Numeric Abstract Domains

Abstract Interpretation Course

Antoine Miné

ENS & CNRS
Paris, France

26–27 November 2013

East China Normal University
Shanghai, P.R. China

Numeric abstract domains

Goal: abstract sets of points in $\mathcal{P}(\mathbb{R}^n)$

Applications:

- infer invariants on numeric variables
  - absence of arithmetic overflow, buffer overflow, division by 0
  - code optimization
- pointer analysis with pointer arithmetic
  (pointer offset $\simeq$ numeric variable)
- string analysis
  (buffer length, offset, position of 0, are numeric quantities)
- cost analysis
  (time, memory consumption are numeric quantities)
- termination analysis
  (infer ranking functions with value in $\mathbb{N}$)
- dependency analysis on array operations
  (automatic parallelization of loops)

Overview

Course gloal: present classic numeric abstract domains

- semantic aspects
  - choose the properties of interest: $\mathcal{D}^\sharp$
  - define sound / optimal transfer functions: $F^\sharp$
- algorithmic aspects
  - propose data-structure representation of $\mathcal{D}^\sharp$
  - implement algorithms for $F^\sharp$
    (further approximations for complexity reasons)
- mix two kinds of approximations
  - static approximation (choice of $\mathcal{D}^\sharp$, $F^\sharp$)
  - dynamic approximation (choice of $\triangledown$)

Numeric semantics

Syntax: expressions

Arithmetic expressions:

$\texttt{exp}$	$::=$	$\texttt{V}$	variable $\texttt{V} \in \mathbb{V}$
	$\mid$	$-\texttt{exp}$	negation
	$\mid$	$\texttt{exp} \diamond \texttt{exp}$	binary operation: $\diamond \in \{+, -, \times, /\}$
	$\mid$	$[c, c']$	constant range, $c, c' \in \mathbb{I} \cup \{\pm\infty\}$ (c is a shorthand for $[c, c]$)

- fixed, finite set of variables $\mathbb{V}$
- one datatype: scalars in $\mathbb{I} \in \{\mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$
  (later extended to floats $\mathbb{F}$)
- non-deterministic
  ($[c, c']$ models inputs, non-deterministic choices, stubs)

Syntax: commands and programs

Commands:

$$\begin{array}{lcll} \texttt{com} & ::= & \texttt{V := exp} & \text{assignment into } \texttt{V} \in \mathbb{V} \\ & | & \texttt{exp} \bowtie 0 & \text{test, } \bowtie \in \{=, <, >, \leq, \geq, \neq\} \end{array}$$

programs: control-flow graphs

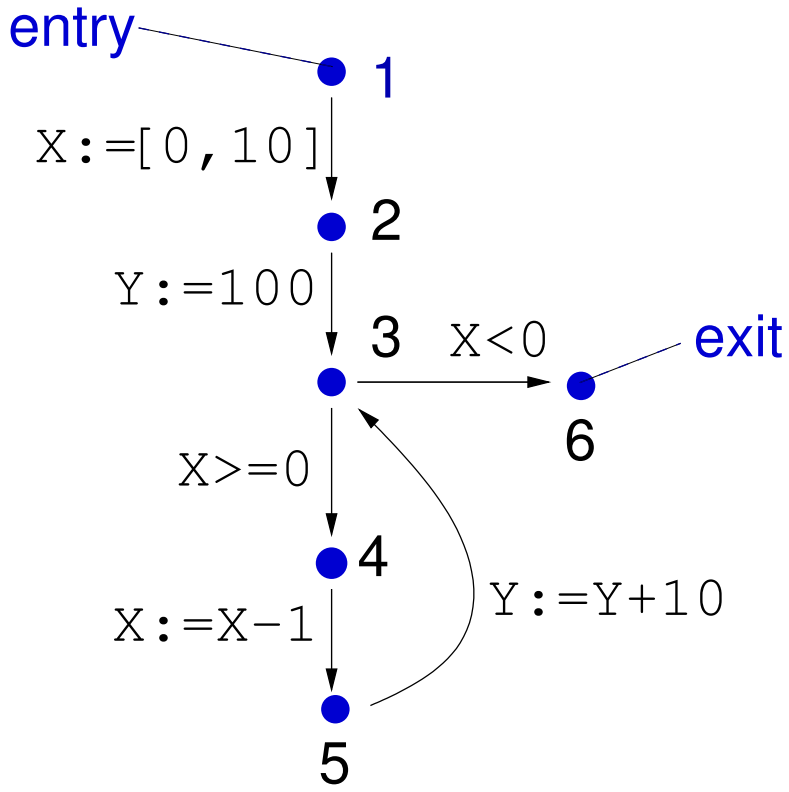
$$P \stackrel{\text{def}}{=} (L, e, A) \quad \left| \quad \begin{array}{ll} L & \text{program points (labels)} \\ e & \text{entry point: } e \in L \\ A & \text{arcs: } A \subseteq L \times \texttt{com} \times L \end{array} \right.$$

- no procedure

Example

```
1X:=[0,10];2
 Y:=100;
 while 3X>=0 do4
   X:=X-1;5
   Y:=Y+10
 done6
```

structured program



control flow graph

(loop X from a value ≤ 10 down to 0, at each iteration add 10 to Y)

Concrete semantics

Concrete semantics: expressions

Semantics of expressions: $\mathtt{E}[\![\, e \,]\!] : (\mathbb{V} \to \mathbb{I}) \to \mathcal{P}(\mathbb{I})$

$$
\begin{array}{lcl}
\mathtt{E}[\![\, [c, c'] \,]\!]\, \rho & \stackrel{\text{def}}{=} & \{\, x \in \mathbb{I} \mid c \leq x \leq c' \,\} \\
\mathtt{E}[\![\, \mathtt{V} \,]\!]\, \rho & \stackrel{\text{def}}{=} & \{\, \rho(\mathtt{V}) \,\} \\
\mathtt{E}[\![\, -e \,]\!]\, \rho & \stackrel{\text{def}}{=} & \{\, -v \mid v \in \mathtt{E}[\![\, e \,]\!]\, \rho \,\} \\
\mathtt{E}[\![\, e_1 + e_2 \,]\!]\, \rho & \stackrel{\text{def}}{=} & \{\, v_1 + v_2 \mid v_1 \in \mathtt{E}[\![\, e_1 \,]\!]\, \rho, v_2 \in \mathtt{E}[\![\, e_2 \,]\!]\, \rho \,\} \\
\mathtt{E}[\![\, e_1 - e_2 \,]\!]\, \rho & \stackrel{\text{def}}{=} & \{\, v_1 - v_2 \mid v_1 \in \mathtt{E}[\![\, e_1 \,]\!]\, \rho, v_2 \in \mathtt{E}[\![\, e_2 \,]\!]\, \rho \,\} \\
\mathtt{E}[\![\, e_1 \times e_2 \,]\!]\, \rho & \stackrel{\text{def}}{=} & \{\, v_1 \times v_2 \mid v_1 \in \mathtt{E}[\![\, e_1 \,]\!]\, \rho, v_2 \in \mathtt{E}[\![\, e_2 \,]\!]\, \rho \,\} \\
\mathtt{E}[\![\, e_1 / e_2 \,]\!]\, \rho & \stackrel{\text{def}}{=} & \{\, v_1 / v_2 \mid v_1 \in \mathtt{E}[\![\, e_1 \,]\!]\, \rho, v_2 \in \mathtt{E}[\![\, e_2 \,]\!]\, \rho,\ v_2 \neq 0 \,\}
\end{array}
$$

- takes as argument an environment $\rho \in \mathbb{V} \to I$
- returns a set of values: non-deterministic, due to inputs $[c, c']$
- defined by structural induction, on the syntax of expressions
- divisions by zero stop the program: $\mathtt{E}[\![\, e/0 \,]\!]\, \rho = \emptyset$

Concrete semantics: commands

Semantics of commands: $\mathsf{C}[\![c]\!] : \mathcal{D} \to \mathcal{D}$ where $\mathcal{D} \stackrel{\text{def}}{=} \mathcal{P}(\mathbb{V} \to \mathbb{I})$

A transfer function for c defines a relation on environments:

$$\begin{array}{lcl} \mathsf{C}[\![\mathtt{V} := e]\!]\,\mathcal{X} & \stackrel{\text{def}}{=} & \{\, \rho[\, \mathtt{V} \mapsto v \,] \mid \rho \in \mathcal{X},\ v \in \mathsf{E}[\![e]\!]\,\rho \,\} \\ \mathsf{C}[\![e \bowtie 0]\!]\,\mathcal{X} & \stackrel{\text{def}}{=} & \{\, \rho \mid \rho \in \mathcal{X},\ \exists v \in \mathsf{E}[\![e]\!]\,\rho : v \bowtie 0 \,\} \end{array}$$

Relates the environments after the execution of a command to the environments before

- assignments evaluate expressions and update variable values
- tests filter environment, keep those that can satisfy the expression

Property: $\cup$−morphism: $\mathsf{C}[\![c]\!]\,\mathcal{X} = \bigcup_{\rho \in \mathcal{X}} \mathsf{C}[\![c]\!]\,\{\rho\}$

Concrete semantics: programs

Semantics of programs: $\mathrm{P}[\![(L, e, A)]\!] : L \to \mathcal{D}$

$\mathrm{P}[\![(L, e, A)]\!]\, \ell$ is the most precise invariant at $\ell \in L$.

It is the smallest solution of a recursive equation system $(\mathcal{X}_\ell)_{\ell \in L}$:

Semantic equation system

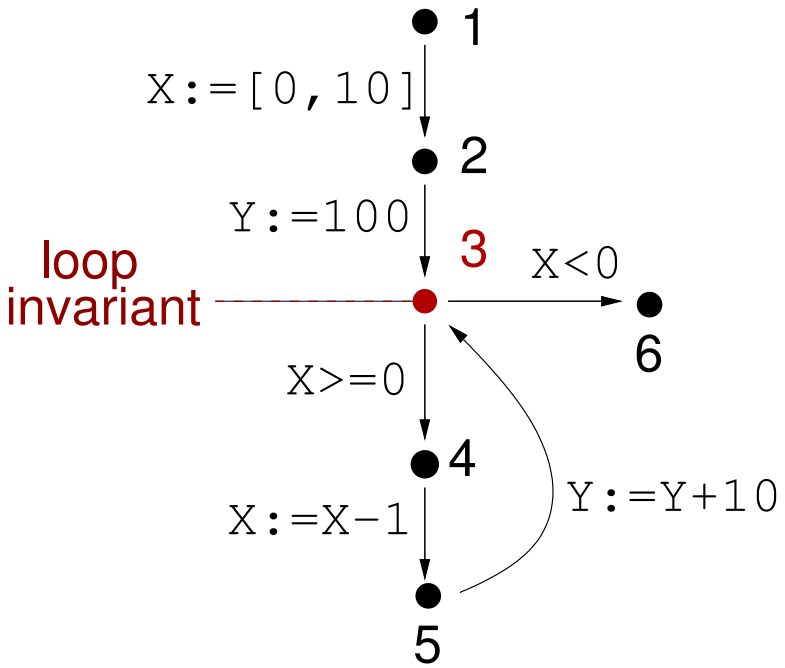
$$\begin{aligned} \mathcal{X}_e &\qquad\qquad\qquad\qquad\qquad\qquad \text{(given initial state)} \\ \mathcal{X}_{\ell \neq e} &= \bigcup_{(\ell', c, \ell) \in A} \mathrm{C}[\![c]\!]\, \mathcal{X}_{\ell'} \qquad \text{(transfer function)} \end{aligned}$$

Tarski's Theorem: the smallest solution exists and is unique [Tarsky 55].

- $(\mathcal{D}, \subseteq, \cup, \cap, \emptyset, (\mathbb{V} \to \mathbb{I}))$ is a complete lattice
- each $M_\ell : \mathcal{X}_\ell \mapsto \bigcup_{(\ell', c, \ell) \in A} \mathrm{C}[\![c]\!]\, \mathcal{X}_{\ell'}$ is monotonic in $\mathcal{D}$

$\Longrightarrow$ the solution is the least fixpoint of $(M_\ell)_{\ell \in L} : \forall \ell\colon M_\ell(\mathcal{X}_\ell) = \mathcal{X}_\ell$

Concrete semantics example



control flow graph

$$\begin{cases} \mathcal{X}_1 = (\{\mathtt{X},\mathtt{Y}\} \to \mathbb{Z}) \\ \mathcal{X}_2 = \mathrm{C}[\![\, \mathtt{X} := [0,10] \,]\!]\, \mathcal{X}_1 \\ \mathcal{X}_3 = \mathrm{C}[\![\, \mathtt{Y} := 100 \,]\!]\, \mathcal{X}_2 \cup \\ \qquad\quad \mathrm{C}[\![\, \mathtt{Y} := \mathtt{Y} + 10 \,]\!]\, \mathcal{X}_5 \\ \mathcal{X}_4 = \mathrm{C}[\![\, \mathtt{X} \geq 0 \,]\!]\, \mathcal{X}_3 \\ \mathcal{X}_5 = \mathrm{C}[\![\, \mathtt{X} := \mathtt{X} - 1 \,]\!]\, \mathcal{X}_4 \\ \mathcal{X}_6 = \mathrm{C}[\![\, \mathtt{X} < 0 \,]\!]\, \mathcal{X}_3 \end{cases}$$

equation system

Loop invariant: circular dependency

$$\mathcal{X}_3 = \{\, \rho \mid \rho(\mathtt{X}) \in [0,10],\ 10\rho(\mathtt{X}) + \rho(\mathtt{Y}) \in [100,200] \cap 10\mathbb{Z} \,\}$$

Concrete resolution

Resolution by increasing iterations:

$$\begin{cases} \mathcal{X}_e^0 & \stackrel{\text{def}}{=} & \mathcal{X}_e \\ \mathcal{X}_{\ell\neq e}^0 & \stackrel{\text{def}}{=} & \emptyset \end{cases} \qquad \begin{cases} \mathcal{X}_e^{n+1} & \stackrel{\text{def}}{=} & \mathcal{X}_e \\ \mathcal{X}_{\ell\neq e}^{n+1} & \stackrel{\text{def}}{=} & \bigcup\limits_{(\ell',c,\ell)\in A} \mathsf{C}[\![\, c \,]\!]\, \mathcal{X}_{\ell'}^n \end{cases}$$

Converges in ω iterations to a least solution,
because each $\mathsf{C}[\![\, c \,]\!]$ is continuous in the CPO $(\mathcal{D}, \subseteq)$

"Kleene" fixpoint theorem [Cousot Cousot 79]

$\Longrightarrow$ constructive definition

Concrete resolution example

iteration 0

$$\begin{cases} \mathcal{X}_1 = \mathbb{Z}^2 & \mathbb{Z}^2 \\ \mathcal{X}_2 = \mathrm{C}[\![\mathtt{X} := [0,10]]\!] \mathcal{X}_1 & \emptyset \\ \mathcal{X}_3 = \mathrm{C}[\![\mathtt{Y} := 100]\!] \mathcal{X}_2 \cup \mathrm{C}[\![\mathtt{Y} := \mathtt{Y} + 10]\!] \mathcal{X}_5 & \emptyset \\ \mathcal{X}_4 = \mathrm{C}[\![\mathtt{X} \geq 0]\!] \mathcal{X}_3 & \emptyset \\ \mathcal{X}_5 = \mathrm{C}[\![\mathtt{X} := \mathtt{X} - 1]\!] \mathcal{X}_4 & \emptyset \\ \mathcal{X}_6 = \mathrm{C}[\![\mathtt{X} < 0]\!] \mathcal{X}_3 & \emptyset \end{cases}$$

Concrete resolution example

iteration 1

$$\begin{cases} \mathcal{X}_1 = \mathbb{Z}^2 & \mathbb{Z}^2 \\ \mathcal{X}_2 = \mathsf{C}[\![\mathtt{X} := [0,10]]\!] \mathcal{X}_1 & [0,10] \times \mathbb{Z} \\ \mathcal{X}_3 = \mathsf{C}[\![\mathtt{Y} := 100]\!] \mathcal{X}_2 \cup \mathsf{C}[\![\mathtt{Y} := \mathtt{Y} + 10]\!] \mathcal{X}_5 & \emptyset \\ \mathcal{X}_4 = \mathsf{C}[\![\mathtt{X} \geq 0]\!] \mathcal{X}_3 & \emptyset \\ \mathcal{X}_5 = \mathsf{C}[\![\mathtt{X} := \mathtt{X} - 1]\!] \mathcal{X}_4 & \emptyset \\ \mathcal{X}_6 = \mathsf{C}[\![\mathtt{X} < 0]\!] \mathcal{X}_3 & \emptyset \end{cases}$$

Concrete resolution example

iteration 2

$$
\begin{cases}
\mathcal{X}_1 = \mathbb{Z}^2 & \mathbb{Z}^2 \\
\mathcal{X}_2 = \mathrm{C}[\![\, \mathtt{X} := [0,10] \,]\!]\, \mathcal{X}_1 & [0,10] \times \mathbb{Z} \\
\mathcal{X}_3 = \mathrm{C}[\![\, \mathtt{Y} := 100 \,]\!]\, \mathcal{X}_2 \cup \mathrm{C}[\![\, \mathtt{Y} := \mathtt{Y} + 10 \,]\!]\, \mathcal{X}_5 & \{\, (0,100), \ldots, (10,100) \,\} \\
\mathcal{X}_4 = \mathrm{C}[\![\, \mathtt{X} \geq 0 \,]\!]\, \mathcal{X}_3 & \emptyset \\
\mathcal{X}_5 = \mathrm{C}[\![\, \mathtt{X} := \mathtt{X} - 1 \,]\!]\, \mathcal{X}_4 & \emptyset \\
\mathcal{X}_6 = \mathrm{C}[\![\, \mathtt{X} < 0 \,]\!]\, \mathcal{X}_3 & \emptyset
\end{cases}
$$

Concrete resolution example

iteration 3

$$
\begin{cases}
\mathcal{X}_1 = \mathbb{Z}^2 & \mathbb{Z}^2 \\
\mathcal{X}_2 = \mathrm{C}[\![\, \mathtt{X} := [0,10] \,]\!]\, \mathcal{X}_1 & [0,10] \times \mathbb{Z} \\
\mathcal{X}_3 = \mathrm{C}[\![\, \mathtt{Y} := 100 \,]\!]\, \mathcal{X}_2 \cup \mathrm{C}[\![\, \mathtt{Y} := \mathtt{Y} + 10 \,]\!]\, \mathcal{X}_5 & \{\, (0,100), \ldots, (10,100) \,\} \\
\mathcal{X}_4 = \mathrm{C}[\![\, \mathtt{X} \geq 0 \,]\!]\, \mathcal{X}_3 & \{\, (0,100), \ldots, (10,100) \,\} \\
\mathcal{X}_5 = \mathrm{C}[\![\, \mathtt{X} := \mathtt{X} - 1 \,]\!]\, \mathcal{X}_4 & \emptyset \\
\mathcal{X}_6 = \mathrm{C}[\![\, \mathtt{X} < 0 \,]\!]\, \mathcal{X}_3 & \emptyset
\end{cases}
$$

Concrete resolution example

iteration 4

$$
\begin{cases}
\mathcal{X}_1 = \mathbb{Z}^2 & \mathbb{Z}^2 \\
\mathcal{X}_2 = \mathtt{C}[\![\, \mathtt{X} := [0,10] \,]\!]\, \mathcal{X}_1 & [0,10] \times \mathbb{Z} \\
\mathcal{X}_3 = \mathtt{C}[\![\, \mathtt{Y} := 100 \,]\!]\, \mathcal{X}_2 \cup \mathtt{C}[\![\, \mathtt{Y} := \mathtt{Y} + 10 \,]\!]\, \mathcal{X}_5 & \{\, (0,100), \ldots, (10,100) \,\} \\
\mathcal{X}_4 = \mathtt{C}[\![\, \mathtt{X} \geq 0 \,]\!]\, \mathcal{X}_3 & \{\, (0,100), \ldots, (10,100) \,\} \\
\mathcal{X}_5 = \mathtt{C}[\![\, \mathtt{X} := \mathtt{X} - 1 \,]\!]\, \mathcal{X}_4 & \{\, (-1,100), \ldots, (9,100) \,\} \\
\mathcal{X}_6 = \mathtt{C}[\![\, \mathtt{X} < 0 \,]\!]\, \mathcal{X}_3 & \emptyset
\end{cases}
$$

Concrete resolution example

iteration 5

$$
\begin{cases}
\mathcal{X}_1 = \mathbb{Z}^2 & \mathbb{Z}^2 \\
\mathcal{X}_2 = \mathrm{C}[\![\, \mathtt{X} := [0,10] \,]\!]\, \mathcal{X}_1 & [0,10] \times \mathbb{Z} \\
\mathcal{X}_3 = \mathrm{C}[\![\, \mathtt{Y} := 100 \,]\!]\, \mathcal{X}_2 \cup \\ \qquad \mathrm{C}[\![\, \mathtt{Y} := \mathtt{Y} + 10 \,]\!]\, \mathcal{X}_5 & \{\, (0,100), \ldots, (10,100), \\ & \quad (-1,110), \ldots, (9,110) \,\} \\
\mathcal{X}_4 = \mathrm{C}[\![\, \mathtt{X} \geq 0 \,]\!]\, \mathcal{X}_3 & \{\, (0,100), \ldots, (10,100) \,\} \\
\mathcal{X}_5 = \mathrm{C}[\![\, \mathtt{X} := \mathtt{X} - 1 \,]\!]\, \mathcal{X}_4 & \{\, (-1,100), \ldots, (9,100) \,\} \\
\mathcal{X}_6 = \mathrm{C}[\![\, \mathtt{X} < 0 \,]\!]\, \mathcal{X}_3 & \emptyset
\end{cases}
$$

Concrete resolution example

iteration 6

$$
\begin{cases}
\mathcal{X}_1 = \mathbb{Z}^2 & \mathbb{Z}^2 \\
\mathcal{X}_2 = \mathrm{C}[\![\mathtt{X} := [0,10]]\!] \mathcal{X}_1 & [0,10] \times \mathbb{Z} \\
\mathcal{X}_3 = \mathrm{C}[\![\mathtt{Y} := 100]\!] \mathcal{X}_2 \cup \\
\qquad \mathrm{C}[\![\mathtt{Y} := \mathtt{Y} + 10]\!] \mathcal{X}_5 & \{ (0,100), \ldots, (10,100), \\
& \quad (-1,110), \ldots, (9,110) \} \\
\mathcal{X}_4 = \mathrm{C}[\![\mathtt{X} \geq 0]\!] \mathcal{X}_3 & \{ (0,100), \ldots, (10,100), \\
& \quad (0,110), \ldots, (9,110) \} \\
\mathcal{X}_5 = \mathrm{C}[\![\mathtt{X} := \mathtt{X} - 1]\!] \mathcal{X}_4 & \{ (-1,100), \ldots, (9,100) \} \\
\mathcal{X}_6 = \mathrm{C}[\![\mathtt{X} < 0]\!] \mathcal{X}_3 & \{ (-1,110) \}
\end{cases}
$$

Concrete resolution example

iteration 7

$$
\begin{cases}
\mathcal{X}_1 = \mathbb{Z}^2 & \mathbb{Z}^2 \\
\mathcal{X}_2 = \mathrm{C}[\![\, \mathtt{X} := [0,10] \,]\!]\, \mathcal{X}_1 & [0,10] \times \mathbb{Z} \\
\mathcal{X}_3 = \mathrm{C}[\![\, \mathtt{Y} := 100 \,]\!]\, \mathcal{X}_2 \cup \\ \qquad \mathrm{C}[\![\, \mathtt{Y} := \mathtt{Y} + 10 \,]\!]\, \mathcal{X}_5 & \{\, (0,100), \ldots, (10,100), \\ & \quad (-1,110), \ldots, (9,110) \,\} \\
\mathcal{X}_4 = \mathrm{C}[\![\, \mathtt{X} \geq 0 \,]\!]\, \mathcal{X}_3 & \{\, (0,100), \ldots, (10,100), \\ & \quad (0,110), \ldots, (9,110) \,\} \\
\mathcal{X}_5 = \mathrm{C}[\![\, \mathtt{X} := \mathtt{X} - 1 \,]\!]\, \mathcal{X}_4 & \{\, (-1,100), \ldots, (9,100), \\ & \quad (-1,110), \ldots, (8,110) \,\} \\
\mathcal{X}_6 = \mathrm{C}[\![\, \mathtt{X} < 0 \,]\!]\, \mathcal{X}_3 & \{\, (-1,110) \,\}
\end{cases}
$$

Concrete resolution example

iteration 8

$$
\begin{cases}
\mathcal{X}_1 = \mathbb{Z}^2 & \mathbb{Z}^2 \\
\mathcal{X}_2 = \mathrm{C}[\![\, \mathtt{X} := [0,10] \,]\!]\, \mathcal{X}_1 & [0,10] \times \mathbb{Z} \\
\mathcal{X}_3 = \mathrm{C}[\![\, \mathtt{Y} := 100 \,]\!]\, \mathcal{X}_2 \cup \\ \qquad \mathrm{C}[\![\, \mathtt{Y} := \mathtt{Y} + 10 \,]\!]\, \mathcal{X}_5 & \{\, (0,100), \ldots, (10,100), \\ & \quad (-1,110), \ldots, (9,110), \\ & \quad (-1,120), \ldots, (8,120) \,\} \\
\mathcal{X}_4 = \mathrm{C}[\![\, \mathtt{X} \geq 0 \,]\!]\, \mathcal{X}_3 & \{\, (0,100), \ldots, (10,100), \\ & \quad (0,110), \ldots, (9,110) \,\} \\
\mathcal{X}_5 = \mathrm{C}[\![\, \mathtt{X} := \mathtt{X} - 1 \,]\!]\, \mathcal{X}_4 & \{\, (-1,100), \ldots, (9,100), \\ & \quad (-1,110), \ldots, (8,110) \,\} \\
\mathcal{X}_6 = \mathrm{C}[\![\, \mathtt{X} < 0 \,]\!]\, \mathcal{X}_3 & \{\, (-1,110) \,\}
\end{cases}
$$

Concrete resolution example

iteration 9

$$
\begin{cases}
\mathcal{X}_1 = \mathbb{Z}^2 & \mathbb{Z}^2 \\
\mathcal{X}_2 = \mathrm{C}[\![\, \mathtt{X} := [0,10] \,]\!]\, \mathcal{X}_1 & [0,10] \times \mathbb{Z} \\
\mathcal{X}_3 = \mathrm{C}[\![\, \mathtt{Y} := 100 \,]\!]\, \mathcal{X}_2 \cup \\ \qquad \mathrm{C}[\![\, \mathtt{Y} := \mathtt{Y} + 10 \,]\!]\, \mathcal{X}_5 & \{\, (0,100), \ldots, (10,100), \\ \qquad (-1,110), \ldots, (9,110), \\ \qquad (-1,120), \ldots, (8,120) \,\} \\
\mathcal{X}_4 = \mathrm{C}[\![\, \mathtt{X} \geq 0 \,]\!]\, \mathcal{X}_3 & \{\, (0,100), \ldots, (10,100), \\ \qquad (0,110), \ldots, (9,110), \\ \qquad (0,120), \ldots, (8,120) \,\} \\
\mathcal{X}_5 = \mathrm{C}[\![\, \mathtt{X} := \mathtt{X} - 1 \,]\!]\, \mathcal{X}_4 & \{\, (-1,100), \ldots, (9,100), \\ \qquad (-1,110), \ldots, (8,110) \,\} \\
\mathcal{X}_6 = \mathrm{C}[\![\, \mathtt{X} < 0 \,]\!]\, \mathcal{X}_3 & \{\, (-1,110), (-1,120) \,\}
\end{cases}
$$

Concrete resolution example

iteration 10

$$
\begin{cases}
\mathcal{X}_1 = \mathbb{Z}^2 & \mathbb{Z}^2 \\
\mathcal{X}_2 = \mathrm{C}[\![\mathtt{X} := [0,10]]\!] \mathcal{X}_1 & [0,10] \times \mathbb{Z} \\
\mathcal{X}_3 = \mathrm{C}[\![\mathtt{Y} := 100]\!] \mathcal{X}_2 \cup \mathrm{C}[\![\mathtt{Y} := \mathtt{Y} + 10]\!] \mathcal{X}_5 & \{ (0,100), \ldots, (10,100), (-1,110), \ldots, (9,110), (-1,120), \ldots, (8,120) \} \\
\mathcal{X}_4 = \mathrm{C}[\![\mathtt{X} \geq 0]\!] \mathcal{X}_3 & \{ (0,100), \ldots, (10,100), (0,110), \ldots, (9,110), (0,120), \ldots, (8,120) \} \\
\mathcal{X}_5 = \mathrm{C}[\![\mathtt{X} := \mathtt{X} - 1]\!] \mathcal{X}_4 & \{ (-1,100), \ldots, (9,100), (-1,110), \ldots, (8,110), (-1,120), \ldots, (7,120) \} \\
\mathcal{X}_6 = \mathrm{C}[\![\mathtt{X} < 0]\!] \mathcal{X}_3 & \{ (-1,110), (-1,120) \}
\end{cases}
$$

Concrete resolution example

iteration ...

$$
\begin{cases}
\mathcal{X}_1 = \mathbb{Z}^2 & \mathbb{Z}^2 \\
\mathcal{X}_2 = \mathrm{C}[\![\mathtt{X} := [0,10]]\!] \mathcal{X}_1 & [0,10] \times \mathbb{Z} \\
\mathcal{X}_3 = \mathrm{C}[\![\mathtt{Y} := 100]\!] \mathcal{X}_2 \cup \\
\qquad \mathrm{C}[\![\mathtt{Y} := \mathtt{Y} + 10]\!] \mathcal{X}_5 & \{ (0,100), \ldots, (10,100), \\
& \quad (-1,110), \ldots, (9,110), \\
& \quad (-1,120), \ldots, (8,120), \ldots \} \\
\mathcal{X}_4 = \mathrm{C}[\![\mathtt{X} \geq 0]\!] \mathcal{X}_3 & \{ (0,100), \ldots, (10,100), \\
& \quad (0,110), \ldots, (9,110), \\
& \quad (0,120), \ldots, (8,120), \ldots \} \\
\mathcal{X}_5 = \mathrm{C}[\![\mathtt{X} := \mathtt{X} - 1]\!] \mathcal{X}_4 & \{ (-1,100), \ldots, (9,100), \\
& \quad (-1,110), \ldots, (8,110), \\
& \quad (-1,120), \ldots, (7,120), \ldots \} \\
\mathcal{X}_6 = \mathrm{C}[\![\mathtt{X} < 0]\!] \mathcal{X}_3 & \{ (-1,110), (-1,120), \ldots \}
\end{cases}
$$

Abstractions

Numeric abstract domains

Representation: given by

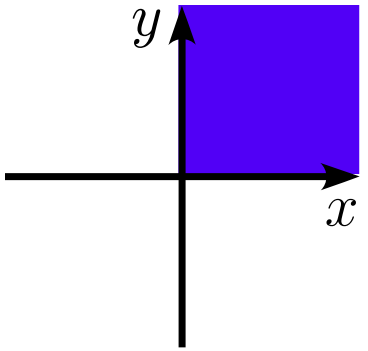
- a set $\mathcal{D}^\sharp$ of machine-representable abstract values
- a partial order $(\mathcal{D}^\sharp, \subseteq^\sharp, \bot^\sharp, \top^\sharp)$
  relating the amount of information given by abstract values
- a concretization function γ: $\mathcal{D}^\sharp \to \mathcal{D}$ $\quad (\mathcal{D} \stackrel{\text{def}}{=} \mathcal{P}(\mathbb{V} \to \mathbb{I}))$
  giving a meaning to each abstract element

Required algebraic properties:

- γ should be monotonic for $\subseteq^\sharp$: $\mathcal{X}^\sharp \subseteq^\sharp \mathcal{Y}^\sharp \Longrightarrow \gamma(\mathcal{X}^\sharp) \subseteq \gamma(\mathcal{Y}^\sharp)$
- $\gamma(\bot^\sharp) = \emptyset$
- $\gamma(\top^\sharp) = \mathbb{V} \to \mathbb{I}$

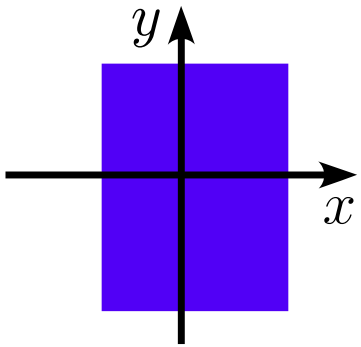
Note: γ need not be one-to-one

Numeric abstract domain examples



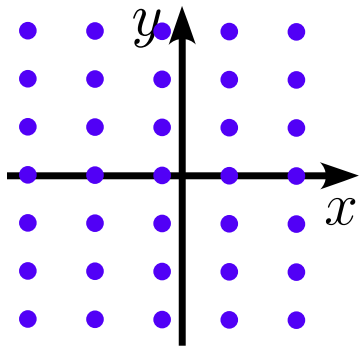
Signs

$x \geq 0$



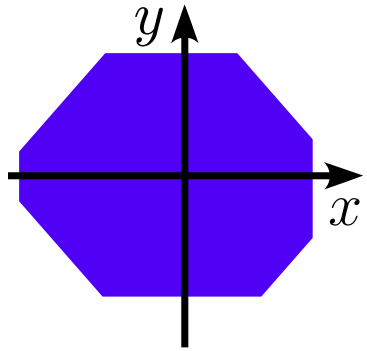
Intervals

$x \in [a, b]$



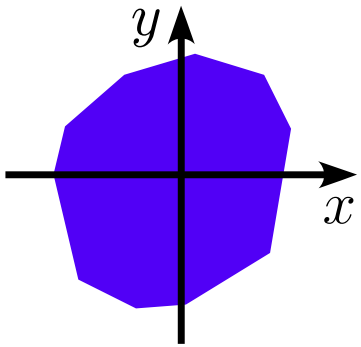
Congruences

$x \in a\mathbb{Z} + b$



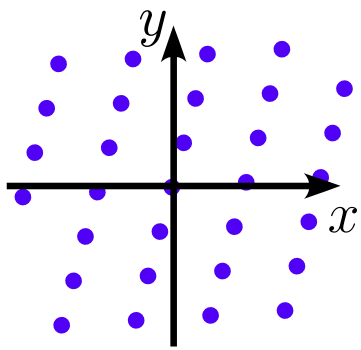
Octagons

$\pm x \pm y \leq c$



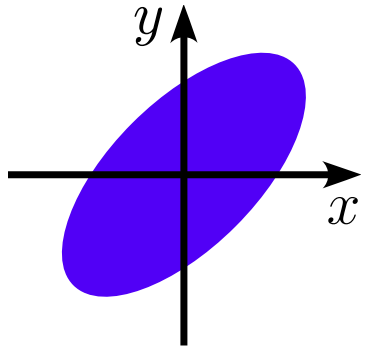
Polyhedra

$\sum_i \alpha_i x_i \leq \beta$



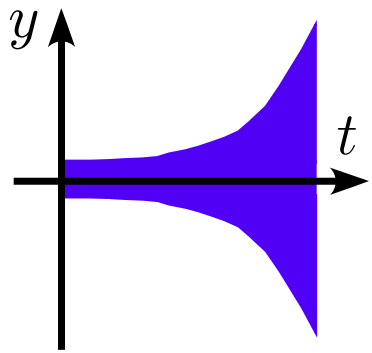
Lattices

$\sum \alpha_i x_i \in a\mathbb{Z} + b$



Ellipsoids

$x^2 + y^2 + \alpha xy \leq \beta$



Exponentials

$|y| \leq \alpha + \beta t^\gamma$

Numeric abstract domains (cont.)

Abstract operators: we require:

- sound, effective, abstract transfer functions $C^\sharp[\![c]\!]$ for all commands c
- sound, effective, abstract set operators $\cup^\sharp$, $\cap^\sharp$
- an algorithm to decide the ordering $\subseteq^\sharp$

Soundness criterion:

$F^\sharp$ is a sound abstraction of a $n-$ary operator F if:

$$\forall \mathcal{X}_1^\sharp, \ldots, \mathcal{X}_n^\sharp \in \mathcal{D}^\sharp: \; F(\gamma(\mathcal{X}_1^\sharp), \ldots, \gamma(\mathcal{X}_n^\sharp)) \subseteq \gamma(F^\sharp(\mathcal{X}_1^\sharp, \ldots, \mathcal{X}_n^\sharp))$$

Optional: Galois connection $(\mathcal{D}, \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\mathcal{D}^\sharp, \subseteq^\sharp)$

- design optimal $F^\sharp$ as $F^\sharp \overset{\text{def}}{=} \alpha \circ F \circ \gamma$ (semantic definition)
- α may not exist (no best abstraction)
- $\alpha \circ F \circ \gamma$ may be too hard to compute (no algorithm)

Abstract semantics

Concrete equation system

$$\mathcal{X} : L \to \mathcal{D} \text{ least solution of } \begin{cases} \mathcal{X}_e \text{ given} \\ \mathcal{X}_{\ell \neq e} = \bigcup_{(\ell', c, \ell) \in A} \mathsf{C}[\![c]\!] \mathcal{X}_{\ell'} \end{cases}$$

Abstract semantics

Concrete equation system

$$\mathcal{X} : L \to \mathcal{D} \text{ least solution of } \begin{cases} \mathcal{X}_e \text{ given} \\ \mathcal{X}_{\ell \neq e} = \bigcup\limits_{(\ell', c, \ell) \in A} \mathsf{C}[\![c]\!] \mathcal{X}_{\ell'} \end{cases}$$

Abstract equation system

$$\mathcal{X}^\sharp : L \to \mathcal{D}^\sharp \text{ any solution of } \begin{cases} \mathcal{X}^\sharp_e \text{ such that } \mathcal{X}_e \subseteq \gamma(\mathcal{X}^\sharp_e) \\ \mathcal{X}^\sharp_{\ell \neq e} \supseteq^\sharp \bigcup\limits^\sharp_{(\ell', c, \ell) \in A} \mathsf{C}^\sharp[\![c]\!] \mathcal{X}^\sharp_{\ell'} \end{cases}$$

Soundness theorem: $\forall \ell \in L : \gamma(\mathcal{X}^\sharp_\ell) \supseteq \mathcal{X}_\ell$

Iteration strategy

Resolution by iterations in $\mathcal{D}^\sharp$:

To effectively solve the abstract system, we need:

- an iteration ordering on abstract equations
- a widening operator $\triangledown$ to speed-up the convergence
  (if there are infinite strictly increasing chains in $\mathcal{D}^\sharp$)

  $\triangledown : (\mathcal{D}^\sharp \times \mathcal{D}^\sharp) \to \mathcal{D}^\sharp$ is a widening if:
  - it is sound: $\gamma(\mathcal{X}^\sharp) \cup \gamma(\mathcal{Y}^\sharp) \subseteq \gamma(\mathcal{X}^\sharp \triangledown \mathcal{Y}^\sharp)$
  - it enforces termination:
    $\forall$ sequence $(\mathcal{Y}_i^\sharp)_{i\in\mathbb{N}}$:
    the sequence $\mathcal{X}_0^\sharp = \mathcal{Y}_0^\sharp,\ \mathcal{X}_{i+1}^\sharp = \mathcal{X}_i^\sharp \triangledown \mathcal{Y}_{i+1}^\sharp$
    stabilizes in finite time: $\exists n < \omega : \mathcal{X}_{n+1}^\sharp = \mathcal{X}_n^\sharp$
    (note: $\exists n : \forall m \geq n : \mathcal{X}_{m+1}^\sharp = \mathcal{X}_m^\sharp$ is not required)

inductive reasoning: generalization from discrete instances
($\neq$ mathematical induction)

Abstract analysis

$\mathcal{W} \subseteq L$ is a set of widening points if every CFG cycle has a point in $\mathcal{W}$

Abstract analysis:

$$\mathcal{X}^{\sharp 0}_e \stackrel{\text{def}}{=} \mathcal{X}^{\sharp}_e \quad \text{such that } \mathcal{X}_e \subseteq \gamma(\mathcal{X}^{\sharp}_e)$$

$$\mathcal{X}^{\sharp 0}_{\ell \neq e} \stackrel{\text{def}}{=} \bot^{\sharp}$$

$$\mathcal{X}^{\sharp n+1}_\ell \stackrel{\text{def}}{=} \begin{cases} \mathcal{X}^{\sharp}_e & \text{if } \ell = e \\ \bigcup^{\sharp}_{(\ell',c,\ell)\in A} \mathrm{C}^{\sharp}[\![c]\!] \mathcal{X}^{\sharp n}_{\ell'} & \text{if } \ell \notin \mathcal{W},\ \ell \neq e \\ \mathcal{X}^{\sharp n}_\ell \mathbin{\triangledown} \bigcup^{\sharp}_{(\ell',c,\ell)\in A} \mathrm{C}^{\sharp}[\![c]\!] \mathcal{X}^{\sharp n}_{\ell'} & \text{if } \ell \in \mathcal{W},\ \ell \neq e \end{cases}$$

- termination: for some δ, $\forall \ell: \mathcal{X}^{\sharp \delta+1}_\ell = \mathcal{X}^{\sharp \delta}_\ell$
- soundness: $\forall \ell \in L: \mathcal{X}_\ell \subseteq \gamma(\mathcal{X}^{\sharp \delta}_\ell)$
- can be refined by decreasing iterations with narrowing $\triangle$ (presented later)
- we apply every equation at each step, but other iteration orders are possible (worklist, chaotic iterations, [Bourdoncle 93])

Abstract analysis (proof)

Proof of soundness:

Suppose that $\forall \ell,\ \mathcal{X}_\ell^{\sharp\delta+1} = \mathcal{X}_\ell^{\sharp\delta}$.

If $\ell = e$, by definition: $\mathcal{X}_e^{\sharp\delta} = \mathcal{X}_e^\sharp$ and $\mathcal{X}_e \subseteq \gamma(\mathcal{X}_e^{\sharp\delta})$.

If $\ell \neq e$, $\ell \notin \mathcal{W}$, then $\mathcal{X}_\ell^{\sharp\delta} = \mathcal{X}_\ell^{\sharp\delta+1} = \cup^\sharp_{(\ell',c,\ell)\in A} \mathsf{C}^\sharp[\![\, c \,]\!]\, \mathcal{X}_{\ell'}^{\sharp\,\delta}$.

By soundness of $\cup^\sharp$ and $\mathsf{C}^\sharp[\![\, c \,]\!]$, $\gamma(\mathcal{X}_\ell^{\sharp\delta}) \supseteq \cup_{(\ell',c,\ell)\in A} \mathsf{C}[\![\, c \,]\!]\, \gamma(\mathcal{X}_{\ell'}^{\sharp\,\delta})$.

If $\ell \neq e$, $\ell \in \mathcal{W}$, then $\mathcal{X}_\ell^{\sharp\delta} = \mathcal{X}_\ell^{\sharp\delta+1} = \mathcal{X}_\ell^{\sharp\delta} \triangledown \cup^\sharp_{(\ell',c,\ell)\in A} \mathsf{C}^\sharp[\![\, c \,]\!]\, \mathcal{X}_{\ell'}^{\sharp\,\delta}$.

By soundness of $\triangledown$, $\gamma(\mathcal{X}_\ell^{\sharp\delta}) \supseteq \gamma(\cup^\sharp_{(\ell',c,\ell)\in A} \mathsf{C}^\sharp[\![\, c \,]\!]\, \mathcal{X}_{\ell'}^{\sharp\,\delta})$,

and so we also have $\gamma(\mathcal{X}_\ell^{\sharp\delta}) \supseteq \cup_{(\ell',c,\ell)\in A} \mathsf{C}[\![\, c \,]\!]\, \gamma(\mathcal{X}_{\ell'}^{\sharp\,\delta})$.

We have proved that $\lambda\ell.\gamma(\mathcal{X}_\ell^{\sharp\delta})$ is a postfixpoint of the concrete equation system.

Hence, it is greater than its least solution.

Abstract analysis (proof)

Proof of termination:

Suppose that the iteration does not terminate in finite time.

Given a label $\ell \in L$, we denote by $i_\ell^1, \ldots, i_\ell^k, \ldots$ the increasing sequence of unstable indices, i.e., such that $\forall k,\ \mathcal{X}_\ell^{\sharp\, i_\ell^k+1} \neq \mathcal{X}_\ell^{\sharp\, i_\ell^k}$.
As the iteration is not stable, $\forall n,\ \exists \ell,\ \mathcal{X}_\ell^{\sharp\, n} \neq \mathcal{X}_\ell^{\sharp\, n+1}$.
Hence, the sequence $(i_\ell^k)_k$ is infinite for at least one $\ell \in L$.

We argue that $\exists \ell \in \mathcal{W}$ such that $(i_\ell^k)_k$ is infinite as, otherwise,
$N = \max\{\, i_\ell^k \mid \ell \in \mathcal{W} \,\} + |L|$ is finite and satisfies:
$\forall n \geq N, \forall \ell \in L,\ \mathcal{X}_\ell^{\sharp\, n} = \mathcal{X}_\ell^{\sharp\, n+1}$, contradicting our assumption.

For such a $\ell \in \mathcal{W}$, consider the subsequence $\mathcal{Y}_k^\sharp = \mathcal{X}_\ell^{\sharp\, i_\ell^k}$ comprised of the unstable iterates of $\mathcal{X}_\ell^\sharp$.
Then $\mathcal{Y}^{\sharp\, k+1} = \mathcal{Y}^{\sharp\, k} \triangledown \mathcal{Z}^{\sharp\, k}$ for some sequence $\mathcal{Z}^{\sharp\, k}$.
The subsequence is infinite and $\forall k,\ \mathcal{Y}^{\sharp\, k+1} \neq \mathcal{Y}^{\sharp\, k}$, which contradicts the definition of $\triangledown$.

Hence, the iteration must terminate in finite time.

Non-relational domains

Cartesian abstraction

Cartesian abstraction (independent attribute analysis)

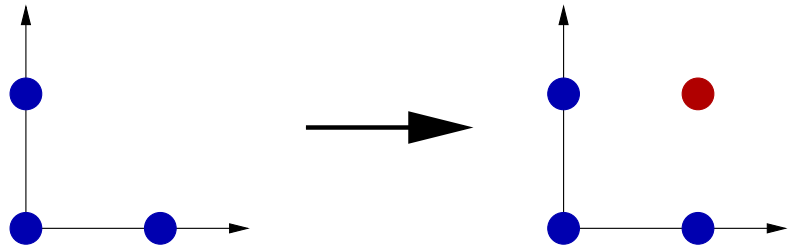
- $(\mathcal{P}(\mathbb{V} \to \mathbb{I}), \subseteq) \underset{\alpha_c}{\overset{\gamma_c}{\leftrightarrows}} (\mathbb{V} \to \mathcal{P}(\mathbb{I}), \dot{\subseteq})$
- $\alpha_c(\mathcal{X}) \overset{\text{def}}{=} \lambda \mathtt{V}.\{\, \rho(\mathtt{V}) \,|\, \rho \in \mathcal{X} \,\}$
- $\gamma_c(\mathcal{X}^\sharp) \overset{\text{def}}{=} \{\, \rho \,|\, \forall \mathtt{V} \in \mathbb{V}: \rho(\mathtt{V}) \in \mathcal{X}^\sharp(\mathtt{V}) \,\}$

Cartesian abstraction

Cartesian abstraction (independent attribute analysis)

- $(\mathcal{P}(\mathbb{V} \to \mathbb{I}), \subseteq) \underset{\alpha_c}{\overset{\gamma_c}{\leftrightarrows}} (\mathbb{V} \to \mathcal{P}(\mathbb{I}), \dot{\subseteq})$
- $\alpha_c(\mathcal{X}) \overset{\text{def}}{=} \lambda \mathtt{V}.\{\, \rho(\mathtt{V}) \mid \rho \in \mathcal{X} \,\}$
- $\gamma_c(\mathcal{X}^\sharp) \overset{\text{def}}{=} \{\, \rho \mid \forall \mathtt{V} \in \mathbb{V}: \rho(\mathtt{V}) \in \mathcal{X}^\sharp(\mathtt{V}) \,\}$

Closure: $c \overset{\text{def}}{=} \gamma_c \circ \alpha_c = \lambda \mathcal{X}.\{\, \rho \mid \forall \mathtt{V} \in \mathbb{V}: \exists \rho' \in \mathcal{X}: \rho(\mathtt{V}) = \rho'(\mathtt{V}) \,\}$



$c(\{(X, Y) \mid X \in \{0, 2\}, Y \in \{0, 2\},\ X + Y \leq 2\}) = \{0, 2\} \times \{0, 2\}$.
"forgets" all relationships between variables

Note: we get a Galois embedding by coalescing $\emptyset$

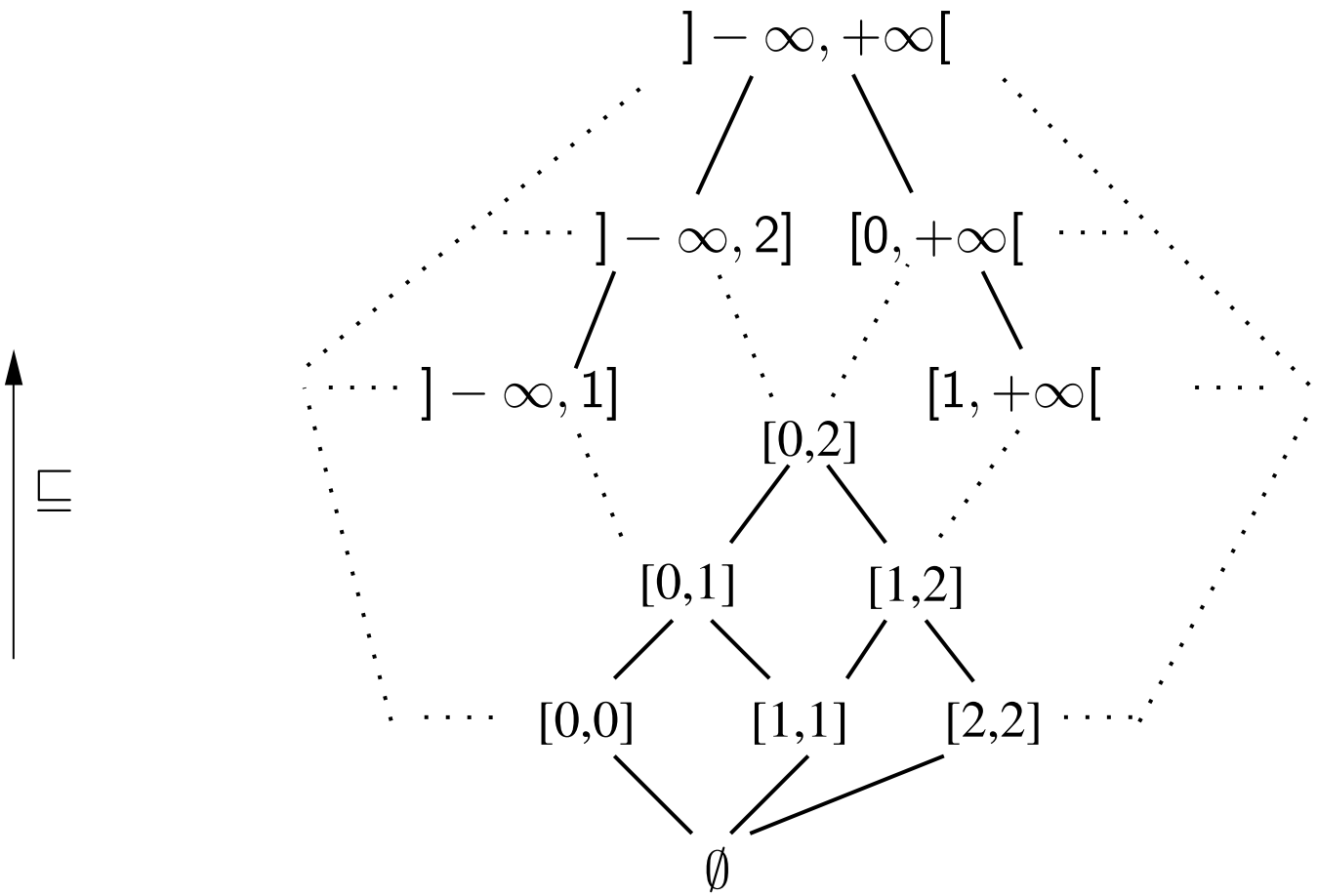
$\mathcal{D}^\sharp \overset{\text{def}}{=} (\mathbb{V} \to (\mathcal{P}(\mathbb{I}) \setminus \{\, \emptyset \,\})) \cup \{\, \emptyset \,\}$

The intervals domain

The interval lattice

Introduced in [Cousot 76]

$$\mathcal{B}^\sharp \stackrel{\text{def}}{=} \{\, [a,b] \mid a \in \mathbb{I} \cup \{-\infty\},\ b \in \mathbb{I} \cup \{+\infty\},\ a \leq b \,\} \cup \{\, \bot^\sharp \,\}$$



Note: intervals are open at infinite bounds $+\infty$, $-\infty$

The interval lattice (cont.)

Galois connection: $\mathcal{P}(\mathbb{I}) \underset{\alpha_b}{\overset{\gamma_b}{\leftrightarrows}} \mathcal{B}^\sharp$

$$\gamma_b([a,b]) \overset{\text{def}}{=} \{ x \in \mathbb{I} \mid a \leq x \leq b \}$$

$$\alpha_b(\mathcal{X}) \overset{\text{def}}{=} \begin{cases} \bot^\sharp & \text{if } \mathcal{X} = \emptyset \\ [\min \mathcal{X}, \max \mathcal{X}] & \text{otherwise} \end{cases}$$

If $\mathbb{I} = \mathbb{Q}$, α_b is not always defined

Partial order:

$$[a,b] \subseteq^\sharp [c,d] \overset{\text{def}}{\iff} a \geq c \text{ and } b \leq d$$

$$\top^\sharp \overset{\text{def}}{=}]-\infty, +\infty[$$

$$[a,b] \cup^\sharp [c,d] \overset{\text{def}}{=} [\min(a,c), \max(b,d)]$$

$$[a,b] \cap^\sharp [c,d] \overset{\text{def}}{=} \begin{cases} [\max(a,c), \min(b,d)] & \text{if } \max \leq \min \\ \bot^\sharp & \text{otherwise} \end{cases}$$

If $\mathbb{I} \neq \mathbb{Q}$, $(\mathcal{B}^\sharp, \subseteq^\sharp, \bot^\sharp, \top^\sharp, \cup^\sharp, \cap^\sharp)$ is a complete lattice

Derived abstract domain

Pointwise lifting to an abstraction of $\mathcal{P}(\mathbb{V} \to \mathbb{I})$:

$$\mathcal{D}^\sharp \stackrel{\text{def}}{=} (\mathbb{V} \to (\mathcal{B}^\sharp \setminus \{ \bot^\sharp \})) \cup \{ \bot^\sharp \}$$

$$\top^\sharp \stackrel{\text{def}}{=} \lambda \mathtt{V}.\top^\sharp$$

$$\mathcal{X}^\sharp \subseteq^\sharp \mathcal{Y}^\sharp \iff^{\text{def}} \mathcal{X}^\sharp = \bot^\sharp \vee (\mathcal{X}^\sharp, \mathcal{Y}^\sharp \neq \bot^\sharp \wedge \forall \mathtt{V}: \mathcal{X}^\sharp(\mathtt{V}) \subseteq^\sharp \mathcal{Y}^\sharp(\mathtt{V}))$$

$$\mathcal{X}^\sharp \cup^\sharp \mathcal{Y}^\sharp \stackrel{\text{def}}{=} \begin{cases} \mathcal{Y}^\sharp & \text{if } \mathcal{X}^\sharp = \bot^\sharp \\ \mathcal{X}^\sharp & \text{if } \mathcal{Y}^\sharp = \bot^\sharp \\ \lambda \mathtt{V}.\mathcal{X}^\sharp(\mathtt{V}) \cup^\sharp \mathcal{Y}^\sharp(\mathtt{V}) & \text{otherwise} \end{cases}$$

$$\mathcal{X}^\sharp \cap^\sharp \mathcal{Y}^\sharp \stackrel{\text{def}}{=} \begin{cases} \bot^\sharp & \text{if } \mathcal{X}^\sharp = \bot^\sharp \text{ or } \mathcal{Y}^\sharp = \bot^\sharp \\ \bot^\sharp & \text{if } \exists \mathtt{V}: \mathcal{X}^\sharp(\mathtt{V}) \cap^\sharp \mathcal{Y}^\sharp(\mathtt{V}) = \bot^\sharp \\ \lambda \mathtt{V}.\mathcal{X}^\sharp(\mathtt{V}) \cap^\sharp \mathcal{Y}^\sharp(\mathtt{V}) & \text{otherwise} \end{cases}$$

Interval abstract arithmetic operators

$$[c, c']^\sharp \stackrel{\text{def}}{=} [c, c']$$

$$-^\sharp [a, b] \stackrel{\text{def}}{=} [-b, -a]$$

$$[a, b] +^\sharp [c, d] \stackrel{\text{def}}{=} [a + c, b + d]$$

$$[a, b] -^\sharp [c, d] \stackrel{\text{def}}{=} [a - d, b - c]$$

$$[a, b] \times^\sharp [c, d] \stackrel{\text{def}}{=} [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$$

$$[a, b] /^\sharp [c, d] \stackrel{\text{def}}{=} \begin{cases} \bot^\sharp & \text{if } c = d = 0 \\ [\min(a/c, a/d, b/c, b/d), \\ \ \max(a/c, a/d, b/c, b/d)] & \text{if } 0 \leq c \\ [-b, -a]/^\sharp[-d, -c] & \text{if } d \leq 0 \\ ([a, b]/^\sharp[c, 0]) \cup^\sharp ([a, b]/^\sharp[0, d]) & \text{otherwise} \end{cases}$$

where $\pm\infty \times 0 = 0, \quad 0/0 = 0, \quad \forall x: x/\pm\infty = 0$
$\forall x > 0: x/0 = +\infty, \quad \forall x < 0: x/0 = -\infty$

Operators are strict: $-^\sharp \bot^\sharp = \bot^\sharp$, $[a, b] +^\sharp \bot^\sharp = \bot^\sharp$, etc.

Interval abstract assignment

Based on interval arithmetic [Moore 66]

Abstract evaluation of expressions: $\mathrm{E}^\sharp[\![e]\!] : \mathcal{D}^\sharp \to \mathcal{B}^\sharp$

$$\mathrm{E}^\sharp[\![e]\!]\, \bot^\sharp \stackrel{\text{def}}{=} \bot^\sharp$$

if $\mathcal{X}^\sharp \neq \bot^\sharp$:

$$\begin{aligned}
\mathrm{E}^\sharp[\![[c, c']]\!]\, \mathcal{X}^\sharp &\stackrel{\text{def}}{=} [c, c']^\sharp \\
\mathrm{E}^\sharp[\![\mathtt{V}]\!]\, \mathcal{X}^\sharp &\stackrel{\text{def}}{=} \mathcal{X}^\sharp(\mathtt{V}) \\
\mathrm{E}^\sharp[\![-e]\!]\, \mathcal{X}^\sharp &\stackrel{\text{def}}{=} -^\sharp\, \mathrm{E}^\sharp[\![e]\!]\, \mathcal{X}^\sharp \\
\mathrm{E}^\sharp[\![e_1 + e_2]\!]\, \mathcal{X}^\sharp &\stackrel{\text{def}}{=} \mathrm{E}^\sharp[\![e_1]\!]\, \mathcal{X}^\sharp +^\sharp \mathrm{E}^\sharp[\![e_2]\!]\, \mathcal{X}^\sharp \\
&\vdots
\end{aligned}$$

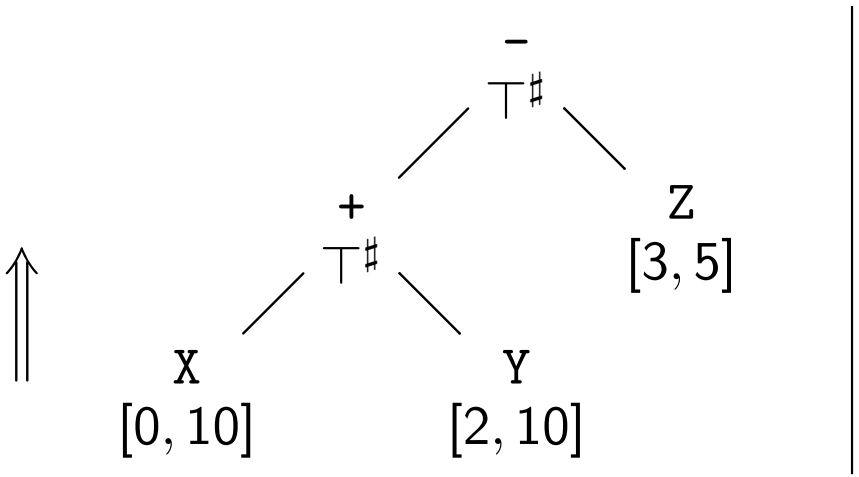
Abstract assignment:

$$\mathrm{C}^\sharp[\![\mathtt{V} := e]\!]\, \mathcal{X}^\sharp \stackrel{\text{def}}{=} \begin{cases} \bot^\sharp & \text{if } \mathcal{V}^\sharp = \bot^\sharp \\ \mathcal{X}^\sharp[\mathtt{V} \mapsto \mathcal{V}^\sharp] & \text{otherwise} \end{cases}$$

where $\mathcal{V}^\sharp = \mathrm{E}^\sharp[\![e]\!]\, \mathcal{X}^\sharp$.

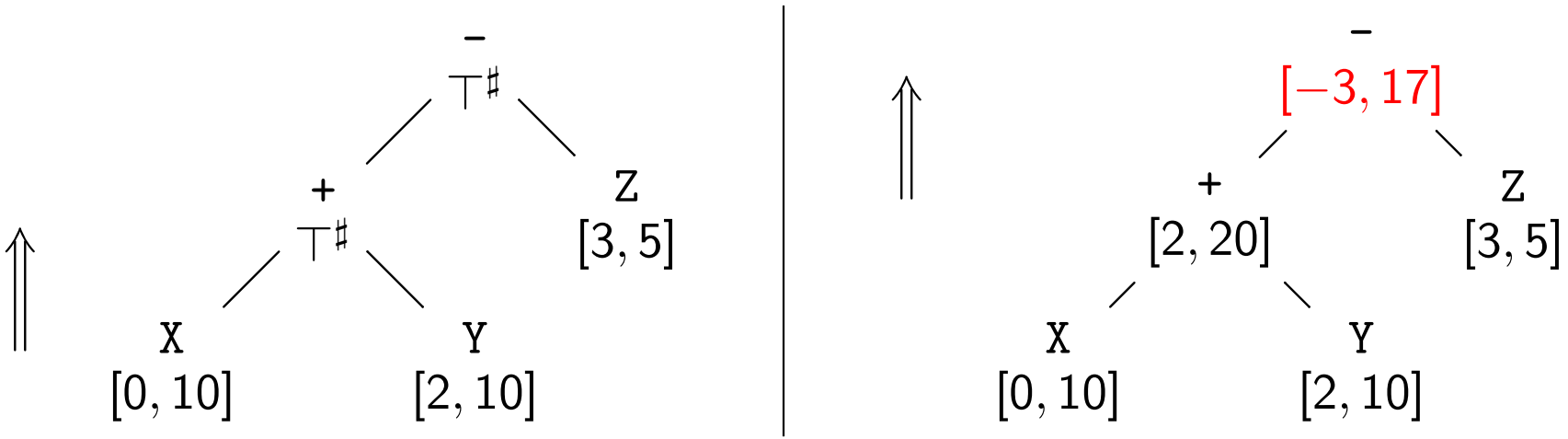
Interval assignment example

Example: $\mathcal{Y}^\sharp \stackrel{\text{def}}{=} \mathrm{C}^\sharp[\![\mathtt{X} := \mathtt{X} + \mathtt{Y} - \mathtt{Z}]\!] \mathcal{X}^\sharp$
with $\mathcal{X}^\sharp = \{ \mathtt{X} \mapsto [0, 10], \mathtt{Y} \mapsto [2, 10], \mathtt{Z} \mapsto [3, 5] \}$



Interval assignment example

Example: $\mathcal{Y}^\sharp \stackrel{\text{def}}{=} \mathrm{C}^\sharp[\![\mathtt{X} := \mathtt{X} + \mathtt{Y} - \mathtt{Z}]\!] \mathcal{X}^\sharp$
with $\mathcal{X}^\sharp = \{ \mathtt{X} \mapsto [0, 10], \mathtt{Y} \mapsto [2, 10], \mathtt{Z} \mapsto [3, 5] \}$



$\Longrightarrow \mathcal{Y}^\sharp = \{ \mathtt{X} \mapsto [-3, 17], \mathtt{Y} \mapsto [2, 10], \mathtt{Z} \mapsto [3, 5] \}$

Optimality

Arithmetic operator optimality

$[c, c']^\sharp$, $+^\sharp$, $-^\sharp$, $\times^\sharp$, $/^\sharp$ are optimal in $\mathcal{B}^\sharp$ as:

$X^\sharp \diamond^\sharp Y^\sharp = \alpha_b(\{\, x \diamond y \mid x \in \gamma_b(X^\sharp),\ y \in \gamma_b(Y^\sharp) \,\})$

Assignment optimality

Nevertheless, $\mathrm{C}^\sharp[\![\, \mathtt{V} := e \,]\!]$ is not optimal in $\mathcal{D}^\sharp$

- e.g.: $\mathrm{C}^\sharp[\![\, \mathtt{V} := \mathtt{X} - \mathtt{X} \,]\!]\, \mathcal{X}^\sharp$ where $\mathcal{X}^\sharp := \{\mathtt{X} \mapsto [a, b]\}$
  gives $\mathtt{V} \in [a, b] -^\sharp [a, b] = [a - b, b - a]$
  When $a < b$, we have $[a - b, b - a] \neq [0, 0]$
- $\mathrm{C}^\sharp[\![\, \mathtt{V} := e \,]\!]$ is optimal if each variable occurs at most once in e

Note: $\cup^\sharp$ is optimal; $\cap^\sharp$ and $\subseteq^\sharp$ are exact

Interval abstract tests (non-generic)

If $\mathcal{X}^\sharp(\mathtt{X}) = [a, b]$ and $\mathcal{X}^\sharp(\mathtt{Y}) = [c, d]$, we can define:

$$\mathrm{C}^\sharp[\![\, \mathtt{X} - c \leq 0 \,]\!]\, \mathcal{X}^\sharp \quad \stackrel{\text{def}}{=} \quad \begin{cases} \bot^\sharp & \text{if } a > c \\ \mathcal{X}^\sharp[\, \mathtt{X} \mapsto [a, \min(b, c)] \,] & \text{otherwise} \end{cases}$$

$$\mathrm{C}^\sharp[\![\, \mathtt{X} - \mathtt{Y} \leq 0 \,]\!]\, \mathcal{X}^\sharp \quad \stackrel{\text{def}}{=} \quad \begin{cases} \bot^\sharp & \text{if } a > d \\ \mathcal{X}^\sharp[\, \mathtt{X} \mapsto [a, \min(b, d)], \\ \qquad \mathtt{Y} \mapsto [\max(c, a), d] \,] & \text{otherwise} \end{cases}$$

$$\mathrm{C}^\sharp[\![\, e \bowtie 0 \,]\!]\, \mathcal{X}^\sharp \quad \stackrel{\text{def}}{=} \quad \mathcal{X}^\sharp \quad \text{otherwise}$$

Note: fall-back operators

- $\mathrm{C}^\sharp[\![\, e \bowtie 0 \,]\!]\, \mathcal{X}^\sharp = \mathcal{X}^\sharp$ is always sound
- $\mathrm{C}^\sharp[\![\, \mathtt{X} := e \,]\!]\, \mathcal{X}^\sharp = \mathcal{X}^\sharp[\mathtt{X} \mapsto \top^\sharp]$ is always sound

Backward arithmetic and comparison operators

Given: sound backward arithmetic and comparison operators that refine their argument given the result of the operation

i.e.

$$\mathcal{X}^{\sharp\prime} = \overleftarrow{\leq 0}^{\sharp}(\mathcal{X}^{\sharp}) \Longrightarrow \\ \{\, x \in \gamma_b(\mathcal{X}^{\sharp}) \,|\, x \leq 0 \,\} \subseteq \gamma_b(\mathcal{X}^{\sharp\prime}) \subseteq \gamma_b(\mathcal{X}^{\sharp})$$

$$\mathcal{X}^{\sharp\prime} = \overleftarrow{-}^{\sharp}(\mathcal{X}^{\sharp}, \mathcal{R}^{\sharp}) \Longrightarrow \\ \{\, x \,|\, x \in \gamma_b(\mathcal{X}^{\sharp}),\ -x \in \gamma_b(\mathcal{R}^{\sharp}) \,\} \subseteq \gamma_b(\mathcal{X}^{\sharp\prime}) \subseteq \gamma_b(\mathcal{X}^{\sharp})$$

$$(\mathcal{X}^{\sharp\prime}, \mathcal{Y}^{\sharp\prime}) = \overleftarrow{+}^{\sharp}(\mathcal{X}^{\sharp}, \mathcal{Y}^{\sharp}, \mathcal{R}^{\sharp}) \Longrightarrow \\ \{\, x \in \gamma_b(\mathcal{X}^{\sharp}) \,|\, \exists y \in \gamma_b(\mathcal{Y}^{\sharp}) : x + y \in \gamma_b(\mathcal{R}^{\sharp}) \,\} \subseteq \gamma_b(\mathcal{X}^{\sharp\prime}) \subseteq \gamma_b(\mathcal{X}^{\sharp}) \\ \{\, y \in \gamma_b(\mathcal{Y}^{\sharp}) \,|\, \exists x \in \gamma_b(\mathcal{X}^{\sharp}) : x + y \in \gamma_b(\mathcal{R}^{\sharp}) \,\} \subseteq \gamma_b(\mathcal{Y}^{\sharp\prime}) \subseteq \gamma_b(\mathcal{Y}^{\sharp})$$

$$\vdots$$

Note: best backward operators can be designed with α_b:
e.g. for $\overleftarrow{+}^{\sharp}$: $\mathcal{X}^{\sharp\prime} = \alpha_b(\{\, x \in \gamma_b(\mathcal{X}^{\sharp}) \,|\, \exists y \in \gamma_b(\mathcal{Y}^{\sharp}) : x + y \in \gamma_b(\mathcal{R}^{\sharp}) \,\})$

Generic backward operator construction

Synthesizing (non optimal) backward arithmetic operators from forward arithmetic operators

$$\overleftarrow{\leq 0}^\sharp(\mathcal{X}^\sharp) \stackrel{\text{def}}{=} \mathcal{X}^\sharp \cap^\sharp]-\infty, 0]^\sharp$$

$$\overleftarrow{-}^\sharp(\mathcal{X}^\sharp, \mathcal{R}^\sharp) \stackrel{\text{def}}{=} \mathcal{X}^\sharp \cap^\sharp (-^\sharp \mathcal{R}^\sharp)$$

$$\overleftarrow{+}^\sharp(\mathcal{X}^\sharp, \mathcal{Y}^\sharp, \mathcal{R}^\sharp) \stackrel{\text{def}}{=} (\mathcal{X}^\sharp \cap^\sharp (\mathcal{R}^\sharp -^\sharp \mathcal{Y}^\sharp),\ \mathcal{Y}^\sharp \cap^\sharp (\mathcal{R}^\sharp -^\sharp \mathcal{X}^\sharp))$$

$$\overleftarrow{-}^\sharp(\mathcal{X}^\sharp, \mathcal{Y}^\sharp, \mathcal{R}^\sharp) \stackrel{\text{def}}{=} (\mathcal{X}^\sharp \cap^\sharp (\mathcal{R}^\sharp +^\sharp \mathcal{Y}^\sharp),\ \mathcal{Y}^\sharp \cap^\sharp (\mathcal{X}^\sharp -^\sharp \mathcal{R}^\sharp))$$

$$\overleftarrow{\times}^\sharp(\mathcal{X}^\sharp, \mathcal{Y}^\sharp, \mathcal{R}^\sharp) \stackrel{\text{def}}{=} (\mathcal{X}^\sharp \cap^\sharp (\mathcal{R}^\sharp /^\sharp \mathcal{Y}^\sharp),\ \mathcal{Y}^\sharp \cap^\sharp (\mathcal{R}^\sharp /^\sharp \mathcal{X}^\sharp))$$

$$\overleftarrow{/}^\sharp(\mathcal{X}^\sharp, \mathcal{Y}^\sharp, \mathcal{R}^\sharp) \stackrel{\text{def}}{=} (\mathcal{X}^\sharp \cap^\sharp (\mathcal{S}^\sharp \times^\sharp \mathcal{Y}^\sharp),\ \mathcal{Y}^\sharp \cap^\sharp ((\mathcal{X}^\sharp /^\sharp \mathcal{S}^\sharp) \cup^\sharp [0,0]^\sharp))$$

$$\text{where } \mathcal{S}^\sharp = \begin{cases} \mathcal{R}^\sharp & \text{if } \mathbb{I} \neq \mathbb{Z} \\ \mathcal{R}^\sharp +^\sharp [-1,1]^\sharp & \text{if } \mathbb{I} = \mathbb{Z} \text{ (as / rounds)} \end{cases}$$

Note: $\overleftarrow{\diamond}^\sharp(\mathcal{X}^\sharp, \mathcal{Y}^\sharp, \mathcal{R}^\sharp) = (\mathcal{X}^\sharp, \mathcal{Y}^\sharp)$ is always sound (no refinement)

Interval backward operators

Applying the generic construction to the interval domain:

$$\overleftarrow{\leq 0}^\sharp([a,b]) \stackrel{\text{def}}{=} \begin{cases} [a, \min(b,0)] & \text{if } a \geq 0 \\ \bot^\sharp & \text{otherwise} \end{cases}$$

$$\overleftarrow{-}^\sharp([a,b],[r,s]) \stackrel{\text{def}}{=} [a,b] \cap^\sharp [-s,-r]$$

$$\overleftarrow{+}^\sharp([a,b],[c,d],[r,s]) \stackrel{\text{def}}{=} ([a,b] \cap^\sharp [r-d, s-c], [c,d] \cap^\sharp [r-b, s-a])$$

...

Interval abstract tests (general)

Abstract test algorithm: $\mathrm{C}^\sharp[\![e \bowtie 0]\!]\,\mathcal{X}^\sharp$

Associate to each expression node an abstract value in $\mathcal{B}^\sharp$ using two traversals of the expression tree:

- first, a bottom-up evaluation using forward operators $\diamond^\sharp$
- apply $\overleftarrow{\bowtie}\,0^\sharp$ to the root
- then, a top-down refinement using backward operators $\overleftarrow{\diamond}^\sharp$

For each expression leaf, we get an abstract value $\mathcal{V}^\sharp$:

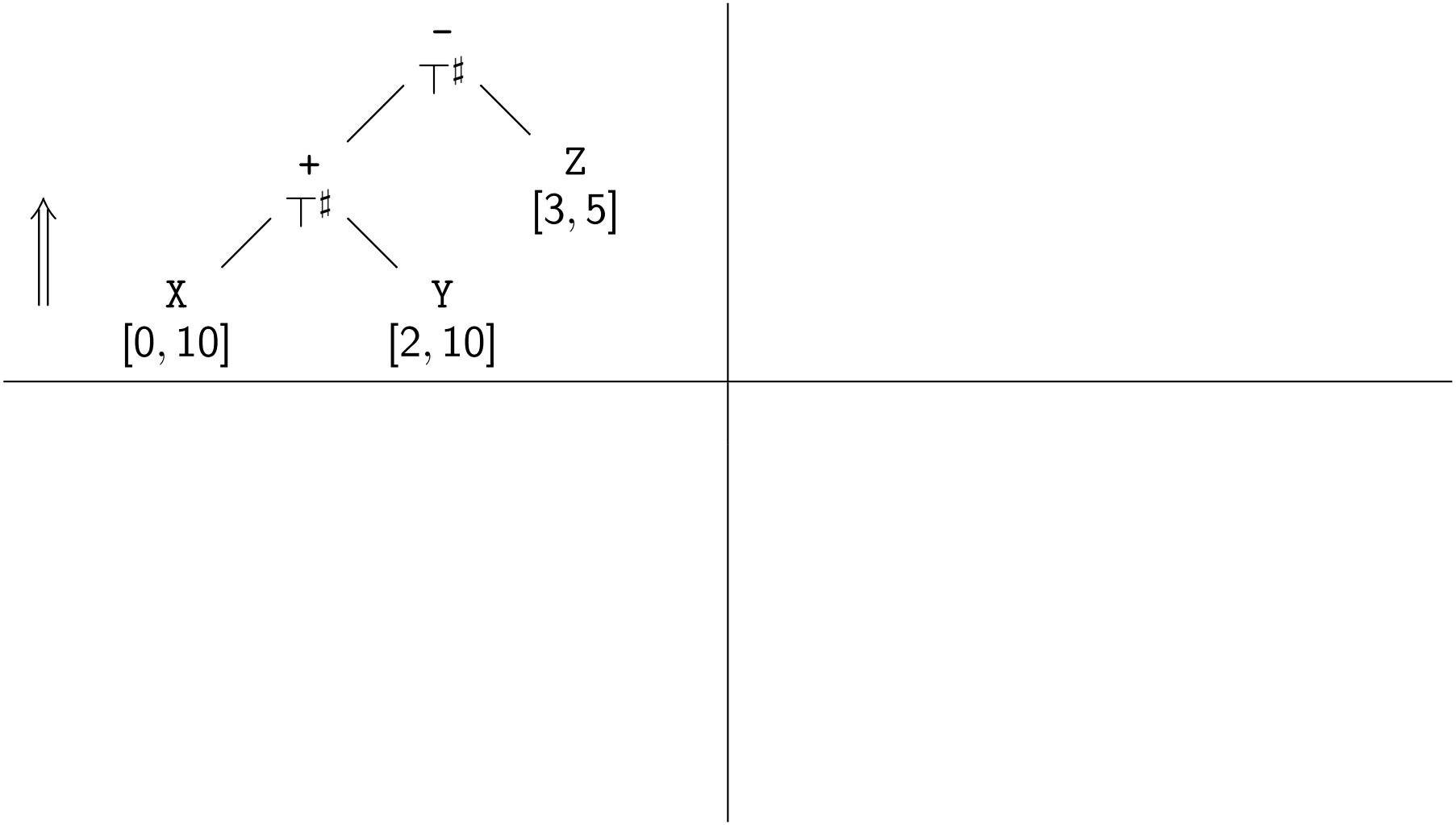
- for a variable V, replace $\mathcal{X}^\sharp(\mathtt{V})$ with $\mathcal{X}^\sharp(\mathtt{V}) \cap^\sharp \mathcal{V}^\sharp$
- for a constant $[c, c']$, check that $[c, c']^\sharp \cap^\sharp \mathcal{V}^\sharp \neq \bot^\sharp$
- $\Longrightarrow$ return $\bot^\sharp$ if some $\cap^\sharp \mathcal{V}^\sharp$ returns $\bot^\sharp$

Improvement: local iterations [Granger 92]

Similar to constraint programming (HC4)

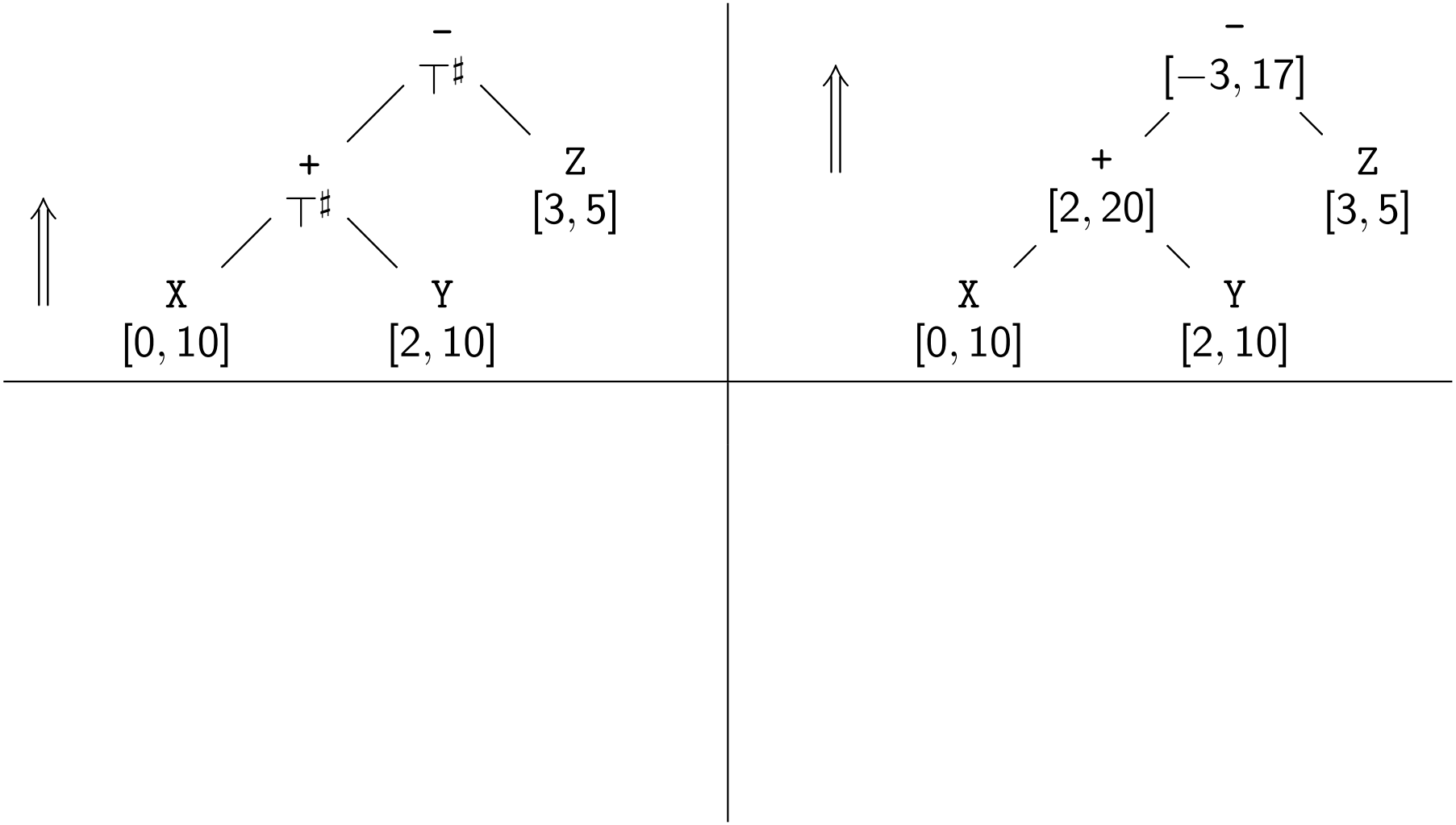
Interval test example

Example: $C^\sharp[\![\mathtt{X} + \mathtt{Y} - \mathtt{Z} \leq 0]\!] \mathcal{X}^\sharp$
with $\mathcal{X}^\sharp = \{ \mathtt{X} \mapsto [0, 10], \mathtt{Y} \mapsto [2, 10], \mathtt{Z} \mapsto [3, 5] \}$



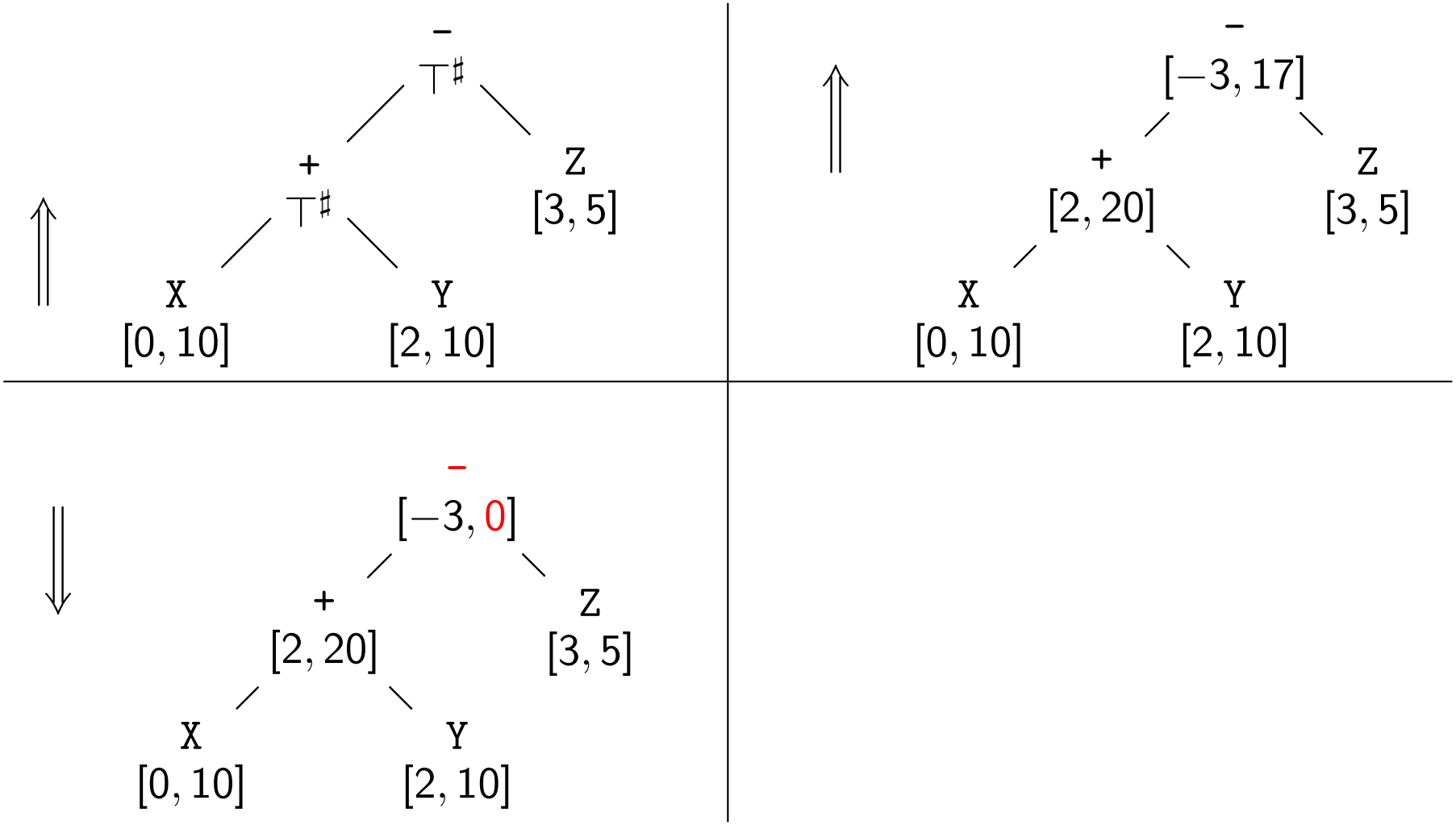
Interval test example

Example: $C^\sharp[\![\mathtt{X} + \mathtt{Y} - \mathtt{Z} \leq 0]\!] \mathcal{X}^\sharp$
with $\mathcal{X}^\sharp = \{ \mathtt{X} \mapsto [0, 10], \mathtt{Y} \mapsto [2, 10], \mathtt{Z} \mapsto [3, 5] \}$



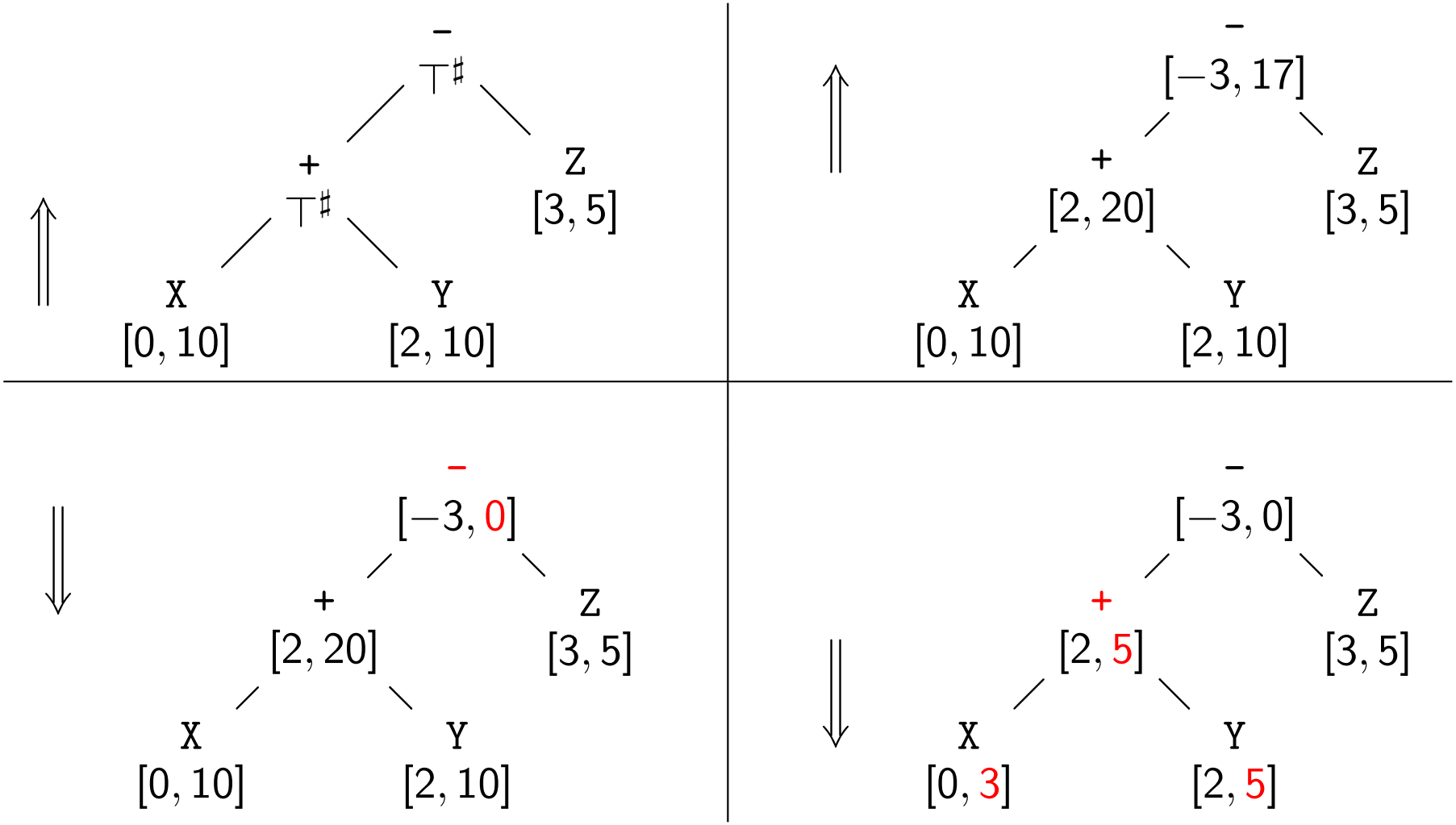
Interval test example

Example: $C^\sharp[\![X + Y - Z \leq 0]\!] \mathcal{X}^\sharp$
with $\mathcal{X}^\sharp = \{ X \mapsto [0, 10], Y \mapsto [2, 10], Z \mapsto [3, 5] \}$



Interval test example

Example: $C^\sharp[\![X + Y - Z \leq 0]\!] \mathcal{X}^\sharp$
with $\mathcal{X}^\sharp = \{ X \mapsto [0, 10], Y \mapsto [2, 10], Z \mapsto [3, 5] \}$



Interval widening

Widening on non-relational domains:

Given a value widening $\triangledown: \mathcal{B}^\sharp \times \mathcal{B}^\sharp \to \mathcal{B}^\sharp$,
we extend it point-wisely into a widening $\triangledown: \mathcal{D}^\sharp \times \mathcal{D}^\sharp \to \mathcal{D}^\sharp$:

$$\mathcal{X}^\sharp \triangledown \mathcal{Y}^\sharp \stackrel{\text{def}}{=} \lambda \mathtt{v}.(\mathcal{X}^\sharp(\mathtt{v}) \triangledown \mathcal{Y}^\sharp(\mathtt{v}))$$

Interval widening example:

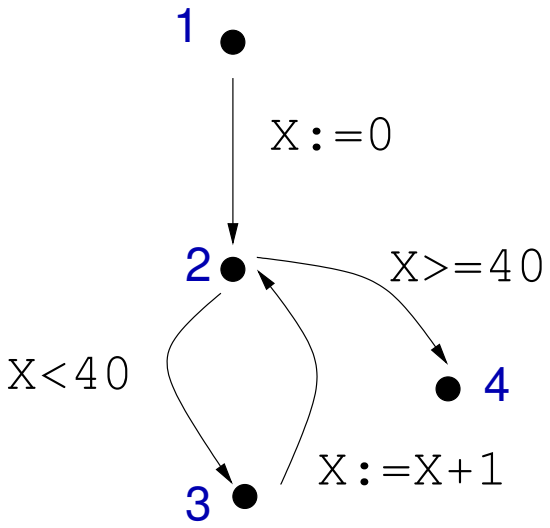
$$\bot^\sharp \quad \triangledown \quad X^\sharp \quad \stackrel{\text{def}}{=} \quad X^\sharp$$

$$[a,b] \quad \triangledown \quad [c,d] \quad \stackrel{\text{def}}{=} \quad \left[\begin{cases} a & \text{if } a \leq c \\ -\infty & \text{otherwise} \end{cases} , \begin{cases} b & \text{if } b \geq d \\ +\infty & \text{otherwise} \end{cases} \right]$$

Unstable bounds are set to $\pm\infty$

Analysis with widening example

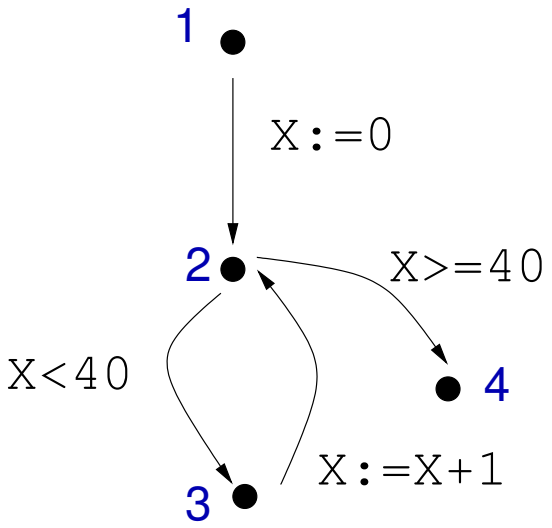
Analysis example with $\mathcal{W} = \{2\}$



ℓ	$\mathcal{X}_\ell^{\sharp 0}$					
1	$\top^\sharp$					
2 $\triangledown$	$\bot^\sharp$					
3	$\bot^\sharp$					
4	$\bot^\sharp$					

Analysis with widening example

Analysis example with $\mathcal{W} = \{2\}$



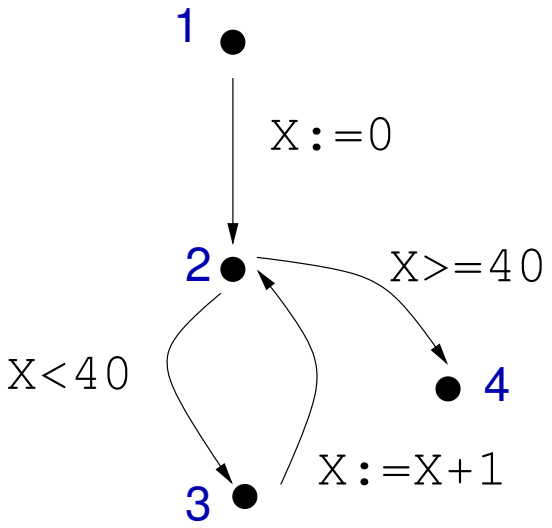
ℓ	$\mathcal{X}_\ell^{\sharp 0}$	$\mathcal{X}_\ell^{\sharp 1}$				
1	$\top^\sharp$	$\top^\sharp$				
2 $\triangledown$	$\bot^\sharp$	$= 0$				
3	$\bot^\sharp$	$\bot^\sharp$				
4	$\bot^\sharp$	$\bot^\sharp$				

More precisely, at the widening point:

$$\mathcal{X}_2^{\sharp 1} = \quad \bot^\sharp \triangledown ([0,0] \cup^\sharp \bot^\sharp) \quad = \quad \bot^\sharp \triangledown [0,0] \quad = [0,0]$$

Analysis with widening example

Analysis example with $\mathcal{W} = \{2\}$



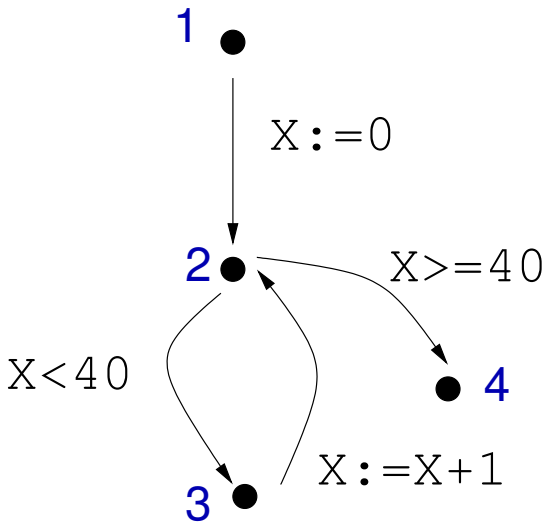
ℓ	$\mathcal{X}_\ell^{\sharp 0}$	$\mathcal{X}_\ell^{\sharp 1}$	$\mathcal{X}_\ell^{\sharp 2}$			
1	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$			
2 $\triangledown$	$\bot^\sharp$	$= 0$	$= 0$			
3	$\bot^\sharp$	$\bot^\sharp$	$= 0$			
4	$\bot^\sharp$	$\bot^\sharp$	$\bot^\sharp$			

More precisely, at the widening point:

$$\mathcal{X}_2^{\sharp 1} = \bot^\sharp \triangledown ([0,0] \cup^\sharp \bot^\sharp) = \bot^\sharp \triangledown [0,0] = [0,0]$$
$$\mathcal{X}_2^{\sharp 2} = [0,0] \triangledown ([0,0] \cup^\sharp \bot^\sharp) = [0,0] \triangledown [0,0] = [0,0]$$

Analysis with widening example

Analysis example with $\mathcal{W} = \{2\}$



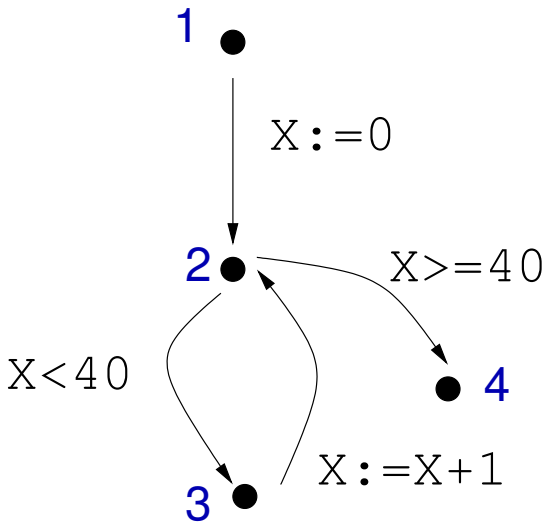
ℓ	$\mathcal{X}_\ell^{\sharp 0}$	$\mathcal{X}_\ell^{\sharp 1}$	$\mathcal{X}_\ell^{\sharp 2}$	$\mathcal{X}_\ell^{\sharp 3}$		
1	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$		
2 $\triangledown$	$\bot^\sharp$	$= 0$	$= 0$	≥ 0		
3	$\bot^\sharp$	$\bot^\sharp$	$= 0$	$= 0$		
4	$\bot^\sharp$	$\bot^\sharp$	$\bot^\sharp$	$\bot^\sharp$		

More precisely, at the widening point:

$$\begin{aligned}
\mathcal{X}_2^{\sharp 1} &= \bot^\sharp \triangledown ([0,0] \cup^\sharp \bot^\sharp) &&= \bot^\sharp \triangledown [0,0] &&= [0,0] \\
\mathcal{X}_2^{\sharp 2} &= [0,0] \triangledown ([0,0] \cup^\sharp \bot^\sharp) &&= [0,0] \triangledown [0,0] &&= [0,0] \\
\mathcal{X}_2^{\sharp 3} &= [0,0] \triangledown ([0,0] \cup^\sharp [1,1]) &&= [0,0] \triangledown [0,1] &&= [0,+\infty[
\end{aligned}$$

Analysis with widening example

Analysis example with $\mathcal{W} = \{2\}$



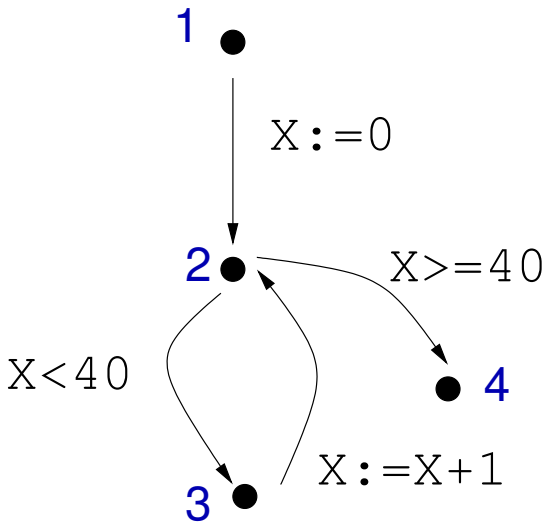
ℓ	$\mathcal{X}_\ell^{\sharp 0}$	$\mathcal{X}_\ell^{\sharp 1}$	$\mathcal{X}_\ell^{\sharp 2}$	$\mathcal{X}_\ell^{\sharp 3}$	$\mathcal{X}_\ell^{\sharp 4}$
1	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$
2 ∇	$\bot^\sharp$	$= 0$	$= 0$	≥ 0	≥ 0
3	$\bot^\sharp$	$\bot^\sharp$	$= 0$	$= 0$	$\in [0, 39]$
4	$\bot^\sharp$	$\bot^\sharp$	$\bot^\sharp$	$\bot^\sharp$	≥ 40

More precisely, at the widening point:

$$\begin{aligned}
\mathcal{X}_2^{\sharp 1} &= \bot^\sharp \nabla ([0,0] \cup^\sharp \bot^\sharp) &&= \bot^\sharp \nabla [0,0] &&= [0,0] \\
\mathcal{X}_2^{\sharp 2} &= [0,0] \nabla ([0,0] \cup^\sharp \bot^\sharp) &&= [0,0] \nabla [0,0] &&= [0,0] \\
\mathcal{X}_2^{\sharp 3} &= [0,0] \nabla ([0,0] \cup^\sharp [1,1]) &&= [0,0] \nabla [0,1] &&= [0,+\infty[\\
\mathcal{X}_2^{\sharp 4} &= [0,+\infty[\nabla ([0,0] \cup^\sharp [1,40]) &&= [0,+\infty[\nabla [0,40] &&= [0,+\infty[
\end{aligned}$$

Analysis with widening example

Analysis example with $\mathcal{W} = \{2\}$



ℓ	$\mathcal{X}_\ell^{\sharp 0}$	$\mathcal{X}_\ell^{\sharp 1}$	$\mathcal{X}_\ell^{\sharp 2}$	$\mathcal{X}_\ell^{\sharp 3}$	$\mathcal{X}_\ell^{\sharp 4}$	$\mathcal{X}_\ell^{\sharp 5}$
1	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$
2 $\triangledown$	$\bot^\sharp$	$= 0$	$= 0$	≥ 0	≥ 0	≥ 0
3	$\bot^\sharp$	$\bot^\sharp$	$= 0$	$= 0$	$\in [0, 39]$	$\in [0, 39]$
4	$\bot^\sharp$	$\bot^\sharp$	$\bot^\sharp$	$\bot^\sharp$	≥ 40	≥ 40

More precisely, at the widening point:

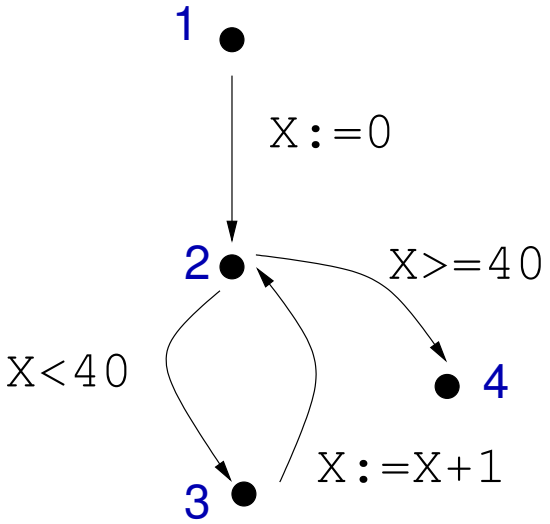
$$\begin{aligned}
\mathcal{X}_2^{\sharp 1} &= \bot^\sharp \triangledown ([0,0] \cup^\sharp \bot^\sharp) &&= \bot^\sharp \triangledown [0,0] &&= [0,0] \\
\mathcal{X}_2^{\sharp 2} &= [0,0] \triangledown ([0,0] \cup^\sharp \bot^\sharp) &&= [0,0] \triangledown [0,0] &&= [0,0] \\
\mathcal{X}_2^{\sharp 3} &= [0,0] \triangledown ([0,0] \cup^\sharp [1,1]) &&= [0,0] \triangledown [0,1] &&= [0,+\infty[\\
\mathcal{X}_2^{\sharp 4} &= [0,+\infty[\triangledown ([0,0] \cup^\sharp [1,40]) &&= [0,+\infty[\triangledown [0,40] &&= [0,+\infty[
\end{aligned}$$

Note that the most precise interval abstraction would be $\mathtt{X} \in [0, 40]$ at 2, and $\mathtt{X} = 40$ at 4

Influence of the widening point and iteration strategy

Changing $\mathcal{W}$ changes the analysis result

Example: The analysis is less precise for $\mathcal{W} = \{3\}$.



ℓ	$\mathcal{X}_\ell^{\sharp 1}$	$\mathcal{X}_\ell^{\sharp 2}$	$\mathcal{X}_\ell^{\sharp 3}$	$\mathcal{X}_\ell^{\sharp 4}$	$\mathcal{X}_\ell^{\sharp 5}$	$\mathcal{X}_\ell^{\sharp 6}$
1	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$
2	$= 0$	$= 0$	$\in [0,1]$	$\in [0,1]$	≥ 0	≥ 0
3 $\triangledown$	$\bot^\sharp$	$= 0$	$= 0$	≥ 0	≥ 0	≥ 0
4	$\bot^\sharp$	$\bot^\sharp$	$\bot^\sharp$	$\bot^\sharp$	$\bot^\sharp$	≥ 40

Intuition: extrapolation to $+\infty$ is no longer contained by the tests.

Chaotic iterations

Changing the iteration order changes the analysis result in the presence of a widening.

Narrowing

Using a widening makes the analysis less precise

Some precision can be retrieved by using a narrowing $\vartriangle$

Definition: narrowing $\vartriangle$

Binary operator $\mathcal{D}^\sharp \times \mathcal{D}^\sharp \to \mathcal{D}^\sharp$ such that:

- $(\mathcal{X}^\sharp \cap^\sharp \mathcal{Y}^\sharp) \subseteq^\sharp (\mathcal{X}^\sharp \vartriangle \mathcal{Y}^\sharp) \subseteq^\sharp \mathcal{X}^\sharp$
- for all sequences $(\mathcal{X}_i^\sharp)$, the decreasing sequence $(\mathcal{Y}_i^\sharp)$ defined by $\begin{cases} \mathcal{Y}_0^\sharp & \stackrel{\text{def}}{=} & \mathcal{X}_0^\sharp \\ \mathcal{Y}_{i+1}^\sharp & \stackrel{\text{def}}{=} & \mathcal{Y}_i^\sharp \vartriangle \mathcal{X}_{i+1}^\sharp \end{cases}$ is stationary

This is not the dual of a widening!

Narrowing examples

Interval narrowing:

$$[a,b] \vartriangle [c,d] \stackrel{\text{def}}{=} \left[\begin{cases} c & \text{if } a = -\infty \\ a & \text{otherwise} \end{cases} , \begin{cases} d & \text{if } b = +\infty \\ b & \text{otherwise} \end{cases} \right]$$

(refine only infinite bounds)

Point-wise extension to $\mathcal{D}^\sharp$: $\mathcal{X}^\sharp \vartriangle \mathcal{Y}^\sharp \stackrel{\text{def}}{=} \lambda \mathtt{V}.(\mathcal{X}^\sharp(\mathtt{V}) \vartriangle \mathcal{Y}^\sharp(\mathtt{V}))$

Finite-time intersection narrowing: (works for all domains)

$$\mathcal{X}^{\sharp i} \vartriangle \mathcal{Y}^\sharp \stackrel{\text{def}}{=} \begin{cases} \mathcal{X}^{\sharp i} \cap^\sharp \mathcal{Y}^\sharp & \text{if } i \leq N \\ \mathcal{X}^{\sharp i} & \text{if } i > N \end{cases}$$

Iterations with narrowing

Let $\mathcal{X}_\ell^{\sharp\delta}$ be the result after widening stabilisation; we have:

$$\mathcal{X}_\ell^{\sharp\delta} \supseteq^\sharp \begin{cases} \top^\sharp & \text{if } \ell = e \\ \bigcup\limits_{(\ell',c,\ell)\in A}^\sharp \mathrm{C}^\sharp[\![\, c \,]\!]\, \mathcal{X}_{\ell'}^{\sharp\delta} & \text{if } \ell \neq e \end{cases}$$

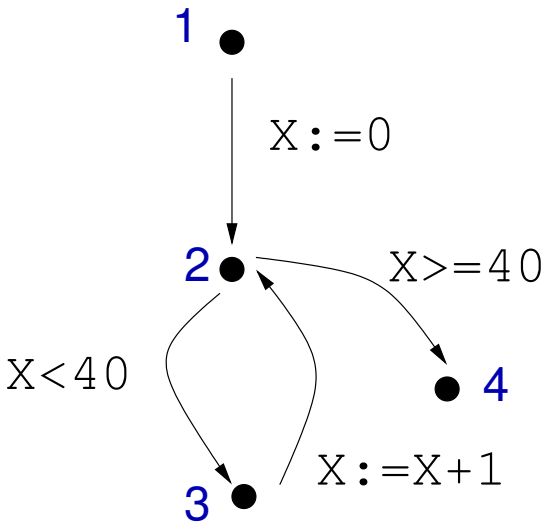
The following sequence is computed:

$$\mathcal{Y}_\ell^{\sharp 0} \stackrel{\text{def}}{=} \mathcal{X}_\ell^{\sharp\delta} \qquad \mathcal{Y}_\ell^{\sharp i+1} \stackrel{\text{def}}{=} \begin{cases} \top^\sharp & \text{if } \ell = e \\ \bigcup\limits_{(\ell',c,\ell)\in A}^\sharp \mathrm{C}^\sharp[\![\, c \,]\!]\, \mathcal{Y}_{\ell'}^{\sharp i} & \text{if } \ell \notin \mathcal{W} \\ \mathcal{Y}_\ell^{\sharp i} \mathbin{\triangle} \bigcup\limits_{(\ell',c,\ell)\in A}^\sharp \mathrm{C}^\sharp[\![\, c \,]\!]\, \mathcal{Y}_{\ell'}^{\sharp i} & \text{if } \ell \in \mathcal{W} \end{cases}$$

- the sequence $(\mathcal{Y}_\ell^{\sharp i})$ is decreasing and converges in finite time
- all $(\mathcal{Y}_\ell^{\sharp i})$ are solutions of the abstract semantical system

Analysis with narrowing example

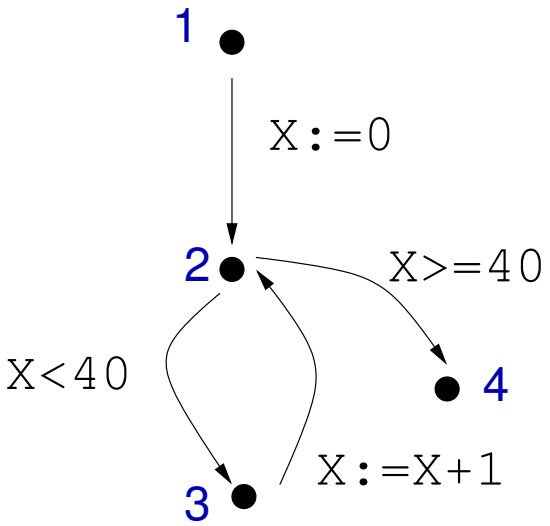
Example with $\mathcal{W} = \{2\}$



ℓ	$\mathcal{Y}_\ell^{\sharp 0}$			
1	$\top^\sharp$			
2 △	≥ 0			
3	$\in [0, 39]$			
4	≥ 40			

Analysis with narrowing example

Example with $\mathcal{W} = \{2\}$



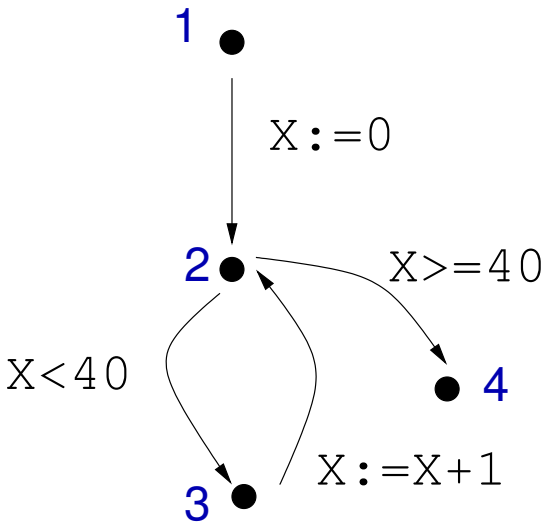
ℓ	$\mathcal{Y}_\ell^{\sharp 0}$	$\mathcal{Y}_\ell^{\sharp 1}$		
1	$\top^\sharp$	$\top^\sharp$		
2 $\triangle$	≥ 0	$\in [0, 40]$		
3	$\in [0, 39]$	$\in [0, 39]$		
4	≥ 40	≥ 40		

Narrowing at 2 gives:

$$\mathcal{Y}_2^{\sharp 1} = [0, +\infty[\triangle ([0, 0] \cup^\sharp [1, 40]) = [0, +\infty[\triangle [0, 40] = [0, 40]$$

Analysis with narrowing example

Example with $\mathcal{W} = \{2\}$



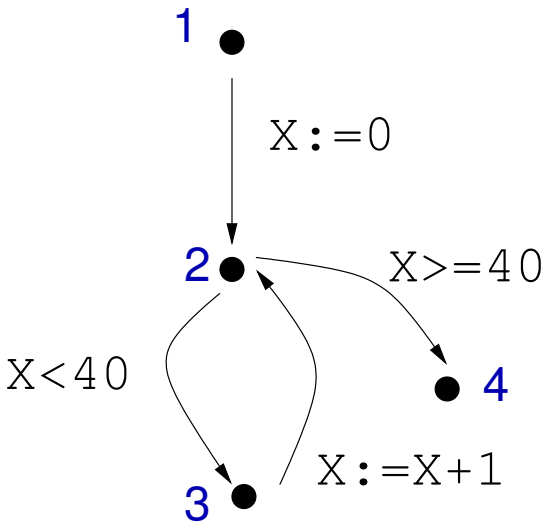
ℓ	$\mathcal{Y}_\ell^{\sharp 0}$	$\mathcal{Y}_\ell^{\sharp 1}$	$\mathcal{Y}_\ell^{\sharp 2}$
1	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$
2 $\triangle$	≥ 0	$\in [0, 40]$	$\in [0, 40]$
3	$\in [0, 39]$	$\in [0, 39]$	$\in [0, 39]$
4	≥ 40	≥ 40	$= 40$

Narrowing at 2 gives:

$$\mathcal{Y}_2^{\sharp 1} = [0, +\infty[\triangle ([0, 0] \cup^\sharp [1, 40]) = [0, +\infty[\triangle [0, 40] = [0, 40]$$
$$\mathcal{Y}_2^{\sharp 2} = \quad [0, 40] \triangle ([0, 0] \cup^\sharp [1, 40]) = \quad [0, 40] \triangle [0, 40] = [0, 40]$$

Analysis with narrowing example

Example with $\mathcal{W} = \{2\}$



ℓ	$\mathcal{Y}_\ell^{\sharp 0}$	$\mathcal{Y}_\ell^{\sharp 1}$	$\mathcal{Y}_\ell^{\sharp 2}$	$\mathcal{Y}_\ell^{\sharp 3}$
1	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$
2 $\triangle$	≥ 0	$\in [0, 40]$	$\in [0, 40]$	$\in [0, 40]$
3	$\in [0, 39]$	$\in [0, 39]$	$\in [0, 39]$	$\in [0, 39]$
4	≥ 40	≥ 40	$= 40$	$= 40$

Narrowing at 2 gives:

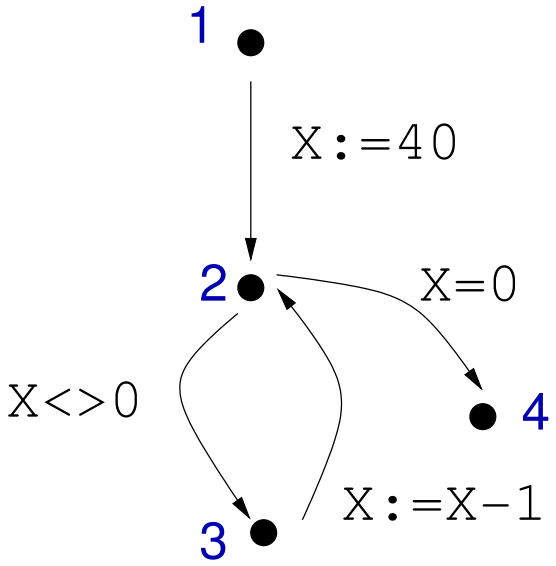
$$\mathcal{Y}_2^{\sharp 1} = [0, +\infty[\triangle ([0, 0] \cup^\sharp [1, 40]) = [0, +\infty[\triangle [0, 40] = [0, 40]$$
$$\mathcal{Y}_2^{\sharp 2} = \quad [0, 40] \triangle ([0, 0] \cup^\sharp [1, 40]) = \quad [0, 40] \triangle [0, 40] = [0, 40]$$

Then $\mathcal{Y}_2^{\sharp 2} : \mathtt{X} \in [0, 40]$ gives $\mathcal{Y}_4^{\sharp 3} : \mathtt{X} = 40$

We found the most precise invariants!

Improving the widening

Example of imprecise analysis

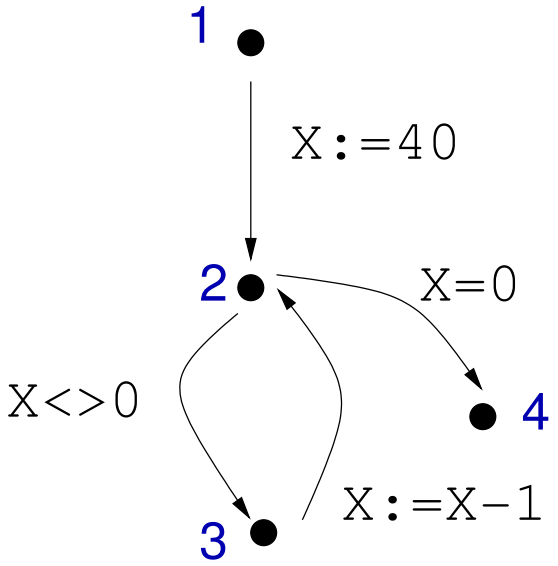


ℓ	intervals with ∇
1	$\top^\sharp$
2 ∇	$\mathtt{X} \leq 40$
3	$\mathtt{X} \leq 40$
4	$\mathtt{X} = 0$

The interval domain cannot prove that $\mathtt{X} \geq 0$ at 2,

Improving the widening

Example of imprecise analysis

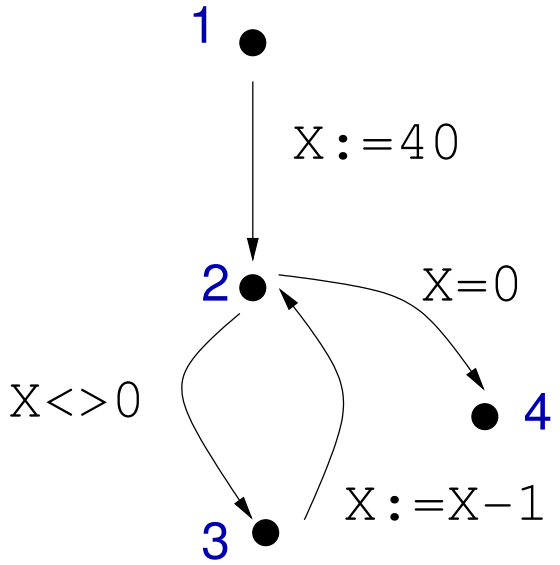


ℓ	intervals with ∇	signs
1	$\top^\sharp$	$\top^\sharp$
2 ∇	$X \leq 40$	$X \geq 0$
3	$X \leq 40$	$X > 0$
4	$X = 0$	$X = 0$

The interval domain cannot prove that $X \geq 0$ at 2, while the (less powerful) sign domain can!

Improving the widening

Example of imprecise analysis



ℓ	intervals with ∇	signs	intervals with ∇'
1	$\top^\sharp$	$\top^\sharp$	$\top^\sharp$
2 ∇	$\mathtt{X} \leq 40$	$\mathtt{X} \geq 0$	$\mathtt{X} \in [0, 40]$
3	$\mathtt{X} \leq 40$	$\mathtt{X} > 0$	$\mathtt{X} \in [0, 40]$
4	$\mathtt{X} = 0$	$\mathtt{X} = 0$	$\mathtt{X} = 0$

The interval domain cannot prove that $\mathtt{X} \geq 0$ at 2, while the (less powerful) sign domain can!

Solution: improve the interval widening

$$[a, b] \nabla' [c, d] \stackrel{\text{def}}{=} \left[\begin{cases} a & \text{if } a \leq c \\ 0 & \text{if } 0 \leq c < a \\ -\infty & \text{otherwise} \end{cases} , \begin{cases} b & \text{if } b \geq d \\ 0 & \text{if } 0 \geq b > d \\ +\infty & \text{otherwise} \end{cases} \right]$$

(∇' checks the stability of 0)

Widening with thresholds

```
X:=0;
while • 1=1 do
  if [0,1]=0 then
    X:=X+1;
    if X>99 then X:=0 fi
  fi
done
```

We wish to prove that $X \in [0, 99]$ at •

- widening at • finds the loop invariant $\mathtt{X} \in [0, +\infty[$
  $\mathcal{X}^\sharp_\bullet = [0,0] \triangledown ([0,0] \cup^\sharp [0,1]) = [0,0] \triangledown [0,1] = [0, +\infty[$
- narrowing is unable to refine the invariant:
  $\mathcal{Y}^\sharp_\bullet = [0, +\infty[\triangle ([0,0] \cup^\sharp [0, +\infty[) = [0, +\infty[$

  (the code that limits X is not executed at every loop iteration)

Widening with thresholds (cont.)

Solution:

Choose a finite set T of thresholds, containing $+\infty$ and $-\infty$.

Definition: widening with thresholds ∇^T

$$[a,b] \nabla^T [c,d] \stackrel{\text{def}}{=} \left[\begin{cases} a & \text{if } a \leq c \\ \max\{x \in T \mid x \leq c\} & \text{otherwise} \end{cases}, \right.$$

$$\left. \begin{cases} b & \text{if } b \geq d \\ \min\{x \in T \mid x \geq d\} & \text{otherwise} \end{cases} \right]$$

The widening tests and stops at the first stable bound in T

Widening with thresholds (cont.)

Applications:

- On the previous example, we find:
  $\mathtt{X} \in [\, 0,\ \min\{x \in T \mid x \geq 99\}\,]$
- Useful when it is easy to find a 'good' set T
  Example: array bound-checking
- Useful if an over-approximation of the bound is sufficient
  Example: arithmetic overflow checking

Limitations: only works if some non-∞ bound in T is stable

Example: with $T = \{\, 5, 15 \,\}$

```
while 1=1 do
  X:=X+1;
  if X>10 then X=0 fi
done
```

15 is stable

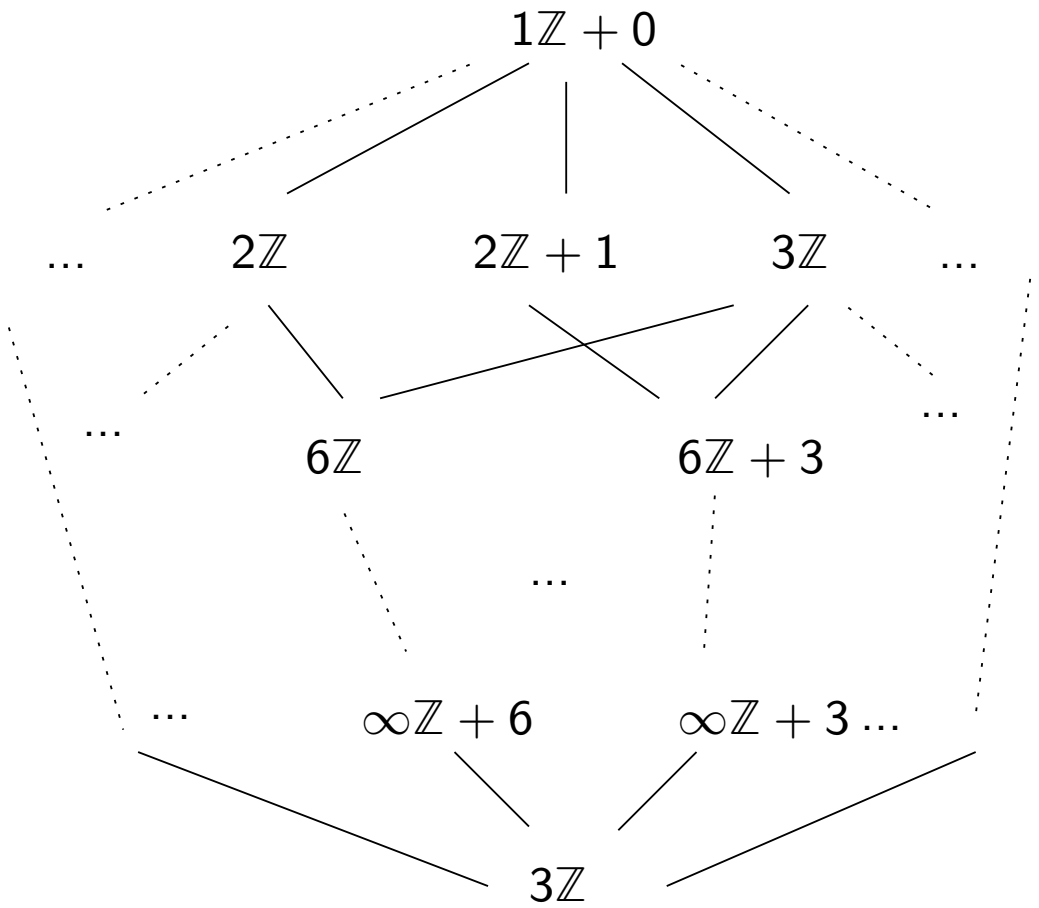
```
while 1=1 do
  X:=X+1;
  if X<>10 then X=0 fi
done
```

no stable bound

The congruence domain

The integer congruence lattice

$$\mathcal{B}^\sharp \stackrel{\text{def}}{=} \{ (a\mathbb{Z} + b) \mid a \in \mathbb{N}^* \cup \{\infty\},\ b \in \mathbb{Z} \} \cup \{ \bot^\sharp \}$$



Introduced by [Granger 89].

We take $\mathbb{I} = \mathbb{Z}$.

The integer congruence lattice (cont.)

Concretization:

$$\gamma_b(\mathcal{X}^\sharp) \stackrel{\text{def}}{=} \begin{cases} \{ ak + b \mid k \in \mathbb{Z} \} & \text{if } \mathcal{X}^\sharp = (a\mathbb{Z} + b),\ a \neq \infty \\ \{ b \} & \text{if } \mathcal{X}^\sharp = (\infty\mathbb{Z} + b) \\ \emptyset & \text{if } \mathcal{X}^\sharp = \bot^\sharp \end{cases}$$

γ_b is not injective: $\gamma_b(2\mathbb{Z} + 1) = \gamma_b(2\mathbb{Z} + 3)$.

Definitions:

Given $x, x' \in \mathbb{Z},\ y, y' \in \mathbb{N}^* \cup \{ \infty \}$, we define:

- $y / y' \iff^{\text{def}} y$ divides y' $(\exists k \in \mathbb{N}^*,\ y' = ky)$ or $y' = \infty$
- $x \equiv x'\ [y] \iff^{\text{def}} x \neq x'$ and $y / |x - x'|$, or $x = x'$
- $\vee$ is the LCM, extended with $y \vee \infty \stackrel{\text{def}}{=} \infty \vee y \stackrel{\text{def}}{=} \infty$
- $\wedge$ is the GCD, extended with $y \wedge \infty \stackrel{\text{def}}{=} \infty \wedge y \stackrel{\text{def}}{=} y$

$(\mathbb{N}^* \cup \{ \infty \}, /, \vee, \wedge, 1, \infty)$ is a complete distributive lattice.

Abstract congruence operators

Complete lattice structure on $\mathcal{B}^\sharp$:

- $(a\mathbb{Z}+b) \subseteq^\sharp (a'\mathbb{Z}+b') \iff^{\text{def}} a'/a$ and $b \equiv b'\ [a']$
- $\top^\sharp \stackrel{\text{def}}{=} (1\mathbb{Z}+0)$
- $(a\mathbb{Z}+b) \cup^\sharp (a'\mathbb{Z}+b') \stackrel{\text{def}}{=} (a \wedge a' \wedge |b-b'|)\mathbb{Z}+b$
- $(a\mathbb{Z}+b) \cap^\sharp (a'\mathbb{Z}+b') \stackrel{\text{def}}{=} \begin{cases} (a \vee a')\mathbb{Z}+b'' & \text{if } b \equiv b'\ [a \wedge a'] \\ \bot^\sharp & \text{otherwise} \end{cases}$

  b'' such that $b'' \equiv b\ [a \vee a'] \equiv b'\ [a \vee a']$ is given by Bezout's Theorem.

Galois connection: $\alpha_b(\mathcal{X}) = \bigcup_{c \in \mathcal{X}}^\sharp (\infty\mathbb{Z}+c)$

(up to equivalence $a\mathbb{Z}+b \equiv a'\mathbb{Z}+b' \iff^{\text{def}} a = a' \wedge b \equiv b'\ [a]$)

Abstract congruence operators (cont.)

Arithmetic operators:

$$[c, c']^\sharp \stackrel{\text{def}}{=} \begin{cases} \infty\mathbb{Z} + c & \text{if } c = c' \\ \top^\sharp & \text{otherwise} \end{cases}$$

$$-^\sharp (a\mathbb{Z} + b) \stackrel{\text{def}}{=} a\mathbb{Z} + (-b)$$

$$(a\mathbb{Z} + b) +^\sharp (a'\mathbb{Z} + b') \stackrel{\text{def}}{=} (a \wedge a')\mathbb{Z} + (b + b')$$

$$(a\mathbb{Z} + b) -^\sharp (a'\mathbb{Z} + b') \stackrel{\text{def}}{=} (a \wedge a')\mathbb{Z} + (b - b')$$

$$(a\mathbb{Z} + b) \times^\sharp (a'\mathbb{Z} + b') \stackrel{\text{def}}{=} (aa' \wedge ab' \wedge a'b)\mathbb{Z} + bb'$$

$$(a\mathbb{Z} + b) /^\sharp (a'\mathbb{Z} + b') \stackrel{\text{def}}{=}$$

$$\begin{cases} \bot^\sharp & \text{if } a'\mathbb{Z} + b' = \infty\mathbb{Z} + 0 \\ (a/|b'|)\mathbb{Z} + (b/b') & \text{if } a' = 0,\ b' \neq 0,\ b' | a, \text{ and } b' | b \\ \top^\sharp & \text{otherwise (not optimal)} \end{cases}$$

Abstract congruence operators (cont.)

Test operators:

$$\overleftarrow{\leq 0}^\sharp (a\mathbb{Z}+b) \stackrel{\text{def}}{=} \begin{cases} \bot^\sharp & \text{if } a=\infty,\ b>0 \\ a\mathbb{Z}+b & \text{otherwise} \end{cases}$$

$$\vdots$$

Note: better than the generic $\overleftarrow{\leq 0}^\sharp (\mathcal{X}^\sharp) \stackrel{\text{def}}{=} \mathcal{X}^\sharp \cap^\sharp]-\infty, 0]^\sharp = \mathcal{X}^\sharp$

Extrapolation operators:

- no infinite increasing chain $\Longrightarrow$ no need for ∇
- infinite decreasing chains $\Longrightarrow$ $\triangle$ needed

$$(a\mathbb{Z}+b) \ \triangle \ (a'\mathbb{Z}+b') \stackrel{\text{def}}{=} \begin{cases} a'\mathbb{Z}+b' & \text{if } a=1 \\ a\mathbb{Z}+b & \text{otherwise} \end{cases}$$

Congruence analysis example

```
X:=0; Y:=2;
while • X<40 do
  X:=X+2;
  if X<5 then Y:=Y+18 fi;
  if X>8 then Y:=Y-30 fi
done
```

We find, at •, the loop invariant $\begin{cases} \mathtt{X} \in 2\mathbb{Z} \\ \mathtt{Y} \in 6\mathbb{Z}+2 \end{cases}$

Relational domains

The need for relational domains

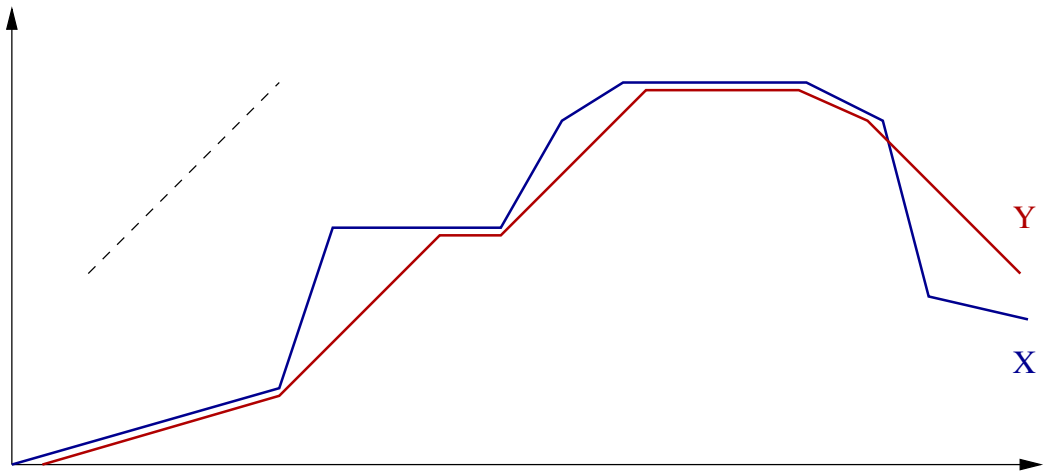
Accumulated loss of precision

Non-relation domains cannot represent variable relationships

Rate limiter

```
Y:=0; while • 1=1 do
  X:=[-128,128]; D:=[0,16];
  S:=Y; Y:=X; R:=X-S;
  if R<=-D then Y:=S-D fi;
  if R>=D then Y:=S+D fi
done
```

X: input signal
Y: output signal
S: last output
R: delta Y-S
D: max. allowed for |R|



Accumulated loss of precision

Non-relation domains cannot represent variable relationships

Rate limiter

```
Y:=0; while • 1=1 do
  X:=[-128,128]; D:=[0,16];
  S:=Y; Y:=X; R:=X-S;
  if R<=-D then Y:=S-D fi;
  if R>=D then Y:=S+D fi
done
```

X: input signal
Y: output signal
S: last output
R: delta Y-S
D: max. allowed for |R|

Iterations in the interval domain (without widening):

$\mathcal{X}_\bullet^{\sharp 0}$	$\mathcal{X}_\bullet^{\sharp 1}$	$\mathcal{X}_\bullet^{\sharp 2}$	...	$\mathcal{X}_\bullet^{\sharp n}$
$\mathtt{Y} = 0$	$\lvert\mathtt{Y}\rvert \leq 144$	$\lvert\mathtt{Y}\rvert \leq 160$	...	$\lvert\mathtt{Y}\rvert \leq 128 + 16n$

In fact, $\mathtt{Y} \in [-128, 128]$ always holds.

To prove that, e.g. $\mathtt{Y} \geq -128$, we must be able to:

- represent the properties $\mathtt{R} = \mathtt{X} - \mathtt{S}$ and $\mathtt{R} \leq -\mathtt{D}$
- combine them to deduce $\mathtt{S} - \mathtt{X} \geq \mathtt{D}$, and then $\mathtt{Y} = \mathtt{S} - \mathtt{D} \geq \mathtt{X}$

The need for relational loop invariants

To prove some invariant after the end of a loop,
we often need to find a loop invariant of a more complex form

relational loop invariant

```
X:=0; I:=1;
while • I<5000 do
  if [0,1]=1 then X:=X+1 else X:=X-1 fi;
  I:=I+1
done ♦
```

A non-relational analysis finds at ♦ that $\mathtt{I} = 5000$ and $\mathtt{X} \in \mathbb{Z}$

The best invariant is: $(\mathtt{I} = 5000) \wedge (\mathtt{X} \in [-4999, 4999]) \wedge (\mathtt{X} \equiv 0\ [2])$

To find this non-relational invariant, we must find a relational loop invariant at •: $(-\mathtt{I} < \mathtt{X} < \mathtt{I}) \wedge (\mathtt{X} + \mathtt{I} \equiv 1\ [2]) \wedge (\mathtt{I} \in [1, 5000])$,
and apply the loop exit condition $\mathrm{C}^\sharp[\![\, \mathtt{I} >= 5000 \,]\!]$

Modular analysis

store the maximum of X,Y,0 into Z

```
max(X,Y,Z)

  Z :=X ;
  if Y  > Z  then Z :=Y ;
  if Z  < 0 then Z :=0;
```

Modular analysis:

- analyze a procedure once (procedure summary)
- reuse the summary at each call site (instantiation)

  $\Longrightarrow$ improved efficiency

Modular analysis

store the maximum of X,Y,0 into Z'

```
max(X,Y,Z)
  X':=X; Y':=Y; Z':=Z;
  Z':=X';
  if Y' > Z' then Z':=Y';
  if Z' < 0 then Z':=0;
```

$(Z' \geq X \wedge Z' \geq Y \wedge Z' \geq 0 \wedge X' = X \wedge Y' = Y)$

Modular analysis:

- analyze a procedure once (procedure summary)
- reuse the summary at each call site (instantiation)
  $\Longrightarrow$ improved efficiency
- infer a relation between input X,Y,Z and output X′,Y′,Z′ values
  $\mathcal{P}((\mathbb{V} \to \mathbb{R}) \times (\mathbb{V} \to \mathbb{R})) \equiv \mathcal{P}((\mathbb{V} \times \mathbb{V}) \to \mathbb{R})$
- requires inferring relational information

[Jeannet 09]

The polyhedra domain

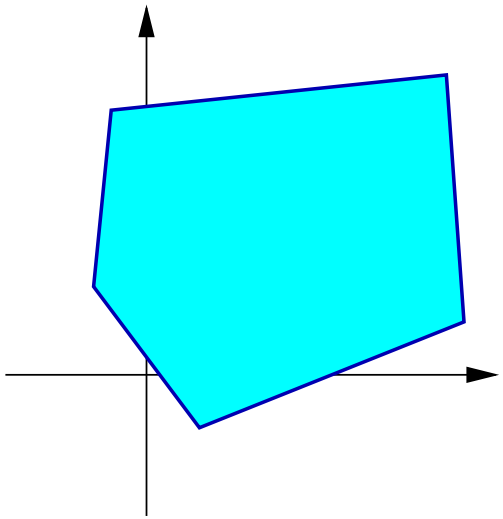
The polyhedra domain

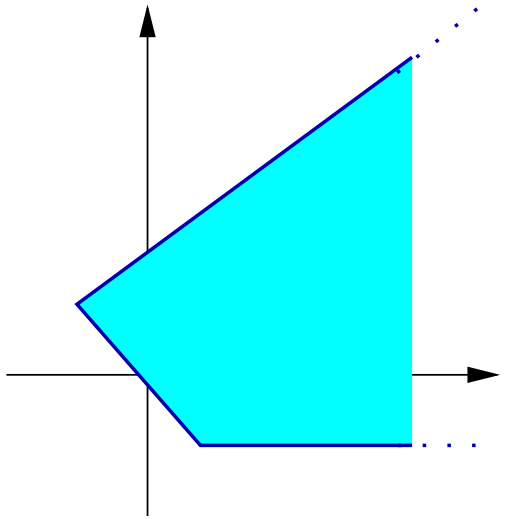
Here, $\mathbb{I} \in \{\mathbb{Q}, \mathbb{R}\}$

We look for invariants of the form: $\bigwedge_j \left(\sum_{i=1}^{n} \alpha_{ij} \mathtt{V}_i \geq \beta_j \right)$

We use the polyhedra domain proposed by [Cousot Halbwachs 78]:

$\mathcal{D}^\sharp \stackrel{\text{def}}{=} \{\text{closed convex polyhedra of } \mathbb{V} \to \mathbb{I}\}$





Note: polyhedra need not be bounded ($\neq$ polytopes)

Double description of polyhedra

Polyhedra have dual representations (Weyl–Minkowski Theorem) (see [Schriver 86])

Constraint representation

$\langle \mathbf{M}, \vec{C} \rangle$ with $\mathbf{M} \in \mathbb{I}^{m \times n}$ and $\vec{C} \in \mathbb{I}^m$

represents: $\gamma(\langle \mathbf{M}, \vec{C} \rangle) \stackrel{\text{def}}{=} \{ \vec{V} \mid \mathbf{M} \times \vec{V} \geq \vec{C} \}$

We will also often use a constraint set notation $\{ \sum_i \alpha_{ij} \mathtt{V}_i \geq \beta_j \}$

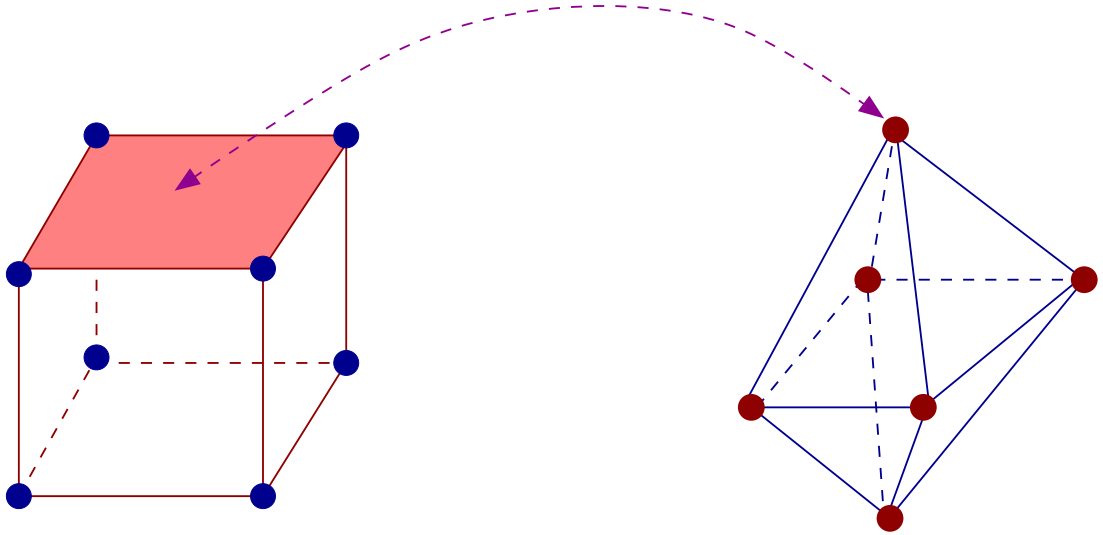
Generator representation

$[\mathbf{P}, \mathbf{R}]$ where

- $\mathbf{P} \in \mathbb{I}^{n \times p}$ is a set of p points: $\vec{P}_1, \ldots, \vec{P}_p$
- $\mathbf{R} \in \mathbb{I}^{n \times r}$ is a set of r rays: $\vec{R}_1, \ldots, \vec{R}_r$

$$\gamma([\mathbf{P}, \mathbf{R}]) \stackrel{\text{def}}{=} \left\{ \left(\sum_{j=1}^{p} \alpha_j \vec{P}_j \right) + \left(\sum_{j=1}^{r} \beta_j \vec{R}_j \right) \mid \forall j, \alpha_j, \beta_j \geq 0 : \sum_{j=1}^{p} \alpha_j = 1 \right\}$$

Duality in polyhedra



P $\qquad\qquad$ P^*

$0x + 0y + 1z \leq 1 \quad \Longleftrightarrow \quad (0, 0, 1)$

Duality:

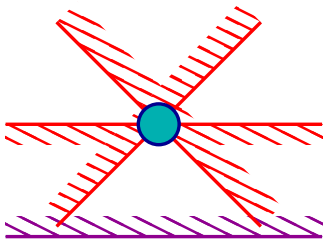
- the generators of P^* are the constraints of P
- the constraints of P^* are the generators of P
- $P^{**} = P$

Polyhedra representations

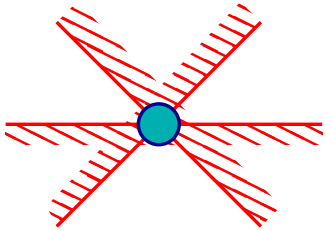
Minimal representations

- A constraint / generator system is minimal if no constraint / generator can be omitted without changing the concretization
- Minimal representations are not unique

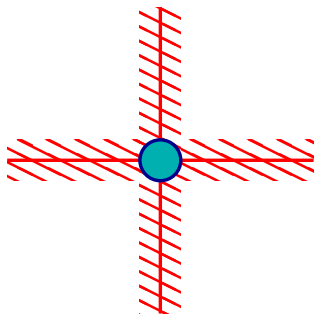
Example: constraint representation for a point



(a)



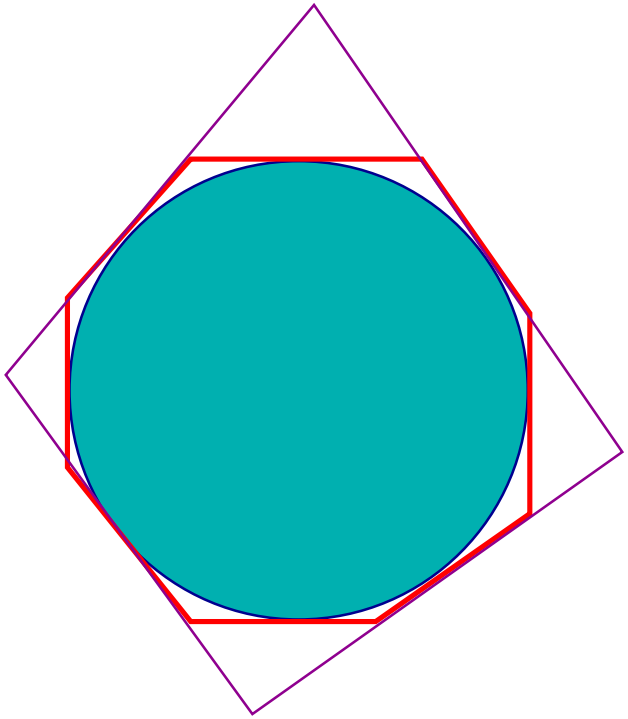
(b)



(c)

- (a) $y + x \geq 0, y - x \geq 0, y \leq 0, y \geq -5$ (non mimimal)
- (b) $y + x \geq 0, y - x \geq 0, y \leq 0$ (minimal)
- (c) $x \leq 0, x \geq 0, y \leq 0, y \geq 0$ (minimal)

Polyhedra representations (cont.)



- No memory bound on the representations (even minimal ones)
- No best abstraction α

Chernikova's algorithm

Algorithm by [Chernikova 68], improved by [LeVerge 92] to switch from a constraint system to an equivalent generator system

Why? Most operators are easier on one representation

Notes:

- By duality, we can use the same algorithm to switch from generators to constraints
- The minimal generator system can be exponential in the original constraint system (e.g., hypercube: $2n$ constraints, 2^n vertices)
- Equality constraints and lines (pairs of opposed rays) may be handled separately and more efficiently
- Chernikova's algorithm minimizes the representation on-the-fly (not presented here)

Algorithm: incrementally add constraints one by one

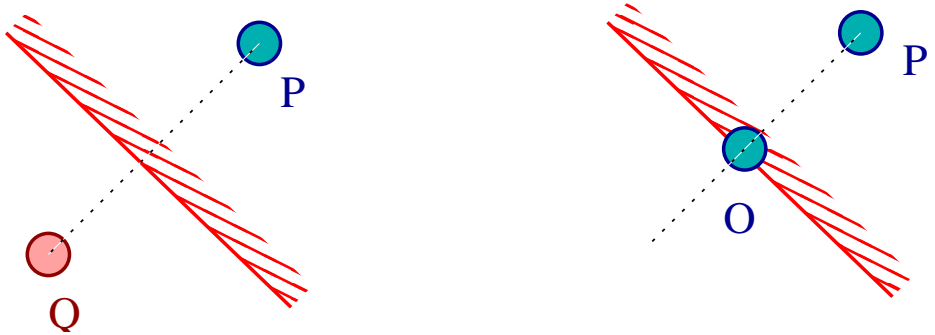
Start with:
$$\begin{cases} \mathbf{P}_0 = \{ (0, \ldots, 0) \} & \text{(origin)} \\ \mathbf{R}_0 = \{ \vec{x}_i, -\vec{x}_i \mid 1 \leq i \leq n \} & \text{(axes)} \end{cases}$$

Chernikova's algorithm (cont.)

Update $[\mathbf{P}_{k-1}, \mathbf{R}_{k-1}]$ to $[\mathbf{P}_k, \mathbf{R}_k]$
by adding one constraint $\vec{M}_k \cdot \vec{V} \geq C_k \in \langle \mathbf{M}, \vec{C} \rangle$:

start with $\mathbf{P}_k = \mathbf{R}_k = \emptyset$,

- for any $\vec{P} \in \mathbf{P}_{k-1}$ s.t. $\vec{M}_k \cdot \vec{P} \geq C_k$, add $\vec{P}$ to $\mathbf{P}_k$
- for any $\vec{R} \in \mathbf{R}_{k-1}$ s.t. $\vec{M}_k \cdot \vec{R} \geq 0$, add $\vec{R}$ to $\mathbf{R}_k$
- for any $\vec{P}, \vec{Q} \in \mathbf{P}_{k-1}$ s.t. $\vec{M}_k \cdot \vec{P} > C_k$ and $\vec{M}_k \cdot \vec{Q} < C_k$, add to $\mathbf{P}_k$:
  $\vec{O} \stackrel{\text{def}}{=} \frac{C_k - \vec{M}_k \cdot \vec{Q}}{\vec{M}_k \cdot \vec{P} - \vec{M}_k \cdot \vec{Q}} \vec{P} - \frac{C_k - \vec{M}_k \cdot \vec{P}}{\vec{M}_k \cdot \vec{P} - \vec{M}_k \cdot \vec{Q}} \vec{Q}$



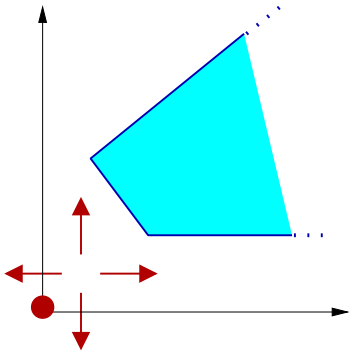
Chernikova's algorithm (cont.)

- for any $\vec{R}, \vec{S} \in \mathbf{R}_{k-1}$ s.t. $\vec{M}_k \cdot \vec{R} > 0$ and $\vec{M}_k \cdot \vec{S} < 0$, add to $\mathbf{R}_k$:
  $\vec{O} \stackrel{\text{def}}{=} (\vec{M}_k \cdot \vec{S})\vec{R} - (\vec{M}_k \cdot \vec{R})\vec{S}$

- for any $\vec{P} \in \mathbf{P}_{k-1}$, $\vec{R} \in \mathbf{R}_{k-1}$ s.t. either $\vec{M}_k \cdot \vec{P} > C_k$ and $\vec{M}_k \cdot \vec{R} < 0$, or $\vec{M}_k \cdot \vec{P} < C_k$ and $\vec{M}_k \cdot \vec{R} > 0$, add to $\mathbf{P}_k$:
  $\vec{O} \stackrel{\text{def}}{=} \vec{P} + \frac{C_k - \vec{M}_k \cdot \vec{P}}{\vec{M}_k \cdot \vec{R}} \vec{R}$

Chernikova's algorithm example

Example:

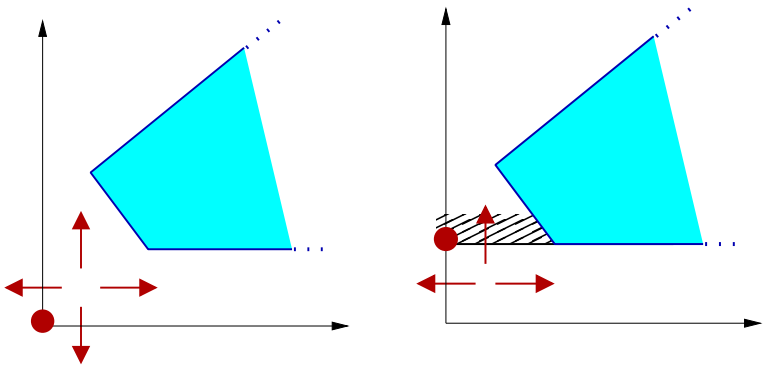


(0)

$$\mathbf{P}_0 = \{(0,0)\} \qquad \mathbf{R}_0 = \{(1,0),\ (-1,0),\ (0,1),\ (0,-1)\}$$

Chernikova's algorithm example

Example:



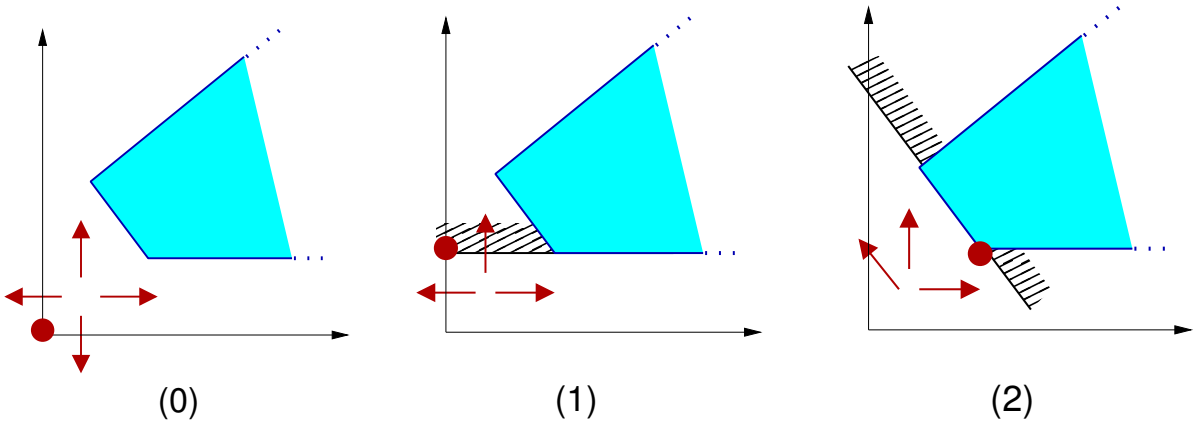
$\mathtt{Y} \geq 1$

$\mathbf{P}_0 = \{(0,0)\}$
$\mathbf{P}_1 = \{(0,1)\}$

$\mathbf{R}_0 = \{(1,0), (-1,0), (0,1), (0,-1)\}$
$\mathbf{R}_1 = \{(1,0), (-1,0), (0,1)\}$

Chernikova's algorithm example

Example:



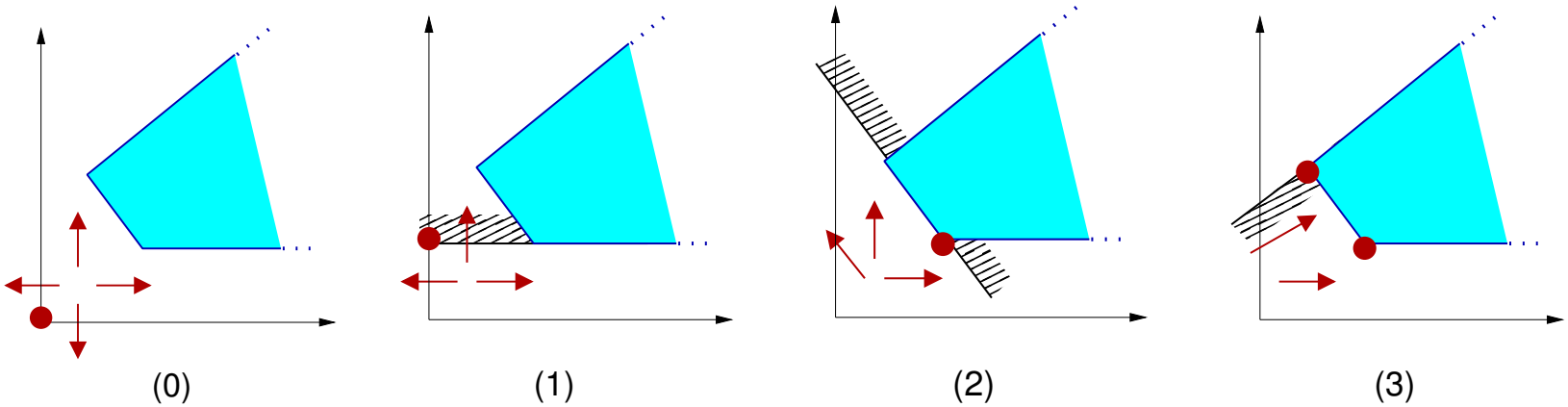
$\mathtt{Y} \geq 1$
$\mathtt{X} + \mathtt{Y} \geq 3$

$\mathbf{P}_0 = \{(0,0)\}$
$\mathbf{P}_1 = \{(0,1)\}$
$\mathbf{P}_2 = \{(2,1)\}$

$\mathbf{R}_0 = \{(1,0), (-1,0), (0,1), (0,-1)\}$
$\mathbf{R}_1 = \{(1,0), (-1,0), (0,1)\}$
$\mathbf{R}_2 = \{(1,0), (-1,1), (0,1)\}$

Chernikova's algorithm example

Example:



	$\mathbf{P}_0 = \{(0,0)\}$	$\mathbf{R}_0 = \{(1,0), (-1,0), (0,1), (0,-1)\}$
$\mathtt{Y} \geq 1$	$\mathbf{P}_1 = \{(0,1)\}$	$\mathbf{R}_1 = \{(1,0), (-1,0), (0,1)\}$
$\mathtt{X} + \mathtt{Y} \geq 3$	$\mathbf{P}_2 = \{(2,1)\}$	$\mathbf{R}_2 = \{(1,0), (-1,1), (0,1)\}$
$\mathtt{X} - \mathtt{Y} \leq 1$	$\mathbf{P}_3 = \{(2,1), (1,2)\}$	$\mathbf{R}_3 = \{(0,1), (1,1)\}$

Operators on polyhedra

Given $\mathcal{X}^\sharp, \mathcal{Y}^\sharp \neq \bot^\sharp$, we define:

$$\mathcal{X}^\sharp \subseteq^\sharp \mathcal{Y}^\sharp \quad \overset{\text{def}}{\iff} \quad \begin{cases} \forall \vec{P} \in \mathbf{P}_{\mathcal{X}^\sharp} : \mathbf{M}_{\mathcal{Y}^\sharp} \times \vec{P} \geq \vec{C}_{\mathcal{Y}^\sharp} \\ \forall \vec{R} \in \mathbf{R}_{\mathcal{X}^\sharp} : \mathbf{M}_{\mathcal{Y}^\sharp} \times \vec{R} \geq \vec{0} \end{cases}$$

$$\mathcal{X}^\sharp =^\sharp \mathcal{Y}^\sharp \quad \overset{\text{def}}{\iff} \quad \mathcal{X}^\sharp \subseteq^\sharp \mathcal{Y}^\sharp \quad \text{and} \quad \mathcal{Y}^\sharp \subseteq^\sharp \mathcal{X}^\sharp$$

$$\mathcal{X}^\sharp \cap^\sharp \mathcal{Y}^\sharp \quad \overset{\text{def}}{=} \quad \left\langle \begin{bmatrix} \mathbf{M}_{\mathcal{X}^\sharp} \\ \mathbf{M}_{\mathcal{Y}^\sharp} \end{bmatrix}, \begin{bmatrix} \vec{C}_{\mathcal{X}^\sharp} \\ \vec{C}_{\mathcal{Y}^\sharp} \end{bmatrix} \right\rangle \quad \text{(join constraint sets)}$$

$$\mathcal{X}^\sharp \cup^\sharp \mathcal{Y}^\sharp \quad \overset{\text{def}}{=} \quad [\,[\mathbf{P}_{\mathcal{X}^\sharp}\ \mathbf{P}_{\mathcal{Y}^\sharp}],\ [\mathbf{R}_{\mathcal{X}^\sharp}\ \mathbf{R}_{\mathcal{Y}^\sharp}]\,] \quad \text{(join generator sets)}$$

Remarks:

- $\subseteq^\sharp$, $=^\sharp$ and $\cap^\sharp$ are exact
- $\cup^\sharp$ is optimal: we get the topological closure of the convex hull of $\gamma(\mathcal{X}^\sharp) \cup \gamma(\mathcal{Y}^\sharp)$

Operators on polyhedra (cont.)

$$\mathrm{C}^\sharp[\![\textstyle\sum_i \alpha_i \mathtt{V}_i + \beta \geq 0]\!]\, \mathcal{X}^\sharp \stackrel{\text{def}}{=} \left\langle \begin{bmatrix} \mathbf{M}_{\mathcal{X}^\sharp} \\ \alpha_1 \cdots \alpha_n \end{bmatrix}, \begin{bmatrix} \vec{C}_{\mathcal{X}^\sharp} \\ -\beta \end{bmatrix} \right\rangle$$

$$\mathrm{C}^\sharp[\![\mathtt{V}_j :=]-\infty, +\infty[ ]\!]\, \mathcal{X}^\sharp \stackrel{\text{def}}{=} [\, \mathbf{P}_{\mathcal{X}^\sharp},\ [\, \mathbf{R}_{\mathcal{X}^\sharp}\ \ \vec{x}_j\ \ (-\vec{x}_j)\,]\,]$$

$$\mathrm{C}^\sharp[\![\mathtt{V}_j := \textstyle\sum_i \alpha_i \mathtt{V}_i + \beta]\!]\, \mathcal{X}^\sharp \stackrel{\text{def}}{=}$$

if $\alpha_j = 0$, $(\mathrm{C}^\sharp[\![\sum_i \alpha_i \mathtt{V}_i - \mathtt{V}_j + \beta = 0]\!] \circ \mathrm{C}^\sharp[\![\mathtt{V}_j :=]-\infty, +\infty[ ]\!])\mathcal{X}^\sharp$

if $\alpha_j \neq 0$, $\langle \mathbf{M}, \vec{C} \rangle$ where $\mathtt{V}_j$ is replaced with $\frac{1}{\alpha_j}(\mathtt{V}_j - \sum_{i \neq j} \alpha_i \mathtt{V}_i - \beta)$

Remarks:

- $\mathrm{C}^\sharp[\![\sum_i \alpha_i \mathtt{V}_i + \beta \geq 0]\!]$, $\mathrm{C}^\sharp[\![\mathtt{V}_j := \sum_i \alpha_i \mathtt{V}_i + \beta]\!]\, \mathcal{X}$ and $\mathrm{C}^\sharp[\![\mathtt{V}_j :=]-\infty, +\infty[ ]\!]$ are exact
- We can also define $\mathrm{C}^\sharp[\![\mathtt{V}_j := \sum_i \alpha_i \mathtt{V}_i + \beta]\!]$ on a generator system
- fall-back to $\mathrm{C}^\sharp[\![e \bowtie 0]\!]\, \mathcal{X}^\sharp \stackrel{\text{def}}{=} \mathcal{X}^\sharp$ and $\mathrm{C}^\sharp[\![\mathtt{V}_j := e]\!]\, \mathcal{X}^\sharp \stackrel{\text{def}}{=} \mathrm{C}^\sharp[\![\mathtt{V}_j :=]-\infty, +\infty[ ]\!]\, \mathcal{X}^\sharp$

Polyhedra widening

$\mathcal{D}^\sharp$ has strictly increasing infinite chains $\Longrightarrow$ we need a widening

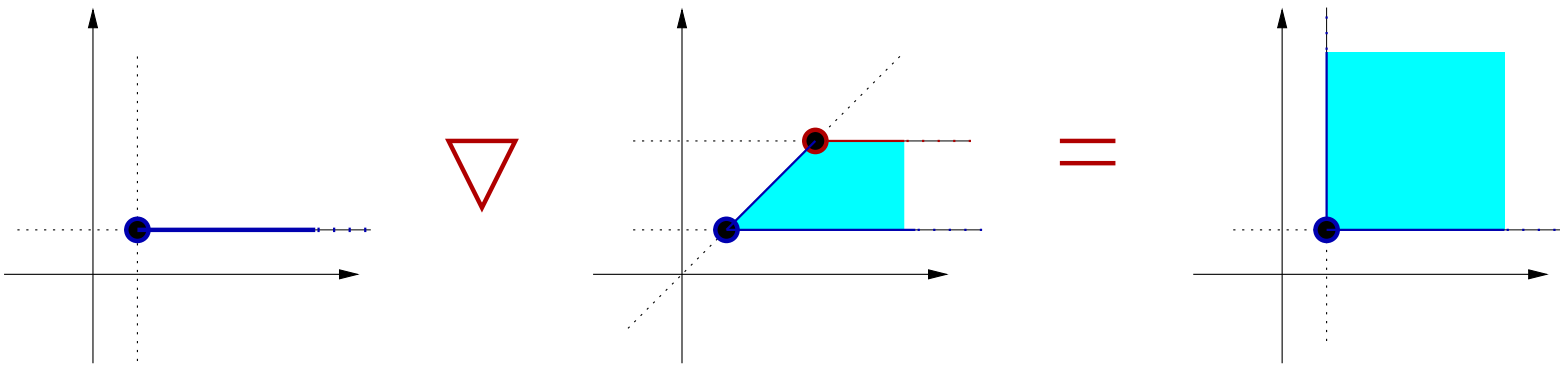
Definition:

Take $\mathcal{X}^\sharp$ and $\mathcal{Y}^\sharp$ in minimal constraint-set form

$$\mathcal{X}^\sharp \mathbin{\triangledown} \mathcal{Y}^\sharp \stackrel{\text{def}}{=} \{\, c \in \mathcal{X}^\sharp \mid \mathcal{Y}^\sharp \subseteq^\sharp \{c\} \,\}$$

We suppress any unstable constraint $c \in \mathcal{X}^\sharp$, i.e., $\mathcal{Y}^\sharp \not\subseteq^\sharp \{c\}$

Example:



Polyhedra widening

$\mathcal{D}^\sharp$ has strictly increasing infinite chains $\Longrightarrow$ we need a widening

Definition:

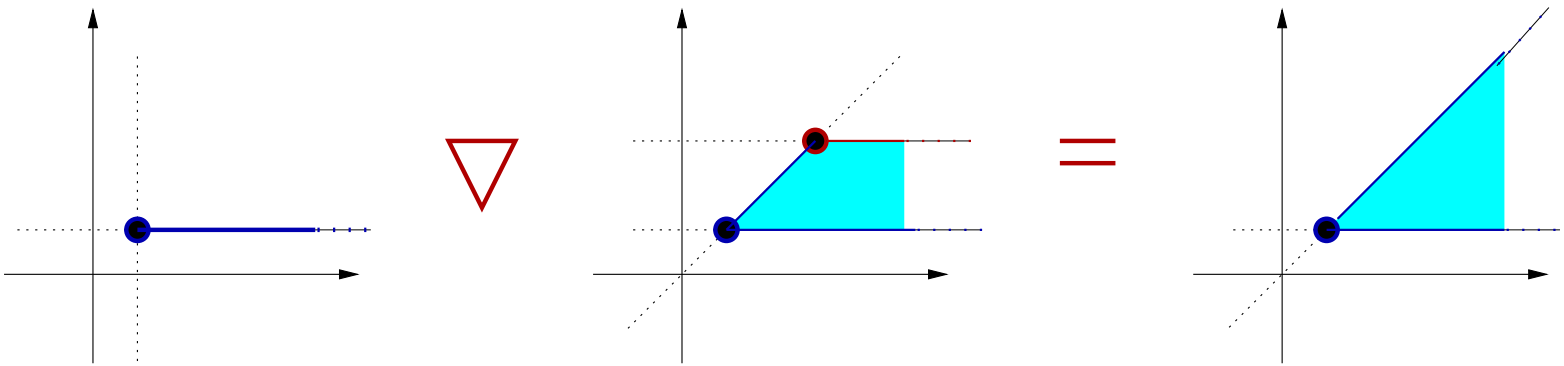
Take $\mathcal{X}^\sharp$ and $\mathcal{Y}^\sharp$ in minimal constraint-set form

$$\begin{aligned} \mathcal{X}^\sharp \triangledown \mathcal{Y}^\sharp \quad &\stackrel{\text{def}}{=} \quad \{\, c \in \mathcal{X}^\sharp \mid \mathcal{Y}^\sharp \subseteq^\sharp \{c\} \,\} \\ &\cup \quad \{\, c \in \mathcal{Y}^\sharp \mid \exists c' \in \mathcal{X}^\sharp : \mathcal{X}^\sharp =^\sharp (\mathcal{X}^\sharp \setminus c') \cup \{c\} \,\} \end{aligned}$$

We suppress any unstable constraint $c \in \mathcal{X}^\sharp$, i.e., $\mathcal{Y}^\sharp \not\subseteq^\sharp \{c\}$

We also keep constraints $c \in \mathcal{Y}^\sharp$ equivalent to those in $\mathcal{X}^\sharp$, i.e., when $\exists c' \in \mathcal{X}^\sharp : \mathcal{X}^\sharp =^\sharp (\mathcal{X}^\sharp \setminus c') \cup \{c\}$

Example:



Example analysis

Example program

```
X:=2; I:=0;
while • I<10 do
  if [0,1]=0 then X:=X+2 else X:=X-3 fi;
  I:=I+1
done♦
```

We use a finite number (one) of intersections $\cap^\sharp$ as narrowing

Iterations with widening and narrowing at • give:

$\mathcal{X}_\bullet^{\sharp 1} = \{\mathtt{X} = 2, \mathtt{I} = 0\}$

Example analysis

Example program

```
X:=2; I:=0;
while • I<10 do
  if [0,1]=0 then X:=X+2 else X:=X-3 fi;
  I:=I+1
done♦
```

We use a finite number (one) of intersections $\cap^\sharp$ as narrowing

Iterations with widening and narrowing at • give:

$$\begin{aligned}
\mathcal{X}_\bullet^{\sharp 1} &= \{\mathtt{X} = 2, \mathtt{I} = 0\} \\
\mathcal{X}_\bullet^{\sharp 2} &= \{\mathtt{X} = 2, \mathtt{I} = 0\} \mathbin{\triangledown} (\{\mathtt{X} = 2, \mathtt{I} = 0\} \cup^\sharp \{\mathtt{X} \in [-1, 4],\ \mathtt{I} = 1\}) \\
&= \{\mathtt{X} = 2, \mathtt{I} = 0\} \mathbin{\triangledown} \{\mathtt{I} \in [0, 1],\ 2 - 3\mathtt{I} \leq \mathtt{X} \leq 2\mathtt{I} + 2\ \} \\
&= \{\mathtt{I} \geq 0,\ 2 - 3\mathtt{I} \leq \mathtt{X} \leq 2\mathtt{I} + 2\}
\end{aligned}$$

Example analysis

Example program

```
X:=2; I:=0;
while • I<10 do
  if [0,1]=0 then X:=X+2 else X:=X-3 fi;
  I:=I+1
done♦
```

We use a finite number (one) of intersections $\cap^\sharp$ as narrowing

Iterations with widening and narrowing at • give:

$$
\begin{aligned}
\mathcal{X}_\bullet^{\sharp 1} &= \{\mathtt{X} = 2, \mathtt{I} = 0\} \\
\mathcal{X}_\bullet^{\sharp 2} &= \{\mathtt{X} = 2, \mathtt{I} = 0\} \mathbin{\triangledown} (\{\mathtt{X} = 2, \mathtt{I} = 0\} \cup^\sharp \{\mathtt{X} \in [-1, 4],\ \mathtt{I} = 1\}) \\
&= \{\mathtt{X} = 2, \mathtt{I} = 0\} \mathbin{\triangledown} \{\mathtt{I} \in [0, 1],\ 2 - 3\mathtt{I} \leq \mathtt{X} \leq 2\mathtt{I} + 2\ \} \\
&= \{\mathtt{I} \geq 0,\ 2 - 3\mathtt{I} \leq \mathtt{X} \leq 2\mathtt{I} + 2\} \\
\mathcal{X}_\bullet^{\sharp 3} &= \{\mathtt{I} \geq 0,\ 2 - 3\mathtt{I} \leq \mathtt{X} \leq 2\mathtt{I} + 2\} \cap^\sharp \\
&\quad (\{\mathtt{X} = 2, \mathtt{I} = 0\} \cup^\sharp \{\mathtt{I} \in [1, 10],\ 2 - 3\mathtt{I} \leq \mathtt{X} \leq 2\mathtt{I} + 2\}) \\
&= \{\mathtt{I} \in [0, 10],\ 2 - 3\mathtt{I} \leq \mathtt{X} \leq 2\mathtt{I} + 2\}
\end{aligned}
$$

Example analysis

Example program

```
X:=2; I:=0;
while • I<10 do
  if [0,1]=0 then X:=X+2 else X:=X-3 fi;
  I:=I+1
done♦
```

We use a finite number (one) of intersections $\cap^\sharp$ as narrowing

Iterations with widening and narrowing at • give:

$$
\begin{aligned}
\mathcal{X}_\bullet^{\sharp 1} &= \{\mathtt{X} = 2, \mathtt{I} = 0\} \\
\mathcal{X}_\bullet^{\sharp 2} &= \{\mathtt{X} = 2, \mathtt{I} = 0\} \mathbin{\triangledown} (\{\mathtt{X} = 2, \mathtt{I} = 0\} \cup^\sharp \{\mathtt{X} \in [-1, 4],\ \mathtt{I} = 1\}) \\
&= \{\mathtt{X} = 2, \mathtt{I} = 0\} \mathbin{\triangledown} \{\mathtt{I} \in [0, 1],\ 2 - 3\mathtt{I} \leq \mathtt{X} \leq 2\mathtt{I} + 2\ \} \\
&= \{\mathtt{I} \geq 0,\ 2 - 3\mathtt{I} \leq \mathtt{X} \leq 2\mathtt{I} + 2\} \\
\mathcal{X}_\bullet^{\sharp 3} &= \{\mathtt{I} \geq 0,\ 2 - 3\mathtt{I} \leq \mathtt{X} \leq 2\mathtt{I} + 2\} \cap^\sharp \\
&\quad (\{\mathtt{X} = 2, \mathtt{I} = 0\} \cup^\sharp \{\mathtt{I} \in [1, 10],\ 2 - 3\mathtt{I} \leq \mathtt{X} \leq 2\mathtt{I} + 2\}) \\
&= \{\mathtt{I} \in [0, 10],\ 2 - 3\mathtt{I} \leq \mathtt{X} \leq 2\mathtt{I} + 2\}
\end{aligned}
$$

At ♦ we find eventually: $\mathtt{I} = 10 \wedge \mathtt{X} \in [-28, 22]$

Other polyhedra widenings

Widening with thresholds:

Given a finite set T of constraints, we add to $\mathcal{X}^\sharp \triangledown \mathcal{Y}^\sharp$ all the constraints from T satisfied by both $\mathcal{X}^\sharp$ and $\mathcal{Y}^\sharp$

Delayed widening:

We replace $\mathcal{X}^\sharp \triangledown \mathcal{Y}^\sharp$ with $\mathcal{X}^\sharp \cup^\sharp \mathcal{Y}^\sharp$ a finite number of times
(this works for any widening and abstract domain)

See also [Bagnara 03]

(Note: no polyhedra narrowing, except finite-time intersection)

Constraint-only polyhedron domain

It is possible to use only the constraint representation:

- avoids the cost of Chernikova's algorithm
- avoids exponential generator systems (hypercubes)

The core operations are: projection and redundancy removal

Projection: using Fourier-Motzkin elimination

Fourier$(\mathcal{X}^\sharp, \mathtt{V}_k)$ eliminates $\mathtt{V}_k$ from all the constraints in $\mathcal{X}^\sharp$:

$$\begin{array}{l} \textit{Fourier}(\mathcal{X}^\sharp, \mathtt{V}_k) \stackrel{\text{def}}{=} \\ \quad \{\ (\sum_i \alpha_i \mathtt{V}_i \geq \beta) \in \mathcal{X}^\sharp \mid \alpha_k = 0\ \} \cup \\ \quad \{\ (-\alpha_k^-)c^+ + \alpha_k^+ c^- \mid \begin{array}{l} c^+ = (\sum_i \alpha_i^+ \mathtt{V}_i \geq \beta^+) \in \mathcal{X}^\sharp,\ \alpha_k^+ > 0, \\ c^- = (\sum_i \alpha_i^- \mathtt{V}_i \geq \beta^-) \in \mathcal{X}^\sharp,\ \alpha_k^- < 0\ \} \end{array} \end{array}$$

we then have:

$$\gamma(\textit{Fourier}(\mathcal{X}^\sharp, \mathtt{V}_k)) = \{\ \vec{x}[\mathtt{V}_k \mapsto v] \mid v \in \mathbb{I},\ \vec{x} \in \gamma(\mathcal{X}^\sharp)\ \}$$

Constraint-only polyhedron domain (cont.)

Fourier causes a quadratic growth in constraint number
Most such constraints are redundant

Redundancy removal: using linear programming [Schriver 86]

Let $\mathit{simplex}(\mathcal{Y}^\sharp, \vec{v}) \stackrel{\text{def}}{=} \min \{ \vec{v} \cdot \vec{y} \mid \vec{y} \in \gamma(\mathcal{Y}^\sharp) \}$

If $c = (\vec{\alpha} \cdot \vec{\mathbb{V}} \geq \beta) \in \mathcal{X}^\sharp$ and $\beta \leq \mathit{simplex}(\mathcal{X}^\sharp \setminus \{c\}, \vec{\alpha})$,
then c can be safely removed from $\mathcal{X}^\sharp$
(iterate over all constraints)

Note: running *simplex* many times can be become costly

- use fast syntactic checks first
- check against the bounding-box first

Constraint-only polyhedron domain (cont.)

Constraint-only abstract operators:

$\mathcal{X}^\sharp \subseteq^\sharp \mathcal{Y}^\sharp \iff^{\text{def}} \forall(\vec{\alpha}\cdot\vec{\mathtt{V}} \geq \beta) \in \mathcal{Y}^\sharp : \mathit{simplex}(\mathcal{X}^\sharp, \vec{\alpha}) \geq \beta$

$\mathcal{X}^\sharp =^\sharp \mathcal{Y}^\sharp \iff^{\text{def}} \mathcal{X}^\sharp \subseteq^\sharp \mathcal{Y}^\sharp$ and $\mathcal{Y}^\sharp \subseteq^\sharp \mathcal{X}^\sharp$

$\mathcal{X}^\sharp \cap^\sharp \mathcal{Y}^\sharp \stackrel{\text{def}}{=} \mathcal{X}^\sharp \cup \mathcal{Y}^\sharp$ (join constraint sets)

$\mathrm{C}^\sharp[\![\, \mathtt{V}_j :=]-\infty, +\infty[\,]\!]\, \mathcal{X}^\sharp \stackrel{\text{def}}{=} \mathit{Fourier}(\mathcal{X}^\sharp, \mathtt{V}_j)$

$\mathrm{C}^\sharp[\![\, \sum_i \alpha_i \mathtt{V}_i + \beta \geq 0 \,]\!]\, \mathcal{X}^\sharp$ as before

$\mathrm{C}^\sharp[\![\, \mathtt{V}_j := \sum_i \alpha_i \mathtt{V}_i + \beta \,]\!]\, \mathcal{X}^\sharp$ as before

Constraint-only polyhedron domain (cont.)

Constraint-only convex hull:

- Express a point $\vec{\mathtt{V}} \in \mathcal{X}^\sharp \cup^\sharp \mathcal{Y}^\sharp$ as a convex combination:
  $\vec{\mathtt{V}} = \sigma\vec{\mathtt{X}} + \sigma'\vec{\mathtt{Y}}$ for $\vec{\mathtt{X}} \in \mathcal{X}^\sharp$, $\vec{\mathtt{Y}} \in \mathcal{Y}^\sharp$, $\sigma + \sigma' = 1$, $\sigma, \sigma' \geq 0$
- as $\sigma\vec{\mathtt{X}} + \sigma'\vec{\mathtt{Y}}$ is quadratic
  we consider instead: $\vec{\mathtt{V}} = \vec{\mathtt{X}} + \vec{\mathtt{Y}}$ with $\vec{\mathtt{X}}/\sigma \in \mathcal{X}^\sharp$, $\vec{\mathtt{Y}}/\sigma' \in \mathcal{Y}^\sharp$
  i.e., $\vec{\mathtt{X}} \in \sigma\mathcal{X}^\sharp$, $\vec{\mathtt{Y}} \in \sigma'\mathcal{Y}^\sharp$
  (adds closure points on unbounded polyhedra)

Formally:

$\mathcal{X}^\sharp \cup^\sharp \mathcal{Y}^\sharp \stackrel{\text{def}}{=}$

$$\begin{array}{l} \mathit{Fourier}(\ \{\ (\sum_j \alpha_j \mathtt{X}_j - \beta\sigma \geq 0) \mid (\sum_j \alpha_j \mathtt{V}_j \geq \beta) \in \mathcal{X}^\sharp\ \} \quad \cup \\ \qquad \{\ (\sum_j \alpha_j \mathtt{Y}_j - \beta\sigma' \geq 0) \mid (\sum_j \alpha_j \mathtt{V}_j \geq \beta) \in \mathcal{Y}^\sharp\ \} \quad \cup \\ \qquad \{\ \mathtt{V}_j = \mathtt{X}_j + \mathtt{Y}_j \mid \mathtt{V}_j \in \mathbb{V}\ \} \cup \{\ \sigma \geq 0,\ \sigma' \geq 0,\ \sigma + \sigma' = 1\ \}, \\ \qquad \{\ \mathtt{X}_j, \mathtt{Y}_j \mid \mathtt{V}_j \in \mathbb{V}\ \} \cup \{\ \sigma, \sigma'\ \}\ ) \end{array}$$

[Benoi 96]

Weakly relational domains

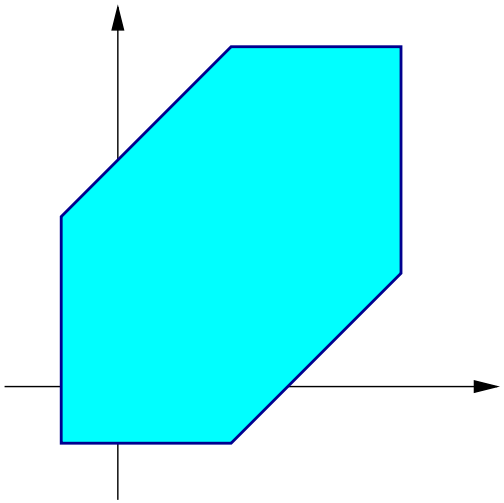
The zone domain

Here, $\mathbb{I} \in \{\mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$

We look for invariants of the form:

$$\bigwedge \mathtt{V}_i - \mathtt{V}_j \leq c \text{ or } \pm \mathtt{V}_i \leq c, \quad c \in \mathbb{I}$$

A subset of $\mathbb{I}^n$ bounded by such constraints is called a **zone**



[Miné 01]

(zones are inspired from *model-checking* of timed automata)

Machine representation

A potential constraint has the form: $\mathtt{V}_j - \mathtt{V}_i \leq c$

Potential graph: directed, weighted graph $\mathcal{G}$

- nodes are labelled with variables in $\mathbb{V}$
- we add an arc with weight c from $\mathtt{V}_i$ to $\mathtt{V}_j$ for each constraint $\mathtt{V}_j - \mathtt{V}_i \leq c$

Difference Bound Matrix (DBM)

Adjacency matrix $\mathbf{m}$ of $\mathcal{G}$:

- $\mathbf{m}$ is square, with size $n \times n$, and elements in $\mathbb{I} \cup \{+\infty\}$
- $m_{ij} = c < +\infty$ denotes the constraint $\mathtt{V}_j - \mathtt{V}_i \leq c$
- $m_{ij} = +\infty$ if there is no upper bound on $\mathtt{V}_j - \mathtt{V}_i$

Concretization:

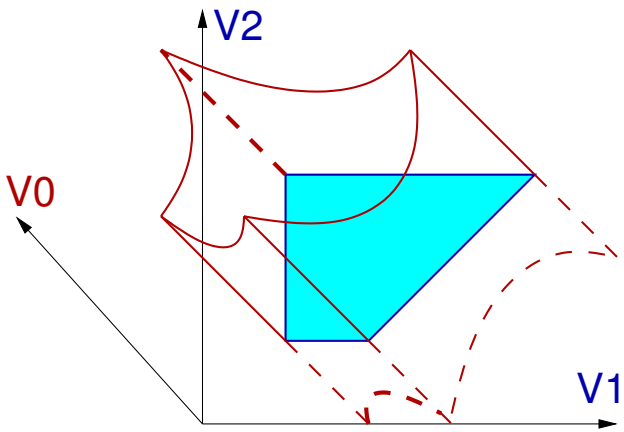
$$\gamma(\mathbf{m}) \stackrel{\text{def}}{=} \{ (v_1, \ldots, v_n) \in \mathbb{I}^n \mid \forall i, j : v_j - v_i \leq m_{ij} \}$$

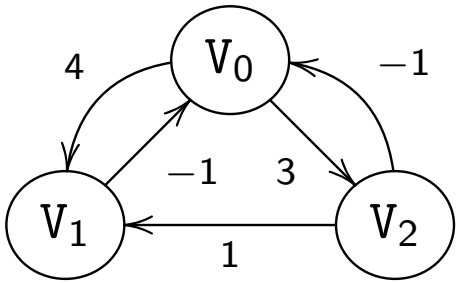
Machine representation (cont.)

Unary constraints add a constant null variable $\mathtt{V}_0$

- $\mathbf{m}$ has size $(n+1) \times (n+1)$
- $\mathtt{V}_i \leq c$ is denoted as $\mathtt{V}_i - \mathtt{V}_0 \leq c$, i.e., $m_{i0} = c$
- $\mathtt{V}_i \geq c$ is denoted as $\mathtt{V}_0 - \mathtt{V}_i \leq -c$, i.e., $m_{0i} = -c$
- γ is now: $\gamma_0(\mathbf{m}) \stackrel{\text{def}}{=} \{ (v_1, \ldots, v_n) \mid (0, v_1, \ldots, v_n) \in \gamma(\mathbf{m}) \}$

Example:





	$\mathtt{V}_0$	$\mathtt{V}_1$	$\mathtt{V}_2$
$\mathtt{V}_0$	$+\infty$	4	3
$\mathtt{V}_1$	-1	$+\infty$	$+\infty$
$\mathtt{V}_2$	-1	1	$+\infty$

The DBM lattice

$\mathcal{D}^\sharp$ contains all DBMs, plus $\bot^\sharp$

$\leq$ on $\mathbb{I} \cup \{+\infty\}$ is extended point-wisely

If $\mathbf{m}, \mathbf{n} \neq \bot^\sharp$:

$$\begin{array}{rcl}
\mathbf{m} \subseteq^\sharp \mathbf{n} & \overset{\text{def}}{\iff} & \forall i,j: m_{ij} \leq n_{ij} \\
\mathbf{m} =^\sharp \mathbf{n} & \overset{\text{def}}{\iff} & \forall i,j: m_{ij} = n_{ij} \\
\left[\mathbf{m} \cap^\sharp \mathbf{n}\right]_{ij} & \overset{\text{def}}{=} & \min(m_{ij}, n_{ij}) \\
\left[\mathbf{m} \cup^\sharp \mathbf{n}\right]_{ij} & \overset{\text{def}}{=} & \max(m_{ij}, n_{ij}) \\
\left[\top^\sharp\right]_{ij} & \overset{\text{def}}{=} & +\infty
\end{array}$$

$(\mathcal{D}^\sharp, \subseteq^\sharp, \cup^\sharp, \cap^\sharp, \bot^\sharp, \top^\sharp)$ is a lattice

Remarks:

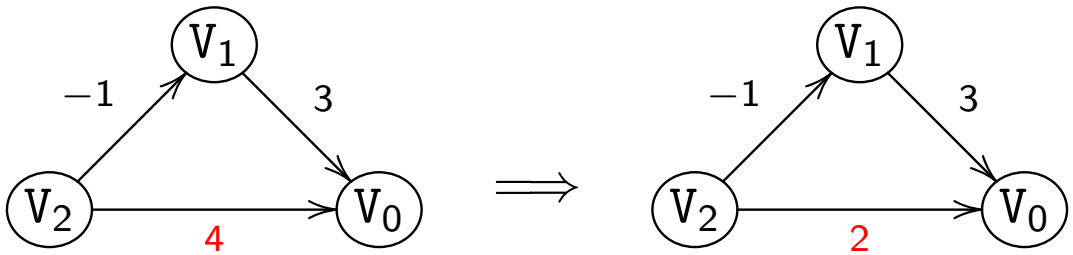
- $\mathcal{D}^\sharp$ is complete if $\leq$ is ($\mathbb{I} = \mathbb{R}$ or $\mathbb{Z}$, but not $\mathbb{Q}$)
- $\mathbf{m} \subseteq^\sharp \mathbf{n} \Longrightarrow \gamma_0(\mathbf{m}) \subseteq \gamma_0(\mathbf{n})$, but not the converse
- $\mathbf{m} =^\sharp \mathbf{n} \Longrightarrow \gamma_0(\mathbf{m}) = \gamma_0(\mathbf{n})$, but not the converse

Normal form, equality and inclusion testing

Issue: how can we compare $\gamma_0(\mathbf{m})$ and $\gamma_0(\mathbf{n})$?

Idea: find a normal form by propagating/tightening constraints

$$\begin{cases} V_0 - V_1 \leq 3 \\ V_1 - V_2 \leq -1 \\ V_0 - V_2 \leq 4 \end{cases} \qquad \begin{cases} V_0 - V_1 \leq 3 \\ V_1 - V_2 \leq -1 \\ V_0 - V_2 \leq 2 \end{cases}$$



Definition: shortest-path closure $\mathbf{m}^*$

$$m^*_{ij} \stackrel{\text{def}}{=} \min_{\substack{N \\ \langle i = i_1, \ldots, i_N = j \rangle}} \sum_{k=1}^{N-1} m_{i_k\, i_{k+1}}$$

Exists only when $\mathbf{m}$ has no cycle with strictly negative weight

Floyd–Warshall algorithm

Properties:

- $\gamma_0(\mathbf{m}) = \emptyset \iff \mathcal{G}$ has a cycle with strictly negative weight
- if $\gamma_0(\mathbf{m}) \neq \emptyset$, the shortest-path graph $\mathbf{m}^*$ is a normal form:
  $$\mathbf{m}^* = \min_{\subseteq^\sharp} \{ \mathbf{n} \mid \gamma_0(\mathbf{m}) = \gamma_0(\mathbf{n}) \}$$
- If $\gamma_0(\mathbf{m}), \gamma_0(\mathbf{n}) \neq \emptyset$, then
  - $\gamma_0(\mathbf{m}) = \gamma_0(\mathbf{n}) \iff \mathbf{m}^* =^\sharp \mathbf{n}^*$
  - $\gamma_0(\mathbf{m}) \subseteq \gamma_0(\mathbf{n}) \iff \mathbf{m}^* \subseteq^\sharp \mathbf{n}$

Floyd–Warshall algorithm

$$\begin{cases} m_{ij}^0 & \stackrel{\text{def}}{=} & m_{ij} \\ m_{ij}^{k+1} & \stackrel{\text{def}}{=} & \min(m_{ij}^k, m_{ik}^k + m_{kj}^k) \end{cases}$$

- If $\gamma_0(\mathbf{m}) \neq \emptyset$, then $\mathbf{m}^* = \mathbf{m}^{n+1}$ (normal form)
- $\gamma_0(\mathbf{m}) = \emptyset \iff \exists i: m_{ii}^{n+1} < 0$ (emptiness testing)
- $\mathbf{m}^{n+1}$ can be computed in $\mathcal{O}(n^3)$ time

Abstract operators

Abstract union $\cup^\sharp$

- $\gamma_0(\mathbf{m} \cup^\sharp \mathbf{n})$ may not be the smallest zone containing $\gamma_0(\mathbf{m})$ and $\gamma_0(\mathbf{n})$
- however, $(\mathbf{m}^*) \cup^\sharp (\mathbf{n}^*)$ is optimal:
  $$(\mathbf{m}^*) \cup^\sharp (\mathbf{n}^*) = \min_{\subseteq^\sharp} \{ \mathbf{o} \mid \gamma_0(\mathbf{o}) \supseteq \gamma_0(\mathbf{m}) \cup \gamma_0(\mathbf{n}) \}$$
  which implies
  $$\gamma_0((\mathbf{m}^*) \cup^\sharp (\mathbf{n}^*)) = \min_{\subseteq} \{ \gamma_0(\mathbf{o}) \mid \gamma_0(\mathbf{o}) \supseteq \gamma_0(\mathbf{m}) \cup \gamma_0(\mathbf{n}) \}$$
- $(\mathbf{m}^*) \cup^\sharp (\mathbf{n}^*)$ is always closed

Abstract intersection $\cap^\sharp$

- $\cap^\sharp$ is always exact: $\gamma_0(\mathbf{m} \cap^\sharp \mathbf{n}) = \gamma_0(\mathbf{m}) \cap \gamma_0(\mathbf{n})$
- $(\mathbf{m}^*) \cap^\sharp (\mathbf{n}^*)$ may not be closed

Remark:

The set of closed matrices with $\bot^\sharp$, and the operations $\subseteq^\sharp$, $\cup^\sharp$, $\lambda \mathbf{m}, \mathbf{n}.(\mathbf{m} \cap^\sharp \mathbf{n})^*$ define a sub-lattice

γ_0 is injective in this sub-lattice

Abstract operators (cont.)

We can define:

$$\left[\mathrm{C}^\sharp[\![\mathtt{V}_{j_0} - \mathtt{V}_{i_0} \leq c]\!]\,\mathbf{m}\right]_{ij} \stackrel{\text{def}}{=} \begin{cases} \min(m_{ij}, c) & \text{if } (i,j) = (i_0, j_0), \\ m_{ij} & \text{otherwise.} \end{cases}$$

$$\left[\mathrm{C}^\sharp[\![\mathtt{V}_{j_0} :=]-\infty, +\infty[ ]\!]\,\mathbf{m}\right]_{ij} \stackrel{\text{def}}{=} \begin{cases} +\infty & \text{if } i = j_0 \text{ or } j = j_0, \\ m^*_{ij} & \text{otherwise.} \end{cases}$$

(not optimal on non-closed arguments)

$$\mathrm{C}^\sharp[\![\mathtt{V}_{j_0} := \mathtt{V}_{i_0} + [a,b]]\!]\,\mathbf{m} \stackrel{\text{def}}{=} (\mathrm{C}^\sharp[\![\mathtt{V}_{j_0} - \mathtt{V}_{i_0} = [a,b]]\!] \circ \mathrm{C}^\sharp[\![\mathtt{V}_{j_0} :=]-\infty, +\infty[ ]\!])\mathbf{m} \quad \text{if } i_0 \neq j_0$$

$$\left[\mathrm{C}^\sharp[\![\mathtt{V}_{j_0} := \mathtt{V}_{j_0} + [a,b]]\!]\,\mathbf{m}\right]_{ij} \stackrel{\text{def}}{=} \begin{cases} m_{ij} - a & \text{if } i = j_0 \text{ and } j \neq j_0 \\ m_{ij} + b & \text{if } i \neq j_0 \text{ and } j = j_0 \\ m_{ij} & \text{otherwise.} \end{cases}$$

($i_0 \neq j_0$; $\mathtt{V}_{i_0}$ can be replaced with 0 by setting $i_0 = 0$)

These transfer functions are exact

Abstract operators (cont.)

Issue: given an arbitrary linear assignment $\mathtt{V}_{j_0} := a_0 + \sum_k a_k \times \mathtt{V}_k$

- there is no exact abstraction, in general
- the best abstraction $\alpha \circ \mathrm{C}[\![c]\!] \circ \gamma$ is costly to compute (e.g. convert to a polyhedron and back, with exponential cost)

Possible solution:

Given a (more general) assignment $e = [a_0, b_0] + \sum_k [a_k, b_k] \times \mathtt{V}_k$
we define an approximate operator as follows:

$$\left[\mathrm{C}^\sharp[\![\mathtt{V}_{j_0} := e]\!]\,\mathbf{m}\right]_{ij} \stackrel{\text{def}}{=} \begin{cases} \max(\mathrm{E}^\sharp[\![e]\!]\,\mathbf{m}) & \text{if } i = 0 \text{ and } j = j_0 \\ -\min(\mathrm{E}^\sharp[\![e]\!]\,\mathbf{m}) & \text{if } i = j_0 \text{ and } j = 0 \\ \max(\mathrm{E}^\sharp[\![e - \mathtt{V}_i]\!]\,\mathbf{m}) & \text{if } i \neq 0, j_0 \text{ and } j = j_0 \\ -\min(\mathrm{E}^\sharp[\![e + \mathtt{V}_j]\!]\,\mathbf{m}) & \text{if } i = j_0 \text{ and } j \neq 0, j_0 \\ m_{ij} & \text{otherwise} \end{cases}$$

where $\mathrm{E}^\sharp[\![e]\!]\,\mathbf{m}$ evaluates e using interval arithmetics with $\mathtt{V}_k \in [-m^*_{k0}, m^*_{0k}]$

Quadratic total cost (plus the cost of closure)

Abstract operators (cont.)

Example:

Argument

$$\begin{cases} 0 \leq \mathtt{Y} \leq 10 \\ 0 \leq \mathtt{Z} \leq 10 \\ 0 \leq \mathtt{Y} - \mathtt{Z} \leq 10 \end{cases}$$

$$\Downarrow \ \mathtt{X} := \mathtt{Y} - \mathtt{Z}$$

$$\begin{cases} -10 \leq \mathtt{X} \leq 10 \\ -20 \leq \mathtt{X} - \mathtt{Y} \leq 10 \\ -20 \leq \mathtt{X} - \mathtt{Z} \leq 10 \end{cases}$$

Intervals

$$\begin{cases} -10 \leq \mathtt{X} \leq 10 \\ -10 \leq \mathtt{X} - \mathtt{Y} \leq 0 \\ -10 \leq \mathtt{X} - \mathtt{Z} \leq 10 \end{cases}$$

Approximate solution

$$\begin{cases} 0 \leq \mathtt{X} \leq 10 \\ -10 \leq \mathtt{X} - \mathtt{Y} \leq 0 \\ -10 \leq \mathtt{X} - \mathtt{Z} \leq 10 \end{cases}$$

Best (polyhedra)

We have a good trade-off between cost and precision

The same idea can be used for tests and backward assignments

Widening and narrowing

The zone domain has both strictly increasing and decreasing infinite chains

Widening ∇

$$[\mathbf{m} \nabla \mathbf{n}]_{ij} \stackrel{\text{def}}{=} \begin{cases} m_{ij} & \text{if } n_{ij} \leq m_{ij} \\ +\infty & \text{otherwise} \end{cases}$$

Unstable constraints are deleted

Narrowing Δ

$$[\mathbf{m} \Delta \mathbf{n}]_{ij} \stackrel{\text{def}}{=} \begin{cases} n_{ij} & \text{if } m_{ij} = +\infty \\ m_{ij} & \text{otherwise} \end{cases}$$

Only $+\infty$ bounds are refined

Remarks:

- We can construct widenings with thresholds
- ∇ (resp. Δ) can be seen as a point-wise extension of an interval widening (resp. narrowing)

Other weakly relational domains

Based on closure:

- **Octagons**: $\pm X \pm Y \leq c$
  extension of zones by symmetry x / $-x$
  uses a slight adaptation of Floyd-Warshall
- **Two-variables per inequality**: $\alpha x + \beta y \leq c$
  (closure algorithm by Nelson)
- **Octahedra**: $\sum \alpha_i V_i \leq c$, $\alpha_i \in \{-1, 0, 1\}$
  (incomplete propagation, to avoid exponential cost)
- **Pentagons**: $X - Y \leq 0$
  (incomplete propagation, aims at linear cost)

Based on linear programming:

- **Template polyhedra**: $\mathbf{M} \times \vec{V} \geq \vec{C}$ for a fixed $\mathbf{M}$

Analysis example with octagons

Rate limiter

```
Y:=0; while • 1=1 do
  X:=[-128,128]; D:=[0,16];
  S:=Y; Y:=X; R:=X-S;
  if R<=-D then Y:=S-D fi;
  if R>=D then Y:=S+D fi
done
```

X:	input signal
Y:	output signal
S:	last output
R:	delta Y-S
D:	max. allowed for |R|

Analysis using:

- the octagon domain
- an abstract operator for $\mathtt{V}_{j_0} := [a_0, b_0] + \sum_k [a_k, b_k] \times \mathtt{V}_k$ similar to the one we defined on zones
- a widening with thresholds T

Result: we prove that $|\mathtt{Y}|$ is bounded by: $\min \{ t \in T \mid t \geq 144 \}$

Note: the polyhedron domain would find $|\mathtt{Y}| \leq 128$ and does not require thresholds, but it is more costly

Floating-point (and non-linearity)

Floating-point uses

Two independent problems:

- **Implement the analyzer using floating-point**

  goal: trade precision for efficiency

  exact rational arithmetics can be costly
  coefficients can grow large (polyhedra)

- **Analyze floating-point programs**

  goal: catch run-time errors caused by rounding
  (overflow, division by 0, . . .)

Also: a floating-point analyzer for floating-point programs

Challenge: how to stay **sound**?

Floating-point semantics

Floating-point numbers

Real computers do not know about $\mathbb{Q}$ and $\mathbb{R}$
They use limited-precision floating-point numbers $\mathbb{F}$

IEEE 754-1985 standard is the most widespread format
(supported by most processors and programming languages)

IEEE Binary representation: a number is a triple $\langle s, e, f \rangle$

- a 1-bit sign s
- a e-bit exponent e, with a bias (e represents $e - \text{bias}$)
- a p-bit fraction $f = .b_1 \dots b_p$ (f represents $\sum_i 2^{-i} b_i$)

IEEE format examples given by the choice of e, bias, p:

32-bit single precision *float*: $\begin{cases} \text{e} = 8 \\ \text{bias} = 127 \\ \text{p} = 23 \end{cases}$

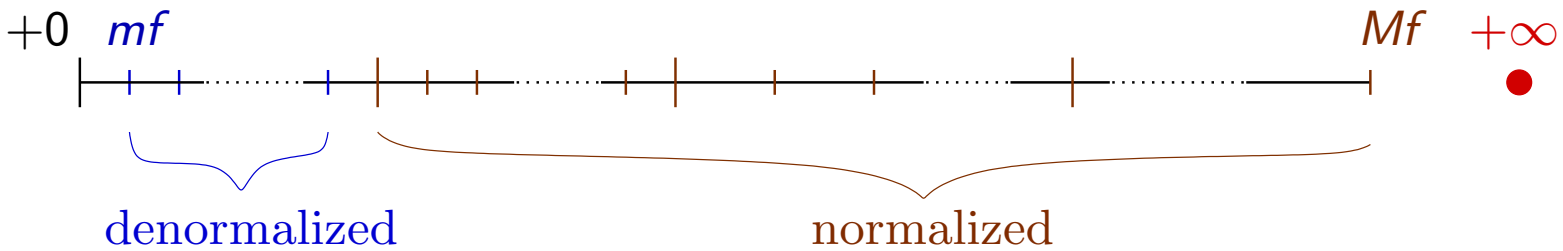
Other widespread formats:
64-bit *double*, 80-bit *double extended*, 128-bit *quad*

Floating-point representation

Semantics $\langle s, e, f \rangle$ represents either:

- a normalized number: $(-1)^s \times 2^{e-\text{bias}} \times 1.f$ (if $1 \leq e \leq 2^{\text{e}} - 2$)
- a denormalized number: $(-1)^s \times 2^{1-\text{bias}} \times 0.f$ (if $e = 0$, $f \neq 0$)
- $+0$ or -0 (if $e = 0$, $f = 0$)
- $+\infty$ or $-\infty$ (if $e = 2^{\text{e}} - 1$, $f = 0$)
- an error code *NaN* (if $e = 2^{\text{e}} - 1$, $f \neq 0$)

Visual representation (positive part)



$mf \stackrel{\text{def}}{=} 2^{1-\text{bias}-\text{p}}$ smallest positive

$Mf \stackrel{\text{def}}{=} (2 - 2^{-\text{p}}) \times 2^{2^{\text{e}}-\text{bias}-2}$ largest non-∞

Floating-point computations

The set of floating-point numbers is not closed under $+$, $-$, $\times$, $/$:

- every result is rounded to a representable float
- an overflow or division by 0 generates $+\infty$ or $-\infty$ (overflow)
- small numbers are truncated to $+0$ or -0 (underflow)
- some operations are invalid ($0/0$, $(+\infty)+(-\infty)$, etc.) and return *NaN*

Simplified semantics:

- overflows and NaNs halt the program with an error Ω
- rounding and underflow are not errors
- we do not distinguish between $+0$ and -0

$\Longrightarrow$ variable values live in a finite subset $\mathbb{F}$ of $\mathbb{R}$, expression values live in $\mathbb{F} \cup \{ \Omega \}$

Floating-point expressions

Floating-point expressions $\mathtt{exp}_f$

$$
\begin{array}{lcll}
\mathtt{exp}_f & ::= & [c, c'] & \text{constant range } c, c' \in \mathbb{F},\ c \leq c' \\
& | & \mathtt{V} & \text{variable } \mathtt{V} \in \mathbb{V} \\
& | & \ominus\ \mathtt{exp}_f & \text{negation} \\
& | & \mathtt{exp}_f \odot_r \mathtt{exp}_f & \text{operator } \odot \in \{\oplus, \ominus, \otimes, \oslash\}
\end{array}
$$

(we use circled operators to distinguish them from operators in $\mathbb{Q}$)

Concrete semantics of expressions

Semantics of rounding: $R_r \colon \mathbb{Q} \to \mathbb{F} \cup \{\Omega\}$.

4 rounding modes r: towards $+\infty$, $-\infty$, 0, or to-nearest n

Example definition:

$$R_{+\infty}(x) \stackrel{\text{def}}{=} \begin{cases} \min \{ y \in \mathbb{F} \mid y \geq x \} & \text{if } x \leq Mf \\ \Omega & \text{if } x > Mf \end{cases}$$

$$R_{-\infty}(x) \stackrel{\text{def}}{=} \begin{cases} \max \{ y \in \mathbb{F} \mid y \leq x \} & \text{if } x \geq -Mf \\ \Omega & \text{if } x < -Mf \end{cases}$$

Notes:

- $\forall x, r,\ R_{-\infty}(x) \leq R_r(x) \leq R_{+\infty}(x)$
- $\forall r, R_r$ is monotonic

Concrete semantics of expressions (cont.)

$\mathsf{E}[\![e_f]\!] : (\mathbb{V} \to \mathbb{F}) \to \mathcal{P}(\mathbb{F} \cup \{\Omega\})$ (expression semantics)

Each operator is evaluated in $\mathbb{R}$ and then rounded using R_r.

$$
\begin{aligned}
\mathsf{E}[\![\mathtt{V}]\!]\, \rho &\stackrel{\text{def}}{=} \{\, \rho(\mathtt{V}) \,\} \\
\mathsf{E}[\![[c, c']]\!]\, \rho &\stackrel{\text{def}}{=} \{\, x \in \mathbb{F} \mid c \leq x \leq c' \,\} \\
\mathsf{E}[\![\ominus e_f]\!]\, \rho &\stackrel{\text{def}}{=} \{\, -x \mid x \in \mathsf{E}[\![e_f]\!]\, \rho \cap \mathbb{F} \,\} \cup (\{\Omega\} \cap \mathsf{E}[\![e_f]\!]\, \rho) \\
\mathsf{E}[\![e_f \odot_r e'_f]\!]\, \rho &\stackrel{\text{def}}{=} \\
&\quad \{\, R_r(x \cdot y) \mid x \in \mathsf{E}[\![e_f]\!]\, \rho \cap \mathbb{F},\ y \in \mathsf{E}[\![e'_f]\!]\, \rho \cap \mathbb{F} \,\} \cup \\
&\quad \{\, \Omega \mid \text{ if } \Omega \in \mathsf{E}[\![e_f]\!]\, \rho \cup \mathsf{E}[\![e'_f]\!]\, \rho \,\} \\
&\quad \{\, \Omega \mid \text{ if } 0 \in \mathsf{E}[\![e'_f]\!]\, \rho \text{ and } \odot = \oslash \,\}
\end{aligned}
$$

$\mathsf{C}[\![c]\!] : \mathcal{P}(\mathbb{V} \to \mathbb{F}) \to \mathcal{P}((\mathbb{V} \to \mathbb{F}) \cup \{\Omega\})$ (command semantics)

$$
\begin{aligned}
\mathsf{C}[\![\mathtt{X} := e_f]\!]\, \mathcal{X} &\stackrel{\text{def}}{=} \{\, \rho[\, \mathtt{X} \mapsto v \,] \mid \rho \in \mathcal{X},\ v \in \mathsf{E}[\![e_f]\!]\, \rho \cap \mathbb{F} \,\} \\
&\quad \cup (\{\Omega\} \cap \mathsf{E}[\![e_f]\!]\, \mathcal{X}) \\
\mathsf{C}[\![e_f \leq 0]\!]\, \mathcal{X} &\stackrel{\text{def}}{=} \{\, \rho \mid \rho \in \mathcal{X},\ \exists v \in \mathsf{E}[\![e_f]\!]\, \rho \cap \mathbb{F},\ v \leq 0 \,\} \\
&\quad \cup (\{\Omega\} \cap \mathsf{E}[\![e_f]\!]\, \mathcal{X})
\end{aligned}
$$

Floating-point interval domain

Representation: $\mathcal{B}^\sharp \stackrel{\text{def}}{=} \{ [a,b] \mid a \in \mathbb{F},\ b \in \mathbb{F},\ a \leq b \} \cup \{ \bot^\sharp \}$

Expression semantics: $\mathrm{E}^\sharp[\![\exp_f]\!] : (\mathbb{V} \to \mathcal{B}^\sharp) \to (\mathcal{B}^\sharp \cup \{ \Omega \})$

$$
\begin{array}{lcl}
[a,b] \oplus^\sharp [a',b'] & \stackrel{\text{def}}{=} & [R_{-\infty}(a+a'), R_{+\infty}(b+b')] \\
[a,b] \ominus^\sharp [a',b'] & \stackrel{\text{def}}{=} & [R_{-\infty}(a-b'), R_{+\infty}(b-a')] \\
[a,b] \otimes^\sharp [a',b'] & \stackrel{\text{def}}{=} & [\ R_{-\infty}(\min(aa', ab', ba', bb')), \\
 & & \quad R_{+\infty}(\max(aa', ab', ba', bb'))]
\end{array}
$$

- We suppose r is unknown and assume a worst case rounding.
- Soundness stems from the monotonicity of $R_{-\infty}$ and $R_{+\infty}$.
- Abstract operators also use float arithmetic (efficiency).

Error management

If some bound in $\mathrm{E}^\sharp[\![\exp_f]\!]$ evaluates to Ω, we

- report the error to the user, and
- continue the evaluation with $\top^\sharp$.

Floating-point analysis example

filter with reinitialisation

```
Z:=0;
while 1=1 do
  if [0,1]=1 then Z:=[-10,10] fi;
  Z:=(0.3 ⊗ Z) ⊕ [-10,10]
done
```

In $\mathbb{R}$, we would have $|Z| < 10/0.7$

Using floats, $|Z|$ is bounded by $B = R_{+\infty}(10/0.7)$

Interval analysis:

A widening with thresholds finds that $|Z| \leq \min \{ x \in T \mid x \geq B \}$

The absence of overflow is proved if T has a value larger than B

Expression linearization

Floating-point issues in relational domains

Relational domains assume many powerful properties on $\mathbb{Q}$: associativity, distributivity,... that are not true on $\mathbb{F}$!

Example: Fourier-Motzkin elimination

$$\mathtt{X} - \mathtt{Y} \leq c \quad \wedge \quad \mathtt{Y} - \mathtt{Z} \leq d \quad \Longrightarrow \quad \mathtt{X} - \mathtt{Z} \leq c + d$$

$$\mathtt{X} \ominus_n \mathtt{Y} \leq c \quad \wedge \quad \mathtt{Y} \ominus_n \mathtt{Z} \leq d \quad \not\Longrightarrow \quad \mathtt{X} \ominus_n \mathtt{Z} \leq c \oplus_n d$$

$$(\mathtt{X} = 1,\ \mathtt{Y} = 10^{38},\ \mathtt{Z} = -1,\ c = \mathtt{X} \ominus_n \mathtt{Y} = -10^{38},$$
$$d = \mathtt{Y} \ominus_n \mathtt{Z} = 10^{38},\ c \oplus_n d = 0,\ \mathtt{X} \ominus_n \mathtt{Z} = 2 > 0)$$

We cannot manipulate float expressions as easily as rational ones!

Solution:

keep representing and manipulating rational expressions

- abstract float expressions from programs into rational ones
- feed them to a rational abstract domain
- (optional) implement the rational domain using floats

Abstraction framework

Issue:

Float expressions are a special case of non-linear expressions (in $\mathbb{Q}$).
Most relational domains can only deal with linear expressions
How can we abstract non-linear assignments such as $\mathtt{X} := \mathtt{Y} \times \mathtt{Z}$?

Idea: replace $\mathtt{Y} \times \mathtt{Z}$ with a sound linear approximation

Framework:

We define an approximation preorder $\preceq$ on expressions:

$$R \models e_1 \preceq e_2 \iff^{\text{def}} \forall \rho \in R: \mathsf{E}[\![e_1]\!] \rho \subseteq \mathsf{E}[\![e_2]\!] \rho$$

Soundness properties if $\gamma(\mathcal{X}^\sharp) \models e \preceq e'$ then:

- $\mathsf{C}[\![\mathtt{V} := e]\!] \gamma(\mathcal{X}^\sharp) \subseteq \gamma(\mathsf{C}^\sharp[\![\mathtt{V} := e']\!] \mathcal{X}^\sharp)$
- $\mathsf{C}[\![e \bowtie 0]\!] \gamma(\mathcal{X}^\sharp) \subseteq \gamma(\mathsf{C}^\sharp[\![e' \bowtie 0]\!] \mathcal{X}^\sharp)$

$\Longrightarrow$ we can now use e' in the abstract instead of e

Affine interval forms

We put expressions in affine interval form: [Miné 04]

$$\mathtt{exp}_\ell ::= [a_0, b_0] + \sum_k [a_k, b_k] \times \mathtt{V}_k$$

Semantics:

$$\mathtt{E}[\![e_\ell]\!] \rho \stackrel{\text{def}}{=} \{ c_0 + \sum_k c_k \times \rho(\mathtt{V}_k) \mid \forall i,\ c_i \in [a_i, b_i] \}$$

(evaluated in $\mathbb{Q}$)

Advantages:

- affine expressions are easy to manipulate
- interval coefficients allow non-determinism in expressions, hence, the opportunity for abstraction
- intervals can easily model rounding errors
- easy to design algorithms for $\mathtt{C}^\sharp[\![\mathtt{X} := e_\ell]\!]$ and $\mathtt{C}^\sharp[\![e_\ell \leq 0]\!]$ in most domains

Affine interval form algebra

Operations on affine interval forms:

- adding $\boxplus$ and subtracting $\boxminus$ two forms
- multiplying $\boxtimes$ and dividing $\boxslash$ a form by an interval

Using interval arithmetic $\oplus^\sharp$, $\ominus^\sharp$, $\otimes^\sharp$, $\oslash^\sharp$:

$$(i_0 + \sum_k i_k \times \mathtt{V}_k) \boxplus (i'_0 + \sum_k i'_k \times \mathtt{V}_k) \stackrel{\text{def}}{=} (i_0 \oplus^\sharp i'_0) + \sum_k (i_k \oplus^\sharp i'_k) \times \mathtt{V}_k$$

$$i \boxtimes (i_0 + \sum_k i_k \times \mathtt{V}_k) \stackrel{\text{def}}{=} (i \otimes^\sharp i_0) + \sum_k (i \otimes^\sharp i_k) \times \mathtt{V}_k$$

$\ldots$

Projection: $\pi_k : \mathcal{D}^\sharp \to \mathtt{exp}_\ell$

We suppose we are given an abstract interval projection operator π_k such that:

$$\pi_k(\mathcal{X}^\sharp) = [a, b] \text{ such that } [a, b] \supseteq \{ \rho(\mathtt{V}_k) \mid \rho \in \gamma(\mathcal{X}^\sharp) \}.$$

Linearization of rational expressions

Intervalization $\iota : (\mathtt{exp}_\ell \times \mathcal{D}^\sharp) \to \mathtt{exp}_\ell$

Flattens the expression into a single interval:

$$\iota(i_0 + \sum_k (i_k \times \mathtt{V_k}), \mathcal{X}^\sharp) \stackrel{\text{def}}{=} i_0 \ +^\sharp \ \sum^\sharp_{b,\,k} (i_k \times^\sharp \pi_k(\mathcal{X}^\sharp))$$

Linearization $\ell : (\mathtt{exp} \times \mathcal{D}^\sharp) \to \mathtt{exp}_\ell$

Defined by induction on the syntax of expressions:

- $\ell(\mathtt{V}, \mathcal{X}^\sharp) \stackrel{\text{def}}{=} [1,1] \times \mathtt{V}$
- $\ell([a,b], \mathcal{X}^\sharp) \stackrel{\text{def}}{=} [a,b]$
- $\ell(e_1 + e_2, \mathcal{X}^\sharp) \stackrel{\text{def}}{=} \ell(e_1, \mathcal{X}^\sharp) \boxplus \ell(e_2, \mathcal{X}^\sharp)$
- $\ell(e_1 - e_2, \mathcal{X}^\sharp) \stackrel{\text{def}}{=} \ell(e_1, \mathcal{X}^\sharp) \boxminus \ell(e_2, \mathcal{X}^\sharp)$
- $\ell(e_1 / e_2, \mathcal{X}^\sharp) \stackrel{\text{def}}{=} \ell(e_1, \mathcal{X}^\sharp) \boxslash \iota(\ell(e_2, \mathcal{X}^\sharp), \mathcal{X}^\sharp)$
- $\ell(e_1 \times e_2, \mathcal{X}^\sharp) \stackrel{\text{def}}{=}$ can be $\begin{cases} \text{either} & \iota(\ell(e_1, \mathcal{X}^\sharp), \mathcal{X}^\sharp) \boxtimes \ell(e_2, \mathcal{X}^\sharp), \\ \text{or} & \iota(\ell(e_2, \mathcal{X}^\sharp), \mathcal{X}^\sharp) \boxtimes \ell(e_1, X^\sharp) \end{cases}$

Linearization application

Property soundness of the linearization:

For any abstract domain $\mathcal{D}^\sharp$, any $\mathcal{X}^\sharp \in \mathcal{D}^\sharp$ and $e \in \mathtt{exp}$, we have:

$$\gamma(\mathcal{X}^\sharp) \models e \preceq \ell(e, \mathcal{X}^\sharp)$$

Remarks:

- ℓ results in a loss of precision
- ℓ is not monotonic for $\preceq$
  (e.g., $\ell(\mathtt{V}/\mathtt{V}, \mathtt{V} \mapsto [1, +\infty]) = [0,1] \times \mathtt{V} \not\preceq 1$)

Application to the octagon domain

```
Y:=[0,+∞];
T:=[-1,1];
X:=T×Y
```

- $\mathtt{T} \times \mathtt{Y}$ is linearized as $[-1, 1] \times \mathtt{Y}$
- we can prove that $|\mathtt{X}| \leq \mathtt{Y}$

Linearization application (cont.)

Application to the interval domain

$\mathrm{C}^\sharp[\![\, \mathtt{V} := \ell(e, \mathcal{X}^\sharp) \,]\!]\, \mathcal{X}^\sharp$ is always more precise than $\mathrm{C}^\sharp[\![\, \mathtt{V} := e \,]\!]\, \mathcal{X}^\sharp$

ℓ simplifies symbolically variables occurring several times

Example: $\mathtt{X} := 2 \times \mathtt{V} - \mathtt{V}$, where $\mathtt{V} \in [a, b]$:

- using vanilla intervals:

  $$\mathrm{E}^\sharp[\![\, 2 \times \mathtt{V} - \mathtt{V} \,]\!]\, (\mathcal{X}^\sharp) = 2 \times^\sharp [a, b] -^\sharp [a, b] = [2a - b, 2b - a]$$

- after linearization $\ell(2 \times \mathtt{V} - \mathtt{V}, \mathcal{X}^\sharp) = \mathtt{V}$, so

  $$\mathrm{E}^\sharp[\![\, \ell(2 \times \mathtt{V} - \mathtt{V}, \mathcal{X}^\sharp) \,]\!]\, \mathcal{X}^\sharp = [a, b]$$

  strictly more precise than $[2a - b, 2b - a]$ when $a \neq b$

Linearization of floating-point expressions

Rounding an affine interval form: (32-bit single precision)

- if the result is normalized: we have a relative error ε with magnitude 2^{-23}:

$$\varepsilon([a_0, b_0] + \sum_k [a_k, b_k] \times \mathtt{V}_k) \stackrel{\text{def}}{=} \max(|a_0|, |b_0|) \times [-2^{-23}, 2^{-23}] + \sum_k (\max(|a_k|, |b_k|) \times [-2^{-23}, 2^{-23}] \times \mathtt{V}_k)$$

- if the result is denormalized, we have an absolute error $\omega \stackrel{\text{def}}{=} [-2^{-149}, 2^{-149}]$.

$\Longrightarrow$ we sum these two sources of rounding errors

Linearization with rounding errors: $\ell : (\exp_f \times \mathcal{D}^\sharp) \to \exp_\ell$

$$\ell(e_1 \oplus_r e_2, \mathcal{X}^\sharp) \stackrel{\text{def}}{=} \ell(e_1, \mathcal{X}^\sharp) \boxplus \ell(e_2, \mathcal{X}^\sharp) \boxplus \varepsilon(\ell(e_1, \mathcal{X}^\sharp)) \boxplus \varepsilon(\ell(e_2, \mathcal{X}^\sharp)) \boxplus \omega$$

$$\ell(e_1 \otimes_r e_2, \mathcal{X}^\sharp) \stackrel{\text{def}}{=} \iota(\ell(e_1, \mathcal{X}^\sharp), \mathcal{X}^\sharp) \boxtimes (\ell(e_2, \mathcal{X}^\sharp) \boxplus \varepsilon(\ell(e_2, \mathcal{X}^\sharp))) \boxplus \omega$$

...

Soundness of the floating-point linearization

Soundness of the linearization

$\forall e: \forall \mathcal{X}^\sharp \in \mathcal{D}^\sharp: \forall \rho \in \gamma(\mathcal{X}^\sharp)$:
if $\Omega \notin \mathsf{E}[\![e]\!] \rho$, then $\mathsf{E}[\![e]\!] \rho \subseteq \mathsf{E}[\![\ell(e, \mathcal{X}^\sharp)]\!] \rho$

Application: $\mathsf{C}^\sharp[\![\mathtt{V} := e]\!] \mathcal{X}^\sharp$

- check that $\Omega \notin \mathsf{E}[\![e]\!] \rho$ for $\rho \in \gamma(\mathcal{X}^\sharp)$ with interval arithmetic
- compute $\mathsf{C}^\sharp[\![\mathtt{V} := e]\!] \mathcal{X}^\sharp$ as $\mathsf{C}^\sharp[\![\mathtt{V} := \ell(e, \mathcal{X}^\sharp)]\!] \mathcal{X}^\sharp$
- (use $\mathsf{C}^\sharp[\![\mathtt{V} := [-Mf, Mf]]\!] \mathcal{X}^\sharp$ if $\Omega \in \mathsf{E}[\![e]\!] \rho$)

Example applications

- Improving the interval domain using symbolic simplification

  Example:

  $\mathtt{Z} := \mathtt{X} \ominus (0.25 \otimes \mathtt{X})$ is linearized into
  $\mathtt{Z} := ([0.749\cdots, 0.750\cdots] \times \mathtt{X}) + 2.35\cdots 10^{-38} \times [-1, 1]$

  If $\mathtt{X} \in [-1, 1]$, we find $|\mathtt{Z}| \leq 0.750\cdots$
  (instead of $|\mathtt{Z}| \leq 1.25\cdots$)

- Allows using relational domains (octagons, etc.)

  Example: floating-point version of the rate limiter
  (single precision)

  The bound of the output $|\mathtt{Y}|$ is the smallest threshold larger than 144.00005 (instead of 144)

Floating-point implementation

Goal: implement abstract domains using floating-point numbers

- more efficient (especially to analyse floating-point programs)
- rounding errors in the algorithms may cause unsoundness!

Simple solution:

round upper-bounds toward $+\infty$, lower bounds toward $-\infty$

Works for:

- intervals ($\oplus^\sharp, \ominus^\sharp, \otimes^\sharp, \ldots$)
- linearization into $\texttt{exp}_\ell$ (based on interval computations)
- zones, octagons (replace $a + b$ with $R_{+\infty}(a + b)$)
- not polyhedra

Sound floating-point polyhedra

Sound floating-point polyhedra

Algorithms to adapt: [Chen 08]

Design sound approximate floating-point algorithms *simplex*$_f$ and *Fourier*$_f$

- linear programming:

$$\mathit{simplex}_f(\mathcal{X}^\sharp, \vec{\alpha}) \leq \mathit{simplex}(\mathcal{X}^\sharp, \vec{\alpha})$$

$$\mathit{simplex}(\mathcal{X}^\sharp, \vec{\alpha}) \stackrel{\text{def}}{=} \min \{ \textstyle\sum_k \alpha_k \rho(\mathtt{V}_k) \mid \rho \in \gamma(\mathcal{X}^\sharp) \}$$

- Fourier-Motzkin elimination:

$$\mathit{Fourier}_f(\mathcal{X}^\sharp, \mathtt{V}_k) \Longleftarrow \mathit{Fourier}(\mathcal{X}^\sharp, \mathtt{V}_k)$$

$$\begin{array}{l}\mathit{Fourier}(\mathcal{X}^\sharp, \mathtt{V}_k) \stackrel{\text{def}}{=} \\ \quad \{ (\sum_i \alpha_i \mathtt{V}_i \geq \beta) \in \mathcal{X}^\sharp \mid \alpha_k = 0 \} \cup \\ \quad \{ (-\alpha_k^-) c^+ + \alpha_k^+ c^- \mid \begin{array}{l} c^+ = (\sum_i \alpha_i^+ \mathtt{V}_i \geq \beta^+) \in \mathcal{X}^\sharp,\ \alpha_k^+ > 0, \\ c^- = (\sum_i \alpha_i^- \mathtt{V}_i \geq \beta^-) \in \mathcal{X}^\sharp,\ \alpha_k^- < 0 \end{array} \}\end{array}$$

Sound floating-point linear programming

Guaranteed linear programming: [Neumaier 04]

Goal: under-approximate $\mu = \min \{ \vec{c} \cdot \vec{x} \mid \mathbf{M} \times \vec{x} \leq \vec{b} \}$

knowing that $\vec{x} \in [\vec{x_l}, \vec{x_h}]$ (bounding-box for $\gamma(\mathcal{X}^\sharp)$)

- compute any approximation $\tilde{\mu}$ of the dual problem:
  $\tilde{\mu} \simeq \mu = \max \{ \vec{b} \cdot \vec{y} \mid {}^t\mathbf{M} \times \vec{y} = \vec{c}, \vec{y} \leq \vec{0} \}$
  and the corresponding vector $\vec{y}$

  (e.g. using an off-the-shelf solver; $\tilde{\mu}$ may over-approximate or under-approximate μ)

- compute with intervals safe bounds $[\vec{r_l}, \vec{r_h}]$ for $\mathbf{A} \times \vec{y} - \vec{c}$:
  $[\vec{r_l}, \vec{r_h}] = ({}^t\mathbf{A} \otimes^\sharp \vec{y}) \ominus^\sharp \vec{c}$

  and then:
  $\nu = \inf((\vec{b} \otimes^\sharp \vec{y}) \ominus^\sharp ([\vec{r_l}, \vec{r_h}] \otimes^\sharp [\vec{x_l}, \vec{x_h}]))$

then: $\nu \leq \mu$

Sound floating-point Fourier-Motzkin elimination

Given:

- $c^+ = (\sum_i \alpha_i^+ \mathtt{V}_i \geq \beta^+)$ with $\alpha_k^+ > 0$
- $c^- = (\sum_i \alpha_i^- \mathtt{V}_i \geq \beta^-)$ with $\alpha_k^- < 0$
- a bounding-box of $\gamma(\mathcal{X}^\sharp)$: $[\vec{x}_l, \vec{x}_h]$

We wish to compute $\sum_{i \neq k} \alpha_i \mathtt{V}_i \geq \beta$ in $\mathbb{F}$
implied by $(-\alpha_k^-)c^+ + \alpha_k^+ c^-$ in $\gamma(\mathcal{X}^\sharp)$

- normalize c^+ and c^- using interval arithmetics:

$$\begin{cases} \mathtt{V}_k + \sum_{i \neq k} (\alpha_i^+ \oslash^\sharp \alpha_k^+)\mathtt{V}_i \geq \beta^+ \oslash^\sharp \alpha_k^+ \\ -\mathtt{V}_k + \sum_{i \neq k} (\alpha_i^- \oslash^\sharp (-\alpha_k^-))\mathtt{V}_i \geq \beta^- \oslash^\sharp (-\alpha_k^-) \end{cases}$$

  (interval affine forms)

- add them using interval arithmetics:

$$\sum_{i \neq k} [a_i, b_i]\mathtt{V}_i \geq [a_0, b_0]$$

  where $[a_i, b_i] = (\alpha_i^+ \oslash^\sharp \alpha_k^+) \ominus^\sharp (\alpha_i^- \oslash^\sharp \alpha_k^-)$,
  $[a_0, b_0] = (\beta^+ \oslash^\sharp \alpha_k^+) \ominus^\sharp (\beta^- \oslash^\sharp \alpha_k^-)$

Sound floating-point Fourier-Motzkin elimination (cont.)

- linearize the interval linear form into $\sum_{i \neq k} \alpha_i \mathtt{V}_i \geq \beta$ where

$$\begin{cases} \alpha_i \in [a_i, b_i] \\ \beta = \sup\left([a_0, b_0] \oplus^\sharp \bigoplus^\sharp_{b,\, i \neq k} (|\alpha_i \ominus^\sharp [a_i, b_i]|) \otimes^\sharp |[\vec{x}_l, \vec{x}_h]|\right) \end{cases}$$

Soundness:

For all choices of $\alpha_i \in [a_i, b_i]$,
$\sum_{i \neq k} \alpha_i \mathtt{V}_k \geq \beta$ holds in $\mathit{Fourier}(\mathcal{X}^\sharp, \mathtt{V}_k)$
(e.g. $\alpha_i = (a_i \oplus b_i) \oslash 2$)

Consequences of rounding

Precision loss:

- projection:

$$\begin{array}{rcl} \gamma(\mathit{Fourier}_f(\mathcal{X}^\sharp, \mathtt{V}_k)) & \supseteq & \{\, \rho[\mathtt{V}_k \mapsto v] \mid v \in \mathbb{Q},\ \rho \in \gamma(\mathcal{X}^\sharp) \,\} \\ & = & \mathsf{C}[\![\, \mathtt{V}_k := [-\infty, +\infty] \,]\!]\, \gamma(\mathcal{X}^\sharp) \end{array}$$

- order:

$$\mathcal{X}^\sharp \subseteq^\sharp \mathcal{Y}^\sharp \Longrightarrow \gamma(\mathcal{X}^\sharp) \subseteq \gamma(\mathcal{Y}^\sharp) \quad (\not\Leftarrow)$$

- join:

$$\gamma(\mathcal{X}^\sharp \cup^\sharp \mathcal{Y}^\sharp) \supseteq \mathit{ConvexHull}_f(\gamma(\mathcal{X}^\sharp) \cup \gamma(\mathcal{Y}^\sharp)) \quad (\neq)$$

Efficiency loss:

- cannot remove all redundant constraints

Floating-point polyhedra widening

Widening ∇:

$$\mathcal{X}^\sharp \nabla \mathcal{Y}^\sharp \stackrel{\text{def}}{=} \{ c \in \mathcal{X}^\sharp \mid \mathcal{Y}^\sharp \subseteq^\sharp \{ c \} \}$$

(drop $\{ c \in \mathcal{Y}^\sharp \mid \exists c' \in \mathcal{X}^\sharp : \mathcal{X}^\sharp =^\sharp (\mathcal{X}^\sharp \setminus c') \cup \{ c \} \}$
as $\mathcal{X}^\sharp$ and $\mathcal{Y}^\sharp$ may have redundant constraints)

Abstraction summary

Floating-point polyhedra analyzer for floating-point programs

expression abstraction		**environment abstraction**
float expression e_f		
$\downarrow$ linearization		$\mathcal{P}(\mathbb{V} \to \mathbb{F})$
affine form e_ℓ in $\mathbb{Q}$		$\downarrow$ abstract domain
$\downarrow$ float implementation		polyhedra in $\mathbb{Q}$
affine form e_ℓ in $\mathbb{F}$	$\longrightarrow$	$\downarrow$ float implementation
		polyhedra in $\mathbb{F}$
		$\downarrow$ widening
		polyhedra in $\mathbb{F}$

Bibliography

Bibliography

[Bagnara 02] **R. Bagnara, E. Ricci, E. Zaffanella & P. M. Hill**. *Possibly not closed convex polyhedra and the Parma Polyhedra Library.* In Proc. SAS'02, LNCS 2477, 213–229, Springer, 2002.

[Bagnara 03] **R. Bagnara, P. Hill, E. Ricci, E. Zaffanella**. *Precise widening operators for convex polyhedra.* In Proc. SAS'03, LNCS 2694, 337-354, Springer, 2003.

[Benoi 96] **F. Benoy & A. King**. *Inferring argument size relationships with CLP(R).* In In Proc. of LOPSTR'96, LNCS 1207, 204–223. Springer, 1996.

[Berdine 07] **J. Berdine, A. Chawdhary, B. Cook, D. Distefano & P. O'Hearn**. *Variance analyses from invariances analyses.* In Proc. POPL'07 211–224, ACM, 2007.

Bibliography (cont.)

[Bourdoncle 93] **F. Bourdoncle**. *Efficient chaotic iteration strategies with widenings.* In Proc. FMPA'93, LNCS 735, 128–141, Springer, 1993.

[Chen 08] **L. Chen, A. Miné & P. Cousot**. *A sound floating-point polyhedra abstract domain.* In Proc. APLAS'08, LNCS 5356, 3–18, Springer, 2008.

[Chernikova 68] **N. V. Chernikova**. *Algorithm for discovering the set of all the solutions of a linear programming problem.* In U.S.S.R. Comput. Math. and Math. Phys., 8(6):282–293, 1968.

[Cousot Cousot 76] **P. Cousot & R. Cousot**. *Static determination of dynamic properties of programs.* In Proc. ISP'76, Dunod, 1976.

[Cousot Halbwachs 78] **P. Cousot & N. Halbwachs**. *Automatic discovery of linear restraints among variables of a program.* In Proc. POPL'78, 84–96, ACM, 1978.

Bibliography (cont.)

[Cousot Cousot 79] **P. Cousot & R. Cousot**. *Constructive versions of Tarski's fixed point theorems.* In Pacific J. of Math., 82(1):43–57, 1979.

[Cousot Cousot 01] **P. Cousot & R. Cousot**. *Compositional separate modular static analysis of programs by abstract interpretation.* In Proc. SSGRR'01, 2001.

[Dor 01] **N. Dor, M. Rodeh & M. Sagiv**. *Cleanness checking of string manipulations in C programs via integer analysis.* In Proc. SAS'01, LNCS 2126, 194–212, Springer, 2001.

[Granger 89] **P. Granger**. *Static analysis of arithmetical congruences.* In JCM, 3(4–5):165–190, 1989.

[Granger 92] **P. Granger**. *Improving the results of static analyses of programs by local decreasing iterations.* In Proc. FSTTCSC'92, LNCS 652, 68–79, Springer, 1992.

Bibliography (cont.)

[Jeannet 09] **B. Jeannet & A. Miné**. *Apron: A library of numerical abstract domains for static analysis.* In Proc. CAV'09, LNCS 5643, 661–667, Springer, 2009, `http://apron.cri.ensmp.fr/library`.

[Le Verge 92] **H. Le Verge**. *A note on Chernikova's algorithm.* In Research Report 1662, INRIA Rocquencourt, 1992.

[Neumaier 04] **A. Neumaier & O. Shcherbina**. *Safe bounds in linear and mixed-integer linear programming.* In Math. Program., 99(2):283–296, 2004.

[Miné 01] **A. Miné**. *The octagon abstract domain.* In Proc. AST'01, 310–319, IEEE, 2001.

[Miné 04] **A. Miné**. *Relational abstract domains for the detection of floating-point run-time errors.* In Proc. ESOP'04, LNCS 2986, 3–17, Springer, 2004.

Bibliography (cont.)

[Moore 66] **R. E. Moore**. *Interval Analysis.* In Englewood Cliff, New Jersey: Prentice-Hall, 1966.

[Schrijver 86] **A. Schrijver**. *Theory of linear and integer programming.* In John Wiley & Sons, Inc., 1986.

[Tarski 55] **A. Tarski**. *A lattice theoretical fixpoint theorem and its applications.* In Pacific J. of Math., 5:285–310, 1955.