An Abstract Domain to Infer Linear Absolute Value Equalities

Liqian Chen   Banghu Yin   Dengping Wei   Ji Wang

National University of Defense Technology, China

25/08/2021 – TASE 2021

Overview

- Motivation
- An abstract domain of linear absolute value equalities
- Implementation and Experiments
- Conclusion

Motivation

Motivation

Goal: numerical static analysis

discover numerical properties of a program statically and automatically

Applications:

- check for runtime errors (e.g., arithmetic overflows, division by zero, array out-of-bounds, etc.)
- optimize programs
- ...

Theoretical framework: **abstract interpretation**

to design static analyses that are

- sound by construction (no behavior is omitted)
- approximate (trade-off between precision and efficiency)

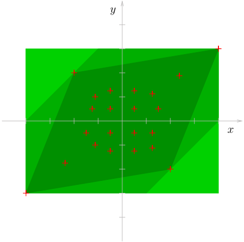
Motivation

Abstract domain: key ingredient of abstract interpretation

- a specific kind of computer-representable properties
  - e.g., a family of constraints
- sound (but maybe incomplete) algorithms for semantic actions
  - e.g., join, meet, widening,...

Numerical abstract domains

- infer relationships among numerical variables
- examples
  - non-relational: **intervals** ($a \leq x \leq b$)
  - weakly relational: **octagons** ($\pm x \pm y \leq c$)
  - strongly relational: **polyhedra** ($\Sigma_k a_k x_k \leq b$)
  - ...

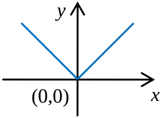


Motivation

Convexity limitations: a motivating example

```
float x, y;
if (x ≥ 0  /* |x| == x */ ) { y := x; }
else       /* |x| == −x */  { y := −x; }
① if (x ≥ 0 /* |x| == x */ )  { assert(y == x);  }
else       /* |x| == −x */  { assert(y == −x); }
}
```

Loc	PolkaEq	AVI	AVE
①	$\top$	$y == \|x\| \wedge$ $y == \|y\|$	$y == \|x\| \wedge$ $y == \|y\|$

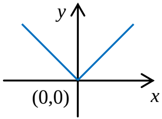


Motivation

Absolute Value (AV): $y = |x|$

- piecewise linear expressiveness

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$



Possible applications

- to encode disjunctions of linear constraints in the program
  - $(x = -1 \vee x = 1) \iff |x| = 1$
- AV functions in C: $abs(), fabs(), \ldots$
- Min/Max functions in C: $fmax(), fmin(), \ldots$
  - e.g., $\max(x, y) = \frac{1}{2}(|x - y| + x + y)$
- ReLU function in neural network
  - $ReLU(x, 0) = \frac{1}{2}(|x| + x)$

Motivation

The domain of linear AV inequalities: $(\Sigma_k a_k x_k + \Sigma_k b_k |x_k| \leq b)$ [Chen et al. ESOP'11]

- idea: extending polyhedra domain $(\Sigma_k a_k x_k \leq b)$ with absolute value
- pros: piecewise linear expressiveness
- cons: exponential complexity

New idea: A domain of linear AV equalities: $(\Sigma_k a_k x_k + \Sigma_k b_k |x_k| = b)$

- goal: less costly but with non-convex expressiveness
- idea: extending the affine (linear) equality domain with absolute value
  - affine (linear) equality domain $(\Sigma_k a_k x_k = b)$: scalable, widely used in practice

An abstract domain of linear absolute value equalities

The AVE abstract domain

An abstract domain of linear absolute value equalities (AVE)

- goal: to infer linear equality relations among values and absolute values of program variables

$$\boxed{\Sigma_k a_k x_k + \Sigma_k b_k |x_k| = c}$$

Domain representation for domain element **P**

- AVE representation: a linear AVE system $Ax + B|x| = c$
- semantics: $\gamma(\mathbf{P}) = \{x \in \mathbb{R}^n \colon Ax + B|x| = c\}$

Topological properties: can be non-convex, even unconnected

- a (possibly empty) affine space (within the orthant boundary) in each orthant
- e.g., $y = |x|$

The AVE abstract domain (representation)

Expressiveness limitation: $\Sigma_k a_k x_k + \Sigma_k b_k |x_k| = c$

- $|\cdot|$ applies to only (single) variables rather than expressions

An example: $\max(x, y) = z$, i.e.,

$$z = \begin{cases} x & \text{if } y \leq x \\ y & \text{if } x < y \end{cases}$$

Expressiveness lifting

- introduce new auxiliary variables to denote expressions inside the AV function
- e.g., $w = x - y \quad \wedge \quad \frac{1}{2}(|w| + x + y) = z$

The AVE abstract domain (HLCP representation)

Horizontal Linear Complementary Problem (HLCP)

- given $M, N \in \mathbb{Q}^{m\times n}$ and $q \in \mathbb{Q}^m$, find $x^+, x^- \in \mathbb{Q}^n$ so that

$$Mx^+ + Nx^- = q \qquad (1)$$
$$x^+, x^- \geq 0 \qquad (2)$$
$$(x^+)^T x^- = 0. \qquad (3)$$

Complementarity condition: condition (3), which implies

$$x_i^+ x_i^- = 0 \qquad \text{for } i = 1, \ldots, n$$

The AVE abstract domain (HLCP representation)

Equivalence of AVEs and HLCPs

Let $x^+ = (\max(x_i, 0))_{i=1}^n$ and $x^- = (\max(-x_i, 0))_{i=1}^n$, so that

$$x^+ \geq 0, x^- \geq 0, (x^+)^T x^- = 0$$

and

$$x = x^+ - x^- \qquad |x| = x^+ + x^-$$
$$x^+ = \frac{1}{2}(x + |x|) \qquad x^- = \frac{1}{2}(|x| - x).$$

Then, AVE

$$Ax + B|x| = c$$

can be reformulated as the following HLCP:

$$(A + B)x^+ + (B - A)x^- = c$$
$$x^+, x^- \geq 0 \wedge (x^+)^T x^- = 0$$

The AVE abstract domain (HLCP representation)

Domain representation (HLCP constraints):

$$Ax^{\pm} = b, \quad x^{\pm} \geq 0, (x^{+})^{T} x^{-} = 0 \qquad (x^{\pm} \in \{x^{+}, x^{-}\})$$

- linear system part: $Ax^{\pm} = b$ in reduced row echelon form

Definition (Reduced row echelon form)

$Ax = b$ where A is of size $m \times n$, is in *reduced row echelon* form if

1) Every row i_0 of A has at least one non-zero entry
2) Let $x^{\pm}_{j_0}$ be the leading variable of row i_0 of A. Then
   - $A_{i_0 j_0} = 1$
   - for all $i > i_0, j \leq j_0, A_{ij} = 0$
   - for all $i < i_0, A_{i j_0} = 0$.

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 2 & 4 \\ 0 & 0 & 1 & 0 & 7 \end{pmatrix}$$

- complementary condition part: standard, with no need of being stored explicitly

The AVE abstract domain (HLCP representation)

Double Description Method for Polyhedra

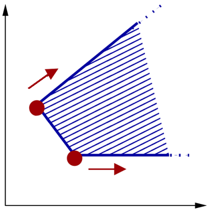
Theorem (Minkowski-Weyl Theorem)

The set $P \subseteq \mathbb{R}^n$ is a polyhedron, iff it is finitely generated, i.e., there exist finite sets $V, R \in \mathbb{R}^n$ such that P can be generated by (V, R):

$$P = \left\{ \sum_{i=1}^{|V|} \lambda_i V_i + \sum_{j=1}^{|R|} \mu_j R_j \;\middle|\; \forall i, \lambda_i \geq 0, \forall j, \mu_j \geq 0, \sum_{i=1}^{|V|} \lambda_i = 1 \right\}$$

Dual representations

- constraint representation: $Ax \leq b$
  - e.g., $\{-y \leq -1, x - y \leq 1, -x - y \leq -3\}$
- generator representation: $G = (V, R)$
  - e.g., $(\{(2,1),(1,2)\}, \quad \{(0,1),(1,1)\})$



Dual conversion: Chernikova's algorithm

The AVE abstract domain (HLCP representation)

Computing complementary generators for HLCP:

$$Mx^+ + Nx^- = c \wedge x^+, x^- \geq 0 \wedge (x^+)^T x^- = 0$$

Step1: $G \leftarrow$ Polyhedra.Cons2Gens $(Mx^+ + Nx^- = c \wedge x^+, x^- \geq 0)$
Step2: $G^c \leftarrow \{g \in G \mid g \text{ satisfies } (x_g^+)^T x_g^- = 0\}$

Dual representations

- HLCP constraint representation:
  $$Mx^+ + Nx^- = c \wedge x^+, x^- \geq 0 \wedge \ (x^+)^T x^- = 0$$
- complementary generator representation: $G^c = (V^c, R^c)$

The AVE abstract domain (operations)

How to implement AVE domain operations for static analysis

- maintain the map between abstract environments over x and abstract environments over x^+, x^-:

$$x = x^+ - x^-, \qquad |x| = x^+ + x^-$$
$$x^+ = \frac{1}{2}(x + |x|), \qquad x^- = \frac{1}{2}(|x| - x)$$

  where x^+, x^- satisfy $x^+ \geq 0, x^- \geq 0, (x^+)^T x^- = 0$

- compute $G^c = (V^c, R^c)$, the set of complementary generators of HLCP system (when needed):

$$Mx^+ + Nx^- = b$$
$$x^+ \geq 0, x^- \geq 0, (x^+)^T x^- = 0$$

The AVE abstract domain (operations)

Domain operations

1. lattice operations
   - meet: $\mathbf{P} \sqcap \mathbf{P}'$ is an AVE domain element whose HLCP constraint representation is

$$\begin{aligned} Mx^{+} + Nx^{-} &= b \\ M'x^{+} + N'x^{-} &= b' \\ x^{+} \geq 0, x^{-} \geq 0, (x^{+})^{T} x^{-} &= 0 \end{aligned}$$

     - where $\{Mx^{+} + Nx^{-} = b, M'x^{+} + N'x^{-} = b'\}$ can be converted into reduced row echelon form via Gaussian elimination

The AVE abstract domain (operations)

Domain operations

1. lattice operations
   - join: $\mathbf{P} \sqcup \mathbf{P}'$ is the least AVE element containing $\mathbf{P}$ and $\mathbf{P}'$, whose set of complementary generators is the union of those of $\mathbf{P}$ and $\mathbf{P}'$: $(V^c \cup V'^c, R^c \cup R'^c)$.
     1. Compute the complementary generator representation (V^c, R^c), (V'^c, R'^c) respectively for $\mathbf{P}$ and $\mathbf{P}'$;
     2. Compute $(V^c \cup V'^c, R^c \cup R'^c)$, and suppose $V^c \cup V'^c = \{v_1, \ldots, v_p\}, R^c \cup R'^c = \{r_1, \ldots, r_q\}$;
     3. Project out variables $\lambda_j(j = 1, \ldots, p), \mu_k(k = 1, \ldots, q)$ (via Gaussian elimination) from the following system:
        $$\begin{cases} (x^+ \ x^-)^T = \sum_{j=1}^{p}(\lambda_j v_j) + \sum_{k=1}^{q}(\mu_k r_k) \\ \sum_{j=1}^{p} \lambda_j = 1 \end{cases}$$
        Suppose we get $\hat{M}x^+ + \hat{N}x^- = \hat{b}$
     4. Finally, the resulting HLCP representation of $\mathbf{P} \sqcup \mathbf{P}'$ is:
        $$\hat{M}x^+ + \hat{N}x^- = \hat{b}$$
        $$x^+ \geq 0, x^- \geq 0, (x^+)^T x^- = 0$$

The AVE abstract domain (example: join)

float x, y;
if $(x \geq 0$ /* $|x| == x$ */) { $y := x$; ① }
else /* $|x| == -x$ */ { $y := -x$; ② }
③ ...

Loc	AVE/HLCP constraints	Complementary generators
①	$\mathbf{P} = \{(x\ y)^T \mid x - y = 0, \lvert x \rvert = x\} = \{(x^+\ x^-\ y^+\ y^-)^T \mid x^+ - y^+ + y^- = 0, x^- = 0, x^{\pm} \geq 0, y^{\pm} \geq 0, x^+x^- = 0, y^+y^- = 0\}$	$\left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}\right\}, \left\{\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}\right\}\right)$
②	$\mathbf{P}' = \{(x\ y)^T \mid -x - y = 0, \lvert x \rvert = -x\} = \{(x^+\ x^-\ y^+\ y^-)^T \mid x^- - y^+ + y^- = 0, x^+ = 0, x^{\pm} \geq 0, y^{\pm} \geq 0, x^+x^- = 0, y^+y^- = 0\}$	$\left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}\right\}, \left\{\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}\right\}\right)$
③	?	?

The AVE abstract domain (example: join)

float x, y;
if $(x \geq 0$ /* $|x| == x$ */) { $y := x$; ① }
else /* $|x| == -x$ */ { $y := -x$; ② }
③ ...

Loc	AVE/HLCP constraints	Complementary generators
①	$\mathbf{P} = \{(x\ y)^T \mid x - y = 0, \lvert x\rvert = x\} = \{(x^+\ x^-\ y^+\ y^-)^T \mid x^+ - y^+ + y^- = 0, x^- = 0, x^\pm \geq 0, y^\pm \geq 0, x^+x^- = 0, y^+y^- = 0\}$	$\left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}\right\}, \left\{\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}\right\}\right)$
②	$\mathbf{P}' = \{(x\ y)^T \mid -x - y = 0, \lvert x\rvert = -x\} = \{(x^+\ x^-\ y^+\ y^-)^T \mid x^- - y^+ + y^- = 0, x^+ = 0, x^\pm \geq 0, y^\pm \geq 0, x^+x^- = 0, y^+y^- = 0\}$	$\left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}\right\}, \left\{\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}\right\}\right)$
③	?	$\left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}\right\}, \left\{\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}\right\}\right)$

The AVE abstract domain (example: join)

float x, y;
if $(x \geq 0$ /* $|x| == x$ */) { $y := x$; ① }
else /* $|x| == -x$ */ { $y := -x$; ② }
③ ...

Loc	AVE/HLCP constraints	Complementary generators
①	...	...
②	...	...
③	?	$\left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}\right\}, \left\{\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}\right\}\right)$

Projecting out λ_1, μ_1, μ_2 (wherein $\lambda_1 = 1$) from

$$\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} = \lambda_1 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \mu_1 \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \mu_2 \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

will result in $x^+ + x^- - y^+ = 0, \quad y^- = 0$.

The AVE abstract domain (example: join)

```
float x, y;
if (x ≥ 0  /* |x| == x */ ) { y := x; ① }
else       /* |x| == −x */  { y := −x; ② }
③ ...
```

Loc	AVE/HLCP constraints	Complementary generators
①	$\mathbf{P} = \{(x\ y)^T \mid x - y = 0, \lvert x\rvert = x\} =$ $\{(x^+\ x^-\ y^+\ y^-)^T \mid x^+ - y^+ + y^- = 0, x^- = 0,$ $x^\pm \geq 0, y^\pm \geq 0, x^+x^- = 0, y^+y^- = 0\}$	$\left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}\right\}, \left\{\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}\right\}\right)$
②	$\mathbf{P}' = \{(x\ y)^T \mid -x - y = 0, \lvert x\rvert = -x\} =$ $\{(x^+\ x^-\ y^+\ y^-)^T \mid x^- - y^+ + y^- = 0, x^+ = 0,$ $x^\pm \geq 0, y^\pm \geq 0, x^+x^- = 0, y^+y^- = 0\}$	$\left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}\right\}, \left\{\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}\right\}\right)$
③	$\mathbf{P}' = \{(x\ y)^T \mid y = \lvert x\rvert,\ \lvert y\rvert = y\} =$ $\{(x^+\ x^-\ y^+\ y^-)^T \mid x^+ + x^- - y^+ = 0, y^- = 0,$ $x^\pm \geq 0, y^\pm \geq 0, x^+x^- = 0, y^+y^- = 0\}$	$\left(\begin{pmatrix} x^+ \\ x^- \\ y^+ \\ y^- \end{pmatrix} : \left\{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}\right\}, \left\{\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}\right\}\right)$

The AVE abstract domain (operations)

Domain operations

2. transfer functions
   - test transfer function: $\tau[\![cx + d|x| = e]\!]^\sharp(\mathbf{P})$, whose HLCP system is defined as

$$\begin{aligned} Mx^+ + Nx^- &= b \\ (c+d)x^+ + (d-c)x^- &= e \\ x^+ \geq 0, x^- \geq 0, (x^+)^T x^- &= 0 \end{aligned}$$

     - where $\{Mx^+ + Nx^- = b, (c+d)x^+ + (d-c)x^- = e\}$ can be converted into reduced row echelon form via Gaussian elimination
   - projection: $\tau[\![x_j := random()]\!]^\sharp(\mathbf{P})$, can be implemented by projecting out x_j^+, x_j^- via Gaussian elimination from

$$Mx^+ + Nx^- = b$$

   - assignment transfer function: $\tau[\![x_j := \Sigma_i a_i x_i + \Sigma_i b_i |x_i| + c]\!]^\sharp(\mathbf{P})$, can be implemented as:

$$\left(\tau[\![x_j := random()]\!]^\sharp \circ \tau[\![\Sigma_i a_i x_i + \Sigma_i b_i |x_i| + c - x_j' = 0]\!]^\sharp(\mathbf{P})\right)[x_j'/x_j]$$

The AVE abstract domain (operations)

Domain operations

3. Extrapolations (Widening):
   - the lattice of linear equalities (in a program) has finite height, and thus we do not need a widening operation for the domain of linear equalities.
   - the intersection of an AVE element with each orthant, results in an affine space, i.e., an element in the domain of linear equalities.
   - the number of the orthants are finite (for a given program)

   $\leadsto$ we also do not need a widening operation for the AVE domain. At each widening point, we use the join operator $\sqcup$ instead of the widening.

Implementation and Experiments

Prototype

Prototype implementation rAVE using:

- GMP (the GNU Multiple Precision arithmetic library)
  - to guarantee the soundness of the implementation
- NewPolka: a rational implementation of the polyhedra domain
  - for Chernikova's algorithm

Interface:

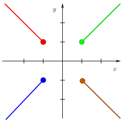
- plugged into the APRON library [Jeannet Miné]
- programs analyzed with INTERPROC [Jeannet et al.]

Comparison with

- PolkaEq [Jeannet]: the linear equality domain in APRON
- rAVI: the domain of linear absolute value inequalities [Chen et al. ESOP11]

Example analyses

real x, y;
assume $x = 1$ or $x = -1$;
assume $y = 1$ or $y = -1$;
while ($true$) {
① if ($x \geq 0$ /* $|x| == x$ */) { $x := x + 1$; }
  else /* $|x| == -x$ */ { $x := x - 1$; }
  if ($y \geq 0$ /* $|y| == y$ */) { $y := y + 1$; }
  else /* $|y| == -y$ */ { $y := y - 1$; }
}



Loc	PolkaEq	rAVE	rAVI
①	$\top$	$\|x\| = \|y\|$	$\|x\| = \|y\| \wedge \|x\| \geq 1$

Preliminary experimental results

Program	PolkaEq		rAVE		rAVI		Invariant	
	#iter.	t(ms)	#iter.	t(ms)	#iter.	t(ms)	PolkaEq vs. rAVE	rAVE vs. rAVI
MotivEx	1	3.3	1	4.0	1	5.1	$\sqsupset$	$=$
AVtest1	3	7.1	4	11.2	3	12.3	$\sqsupset$	$\sqsupset$
Complexity_cav08	3	4.4	4	14.4	4	20.7	$\sqsupset$	$\sqsupset$
Synergy1	3	5.3	3	17.3	4	30.9	$\sqsupset$	$=$
Reverse	3	4.0	3	5.6	4	8.7	$\sqsupset$	$=$
Recwhile	3	3.6	7	24.5	7	31.2	$\sqsupset$	$\sqsupset$
Speed_popl09	3	5.5	4	25.0	4	30.3	$\sqsupset$	$\sqsupset$

- These programs involve non-convex behaviors (such as absolute value functions, max functions, disjunctions, etc.) that are out of the expressiveness of convex domains (including PolkaEq)
- rAVE outputs 1∼6 linear AV equality invariants for each example at loop head
- rAVI also infers certain linear inequalities and linear AV inequalities, which are out of the expressiveness of rAVE

Conclusion

Summary:

- a new abstract domain: **linear absolute value equalities** (AVE)

$$(\Sigma_k a_k x_k + \Sigma_k b_k |x_k| = c)$$

  - idea: extend the affine equality domain with **absolute value**
    - can express non-convex (even unconnected) properties
  - key:
    - making use of the equivalence between AVEs and HLCPs
    - maintaining the **reduced row echelon form** for the linear system part of HLCP representation

    ⇝ at most $2n$ linear AV equalities for a program involving n variables

Future Work

- **for precision**
  - introducing automatically auxiliary variables inside the AV function
  - combining the AVE abstract domain with the interval domain
- **more experiments** on large realistic programs