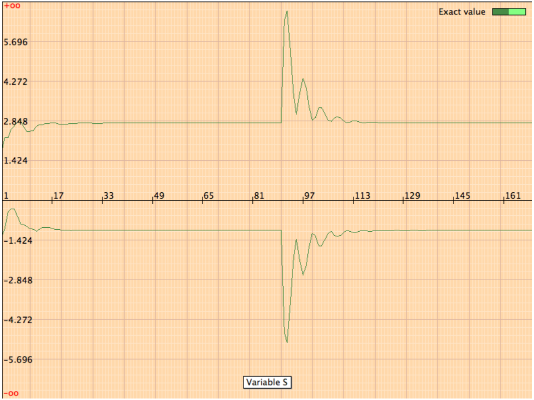
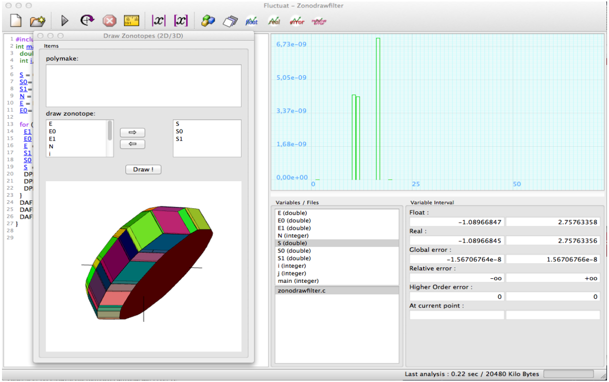


Illustration: a simple order 2 filter

$$S_{n+2} = 0.7E_{n+2} - 1.3E_{n+1} + 1.1E_n + 1.4S_{n+1} - 0.7S_n$$



- A polyhedral approximation of the classical ellipsoidal invariant
- May be inefficient, for convergence, q depending on the largest eigenvalue module
- mixed zonotopic/ellipsoidal invariants ?

Ellipsoidal domains

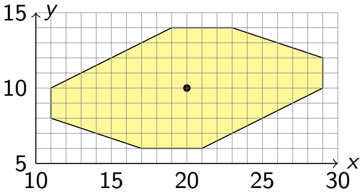
Long history

Kurzhanski in Control Theory (1991), Feret (ESOP 2004), Cousot (VMCAI 2005), Adjé et al. (ESOP 2010), Gawlitza et al. (SAS 2010), Garoche et al. (HSCC 2012) etc.

Extend affine forms to ellipsoidal forms ? Change of norm (norm l_2, or l_p...

$$\hat{x} = x_0 + \sum_{i=1}^{n} \mathbf{x}_i \varepsilon_i, \text{ with } \|\varepsilon\|_\infty = \sup_{i=1,\ldots,n} | \varepsilon_i | \leq 1$$

$$\begin{aligned} x &= 20 - 4\varepsilon_1 + 2\varepsilon_3 + 3\varepsilon_4 \\ y &= 10 - 2\varepsilon_1 + \varepsilon_2 - \varepsilon_4 \end{aligned}$$



Ellipsoidal domains

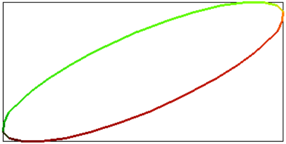
Long history

Kurzhanski in Control Theory (1991), Feret (ESOP 2004), Cousot (VMCAI 2005), Adjé et al. (ESOP 2010), Gawlitza et al. (SAS 2010), Garoche et al. (HSCC 2012) etc.

Extend affine forms to ellipsoidal forms ? Change of norm (norm l_2, or l_p...

$$\hat{x} = x_0 + \sum_{i=1}^{n} \mathbf{x}_i \varepsilon_i, \text{ with } \|\varepsilon\|_2 = \sqrt{\sum_{i=1}^{n} \varepsilon_i^2} \leq 1$$

$$\begin{aligned} x &= 20 - 4\varepsilon_1 + 2\varepsilon_3 + 3\varepsilon_4 \\ y &= 10 - 2\varepsilon_1 + \varepsilon_2 - \varepsilon_4 \end{aligned}$$



Ellipsoidal domains

Long history

Kurzhanski in Control Theory (1991), Feret (ESOP 2004), Cousot (VMCAI 2005), Adjé et al. (ESOP 2010), Gawlitza et al. (SAS 2010), Garoche et al. (HSCC 2012) etc.

Extend affine forms to ellipsoidal forms ? Change of norm (norm l_2, or l_p...

$$\hat{x} = x_0 + \sum_{i=1}^{n} \mathbf{x}_i \varepsilon_i, \text{ with } \|\varepsilon\|_2 = \sqrt{\sum_{i=1}^{n} \varepsilon_i^2} \leq 1$$

Functional order

$X \subseteq Y$ if and only if for all $t \in \mathbb{R}^p$

$$\|(C^X - C^Y)t\|_2 \leq \|P^Y t\|_2 - \|P^X t\|_2$$

Some references

- [CAV 2009] K. Ghorbal, E. Goubault and S. Putot, The Zonotope Abstract Domain Taylor1+ (upper bounds, fixpoint computations, implementation in the Apron library)
- [CAV 2010] K. Ghorbal, E. Goubault and S. Putot, A Logical Product to Zonotope Intersection (interpretation of conditionals - also a variation in: Automatica 2016, Constrained zonotopes: A new tool for set-based estimation and fault detection, Scott, Raimondo, Marseglia, Braatz)
- [SAS 2012] E. Goubault, S. Putot and F. Védrine, Modular Static Analysis with Zonotopes
- [NSAD 2012] E. Goubault, T. Le Gall and S. Putot, An Accurate Join for Zonotopes, Preserving Affine Input/Output Relations
- [FMSD 2016] E. Goubault and S. Putot, A zonotopic framework for functional abstractions (extended version with full abstraction, older versions with more details Arxiv 2008 and Arxiv 2009)

Use of zonotopes for the verification of neural networks

Convolutional neural networks analysis

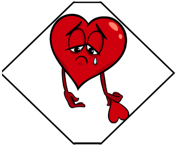
Composition of conditional affine transformations and non-affine activation functions (for example ReLU(x)=max(x,0), etc)

- affine transformations are exactly and efficiently represented in zonotopes
- activation function has to be abstracted

Work at ETH Zurich: several articles on abstract interpretation of such networks using zonotopes, for example

- [IEEE S&P 2018] AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation, T Gehr, M Mirman, D Drachsler-Cohen, P Tsankov, S Chaudhuri, M Vechev

But ... their most recent work (DeepPoly) relying on polyhedra outperforms their previous one with zonotopes! ([POPL 2019] An Abstract Domain for Certifying Neural Networks, G Singh, T Gehr, M Püschel, M Vechev)



Outline of the talk

Zonotopes as a general purpose numerical abstract domain

- Inspired from guaranteed numerical methods (affine arithmetic, Taylor methods)
- A functional abstraction: parametrized sub-polyhedral abstraction with low complexity (allows modular analysis, test generation, etc)
- Set operations and a word on fixpoint computations

Good at expressing (and propagationg) perturbations

- Used for assessing safety and robustness of neural networks (ETH Zurich)
- Finite precision accuracy and robustness analysis (Fluctuat analyzer)

Possible extensions

- An interesting representation of ellipsoids ?
- Under-approximations
- Mixed non-deterministic and probabilistic analysis

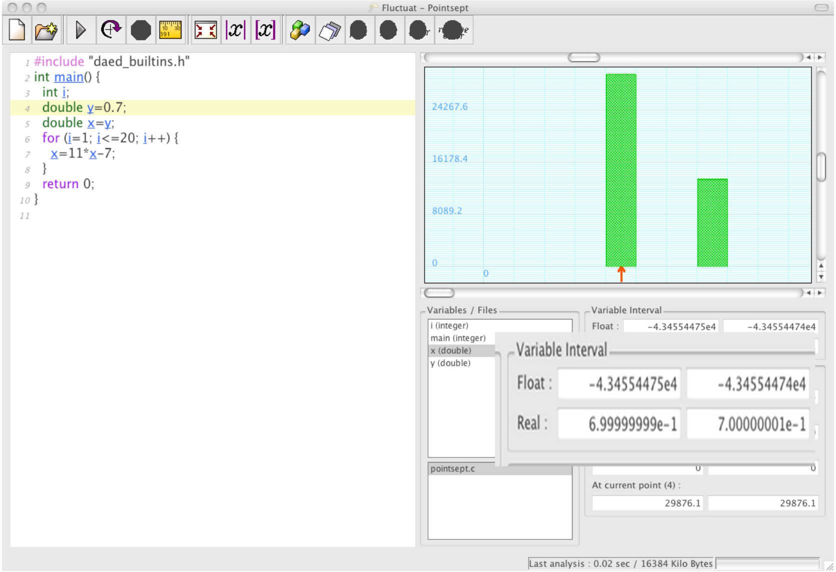
FLUCTUAT: concrete semantics

- IEEE 754 norm on f.p. numbers specifies the rounding error (same is feasible for fixed point semantics)
- Aim: compute rounding errors and their propagation
  - we need the floating-point values
  - relational (thus accurate) analysis more natural on real values
  - for each variable, we compute (f^x, r^x, e^x)
  - then we will abstract each term (real value and errors)

```
float x,y,z;
x = 0.1; // [1]
y = 0.5; // [2]
z = x+y; // [3]
t = x*z; // [4]
```

$$
\begin{aligned}
f^x &= 0.1 + 1.49e^{-9}\ [1] \\
f^y &= 0.5 \\
f^z &= 0.6 + 1.49e^{-9}\ [1] + 2.23e^{-8}\ [3] \\
f^t &= 0.06 + 1.04e^{-9}\ [1] + 2.23e^{-9}\ [3] - 8.94e^{-10}\ [4] - 3.55e^{-17}\ [ho]
\end{aligned}
$$

Example (Fluctuat)



Abstraction in Fluctuat

Abstract value

- For each variable x, a triplet (f^x, r^x, e^x):
  - Interval $\mathbf{f}^x = [\underline{f^x}, \overline{f^x}]$ bounds the finite prec value, $(\underline{f^x}, \overline{f^x}) \in \mathbb{F} \times \mathbb{F}$,
  - Affine forms for real value and error; for simplicity no η symbols

$$f^x = \underbrace{(\alpha_0^x + \bigoplus_i \alpha_i^x \varepsilon_i^r)}_{\text{real value}} + (\underbrace{e_0^x}_{\text{center of the error}} + \underbrace{\bigoplus_l e_l^x \varepsilon_l^e}_{\text{uncertainty on error due to point } l} + \underbrace{\bigoplus_i m_i^x \varepsilon_i^r}_{\text{propag of uncertainty on value at pt } i})$$

- Constraints on noise symbols (interval + equality constraints)
  - for finite precision control flow
  - for real control flow

Second order filters

Filters

Householder

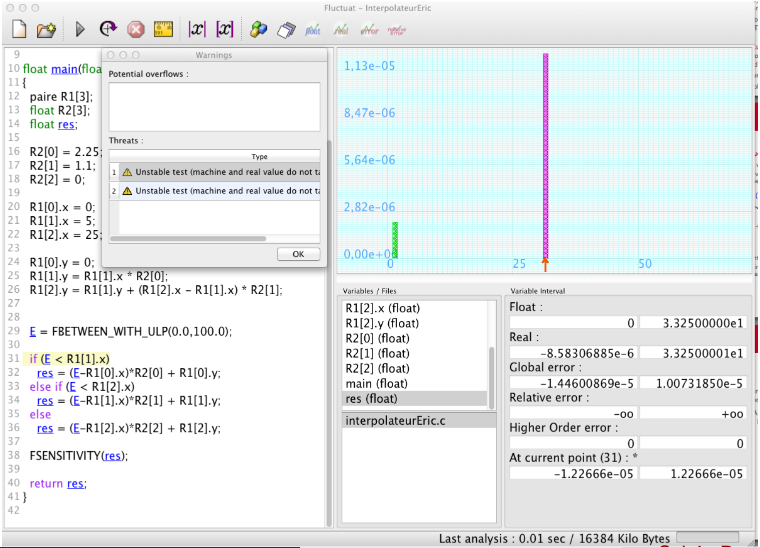
Control flow problems!

Unstable tests: when real and finite precision control flow can be different

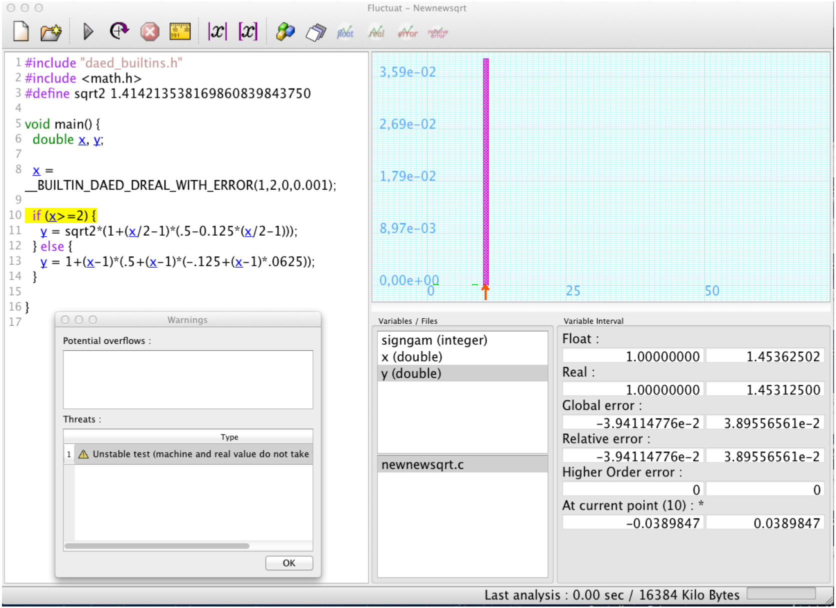
- Error analyses are sound only under the stable test assumption
- When considering large sets of executions, most tests are unstable
- Compute discontinuity error bounds due to unstable tests:
  - makes our error analysis sound in the presence of unstable tests
  - gives a robustness analysis of implementations (in line with work on continuity/robustness analysis of Chaudhuri Gulwani POPL 2010, Majumbar RTSS 2009 etc.)

A typical example of unstable tests: affine interpolators

All tests are unstable, but the implementation is robust, the conditional block does not introduce a discontinuity



But actual discontinuities also occur (sqrt qpproximation)



Abstract domain in Fluctuat

Abstract value at each control point c

- For each variable, affine forms for real value and error:

$$f^x = \underbrace{(\alpha_0^x + \bigoplus_i \alpha_i^x \varepsilon_i^r)}_{\text{real value}} + (\underbrace{e_0^x}_{\text{center of the error}} + \underbrace{\bigoplus_l e_l^x \varepsilon_l^e}_{\text{uncertainty on error due to point } l}$$

$$+ \underbrace{\bigoplus_i m_i^x \varepsilon_i^r}_{\text{propag of uncertainty on value at pt } i})$$

- Constraints on noise symbols coming from interpretation of test condition
  - $\varepsilon^r \in \Phi_r^X$ for real control flow (test on the r^x: constraints on the ε_i^r)
  - $(\varepsilon^r, \varepsilon^e) \in \Phi_f^X$ for finite precision control flow (test on the $f^x = r^x + e^x$: constraints on the ε_i^r and ε_l^e)

Unstable test condition = intersection of constraints $\varepsilon^r \in \Phi_r^X \sqcap \Phi_f^Y$:

- unstable test: <u>for a same execution (same values of the noise symbols ε_i)</u> the control flow is different
- restricts the range of the ε_i: allows us to bound accurately the discontinuity error

Formally, sound abstraction (with discontinuity errors)

Abstract value

An abstract value X, for a program with p variables $x_1, \ldots, x_p$, is a tuple $X = (R^X, E^X, D^X, \Phi_r^X, \Phi_f^X)$ composed of the following affine sets and constraints, for all $k = 1, \ldots, p$:

$$\left\{ \begin{array}{llcll} R^X & : \hat{r}_k^X & = & r_{0,k}^X + \sum_{i=1}^{n} r_{i,k}^X \varepsilon_i^r & \text{where } \varepsilon^r \in \Phi_r^X \\ E^X & : \hat{e}_k^X & = & e_{0,k}^X + \sum_{i=1}^{n} e_{i,k}^X \varepsilon_i^r + \sum_{j=1}^{m} e_{n+j,k}^X \varepsilon_j^e & \text{where } (\varepsilon^r, \varepsilon^e) \in \Phi_f^X \\ D^X & : \hat{d}_k^X & = & d_{0,k}^X + \sum_{i=1}^{o} d_{i,k}^X \varepsilon_i^d & \\ & \hat{f}_k^X & = & \hat{r}_k^X + \hat{e}_k^X & \text{where } (\varepsilon^r, \varepsilon^e) \in \Phi_f^X \end{array} \right.$$

E^X is the propagated rounding error, D^X the propagated discontinuity error

New discontinuity errors computed when joining branches of a possibly unstable test

$Z = X \sqcup Y$ is $Z = (R^Z, E^Z, D^Z, \Phi_r^X \cup \Phi_r^Y, \Phi_f^X \cup \Phi_f^Y)$ such that

$$\left\{ \begin{array}{l} (R^Z, \Phi_r^Z \cup \Phi_f^Z) = (R^X, \Phi_r^X \cup \Phi_f^X) \sqcup (R^Y, \Phi_r^Y \cup \Phi_f^Y) \\ (E^Z, \Phi_f^Z) = (E^X, \Phi_f^X) \sqcup (E^Y, \Phi_f^Y) \\ D^Z = D^X \sqcup D^Y \sqcup (R^X - R^Y, \Phi_f^X \sqcap \Phi_r^Y) \sqcup (R^Y - R^X, \Phi_f^Y \sqcap \Phi_r^X) \end{array} \right.$$

Example: sound unstable test analysis

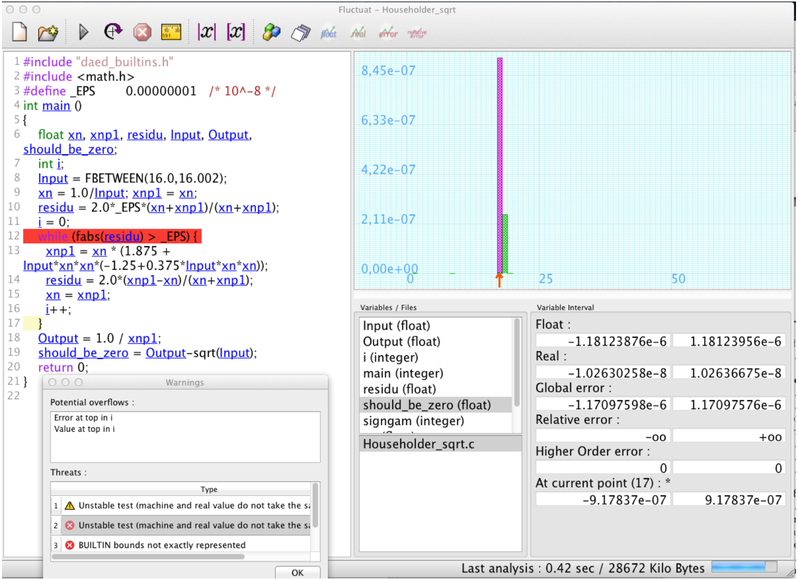
```
int main(void) {
 double x,y;
 x = DREAL_WITH_ERROR(1,3,1.0e-5,1.0e-5);
 if (x <= 2)
   y = x + 2; [1]
 else
   y = x; [2]
}
```

- Before the test: $f^x = (2 + \varepsilon_1) + 10^{-5}$
- Test $x \leq 2$:
  - in reals: $\varepsilon_1 \leq 0$
  - in floats: $\varepsilon_1 + 1.0e^{-5} \leq 0$, ie $\varepsilon_1 \leq -1.0e^{-5}$.
- First unstable test possibility :
  - real takes then branch: $\varepsilon_1 \leq 0$
  - float takes else branch: $\varepsilon_1 > -1.0e^{-5}$
  - unstable test = intersection of constraints: $-1.0e^{-5} < \varepsilon_1 \leq 0$

$$f^y_{[2]} - r^y_{[1]} = (2 + \varepsilon_1 + 1.0e^{-5}) - (4 + \varepsilon_1) = -2 + 1.0e^{-5}.$$

- Second unstable test possibility: conditions $\varepsilon_1 \leq -1.0e^{-5}$ and $\varepsilon_1 > 0$ are non compatible (no unstable test)

Householder algorithm for square root



Some references

Latest abstractions in Fluctuat

- [APLAS 2013] E. Goubault and S. Putot, Robustness analysis of finite precision implementations (handling unstable tests)
- [VMCAI 2011] E. Goubault and S.Putot, Static Analysis of Finite Precision Computations (full zonotopic abstraction with stable test assumption)

Case studies

on industrial code, mostly control code (nuclear plants, automotive industry, aeronautics and space industry etc.)

- [FMICS 2009] D. Delmas, E. Goubault, S. Putot, J. Souyris, K. Tekkal and F. Vedrine, Towards an Industrial Use of FLUCTUAT on Safety-Critical Avionics Software
- [FMICS 2007] E. Goubault, S. Putot, P. Baufreton and J. Gassino, Static Analysis of the Accuracy in Control Systems : Principles and Experiments (nuclear safety applications)

Possible extensions of affine sets / zonotopes

Keep same parameterization $x = \sum_i x_i \varepsilon_i$ but with

- Interval/zonotopic coefficients x_i: generalized affine sets for under-approximation
  - under-approximation = sets of variables values, that are sure to be reached for some inputs in the specified ranges
  - using Kaucher arithmetic extending interval arithmetic over generalized intervals
  - [SAS 2007] E. Goubault and S. Putot, Under-Approximations of Computations in Real Numbers Based on Generalized Affine Arithmetic
  - But also for hybrid systems reachability analysis (VMCAI talk on Sunday!)
- Noise symbols ε_i no longer simply defined as ranging in intervals:
  - ellipsoids: $\|\varepsilon\|_2 \leq 1$ (instead of $\|\varepsilon\|_\infty \leq 1$)
  - probabilistic affine forms: ε_i take values in probability boxes

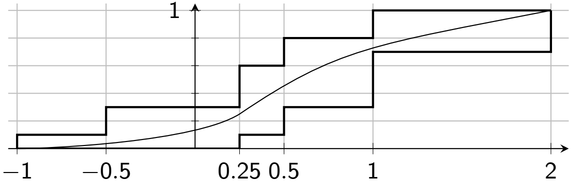
Motivation for a probabilistic extension to affine forms

Typical problem

- Some inputs known to lie in sets (non-deterministic inputs), and some a probability distribution (probabilistic inputs)
    - for example, temperature distribution known but we only know a range for pressure, in some software-driven apparatus
- Inputs may be thought of as given by imprecise probabilities

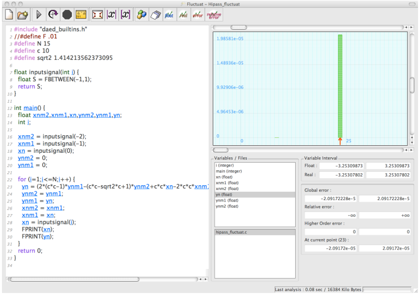
Discrete p-boxes or Dempster-Shafer structures

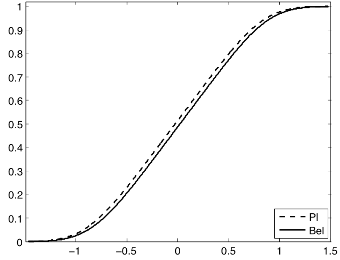
- Generalize probability distributions and interval computations
- Represent sets of probability distributions: between an upper and a lower Cumulative Distribution Function $P(X \leq x)$



Example: recursive filter with independent inputs in [-1,1]

Prove that dangerous worst case occur with very low probability

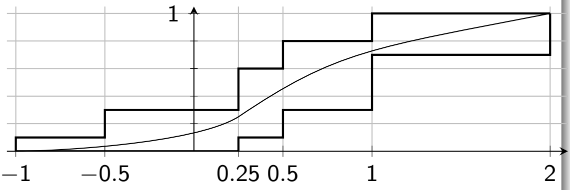




- Deterministic analysis (left): outputs in [-3.25,3.25] (exact)
- Mixed probabilistic/deterministic analysis (right): outputs in [-3.25,3.25], and in [-1,1] with very strong probability (in fact, very close to a Gaussian distribution)

Based on Dempster-Shafer structures (1976)

- Based on a notion of focal elements ($\in F$ - here F is a set of subsets of $\mathbb{R}$):
  - sets of non-deterministic events/values - here sub-intervals of values in [-1,1]
- Weights (positive reals) associated to focal elements ($w : F \to \mathbb{R}^+$)
  - probabilistic information available on the belonging to the focal elements, not to precise events
  - equivalent to having staircase upper and lower probabilities



Example:

$$d = \{\langle[-1, 0.25], 0.1\rangle, \langle[-0.5, 0.5], 0.2\rangle, \langle[0.25, 1], 0.3\rangle,$$
$$\langle[0.5, 1], 0.1\rangle, \langle[0.5, 2], 0.1\rangle, \langle[1, 2], 0.2\rangle\}$$

represents the set of probability distributions with support $[-1, 2]$, where the probability of picking a value between -1 and 0.25 is 0.1, the probability of picking a value between -0.5 and 0.5 is 0.2 etc.

Some computation rules: $z = x \square y$ ($\square$=+,-,$\times$,/ etc.)

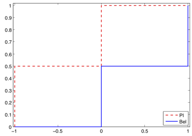
Independent variables x, y

- x (resp. y) given by focal elements F^x (resp. F^y) and weights w^x (resp. w^y)
- Define DS for z: $F^z = \{f^x \square f^y \mid f^x \in F^X, f^y \in F^y\}$ and $w^z(f^x \square f^y) = w^x(f^x) w^y(f^y)$ (and renormalize)

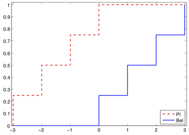
Example

- x with $F^x = \{[-1,0],[0,1]\}$, $w^x([-1,0]) = w^x([0,1]) = \frac{1}{2}$ (approximation of uniform distribution on [-1,1])
- y with $F^y = \{[-2,0],[0,2]\}$, $w^y([-2,0]) = w^y([0,2]) = \frac{1}{2}$

$x; y$	[-2,0], $\frac{1}{2}$	[0,2], $\frac{1}{2}$
[-1,0], $\frac{1}{2}$	[-3,0], $\frac{1}{4}$	[-1,2], $\frac{1}{4}$
[0,1], $\frac{1}{2}$	[-2,1], $\frac{1}{4}$	[0,3], $\frac{1}{4}$



CDF of x



CDF of $x + y$

Dependent variables: more costly and imprecise operations

Our approach

Encode as much deterministic dependencies as possible by affine arithmetic

$$\begin{array}{rcll} S_{n+2} & = & 0.7E_{n+2} - 1.3E_{n+1} + 1.1E_n & \text{independent values} \\ & & +1.4S_{n+1} - 0.7S_n & \text{linear dependancy} \end{array}$$

- because arithmetic on dependent p-boxes / DS is not very efficient

P-forms (probabilistic affine forms)

- Associate a Dempster-Shafer structure to each noise symbol
- ε_i independent of each other, created by inputs
- η_j unknown dependencies with each other and with the ε_i, created by non-linear computation (including branching)
- use of Frechet bounds when dependencies are unknown, easier calculus when variables are known to be independent
- both more accurate and faster than direct DS arithmetic

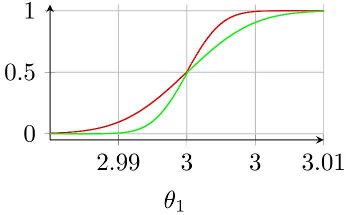
Example: Ferson polynomial

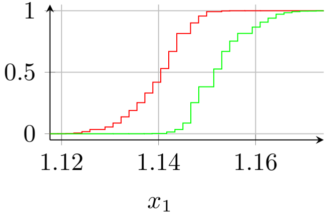
- Example from Enszer, J.A., Lin, Y., Ferson, S., Corliss, G.F., Stadtherr, M.A., "Probability bounds analysis for nonlinear dynamic process models"
- Goal: compute bounds on the solution of the differential equations

$$\dot{x}_1 = \theta_1 x_1 (1 - x_2) \quad \dot{x}_2 = \theta_2 x_2 (x_1 - 1)$$

with initial values $x_1(0) = 1.2$ and $x_2(0) = 1.1$ and uncertain parameters θ_1, θ_2 given by a normal distribution with mean 3 and 1, resp., but with an unknown standard deviation in the range $[-0.01, 0.01]$

- Results with our probabilistic affine forms:





- Application: we can, with high probability, discard some values in the resulting interval. For example, we could show that $P(x_1 \leq 1.13) \leq 0.0552$

References

P-forms:

- [Computing 2012] O. Bouissou, E. Goubault, J. Goubault-Larrecq, S. Putot, A generalization of p-boxes to affine arithmetic
- [VSTTE 2013] A. Adje, O. Bouissou, E. Goubault, J. Goubault-Larrecq, S. Putot, Static Analysis of Programs with Imprecise Probabilistic Inputs

Improving the abstraction:

- [TACAS 2016] O. Bouissou, E. Goubault, S. Putot, A. Chakarov, S. Sankaranarayanan, Uncertainty Propagation using Probabilistic Affine Forms and Concentration of Measure Inequalities,

Application to the analysis of finite precision decision-making programs:

- [EMSOFT 2018] E. Darulova, E. Goubault, D. Lohar, S. Putot, Discrete Choice in the Presence of Numerical Uncertainties

Implementations of zonotope abstract domains

Zonotope abstract domain

- Implemented by K. Ghorbal in the APRON library (http://apron.cri.ensmp.fr/library/, domain named Taylor1+)
- Also some version in Elina http://elina.ethz.ch (used for neural network analysis)

Application in tools for finite precision analysis

- Academic version of FLUCTUAT (proprietary tool of CEA) (can be used through an API from an other analyzer)
- Rosa (TOPLAS 2017, Towards a Compiler for Reals, E. Darulova, V. Kuncak, also relies on affine arithmetic), Daisy (TACAS 2018 Framework for Analysis and Optimization of Numerical Programs, E. Darulova, A Izycheva, F. Nasir, F. Ritter, H. Becker, and R. Bastian)
- PRECiSA (VMCAI 2018, An Abstract Interpretation Framework for the Round-Off Error Analysis of Floating-Point Programs), also uses affine arithmetic among other abstractions