An Abstract Domain to Discover Interval Linear Equalities

Liqian Chen[1,2] Antoine Miné[1,3] Ji Wang[2] Patrick Cousot[1,4]

[1]École Normale Supérieure, Paris, France
[2]National Lab. for Parallel and Distributed Processing, Changsha, China
[3]CNRS, France
[4]CIMS, New York University, New York, NY, USA

17/01/2010 – VMCAI 2010

Overview

- Motivation
- The abstract domain of interval linear equalities
- Early experimental results
- Conclusion

Motivation

Numerical static analysis by abstract interpretation

Numerical static analysis

- discover numerical properties of a program statically and automatically

Theoretical framework: **abstract interpretation**

to design static analyses that are

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

Numerical abstract domains

- infer relationships among numerical variables
- examples
  - Intervals ($a \leq x \leq b$), Octagons ($\pm x \pm y \leq c$), Polyhedra ($\Sigma_k a_k x_k \leq b$)

Motivation

Interval (mathematics)

- to model uncertainty, inexactness
- real-life systems with interval data

Interval coefficients in static analysis:

- interval-based abstractions for programs [Miné 06]
  - non-linear operations: $x * y \rightsquigarrow [\underline{x}, \overline{x}] \times y$
  - floating-point arithmetic:
    $x \oplus_{f,r} y \rightsquigarrow [1-\epsilon, 1+\epsilon] \times x + [1-\epsilon, 1+\epsilon] \times y + [-\varepsilon, \varepsilon]$
- analysis using floating-point implementations
  - real/rational numbers in the analyzed program:
    $\frac{1}{10} \rightsquigarrow [0.99..., 0.10...]$
  - e.g., floating-point convex polyhedra [Chen Miné Cousot 08]

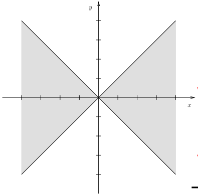
Motivation (cont.)

Interval coefficients for relational abstract domains:

- as the **constant term**: $\Sigma_k a_k x_k = [c_1, c_2]$
  - $c_1 = c_2 \, (c_1, c_2 \in \mathbb{R})$: affine equality
  - $c_1 = -\infty \vee c_2 = +\infty$: linear inequality
  - $c_1 \neq c_2 \, (c_1, c_2 \in \mathbb{R})$: linear stripe
- as **variable coefficients**: non-convex
  - interval polyhedra domain $(\Sigma_k [a_k, b_k] x_k \leq c)$ [Chen et al. 09]
    - rely a lot on LP solvers

$\rightsquigarrow$ A new domain: $(\Sigma_k [a_k, b_k] x_k = [c, d])$

- but lightweight



$$\begin{array}{r} -\mathbf{1} * x + y = 0 \\ \vdots \\ \vee\ \mathbf{0} * x + y = 0 \\ \vdots \\ \vee\ \mathbf{1} * x + y = 0 \\ \hline [-1, 1] x + y = 0 \end{array}$$

Motivation (cont.)

The affine equality domain (Karr's domain, $\Sigma_k a_k x_k = c$) [Karr 76]

- **features**: finite-height, polynomial-time, relational
- **problem**: rational implementations $\rightsquigarrow$ exponentially large numbers
- **our idea**: use floating-point implementations
  - obstacle: pervasive rounding errors
  - e.g., normalizing $3x + y = 1 \rightsquigarrow x + \frac{1}{3}y = \frac{1}{3}$

    $\frac{1}{3}$ (non-representable in floating-point) $\rightsquigarrow$ $[0.33\ldots0,\ 0.33\ldots5]$

affine equality		interval linear equality
$\Sigma_k a_k x_k = c$	$\rightsquigarrow$	$\Sigma_k [\underline{a}_k, \overline{a}_k] x_k = [\underline{c}, \overline{c}]$
$a_k \in \mathbb{Q}$		$\underline{a}_k, \overline{a}_k, \underline{c}, \overline{c} \in \mathbb{F}$

The abstract domain of Interval Linear Equalities (ILE)

Preliminaries

Interval linear system $\mathbf{A}x = \mathbf{b}$

- interval matrix $\mathbf{A} = [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{m\times n} : \underline{A} \leq A \leq \overline{A}\}$
  - where $\underline{A} \in (\mathbb{R} \cup \{-\infty\})^{m\times n}, \overline{A} \in (\mathbb{R} \cup \{+\infty\})^{m\times n}$
- interval vector $\mathbf{b}$: one-column interval matrix
- y is a weak solution of $\mathbf{A}x{=}\mathbf{b}$, if it satisfies $Ay{=}b$ for some $A \in \mathbf{A}$, $b \in \mathbf{b}$

Theorem (From itv linear equalities to linear inequalities: orthant partitioning)

Let $\Sigma_{j=1}^{n}[\underline{A}_{ij}, \overline{A}_{ij}]x_j = [\underline{b}_i, \overline{b}_i]$ be the i-th row of $\mathbf{A}x = \mathbf{b}$. Then $x \in \mathbb{R}^n$ is a weak solution of $\mathbf{A}x = \mathbf{b}$ iff both linear inequalities

$$\begin{cases} \Sigma_{j=1}^{n} A'_{ij}x_j \leq \overline{b}_i \\ -\Sigma_{j=1}^{n} A''_{ij}x_j \leq -\underline{b}_i \end{cases}$$

hold for all $i = 1, \ldots, m$ where

$$A'_{ij} = \begin{cases} \underline{A}_{ij} & \text{if } x_j > 0 \\ 0 & \text{if } x_j = 0 \\ \overline{A}_{ij} & \text{if } x_j < 0 \end{cases} \qquad A''_{ij} = \begin{cases} \overline{A}_{ij} & \text{if } x_j > 0 \\ 0 & \text{if } x_j = 0 \\ \underline{A}_{ij} & \text{if } x_j < 0 \end{cases}$$

Topological properties of interval linear systems

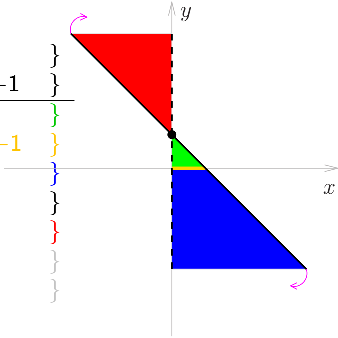
The weak solution set: $\{x \in \mathbb{R}^n \colon \exists A \in \mathbf{A}, \exists b \in \mathbf{b}.\ Ax = b\}$

Topological properties: can be non-convex, unconnected, non-closed

- a (possibly empty) not necessarily closed convex polyhedron in each closed orthant

An example: (for one constraint)

$$\begin{array}{llll}
P = & \{ & [1,+\infty]x + y = 1 & \} \\
= & \{ & [1,+\infty]x + y \le 1, \quad [-\infty,-1]x - y \le -1 & \} \\
\hline
(++) & \{ & 1*x + y \le 1, \quad -\infty*x - y \le -1 & \} \\
(+0) & \{ & 1*x + 0*y \le 1, \quad -\infty*x + 0*y \le -1 & \} \\
(+-) & \{ & 1*x + y \le 1, \quad -\infty*x - y \le -1 & \} \\
(0\ ?) & \{ & 0*x + y \le 1, \quad 0*x - y \le -1 & \} \\
(-+) & \{ & +\infty*x + y \le 1, \quad -1*x - y \le -1 & \} \\
(-0) & \{ & +\infty*x + y \le 1, \quad -1*x - y \le -1 & \} \\
(--) & \{ & +\infty*x + y \le 1, \quad -1*x - y \le -1 & \}
\end{array}$$



The domain of Interval Linear Equalities (ILE)

Domain representation: an ILE element $\mathbf{P}$

- representation: $\mathbf{A}x = \mathbf{b}$ in row echelon form

$$\begin{array}{cccccc} [] & [] & [] & \ldots & [] = [] \\ & \mathbf{0} & 0 & \ldots & 0 = 0 \\ & & [] & \ldots & [] = [] \\ & & & \ldots & \\ & & & & [] = [] \end{array}$$

Definition (Row echelon form)

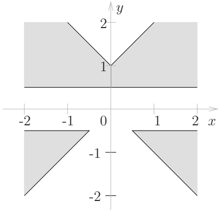
$\mathbf{A}x = \mathbf{b}$ where $\mathbf{A}$ is of size $m \times n$, is in *row echelon* form if

- $m = n$, and
- either x_i is the leading variable of the i-th row, or the i-th row is filled with zeros.

- semantics: $\gamma(\mathbf{P}) = \{x \in \mathbb{R}^n : \exists A \in \mathbf{A}, \exists b \in \mathbf{b}. \, Ax = b\}$

An example:

$$\begin{array}{rcl} [-1,1]x & + & y = [0,1] \\ & & [-1,1]y = 0.5 \end{array}$$



Domain operations: projection

Partial linearization ζ: linearize **interval coefficients** into **scalars**

- given $\varphi : (\Sigma_k[\underline{a}_k, \overline{a}_k]x_k = [\underline{b}, \overline{b}])$,

  $$\zeta(\varphi, x_j, c) \stackrel{\text{def}}{=} \left(c \times x_j + \Sigma_{k \neq j}[\underline{a}_k, \overline{a}_k]x_k = ([\underline{b}, \overline{b}] \boxminus [\underline{a}_j - c, \overline{a}_j - c] \boxtimes [\underline{x}_j, \overline{x}_j])\right)$$

  where c can be any real number.
- E.g., $\varphi : ([0,2]x + y = 2)$ w.r.t. $x, y \in [-2, 4] \stackrel{c=1}{\Longrightarrow} \zeta(\varphi, x, c) : (x + y = [-2, 6])$

Eliminate x_j from a pair of constraints φ, φ': like Gaussian elimination

- 1) $\varphi \dashrightarrow (1 * x_j + \Sigma_{k \neq j}[\underline{a}''_k, \overline{a}''_k]x_k = [\underline{b}'', \overline{b}''])$
  - e.g., $\zeta(\varphi, x_j, c)$ with $c = 1$
- 2) substitute x_j with $([\underline{b}'', \overline{b}''] - \Sigma_{k \neq j}[\underline{a}''_k, \overline{a}''_k]x_k)$ in φ'

  $$\psi : (0 * x_j + \Sigma_{k \neq j}([\underline{a}'_k, \overline{a}'_k] \boxminus [\underline{a}'_j, \overline{a}'_j] \boxtimes [\underline{a}''_k, \overline{a}''_k])x_k = [\underline{b}', \overline{b}'] \boxminus [\underline{a}'_j, \overline{a}'_j] \boxtimes [\underline{b}'', \overline{b}''])$$

Projection (cont.)

Goal: project out x_j from an ILE element $\mathbf{P}$, $\textsc{Project}(\mathbf{P}, x_j)$

$\mathbf{P}' \leftarrow \mathbf{P}$
for $i = 1$ to $j-1$ **do**
  if $([\underline{A}_{ij}, \overline{A}_{ij}] \neq [0,0])$ **then**
    $\varphi \leftarrow \zeta(\mathbf{P}'_i, x_j, c)$ with $c = 0$ {projection by bounds}
    for $k = i+1$ to j **do**
      if $([\underline{A}_{kj}, \overline{A}_{kj}] \neq [0,0])$ **then**
        let φ' be the result by combining $\mathbf{P}'_i$ and $\mathbf{P}'_k$ to eliminate x_j
        if $(\varphi' \preceq \varphi)$ **then** $\varphi \leftarrow \varphi'$
    $\mathbf{P}'_i \leftarrow \varphi$ {φ is the best constraint with leading var x_i that involves no x_j}
$\mathbf{P}'_j \leftarrow [0,0]^{1\times(n+1)}$
return $\mathbf{P}'$

Constraint comparison

Definition (Heuristic metrics)

1) $f_{weight}(\varphi) \stackrel{\text{def}}{=} \sum_k (\overline{a}_k - \underline{a}_k) \times (\overline{x}_k - \underline{x}_k) + (\overline{b} - \underline{b})$,
2) $f_{width}(\varphi) \stackrel{\text{def}}{=} \sum_k (\overline{a}_k - \underline{a}_k) + (\overline{b} - \underline{b})$,
3) $f_{mark}(\varphi) \stackrel{\text{def}}{=} \sum_k \delta(\underline{a}_k, \overline{a}_k) + \delta(\underline{b}, \overline{b})$, where

$$\delta(\underline{d}, \overline{d}) \stackrel{\text{def}}{=} \begin{cases} -1 & \text{if } \underline{d} = \overline{d}, \\ +200 & \text{else if } \underline{d} = -\infty \text{ and } \overline{d} = +\infty, \\ +100 & \text{else if } \underline{d} = -\infty \text{ or } \overline{d} = +\infty, \\ 0 & \text{otherwise.} \end{cases}$$

Definition (Constraint comparison)

We write $\varphi \preceq \varphi'$ if

$$(f_{weight}(\varphi), f_{width}(\varphi), f_{mark}(\varphi)) \leq (f_{weight}(\varphi'), f_{width}(\varphi'), f_{mark}(\varphi'))$$

holds in the sense of lexicographic order.

Note: an affine equality is always $\preceq$ than other kinds of constraints.

Example: $(x + y = 1) \preceq (x + y = [1, 2]) \preceq (x + y = [1, +\infty])$

Join

Joins for known domains

- affine hull for the affine equality domain: affine combination

  $$\sigma_1 z + \sigma_2 z' \text{ with } \sigma_1 + \sigma_2 = 1$$

- convex hull for convex polyhedra domain: convex combination

  $$\sigma_1 z + \sigma_2 z' \text{ with } \sigma_1 + \sigma_2 = 1 \text{ and } \sigma_1, \sigma_2 \geq 0$$

Approximate join based on convex combination for ILE

- given ILE elements $\gamma(\mathbf{P}) = \{z \mid \mathbf{A}z = \mathbf{b}\}, \gamma(\mathbf{P}') = \{z' \mid \mathbf{A}'z' = \mathbf{b}'\}$, we define

$$\left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \exists \sigma_1, \sigma_2 \in \mathbb{R}, z, z' \in \mathbb{R}^n. \\ x = \sigma_1 z + \sigma_2 z' \wedge \sigma_1 + \sigma_2 = 1 \wedge \sigma_1 \geq 0 \wedge \\ \quad \mathbf{A}z = \mathbf{b} \quad \wedge \quad \mathbf{A}'z' = \mathbf{b}' \quad \wedge \sigma_2 \geq 0 \end{array} \right\}$$ [Benoy King 97]

Join (cont.)

Approximate join based on convex combination for ILE

$$\left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \exists \sigma_1, \sigma_2 \in \mathbb{R}, z, z' \in \mathbb{R}^n. \\ x = \sigma_1 z + \sigma_2 z' \wedge \sigma_1 + \sigma_2 = 1 \wedge \sigma_1 \geq 0 \wedge \\ \quad \mathbf{A} z = \mathbf{b} \quad \wedge \;\; \mathbf{A}' z' = \mathbf{b}' \;\; \wedge \sigma_2 \geq 0 \end{array} \right\}$$

$$\xrightarrow[y' = \sigma_2 z']{y = \sigma_1 z} \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \exists \sigma_1, \sigma_2 \in \mathbb{R}, y, y' \in \mathbb{R}^n. \\ x = y + y' \;\; \wedge \sigma_1 + \sigma_2 = 1 \;\; \wedge \sigma_1 \geq 0 \wedge \\ \mathbf{A} y = \sigma_1 \mathbf{b} \;\; \wedge \; \mathbf{A}' y' = \sigma_2 \mathbf{b}' \; \wedge \sigma_2 \geq 0 \end{array} \right\}$$

$$\Longleftrightarrow \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \exists \sigma_1 \in \mathbb{R}, y \in \mathbb{R}^n. \\ \mathbf{A}' x - \mathbf{A}' y + \mathbf{b}' \sigma_1 = \mathbf{b}' \quad \wedge \\ \qquad \mathbf{A}\, y \;-\mathbf{b}\, \sigma_1 = 0 \quad \wedge \\ \qquad\qquad \sigma_1 = [0, 1] \end{array} \right\} \tag{1}$$

Algorithm: projecting out $y(y_1, \ldots, y_n), \sigma_1$ from the row echelon system (1) via PROJECT() yields an ILE element $\mathbf{P} \uplus_w \mathbf{P}'$.

Soundness: $\gamma(\mathbf{P}) \cup \gamma(\mathbf{P}') \subseteq \gamma(\mathbf{P} \uplus_w \mathbf{P}')$.

Note: $\mathbf{P} \uplus_w \mathbf{P}'$ will not miss any affine equality given by affine hull

Join (cont.)

Definition (Interval Combination $\uplus$)

Given $\varphi' : \ (\sum_k [\underline{a}'_k, \overline{a}'_k] \times x_k \ = \ [\underline{b}', \overline{b}'])$ and $\varphi'' : (\sum_k [\underline{a}''_k, \overline{a}''_k] \times x_k = [\underline{b}'', \overline{b}''])$, their *interval combination* is defined as

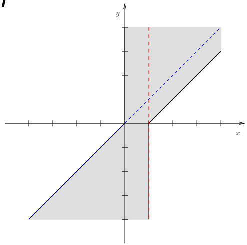
$$\varphi' \uplus \varphi'' \stackrel{\text{def}}{=} \left(\sum_k [\min(\underline{a}'_k, \underline{a}''_k), \max(\overline{a}'_k, \overline{a}''_k)] \times x_k = [\min(\underline{b}', \underline{b}''), \max(\overline{b}', \overline{b}'')]\right).$$

$\rightsquigarrow$ $\mathbf{P}' \uplus \mathbf{P}'' \stackrel{\text{def}}{=} \mathbf{P}$ where $\mathbf{P}_i = \mathbf{P}'_i \uplus \mathbf{P}''_i$ for all $i = 1, \ldots, n$

Soundness: $\gamma(\varphi') \cup \gamma(\varphi'') \subseteq \gamma(\varphi' \uplus \varphi'')$.

Example

Given $\mathbf{P}' = \{x = 1\}$ and $\mathbf{P}'' = \{x - y = 0\}$,
$\mathbf{P} = \mathbf{P}' \uplus \mathbf{P}'' = \{x + [-1, 0]y = [0, 1]\}$ (best!)



Join (cont.)

Definition (Weak Join)

We define a *weak join* operation for the ILE domain as

$$\mathbf{P} \sqcup_w \mathbf{P}' \stackrel{\text{def}}{=} (\mathbf{P} \Cup_w \mathbf{P}') \sqcap_w (\mathbf{P} \uplus \mathbf{P}').$$

Example:

$$\begin{aligned}
\mathbf{P} &= \{I = 2, J - K = 5, [-1,1]K = 1\} \\
\mathbf{P}' &= \{I = 3, J - K = 8, [-1,4]K = 2\}
\end{aligned}$$

$$\begin{aligned}
\mathbf{P} \Cup_w \mathbf{P}' &= \{3I - J + K = 1, J - K = [5,8]\} \\
\mathbf{P} \uplus \mathbf{P}' &= \{I = [2,3], J - K = [5,8], [-1,4]K = [1,2]\} \\
\mathbf{P} \sqcup_w \mathbf{P}' &= \{3I - J + K = 1, J - K = [5,8], [-1,4]K = [1,2]\}
\end{aligned}$$

$$\mathit{AffineHull}(\{I{=}2, J - K{=}5\}, \{I{=}3, J - K{=}8\}) = \{3I - J + K = 1\}$$

$$\mathit{ConvexHull}(\{I{=}2, J - K{=}5\}, \{I{=}3, J - K{=}8\}) = \{3I - J + K = 1, J - K = [5,8]\}$$

Widening

Definition (Widening on a pair of constraints)

Given $\varphi' : (\sum_k [\underline{a}'_k, \overline{a}'_k] x_k = [\underline{b}', \overline{b}'])$ and $\varphi'' : (\sum_k [\underline{a}''_k, \overline{a}''_k] x_k = [\underline{b}'', \overline{b}''])$, we define the *widening* on constraints φ' and φ'' as

$$\varphi' \bigtriangledown_{row} \varphi'' : \left(\sum_k ([\underline{a}'_k, \overline{a}'_k] \bigtriangledown_{itv} [\underline{a}''_k, \overline{a}''_k]) x_k = ([\underline{b}', \overline{b}'] \bigtriangledown_{itv} [\underline{b}'', \overline{b}''])\right)$$

where $\bigtriangledown_{itv}$ is any widening of the interval domain, such as:

$$[\underline{a}, \overline{a}] \bigtriangledown_{itv} [\underline{b}, \overline{b}] = [\underline{a} \leq \underline{b} \, ? \, \underline{a} : -\infty, \ \overline{a} \geq \overline{b} \, ? \, \overline{a} : +\infty]$$

Definition (Widening on ILE elements)

Given two ILE elements $\mathbf{P}' \sqsubseteq \mathbf{P}''$, we define the *widening* as $\mathbf{P}' \bigtriangledown_{ile} \mathbf{P}'' \stackrel{\text{def}}{=} \mathbf{P}$ where

$$\mathbf{P}_i = \begin{cases} \mathbf{P}''_i & \text{if } \mathbf{P}''_i \text{ is an affine equality} \\ \mathbf{P}'_i \bigtriangledown_{row} \mathbf{P}''_i & \text{otherwise} \end{cases}$$

Widening (cont.)

Widening with thresholds ∇^{T}

- T: a finite set of threshold values, including $-\infty$ and $+\infty$
- for the interval domain

$$[\underline{a}, \overline{a}] \nabla^{T}_{itv} [\underline{b}, \overline{b}] = \quad [\underline{a} \le \underline{b} \,?\, \underline{a} : \max\{\ell \in T \mid \ell \le \underline{b}\}, \\ \overline{a} \ge \overline{b} \,?\, \overline{a} : \min\{h \in T \mid h \ge \overline{b}\}]$$

Lifting: $\mathbf{P}' \nabla^{T}_{ile} \mathbf{P}''$ based on ∇^{T}_{itv}

- individual variables $\longrightarrow$ **multiple** variables
- guess not only bounds of the constant term but also the **shape (slope)**

Example:

real x, y;
$x := 0.75 * y + 1$;
while *true* do
① if *random*()
then $x := y + 1$;
else $x := 0.25 * x + 0.5 * y + 1$;
done;

$$\varphi : [1,1]x + [-0.75, -0.75]y = [1,1]$$
$$\varphi' : [1,1]x + [-1, -0.6875]y = [1, 1.25]$$
$$\varphi \nabla_{row} \varphi' : [1,1]x + [-\infty, +\infty]y = [1, +\infty]$$
$$\varphi \nabla^{T}_{row} \varphi' : [1,1]x + [-1, -0.5]y = [1, 1.5]$$
$$(T = \{\pm n \pm 0.5 \mid n \le 2, n \in \mathbb{N}\} \cup \{\pm\infty\})$$

Early experimental results

Prototype

Prototype implementation (*FP*-ILE) using:

- interval arithmetic based on `double`-precision floating-point numbers
  - floats are time and memory efficient
  - still sound: interval arithmetic with outward rounding

Interface:

- plugged into the APRON library
- programs analyzed with INTERPROC

Comparison with:

- *polkaeq*: rational implementation to infer affine equalities
- NewPolka: rational implementation for convex polyhedra domain
- *itvPol*: sound floating-point implementation for interval polyhedra domain

Early Experimental Results

Program	*FP*-ILE			*polkaeq*		Result
name(#vars)	#=	#≃	time(ms)	#=	time(ms)	Invar.
Karr1(3)	1	1	13	1	8	$>$
GS1(4)	2	3	19	2	13	$>$
MOS1(6)	1	1	66	1	33	$>$
Karr1_f(3)	0	2	19	0	9	$>$
Deadcode(2)	1	1	4	0	11	$>$

For these examples, *FP*-ILE misses no affine equality that *polkaeq* finds

Program	*FP*-ILE			NewPolka		*itvPol*			Result	
name(#vars)	#≤	#≃	time	#≤	time	#≤	#≃	time	Invar.	
policy2(2)	3	1	20ms	2	22ms	3	0	46ms	$>$	$>$
policy3(2)	2	2	18ms	2	20ms	2	2	49ms	$>$	$<$
symmetricalstairs(2)	3	0	33ms	3	31ms	2	0	45ms	$<$	$>$
incdec(32)	26	12	32s	×	>1h	×	×	>1h	$>$	$>$
bigjava(44)	18	16	43s	×	>1h	6	4	1206s	$>$	$\neq$

FP-ILE can find interesting interval linear invariants in practice, including commonly used affine equalities, linear stripes, linear inequalities, etc.

Conclusion

Summary:

- a new abstract domain: **interval linear equalities** (ILE)
  - idea: extend the affine equality domain with interval coefficients
  - key: a **row echelon** system of interval linear equalities
  - attractive features:
    - express certain non-convex,unconnected,non-closed properties
    - polynomial-time domain operations
    - sound floating-point implementation
- a time and space efficient alternative to polyhedra-like domains

Future Work:

- improve ILE
  - variable ordering: for precision
  - better strategies for constraint comparison $\preceq$
- relax the row echelon form
  - e.g., allow several constraints per leading variable