Inner and Outer Approximation of Functionals
coming from static analysis
using
Generalized Affine Forms

Eric Goubault and Sylvie Putot

CEA-LIST, MEASI (ModElisation and Analysis of Systems in Interaction)

SCAN 2008, El Paso, TX
Static analysis of programs

- Find outer-approximation of sets of reachable values of variables at some program points
- To ensure absence of runtime errors typically

Example

```c
float x;
x = [0,1]; [1]  \quad x_1 = [0,1]
while (x<=1) {
    x = x - 0.5*x; [3]  \quad x_2 = ]-\infty,1[ \cap (x_1 \cup x_3)
} [4]  \quad x_3 = x_2 - 0.5x_2

x_4 = ]1,\infty[ \cap x_2 (final smallest invariant: x_2 \in [0,1], x_4 = \emptyset)
```
Motivation for this talk

Proof of good behaviour

- Need for **tight** and **correct** outer approximations
  - First part of the talk: How do we find invariant sets? How do we ensure correctness?
  - Based on affine forms - concentrate on real values first

But how pessimistic are the results?

- Joint use of **inner- and outer-approximations** to characterize the quality of analysis results
  - Inner-approximation: sets of values for the variables, that are sure to be reached for some inputs in the specified ranges.
  - (Second part of the talk) Use of affine forms with **generalized intervals** as coefficients
Affine Arithmetic for real numbers

Originally: Comba, de Figueiredo and Stolfi 1993

- A variable $x$ is represented by an affine form $\hat{x}$:

$$\hat{x} = x_0 + x_1\varepsilon_1 + \ldots + x_n\varepsilon_n,$$

where $x_i \in \mathbb{R}$ and $\varepsilon_i$ are independent symbolic variables with unknown value in $[-1, 1]$.

- $x_0 \in \mathbb{R}$ is the central value of the affine form
- the coefficients $x_i \in \mathbb{R}$ are the partial deviations
- the $\varepsilon_i$ are the noise symbols

- The sharing of noise symbols between variables expresses implicit dependency

On top of that...

We want a notion of union (and intersections - outside the scope of this talk) of affine forms since we want to compute invariant forms of particular dynamical systems (programs).
They form sub-polyhedric relations

**Concretization** is a center-symmetric convex polytope

\[
\hat{x} = 20 - 4\varepsilon_1 + 2\varepsilon_3 + 3\varepsilon_4 \\
\hat{y} = 10 - 2\varepsilon_1 + \varepsilon_2 - \varepsilon_4
\]

Define...

\[
\gamma(\hat{x}) = [\alpha_0^x - \|\hat{x}\|, \alpha_0^x + \|\hat{x}\|]
\]

where \(\|\hat{x}\|_1 = \sum_{i=1}^{\infty} |\alpha_i^x|\), (finite, or \(\ell_1\)-convergence)

Also define joint concretisation.
## Assignment

of a variable $x$ whose value is given in a range $[a, b]$ at label $i$, introduces a noise symbol $\varepsilon_i$:

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

## Addition

$$\hat{x} + \hat{y} = (\alpha_0^x + \alpha_0^y) + (\alpha_1^x + \alpha_1^y)\varepsilon_1 + \ldots + (\alpha_n^x + \alpha_n^y)\varepsilon_n$$

For example, with real (exact) coefficients, $f - f = 0$.

## Multiplication

creates a new noise term (can do better):

$$\hat{x} \times \hat{y} = \alpha_0^x \alpha_0^y + \sum_{i=1}^{n} (\alpha_i^x \alpha_0^y + \alpha_i^y \alpha_0^x)\varepsilon_i + \left( \sum_{i=1}^{n} |\alpha_i^x| \cdot \sum_{i=1}^{n} |\alpha_i^y| \right) \varepsilon_{n+1}.$$
Interpretation of unions?

How do we compute...?

...as an affine form $\hat{x}$ the union of for instance:

$$\hat{x} = 3 + \varepsilon_1 + 2\varepsilon_2$$
$$\hat{y} = 1 - 2\varepsilon_1 + \varepsilon_2$$

Problem

- Easy geometric interpretation of union but difficult to find a good notion of “optimal” affine form representing a union
- Unions are some form of non-linear operations
- Our choice: distinguish a noise symbol $\varepsilon_U$ for taking care of uncertainties due to unions (and intersections)
Join operation (see also Goubault/Putot 2008 [4])

Define $z = x \cup y$ by:

\[
\begin{align*}
\alpha_0^z &= mid(\gamma(\hat{x}) \cup \gamma(\hat{y})) \\
\alpha_i^z &= \arg\min |\alpha|, \ \forall i \geq 1 \\
\beta^z &= \sup \gamma(\hat{x} \cup \hat{y}) - \alpha_0^z - \|z\|
\end{align*}
\]

- Intuitively, we keep in the union the minimal common dependencies, the “rest” being put as a coefficient to $\epsilon_U$
- Meet similar...

Where... (“minimal dependency”)

\[
\arg\min_{u \land v \leq \alpha \leq u \lor v} |\alpha| = \{\alpha \in [u \land v, u \lor v], |\alpha| \text{ minimal}\}
\]
Example - again

\[ \hat{x} = 3 + \varepsilon_1 + 2\varepsilon_2 \]
\[ \hat{y} = 1 - 2\varepsilon_1 + \varepsilon_2 \]
\[ \hat{u} = \varepsilon_1 + \varepsilon_2 \]
Example - again

\[ \hat{x} = 3 + \varepsilon_1 + 2\varepsilon_2 \]
\[ \hat{y} = 1 - 2\varepsilon_1 + \varepsilon_2 \]
\[ \hat{u} = \varepsilon_1 + \varepsilon_2 \]

\[ \hat{x} \cup \hat{y} = 2 + \varepsilon_2 + 3\varepsilon_U \]

(Note that \( \gamma(\hat{z}) = [-2, 6] = \gamma(\hat{x}) \cup \gamma(\hat{y})) \)
Example of an invariant for a simple dynamical system/program

Consider:

\[ x_i = f(e_i, e_{i-1}, e_{i-2}, x_{i-1}, x_{i-2}) = 0.7e_i - 1.3e_{i-1} + 1.1e_{i-2} + 1.4x_{i-1} - 0.7x_{i-2} \]

where \( e_i \) are independent inputs between 0 and 1.

Invariant set computation

We use Kleene iteration:

Compute

\[ \hat{x}_i = \hat{x}_{i-1} \cup f(e_i, e_{i-1}, e_{i-2}, \hat{x}_{i-1}, \hat{x}_{i-2}) \]

(in fact, we iterate \( f \) a little bit, by a factor \( k \))
Invariant set

Results

- (k=5) we reach the over-approximation of the enclosure: \([-1.6328,3.2995]\)
- (k=16) we reach \([-1.3,2.8244]\) (in 18 iterations without widening)
- The smallest enclosure is actually \([-1.121240...,2.824318...]\)

Note that this is not limited to independent inputs, or independent initial conditions.
For instance, if all the inputs over time are equal to an unknown number between 0 and 1, the final invariant found with k=16 has concretization \([-0.1008,2.3298]\).
Criteria for correctness

Replace **concrete** variables \( x_i \) and functions \( f \) by affine forms \( \hat{x}_i \)?

[1] Range of individual variables

**Given expressions** \( y_1 = e_1(x_1, \ldots, x_n), \ldots y_m = e_m(x_1, \ldots, x_n) \)

depending on variables \( x_1, \ldots, x_n \), ensure that \( \gamma(\hat{y}_k) \) contains all concrete values \( y_k \) for all possible values of the \( x_j \).

[2] Joint range, given a fixed set of variables and expressions

Same but for the joint concretisation (as a zonotope) \( \gamma(\hat{y}_1, \ldots, \hat{y}_m) \)

[3] Future evaluations (or global consistency)

We want that for all expressions \( f \), the range of \( \hat{f}(\hat{y}_1, \ldots, \hat{y}_m) \) contains all concrete values \( f(y_1, \ldots, y_m) \)

Clearly... [3] ⇒ [2] ⇒ [1]

Converse?
Correctness?

Take (example by Kolev)

\[ \hat{x} = 10 + 5\epsilon_1 + 3\epsilon_2 \]
\[ \hat{y} = 10 - 2\epsilon_1 + \epsilon_3 \]
\[ \hat{z} = 92 + 31\epsilon_1 + 21\epsilon_2 + 2\epsilon_3 + 16\epsilon_4 \quad \text{Kolev multiplication} \]

Question:
Is \( \hat{z} \) a good model for outer-approximating \( \hat{x}\hat{y} \)?
Correctness?

Take (example by Kolev)

\[
\hat{x} = 10 + 5\epsilon_1 + 3\epsilon_2 \\
\hat{y} = 10 - 2\epsilon_1 + \epsilon_3 \\
\hat{z} = 92 + 31\epsilon_1 + 21\epsilon_2 + 2\epsilon_3 + 16\epsilon_4 \quad \text{Kolev multiplication}
\]

Question:

Is \(\hat{z}\) a good model for outer-approximating \(\hat{x}\hat{y}\)?

Here

\[
\gamma(\hat{z}) = [22, 162]
\]

which is a correct range for the multiplication

We have criterion [1]
Joint range

Inner and outer approximations
Joint range

So we do not have [2]...
Joint range and future evaluations

...Nor [3] (of course!)...

Consider (Khalil Ghorbal)

\[
\hat{t} = -4\hat{x} + 0.8\hat{z} - 79 \\
= -45.4 + 4.8\epsilon_1 + 4.8\epsilon_2 + 1.6\epsilon_3 + 12.8\epsilon_4 \in [-69.4, -21.4]
\]

But for \(\epsilon_1 = 0, \epsilon_2 = 1\) and \(\epsilon_3 = 1\),

\[
x = 13, \ y = 11, \ z = 143
\]

so \(t = -16.6 > -21.4\)!
...Nor [3] (of course!)

**Consider (Khalil Ghorbal)**

\[
\hat{t} = -4\hat{x} + 0.8\hat{y} - 79 \\
= -45.4 + 4.8\epsilon_1 + 4.8\epsilon_2 + 1.6\epsilon_3 + 12.8\epsilon_4 \in [-69.4, -21.4]
\]

But for \(\epsilon_1 = 0, \epsilon_2 = 1\) and \(\epsilon_3 = 1\),

\[
x = 13, \ y = 11, \ z = 143
\]

so \(t = -16.6 > -21.4!\)

**But...**

...there are hopefully nice multiplications (SDP based, to appear)
Consider...

\[
\begin{align*}
\hat{x} &= \epsilon_1 \\
\hat{y} &= \epsilon_2 \\
\hat{z} &= f(\hat{x}, \hat{y}) = x + y - \epsilon_4 \\
&= \epsilon_1 + \epsilon_2 - \epsilon_4 \\
&\in [-3, 3]
\end{align*}
\]

\[
\begin{align*}
\hat{x}' &= -\epsilon_1 \\
\hat{y}' &= \frac{1}{2} (\epsilon_3 + \epsilon_4) \\
\hat{z}' &= f(\hat{x}', \hat{y}') = x' + y' - \epsilon_4 \\
&= -\epsilon_1 + \frac{1}{2} (\epsilon_3 - \epsilon_4) \\
&\in [-2, 2]
\end{align*}
\]

Clearly...

The joint concretisations of \((\hat{x}, \hat{y})\) and of \((\hat{x}', \hat{y}')\) are the same (but with different dependencies), whereas the same future evaluation \(f\) does not give the same range on \((\hat{x}, \hat{y})\) and on \((\hat{x}', \hat{y}')\)
Partial conclusion

**Correctness**

- $[3]$ is definitely necessary when **functionals to be evaluated are discovered along the way** (as in static analysis)

**Remark on union**

- Partial order relation $\hat{x} \preceq \hat{y}$ if **all future evaluations** using $\hat{x}$ instead of $\hat{y}$ have smaller concretisation (can be characterized in a simpler manner see also Goubault/Putot 2008 [4])
- Our union operator is a **minimal upper bound** (under some conditions) for this order, reflecting some form of **optimality** under correctness criterion [3]
Correctness

- $[3]$ is definitely necessary when functionals to be evaluated are discovered along the way (as in static analysis).

Remark on union

- Partial order relation $\hat{x} \preceq \hat{y}$ if all future evaluations using $\hat{x}$ instead of $\hat{y}$ have smaller concretisation (can be characterized in a simpler manner see also Goubault/Putot 2008 [4])
- Our union operator is a minimal upper bound (under some conditions) for this order, reflecting some form of optimality under correctness criterion [3]

What about inner-approximations?
**Principle**

- Use more general dependency coefficients
  \[ \tilde{x} = \sum_{i=1}^{n} [a_i, b_i] \varepsilon_i \] (modal interval coefficients)
- Generalized intervals: \( x = [\underline{x}, \overline{x}] \), possibly with \( \underline{x} \geq \overline{x} \).

**First, recap of modal intervals**

- dual \( x = x^* = [\overline{x}, \underline{x}] \) and pro \( x = [\min(\underline{x}, \overline{x}), \max(\underline{x}, \overline{x})] \).
- \( x \) is proper (in \( \mathbb{IR} \)) if \( \underline{x} \leq \overline{x} \), otherwise improper
- Kaucher arithmetic extending classical interval arithmetic
  - For instance same addition
  - But \( [1, 2] \ast [1, -1] = [1, -1] \) whereas \( [1, 2] \ast \text{pro} [1, -1] = [2, -2] \)
Modal intervals/Quantifiers (à la Goldsztejn 2005 [1])

**Classical over-approximated interval computation**

All intervals are proper
\[(\forall x \in x) (\exists z \in z) (f(x) = z).\]

- Let \(f(x) = x^2 - x\), then \(f([2, 3]) = [2, 3]^2 - [2, 3] = [1, 7]\) is interpreted as \((\forall x \in [2, 3]) (\exists z \in [1, 7]) (f(x) = z)\).

**Inner-approximated computation**

All intervals are improper
\[(\forall z \in \text{pro } z) (\exists x \in \text{pro } x) (f(x) = z).\]

- **Application scope is limited** to expressions with no dependency between sub-expressions. An inner-approximation of \(f(x) = x^2 - x\) for \(x \in [2, 3]\) cannot be thus computed.
Inner- and outer-approximations

Example: inner multiplication (using Goldsztejn 2005 [1])

Let \( \hat{x} \) and \( \hat{y} \) be two affine forms (real coeff.) and \( z = x \times y \)

- An inner-approximation, with \((t_1, \ldots, t_n) = (0, \ldots, 0)\), is
  \[
  \hat{z} = \alpha_0^x \alpha_0^y + \sum_{i=1}^{n} (\alpha_i^x \alpha_0^y + \alpha_i^y \alpha_0^x) \varepsilon_i + \left( \sum_{j=1}^{n} (\alpha_i^x \alpha_j^y + \alpha_i^y \alpha_j^x) \varepsilon_j \right) \varepsilon_i
  \]
  - over-approximation of dependencies,
  - \( \alpha_i^z \) contains the tangent \( \frac{\partial z}{\partial \varepsilon_i} \)

An outer-approximation is

\[
\hat{z} = \alpha_0^x \alpha_0^y + \sum_{i=1}^{n} (\alpha_i^x \alpha_0^y + \alpha_i^y \alpha_0^x) \varepsilon_i + \left( \sum_{i=1}^{n} |\alpha_i^x| \right) \left( \sum_{i=1}^{n} |\alpha_i^y| \right) \varepsilon_{n+1},
\]

with a new noise symbol \( \varepsilon_{n+1} \): over-approximation by loss of dependency between linear terms and the non linear term.

The purely affine part of the product is the same.
Consider

\[ f(x) = x^2 - x \text{ when } x \in [2, 3] \text{ (real result } [2, 6]) \]

We find:

\[ \tilde{f}^\varepsilon (\varepsilon_1) = 3.75 + [1.5, 2.5] \varepsilon_1 \]

Inner-approximating concretization

\[ 3.75 + [1.5, 2.5][1, -1] = 3.75 + [1.5, -1.5] = [5.25, 2.25] \]

Outer-approximating concretization

\[ 3.75 + [1.5, 2.5][-1, 1] = 3.75 + [-2.5, 2.5] = [1.25, 6.25] \]

Affine arithmetic (over-approximation)

\[ x^2 - x = [3.75, 4] + 2\varepsilon_1 \text{ (concretization } [1.75, 6]) \]
Generalized affine forms for inner-approximation

Partial order on affine forms with interval coefficients

\( \tilde{x} \sqsubseteq \tilde{y} \) (“\( \tilde{x} \) is better than \( \tilde{y} \)” ) iff \( \forall i \geq 0, \alpha_i^x \sqsubseteq \alpha_i^y \).

\( \forall \tilde{x} \sqsubseteq \tilde{y} \Rightarrow \hat{\gamma}(\tilde{x}) \subseteq \hat{\gamma}(\tilde{y}) \) and \( \hat{\gamma}(\tilde{y}) \subseteq \hat{\gamma}(\tilde{x}) \)

Can be seen as extension of the “future evaluation” order on outer-approximations

Example

\( \tilde{x} = 1 + [2, 4] \varepsilon_1, \tilde{y} = 1 + [1, 5] \varepsilon_1 \Rightarrow \tilde{x} \sqsubseteq \tilde{y} \).

\( \hat{\gamma}(\tilde{x}) = \text{pro} (1 + [2, -2]) = [-1, 3], \hat{\gamma}(\tilde{y}) = [0, 2] \).

\( \hat{\gamma}(\tilde{x}) = 1 + [-4, 4] = [-3, 5], \hat{\gamma}(\tilde{y}) = [-4, 6] \)
Join and meet operations

**Join**

\[ \tilde{z} = \tilde{x} \cup \tilde{y} = (\alpha_0^x \cup \alpha_0^y) + (\alpha_1^x \cup \alpha_1^y)\varepsilon_1 + \ldots + (\alpha_n^x \cup \alpha_n^y)\varepsilon_n. \]

**Meet**

If for \( i \geq 0, \alpha_i^x \cap \alpha_i^y \neq \emptyset \), we can define an inner-approximation of the intersection by

\[ \tilde{z} = \tilde{x} \cap \tilde{y} = (\alpha_0^x \cap \alpha_0^y) + (\alpha_1^x \cap \alpha_1^y)\varepsilon_1 + \ldots + (\alpha_n^x \cap \alpha_n^y)\varepsilon_n. \]

Otherwise, the result is \( \bot \) (possible refinement by propagating instead the constraints induced on the \( \varepsilon_i \)).
Single inner-approximation versus joint inner-approximation versus future evaluations

Our joint concretization

The joint concretization has an a priori weak meaning

\[
\begin{align*}
x_1 &= 5 + \varepsilon_1 \\
x_2 &= 2 + \varepsilon_2 \\
x_3 &= x_1 x_2 \\
&= 10 + [1, 3] \varepsilon_1 + [4, 6] \varepsilon_2 \\
&\subseteq [5, 15] \subseteq [4, 18] \subseteq [3, 19]
\end{align*}
\]

\[\forall z \in [5, 15], \exists \epsilon_1, \epsilon_2, \quad z = x_1 x_2\]

But we can prove...

...that our formulas agree with [1] but also make all future evaluations correct (criterion [3])

Eric Goubault and Sylvie Putot
Inner and outer approximations

Technical conditions ensure that both 2-dim boxes are included in the concrete joint range:

\[
\begin{pmatrix}
  x_1 \\
  x_3
\end{pmatrix} = \begin{pmatrix}
  5 + \epsilon_1^\star + 0\epsilon_2 \\
  10 + [1, 3]\epsilon_1 + [4, 6]\epsilon_2^\star
\end{pmatrix} = \begin{pmatrix}
  [4, 6] \\
  [7, 13]
\end{pmatrix}
\]

\[
\begin{pmatrix}
  x_1 \\
  x_2
\end{pmatrix} = \begin{pmatrix}
  5 + \epsilon_1^\star + 0\epsilon_2 \\
  2 + 0\epsilon_1 + \epsilon_2^\star
\end{pmatrix} = \begin{pmatrix}
  [4, 6] \\
  [1, 3]
\end{pmatrix}
\]

So some surfaces are there inside the joint concretisation... but not possible to characterize a full 3D box inside...
Final conclusion

On correctness...

- For inner-approximations in our framework, criterion [2] is intractable in general:
  - for outer-approximations, still correct when losing dependencies
  - for inner-approximations, we have to outer-approximate dependencies
- The more rigid criterion [3] still applies!

We have a proven general inner-/outer-approximation calculus

- Of course, many details omitted ("splitting" for instance)
Can it be generalized to Taylor models?

Generalized *perturbed* affine forms
using $\epsilon$ symbols?

Floating-point and rounding error estimations
- Existing extension of the abstract domain (NSAD’05, SAS’06) for outer-approximation
- Problematic for inner-approximation

Faster-than-Kleene fixpoint computation
using policy iteration (CAV’05, ESOP’07)
Some references

[1] Alexandre Goldsztejn
Modal Intervals Revisited Part II: A Generalized Interval Mean-Value Extension HAL report number hal-00294222.

Inner Approximation of the Range of Vector-Valued Functions Reliable Computing (Springer), to appear.

Under-Approximations of Computations in Real Numbers Based on Generalized Affine Arithmetic. SAS 2007