Verification of Real-Time Systems
Numerical Abstractions

Jan Reineke

Advanced Lecture, Summer 2015
Recap: From Local to Global Correctness: Kleene Iteration

Idea for Correctness: Abstract Interpretation Cousot, Cousot 1977

Establish a description relation $\Delta$ between the concrete values and their descriptions with:

$$x \Delta a_1 \land a_1 \sqsubseteq a_2 \Rightarrow x \Delta a_2$$

Concretization: $\gamma a = \{ x | x \Delta a \}$

// returns the set of described values

Abstract Domain

Concrete Domain
Recap: Fixpoint Transfer Theorem

Let \((L, \leq)\) and \((L^\#, \leq^\#)\) be two lattices, \(\gamma : L^\# \to L\) a monotone function, and \(F : L \to L\) and \(F^\# : F^\# \to F^\#\) two monotone functions, with

\[
\forall l^\# \in L^\# : \gamma(F^#(l^#)) \geq F(\gamma(l^#)).
\]

Then:

\[
lfp F \leq \gamma(lfp F^\#).
\]
Approximations of an infinite set of points:

\[ \{ \ldots, (19, 77), \ldots, (20, 03), \ldots \} \]
Overview: Numerical Abstractions
Signs (Cousot & Cousot, 1979)

Approximations of an infinite set of points: from above

\[ x \leq 19; y \leq 77; x \geq 20; y \geq 03; x \geq ?; y \geq ? \]

From Below: dual combinations.

Trivial for finite states (liveness model-checking), more difficult for infinite states (variant functions).

Effective computable approximations of an infinite set of points.

Overview: Numerical Abstractions

Intervals (Cousot & Cousot, 1976)


Octagons


Polyhedra


Simple congruences

Overview: Numerical Abstractions
Octagons (Mine, 2001)

Effective computable approximations of an infinite set of points;

\begin{align*}
\begin{cases}
1 \leq x \leq 9 \\
x + y \leq 77 \\
1 \leq y \leq 9 \\
x - y \leq 99
\end{cases}
\end{align*}

\[ x \geq y \]
Overview: Numerical Abstractions

Polyhedra (Cousot & Halbwachs, 1978)

Effective computable approximations of an infinite set of points;

\[ \begin{align*}
19x + 77y &\leq 2004 \\
20x + 03y &\geq 0
\end{align*} \]

\[ \text{Expensive...} \]
Overview: Numerical Abstractions

Simple and Linear Congruences (Granger, 1989+1991)

Effective computable approximations of an infinite set of points;

Octagons

\[ \begin{align*}
& x \not\equiv 1 \mod 2 \\
& x \equiv 0 \mod 2 \\
& \begin{cases}
  x = 19 \mod 77 \\
  y = 20 \mod 99
\end{cases}
\]

\[ \begin{align*}
& 1x + 9y \equiv 7 \mod 8 \\
& 2x - 1y \equiv 9 \mod 9
\end{align*} \]
Numerical Abstractions

Which abstraction is the most precise?

*Depends on questions you want to answer!*
Numerical Abstractions

Which abstraction is the most precise?

*Depends on questions you want to answer!*
Partial Order of Abstractions

- Polyhedra
  - Octagons
    - Intervals
      - Constants
      - Signs
  - Linear Congruences
    - Simple Congruences
      - Parity
Partial Order of Abstractions

Relational domains

- Polyhedra
- Octagons
- Linear Congruences
- Intervals
- Simple Congruences
- Constants
- Signs
- Parity

Independent attribute/non-relational domains
Characteristics of Non-relational Domains

- Non-relational/independent attribute abstraction:
  - Abstract each variable separately
    \[ (\mathcal{P}(\mathbb{Z}), \subseteq) \xrightarrow{\gamma} (\text{NUMERICAL}, \sqsubseteq) \]
  - Maintains no relations between variable values
- Can be lifted to an abstraction of valuations of multiple variables in the expected way:
  \[ (\mathcal{P}(\text{Vars} \rightarrow \mathbb{Z}), \subseteq) \xrightarrow{\gamma_1} (\text{Vars} \rightarrow \mathcal{P}(\mathbb{Z}), \subseteq) \xrightarrow{\gamma_2} (\text{Vars} \rightarrow \text{NUMERICAL}, \sqsubseteq) \]

\[ \alpha_2(f) := \lambda x \in \text{Vars}.\alpha(f(x)) \quad \gamma_2(f^\#) := \lambda x \in \text{Vars}.\gamma(f^\#(x)) \]
The Interval Domain

Abstracts sets of values by enclosing interval

\[ \text{Interval} = \{ [l, u] \mid l \leq u, l \in \mathbb{Z} \cup \{ -\infty \}, u \in \mathbb{Z} \cup \{ \infty \}\} \cup \{ \bot \} \]

where \( \leq \) is appropriately extended from \( \mathbb{Z} \times \mathbb{Z} \) to \( (\mathbb{Z} \cup \{ -\infty \}) \times (\mathbb{Z} \cup \{ \infty \}) \)

Intervals are ordered by inclusion:

\[ \bot \sqsubseteq x \quad \forall x \in \text{Interval} \]

\[ [l, u] \sqsubseteq [l', u'] \text{ if } l' \leq l \land u \leq u' \]

\((\text{Interval}, \sqsubseteq)\) forms a complete lattice.
Concretization and Abstraction of Intervals

- **Concretization:**
  
  \[ \gamma(\bot) = \emptyset \]
  
  \[ \gamma([l, u]) = \{ n \in \mathbb{Z} \mid l \leq n \leq u \} \]

- **Abstraction:**

  \[ \alpha(\emptyset) = \bot \]
  
  \[ \alpha(S) = [\inf S, \sup S] \]

They form a Galois connection.
Interval Arithmetic

Calculating with Intervals:

\[
\begin{align*}
[a, b] + [c, d] &= [a + c, b + d] \\
[a, b] - [c, d] &= [a - d, b - c] \\
[a, b] \times [c, d] &= \min(ac, ad, bc, bd), \max(ac, ad, bc, bd) \\
[a, b] \div [c, d] &= [a, b] \times \frac{1}{d}, \frac{1}{c}, 0 \notin [c, d] \\
\end{align*}
\]
Example: Interval Analysis

\[\begin{align*}
x & \rightarrow [0,3] & x & \rightarrow [0,2] & x & \rightarrow [0,1] & x & \rightarrow [0,0] \\
y & \rightarrow [3,7] & y & \rightarrow [3,5] & y & \rightarrow [3,3] & y & \rightarrow \text{bottom}
\end{align*}\]

Would Octagons determine that \( y \) must be 7 at program point 5?

\[\begin{align*}
\text{Neg}(x < 3) & \rightarrow 5 \\
\text{Pos}(x < 3) & \rightarrow 1 \\
y = y+1 & \rightarrow 2 \\
x = x+1 & \rightarrow 3 \\
y = 2 \times x & \rightarrow 4 \\
\end{align*}\]

Imprecise due to non-relational analysis
Intervals, Hasse diagram

Ascending chain condition is not satisfied!
→ Kleene iteration is not guaranteed to terminate!
Example: Interval Analysis

\[ x \mapsto \bot \]
\[ x \mapsto [0, 0] \]
\[ x \mapsto [0, 1] \]
\[ \ldots \]
\[ x \mapsto [0, 1000] \]

1000 iterations later
Solution: Widening
“Enforce Ascending Chain Condition”

- Widening enforces the ascending chain condition during analysis.
- Accelerates termination by moving up the lattice more quickly.
- May yield imprecise results…

\[ \{ x \mid x \supseteq \text{lfp } F \} \]
A widening $\nabla$ is an operator $\nabla : D \times D \rightarrow D$ such that

1. **Safety**: $x \sqsubseteq (x \nabla y)$ and $y \sqsubseteq (x \nabla y)$

2. **Termination**:

   For all ascending chains $x_0 \sqsubseteq x_1 \sqsubseteq \ldots$ the chain

   $y_0 = x_0$

   $y_{i+1} = y_i \nabla x_{i+1}$

   is finite.
Widening Operator for Intervals

Simplest solution:

\[
\bot \nabla x = x \nabla \bot = x
\]

\[
[l, u] \nabla [l', u'] = \begin{cases} l & : l' \geq l \quad \{ u & : u' \leq u \\
\infty & : l' < l \quad \infty & : u' > u \end{cases}
\]

Example:

\[
\]

\[
[3, 5] \nabla [4, 5] = [3, 5]
\]

\[
[3, 5] \nabla [4, 6] = [3, \infty]
\]

\[
[3, 5] \nabla [2, 6] = [-\infty, \infty]
\]
Example Revisited: Interval Analysis with Simple Widening

Standard Kleene Iteration:
\[ \perp \leq F(\perp) \leq F^2(\perp) \leq F^3(\perp) \leq \ldots \]

Kleene Iteration with Widening:
\[ F_\triangledown(x) := x \triangledown F(x) \]
\[ \perp \leq F_\triangledown(\perp) \leq F^2_\triangledown(\perp) \leq F^3_\triangledown(\perp) \leq \ldots \]

\[ \begin{align*}
  x &\mapsto [0, 0] \\
  x &\mapsto [0, \infty]
\end{align*} \]

\[ \text{Do we need to apply widening at all program points?} \]

\[ \rightarrow \text{Quick termination but imprecise result!} \]
More Sophisticated Widening for Intervals

Define set of jump points (barriers) based on constants appearing in program, e.g.:

\[ \mathcal{J} = \{-\infty, 0, 1, 1000, \infty\} \]

Intuition: “Don’t jump to –infty, +infty immediately but only to next jump point.”

\[ [l, u] \nabla [l', u'] = \begin{cases} 
    l & : l' \geq l \\
    \max\{x \in \mathcal{J} \mid x \leq l'\} & : l' < l' \\
    u & : u' \leq u \\
    \min\{x \in \mathcal{J} \mid x \geq u'\} & : u' > u 
\end{cases} \]
Example Revisited:
Interval Analysis with Sophisticated Widening

$\rightarrow$ More precise, potentially terminates more slowly.
Another Example: Interval Analysis with Sophisticated Widening

\[ x \mapsto [0, 0] \]
\[ x \mapsto [0, 1] \]
\[ x \mapsto [0, 1000] \]

\[ y \mapsto [2, 2] \]
\[ y \mapsto [2, 1000] \]
\[ y \mapsto [2, \infty] \]

Would be \([2, 2000]\) in least fixed point, but 2000 does not appear in the program…
Narrowing: Recovering Precision

- Widening may yield imprecise results by overshooting the least fixed point.
- Narrowing is used to approach the least fixed point from above.

Possible problem: infinite descending chains
Is it really a problem?
Narrowing:
Recovering Precision

Widening terminates at a point \( x \sqsupseteq \text{lfp } F \).
We can iterate:
\[
\begin{align*}
x_0 &= x \\
x_{i+1} &= F(x_i) \cap x_i
\end{align*}
\]

Safety:
By monotonicity we know \( F(x) \sqsupseteq F(\text{lfp } F) = \text{lfp } F \).
By induction we can easily show that \( x_i \sqsupseteq \text{lfp } F \) for all \( i \).

Termination:
Depends on existence of infinite descending chains.
Narrowing: Formal Requirement

A narrowing $\Delta$ is an operator $\Delta : D \times D \rightarrow D$ such that

1. **Safety**: $l \sqsubseteq x$ and $l \sqsubseteq y \implies l \sqsubseteq (x \Delta y) \sqsubseteq x$

2. **Termination**:
   
   for all descending chains $x_0 \sqsupseteq x_1 \sqsupseteq \ldots$ the chain
   
   $y_0 = x_0$
   
   $y_{i+1} = y_i \Delta x_{i+1}$

   is finite.

*Is $\sqcap$ (“meet”) a narrowing operator on intervals?*
Another Example Revisited: Interval Analysis with Widening and Narrowing

Result after Widening:
\[
\begin{align*}
x & \mapsto [0, 0] \\
x & \mapsto [0, 1] \\
x & \mapsto [0, 1000] \\
y & \mapsto [2, 2] \\
y & \mapsto [2, 1000] \\
y & \mapsto [2, \infty]
\end{align*}
\]

Result after Narrowing:
\[
\begin{align*}
x & \mapsto [1000, 1000] \\
y & \mapsto [3, 2001] \\
x & \mapsto [0, 999] \\
x & \mapsto [1, 1000] \\
y & \mapsto [2, 2000]
\end{align*}
\]

→ Precisely the least fixed point!
Applications of Numerical Domains

As input to other analyses:
- Cache Analysis
- Dependencies between memory accesses

Loop Bound Analysis:
- Instrument program with loop iteration counters
- Determine maximal value of counter
- Requires relational analysis
Reduction: Loop Bound Analysis to Value Analysis

Instrument program with counters of loop iterations and other interesting events
Summary

- Interval Analysis: A non-relational value analysis
- Widenings for termination in the presence of Infinite Ascending Chains
- Narrowings to recover precision
- Basic Approach to Loop Bound Analysis based on Value Analysis
Outlook

- Cache Abstractions
- Schedulability Analysis
- Cache-Related Preemption Delay
- Predictable Microarchitectures