Relational
Numerical Abstract Domains

MPRI 2–6: Abstract Interpretation,
application to verification and static analysis

Antoine Miné

CNRS, École normale supérieure

course 3, 2012–2013

Outline

- The need for relational domains
- Presentation of a few relational numerical abstract domains
  - linear equality domains
  - polyhedra domain
  - weakly relational domains: zones, octagons
- Handling non-linear expressions
- Bibliography

Shortcomings of non-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|

Iterations in the interval domain (without widening):

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

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

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

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

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 $C^\sharp[\![\, \mathtt{I} >= 5000 \,]\!]$.

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

[Anco10], [Jean09]

Reminders

Syntax

Fixed finite set of variables $\mathbb{V}$,
with value in $\mathbb{I}$, $\mathbb{I} \in \{\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{M}, \mathbb{F}\}$

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 $[c, c]$

commands:

$\texttt{com}$	$::=$	$\texttt{V := exp}$	assignment into $\texttt{V} \in \mathbb{V}$
	$\mid$	$\texttt{exp} \bowtie 0$	test, $\bowtie \in \{=, <, >, <=, >=, <>\}$

Concrete semantics

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

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

Forward commands: $\mathsf{C}[\![\, c \,]\!] : \mathcal{P}(\mathbb{V} \to \mathbb{I}) \to \mathcal{P}(\mathbb{V} \to \mathbb{I})$

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

Backward commands: $\mathsf{C}[\![\, \overleftarrow{c} \,]\!] : \mathcal{P}(\mathbb{V} \to \mathbb{I}) \to \mathcal{P}(\mathbb{V} \to \mathbb{I})$

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

Abstract domain

- Abstract elements:
  - $\mathcal{D}^\sharp$, a set of computer-representable elements
  - $\gamma : \mathcal{D}^\sharp \to \mathcal{D}$ concretization
  - $\subseteq^\sharp$, an approximation order: $\mathcal{X}^\sharp \subseteq^\sharp \mathcal{Y}^\sharp \implies \gamma(\mathcal{X}^\sharp) \subseteq \gamma(\mathcal{Y}^\sharp)$

- Abstract operators:
  - $\mathsf{C}^\sharp[\![c]\!]$ such that $\mathsf{C}[\![c]\!] \gamma(\mathcal{X}^\sharp) \subseteq \gamma(\mathsf{C}^\sharp[\![c]\!] \mathcal{X}^\sharp)$
  - $\cup^\sharp$ such that $\gamma(\mathcal{X}^\sharp) \cup \gamma(\mathcal{Y}^\sharp) \subseteq \gamma(\mathcal{X}^\sharp \cup^\sharp \mathcal{Y}^\sharp)$
  - $\cap^\sharp$ such that $\gamma(\mathcal{X}^\sharp) \cap \gamma(\mathcal{Y}^\sharp) \subseteq \gamma(\mathcal{X}^\sharp \cap^\sharp \mathcal{Y}^\sharp)$
  - $\mathsf{C}^\sharp[\![\overleftarrow{c}]\!]$ such that $\gamma(\mathcal{X}^\sharp) \cap \mathsf{C}[\![\overleftarrow{c}]\!] \gamma(\mathcal{R}^\sharp) \subseteq \gamma(\mathsf{C}^\sharp[\![\overleftarrow{c}]\!] (\mathcal{X}^\sharp, \mathcal{R}^\sharp))$

- Fixpoint extrapolation:
  - $\triangledown : (\mathcal{D}^\sharp \times \mathcal{D}^\sharp) \to \mathcal{D}^\sharp$ widening
  - $\vartriangle : (\mathcal{D}^\sharp \times \mathcal{D}^\sharp) \to \mathcal{D}^\sharp$ narrowing

Linear equality domains

The affine equality domain

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

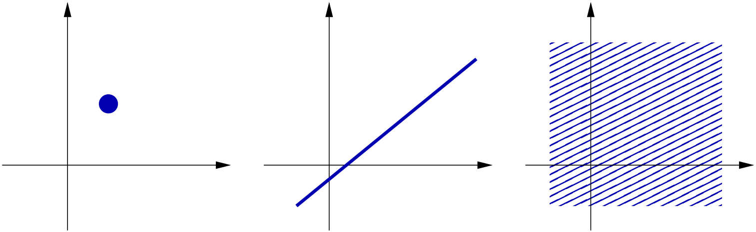
We look for invariants of the form:

$$\bigwedge_j \left(\sum_{i=1}^{n} \alpha_{ij} \mathtt{V}_i = \beta_j\right), \ \alpha_{ij}, \beta_j \in \mathbb{I}$$

where all the α_{ij} and β_j are inferred automatically.

We use a domain of affine spaces proposed by [Karr76]:

$$\mathcal{D}^\sharp \stackrel{\text{def}}{=} \{ \text{ affine subspaces of } \mathbb{V} \to \mathbb{I} \ \}$$



Affine equality representation

Machine representation: an affine subspace is represented as

- either the constant $\bot^\sharp$,
- or a pair $\langle \mathbf{M}, \vec{C} \rangle$ where
  - $\mathbf{M} \in \mathbb{I}^{m \times n}$ is a $m \times n$ matrix, $n = |\mathbb{V}|$ and $m \leq n$,
  - $\vec{C} \in \mathbb{I}^m$ is a row-vector with m rows.

$\langle \mathbf{M}, \vec{C} \rangle$ represents an equation system, with solutions:

$$\gamma(\langle \mathbf{M}, \vec{C} \rangle) \stackrel{\text{def}}{=} \{ \vec{V} \in \mathbb{I}^n \mid \mathbf{M} \times \vec{V} = \vec{C} \}$$

$\mathbf{M}$ should be in row echelon form:

- $\forall i \leq m,\ \exists k_i$ such that $M_{ik_i} = 1$
  and $\forall c < k_i, M_{ic} = 0$, $\forall l \neq i,\ M_{lk_i} = 0$,
- if $i < i'$ then $k_i < k_{i'}$.

Remarks:

- the representation is unique,
- as $m \leq n = |\mathbb{V}|$, the memory cost is in $\mathcal{O}(n^2)$ at worst,
- $\top^\sharp$ is represented as the empty equation system: $m = 0$.

Normalisation and emptiness testing

Let $\mathbf{M} \times \vec{V} = \vec{C}$ be a system, not necessarily in normal form.

The Gaussian reduction tells in $\mathcal{O}(n^3)$ time:

- whether the system is satisfiable, and in that case
- gives an equivalent system in normal form.

i.e. returns an element in $\mathcal{D}^\sharp$.

Example:

$$\left\{ \begin{array}{rcrcrcr} 2\mathtt{X} & + & \mathtt{Y} & + & \mathtt{Z} & = & 19 \\ 2\mathtt{X} & + & \mathtt{Y} & - & \mathtt{Z} & = & 9 \\ & & & & 3\mathtt{Z} & = & 15 \end{array} \right.$$

$$\Downarrow$$

$$\left\{ \begin{array}{rcrcrcr} \mathtt{X} & + & 0.5\mathtt{Y} & & & = & 7 \\ & & & & \mathtt{Z} & = & 5 \end{array} \right.$$

Normalisation and emptiness testing (cont.)

Gaussian reduction algorithm: $Gauss(\langle \mathbf{M}, \vec{C} \rangle)$

$r := 0$ *(rank r)*
for c from 1 to n *(column c)*
    if $\exists \ell > r,\ M_{\ell c} \neq 0$ *(pivot ℓ)*
        $r := r + 1$
        swap $\langle \vec{M}_\ell, C_\ell \rangle$ and $\langle \vec{M}_r, C_r \rangle$
        divide $\langle \vec{M}_r, C_r \rangle$ by M_{rc}
        for j from 1 to n, $j \neq r$
            replace $\langle \vec{M}_j, C_j \rangle$ with $\langle \vec{M}_j, C_j \rangle - M_{jc} \langle \vec{M}_r, C_r \rangle$

if $\exists \ell,\ \langle \vec{M}_\ell, C_\ell \rangle = \langle 0, \ldots, 0, c \rangle, c \neq 0$
    then return *unsatisfiable*

remove all rows $\langle \vec{M}_\ell, C_\ell \rangle$ that equal $\langle 0, \ldots, 0, 0 \rangle$

Affine equality operators

Applications

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

$$\mathcal{X}^\sharp \cap^\sharp \mathcal{Y}^\sharp \stackrel{\text{def}}{=} \textit{Gauss}\left(\left\langle \begin{bmatrix} \mathbf{M}_{\mathcal{X}^\sharp} \\ \mathbf{M}_{\mathcal{Y}^\sharp} \end{bmatrix}, \begin{bmatrix} \vec{C}_{\mathcal{X}^\sharp} \\ \vec{C}_{\mathcal{Y}^\sharp} \end{bmatrix} \right\rangle\right)$$

$$\mathcal{X}^\sharp =^\sharp \mathcal{Y}^\sharp \stackrel{\text{def}}{\iff} \mathbf{M}_{\mathcal{X}^\sharp} = \mathbf{M}_{\mathcal{Y}^\sharp} \quad \text{and} \quad \vec{C}_{\mathcal{X}^\sharp} = \vec{C}_{\mathcal{Y}^\sharp}$$

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

$$C^\sharp[\![\textstyle\sum_j \alpha_j \mathtt{V}_j - \beta = 0]\!] \mathcal{X}^\sharp \stackrel{\text{def}}{=} \textit{Gauss}\left(\left\langle \begin{bmatrix} \mathbf{M}_{\mathcal{X}^\sharp} \\ \alpha_1 \cdots \alpha_n \end{bmatrix}, \begin{bmatrix} \vec{C}_{\mathcal{X}^\sharp} \\ \beta \end{bmatrix} \right\rangle\right)$$

$$C^\sharp[\![e \bowtie 0]\!] \mathcal{X}^\sharp \stackrel{\text{def}}{=} \mathcal{X}^\sharp \qquad \text{for other tests}$$

Remark:

$\subseteq^\sharp$, $=^\sharp$, $\cap^\sharp$, $=^\sharp$ and $C^\sharp[\![\sum_j \alpha_j \mathtt{V}_j - \beta = 0]\!]$ are exact:

$$\mathcal{X}^\sharp \subseteq^\sharp \mathcal{Y}^\sharp \iff \gamma(\mathcal{X}^\sharp) \subseteq \gamma(\mathcal{Y}^\sharp), \quad \gamma(\mathcal{X}^\sharp \cap^\sharp \mathcal{Y}^\sharp) = \gamma(\mathcal{X}^\sharp) \cap \gamma(\mathcal{Y}^\sharp), \ldots$$

Generator representation

Generator representation

An affine subspace can also be represented as a set of vector generators $\vec{G}_1, \ldots, \vec{G}_m$ and an origin point $\vec{O}$, denoted as $[\mathbf{G}, \vec{O}]$.

$$\gamma([\mathbf{G}, \vec{O}]) \stackrel{\text{def}}{=} \{\, \mathbf{G} \times \vec{\lambda} + \vec{O} \mid \vec{\lambda} \in \mathbb{I}^m \,\} \qquad (\mathbf{G} \in \mathbb{I}^{n \times m},\ \vec{O} \in \mathbb{I}^n)$$

We can switch between a generator and a constraint representation:

- From generators to constraints: $\langle \mathbf{M}, \vec{C} \rangle = \mathit{Cons}([\mathbf{G}, \vec{O}])$

  Write the system $\vec{V} = \mathbf{G} \times \vec{\lambda} + \vec{O}$ with variables $\vec{V}$, $\vec{\lambda}$.
  Solve it in $\vec{\lambda}$ (by row operations).
  Keep the constraints involving only $\vec{V}$.

  e.g. $$\left\{ \begin{array}{rcl} \mathtt{X} & = & \lambda + 2 \\ \mathtt{Y} & = & 2\lambda + \mu + 3 \\ \mathtt{Z} & = & \mu \end{array} \right. \Longrightarrow \left\{ \begin{array}{rcl} \mathtt{X} - 2 & = & \lambda \\ -2\mathtt{X} + \mathtt{Y} + 1 & = & \mu \\ 2\mathtt{X} - \mathtt{Y} + \mathtt{Z} - 1 & = & 0 \end{array} \right.$$

  The result is: $2\mathtt{X} - \mathtt{Y} + \mathtt{Z} = 1$.

Generator representation (cont.)

- From constraints to generators: $[\mathbf{G}, \vec{O}] \stackrel{\text{def}}{=} \mathit{Gen}(\langle \mathbf{M}, \vec{C} \rangle)$

  Assume $\langle \mathbf{M}, \vec{C} \rangle$ is normalized.
  For each non-leading variable V, assign a distinct $\lambda_{\mathtt{V}}$,
  solve leading variables in terms of non-leading ones.

  e.g. $\left\{ \begin{array}{rcl} \mathtt{X} + 0.5\mathtt{Y} & = & 7 \\ \mathtt{Z} & = & 5 \end{array} \right. \implies \begin{bmatrix} -0.5 \\ 1 \\ 0 \end{bmatrix} \lambda_{\mathtt{Y}} + \begin{bmatrix} 7 \\ 0 \\ 5 \end{bmatrix}$

Affine equality operators (cont.)

Applications

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

$$\mathcal{X}^\sharp \cup^\sharp \mathcal{Y}^\sharp \stackrel{\text{def}}{=} \mathit{Cons}\left(\mathit{Gauss}\left(\left[\left[\begin{array}{ccc} \mathbf{G}_{\mathcal{X}^\sharp} & \mathbf{G}_{\mathcal{Y}^\sharp} & (\vec{O}_{\mathcal{Y}^\sharp} - \vec{O}_{\mathcal{X}^\sharp}) \end{array}\right], \vec{O}_{\mathcal{X}^\sharp}\right]\right)\right)$$

$$C^\sharp[\![\, \mathtt{V}_j :=]-\infty,+\infty[\,]\!]\, \mathcal{X}^\sharp \stackrel{\text{def}}{=} \mathit{Cons}\left(\mathit{Gauss}\left(\left[\left[\begin{array}{cc} \mathbf{G}_{\mathcal{X}^\sharp} & \vec{x}_j \end{array}\right], \vec{O}_{\mathcal{X}^\sharp}\right]\right)\right)$$

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

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

if $\alpha_j \neq 0, \mathcal{X}^\sharp$ where $\mathtt{V}_j$ is replaced with $(\mathtt{V}_j - \sum_{i \neq j} \alpha_i \mathtt{V}_i - \beta)/\alpha_j$

$C^\sharp[\![\, \mathtt{V}_j := e \,]\!]\, \mathcal{X}^\sharp \stackrel{\text{def}}{=} C^\sharp[\![\, \mathtt{V}_j :=]-\infty,+\infty[\,]\!]\, \mathcal{X}^\sharp$ for other assignments

Remarks:

- $\cup^\sharp$ is optimal, but not exact.
- $C^\sharp[\![\, \mathtt{V}_j := \sum_i \alpha_i \mathtt{V}_i + \beta \,]\!]$ and $C^\sharp[\![\, \mathtt{V}_j :=]-\infty,+\infty[\,]\!]$ are exact.

Affine equality operators (cont.)

Backward assignments:

$$C^\sharp[\![\overleftarrow{V_j :=]-\infty,+\infty[}]\!](\mathcal{X}^\sharp,\mathcal{R}^\sharp) \stackrel{\text{def}}{=} \mathcal{X}^\sharp \cap^\sharp (C^\sharp[\![V_j :=]-\infty,+\infty[ ]\!]\mathcal{R}^\sharp)$$

$$C^\sharp[\![\overleftarrow{V_j := \sum_i \alpha_i V_i + \beta}]\!](\mathcal{X}^\sharp,\mathcal{R}^\sharp) \stackrel{\text{def}}{=}$$

$\mathcal{X}^\sharp \cap^\sharp (\mathcal{R}^\sharp$ where V_j is replaced with $(\sum_i \alpha_i V_i + \beta))$

$$C^\sharp[\![\overleftarrow{V_j := e}]\!](\mathcal{X}^\sharp,\mathcal{R}^\sharp) \stackrel{\text{def}}{=} C^\sharp[\![\overleftarrow{V_j :=]-\infty,+\infty[}]\!](\mathcal{X}^\sharp,\mathcal{R}^\sharp)$$

for other assignments

Remarks:

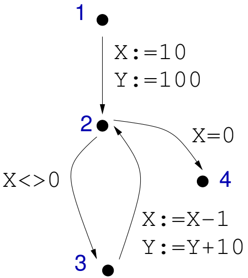
- $C^\sharp[\![\overleftarrow{V_j := \sum_i \alpha_i V_i + \beta}]\!]$ and $C^\sharp[\![\overleftarrow{V_j :=]-\infty,+\infty[}]\!]$ are exact
- a backward assignment can be seen as a substitution wrt. constraints (similar to weakest preconditions [Dijk75])

Analysis example

No infinite increasing chain: we can iterate without widening.

Forward analysis example:

```
¹X:=10; Y:=100;
while ²X<>0 do³
  X:=X-1;
  Y:=Y+10
done⁴
```



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	$\bot^\sharp$	$(10,100)$	$(10,100)$	$10\mathtt{X}+\mathtt{Y}=200$	$10\mathtt{X}+\mathtt{Y}=200$
3	$\bot^\sharp$	$\bot^\sharp$	$(10,100)$	$(10,100)$	$10\mathtt{X}+\mathtt{Y}=200$
4	$\bot^\sharp$	$\bot^\sharp$	$\bot^\sharp$	$\bot^\sharp$	$(0,200)$

Note in particular:

$$\mathcal{X}_2^{\sharp 3} = \{(10,100)\} \cup^\sharp \{(9,110)\} = \{\ (\mathtt{X},\mathtt{Y}) \mid 10\mathtt{X}+\mathtt{Y}=200\ \}$$

Constraint-only equality domain

In fact [Karr76] does not use the generator representation.
(rationale: few constraints but many generators in practice)

We need to redefine two operators: forgetting and union.

- $C^\sharp[\![\, V_j :=]-\infty, +\infty[\,]\!]$

  Pick the row $\langle \vec{M}_i, C_i \rangle$ such that $M_{ij} \neq 0$ and i maximal.
  Use it to eliminate all non-0 occurrences of V_j in $\mathbf{M}$.
  Then remove the row $\langle \vec{M}_i, C_i \rangle$.

  e.g. forgetting Z: $\left\{ \begin{array}{l} X \quad + Z = 10 \\ \quad Y + Z = 7 \end{array} \right. \implies \{\ X - Y = 3$

  The operator is exact.

Constraint-only equality domain (cont.)

- $\langle \mathbf{M}, \vec{C} \rangle \cup^\sharp \langle \mathbf{N}, \vec{D} \rangle$

  Idea: unify columns 1 to n in $\langle \mathbf{M}, \vec{C} \rangle$ and $\langle \mathbf{N}, \vec{D} \rangle$ using row operations.

  e.g. unify columns ${}^t(\vec{0}\ 1\ \vec{0})$ and ${}^t(\vec{\beta}\ 0\ \vec{0})$.

$$\begin{pmatrix} \mathbf{R} & \vec{0} & \mathbf{M_1} \\ \vec{0} & 1 & \vec{M_2} \\ \mathbf{0} & \vec{0} & \mathbf{M_3} \end{pmatrix}, \begin{pmatrix} \mathbf{R} & \vec{\beta} & \mathbf{N_1} \\ \vec{0} & 0 & \vec{N_2} \\ \mathbf{0} & \vec{0} & \mathbf{N_3} \end{pmatrix} \Longrightarrow \begin{pmatrix} \mathbf{R} & \vec{\beta} & \mathbf{M'_1} \\ \vec{0} & 0 & \vec{0} \\ \mathbf{0} & \vec{0} & \mathbf{M_3} \end{pmatrix}, \begin{pmatrix} \mathbf{R} & \vec{\beta} & \mathbf{N_1} \\ \vec{0} & 0 & \vec{N_2} \\ \mathbf{0} & \vec{0} & \mathbf{N_3} \end{pmatrix}$$

  Use the row $(\vec{0}\ 1\ \vec{M_2})$ to create β in the left argument.
  Then remove the row $(\vec{0}\ 1\ \vec{M_2})$.
  The right argument is unchanged.

  Unifying ${}^t(\vec{\alpha}\ 0\ \vec{0})$ and ${}^t(\vec{\beta}\ 0\ \vec{0})$ is a bit more complicated...

A note on integers

Suppose now that $\mathbb{I} = \mathbb{Z}$.

- $\mathbb{Z}$ is not closed under affine operations: $(x/y) \times y \neq x$,
- Gaussian reduction implemented in $\mathbb{Z}$ is unsound.
  (e.g. unsound normalization $2\mathtt{X} + \mathtt{Y} = 19 \not\Longrightarrow \mathtt{X} = 9$, by truncation)

One possible solution

- keep a representation using matrices with coefficients in $\mathbb{Q}$,
- keep all abstract operators as in $\mathbb{Q}$,
- change the concretization into: $\gamma_{\mathbb{Z}}(\mathcal{X}^\sharp) \stackrel{\text{def}}{=} \gamma(\mathcal{X}^\sharp) \cap \mathbb{Z}^n$.

With respect to $\gamma_{\mathbb{Z}}$, the operators are no longer best / exact.

Example: where $\mathcal{X}^\sharp$ is the equation $\mathtt{Y} = 2\mathtt{X}$

- $\gamma_{\mathbb{Z}}(\mathcal{X}^\sharp) = \{ (\mathtt{X},\mathtt{Y}) \mid \mathtt{X} \in \mathbb{Z},\ \mathtt{Y} = 2\mathtt{X} \}$
- $(\mathsf{C}[\![\mathtt{X} := 0]\!] \circ \gamma_{\mathbb{Z}})\mathcal{X}^\sharp = \{ (\mathtt{X},\mathtt{Y}) \mid \mathtt{X} = 0,\ \mathtt{Y}$ is even $\}$
- $(\gamma_{\mathbb{Z}} \circ \mathsf{C}^\sharp[\![\mathtt{X} := 0]\!])\mathcal{X}^\sharp = \{ (\mathtt{X},\mathtt{Y}) \mid \mathtt{X} = 0,\ \mathtt{Y} \in \mathbb{Z} \}$

The analysis forgets the "intergerness" of variables.

The congruence equality domain

Now, $\mathbb{I} = \mathbb{Z}$.

We look for invariants of the form: $\bigwedge_j \left(\sum_{i=1}^{n} m_{ij} \mathtt{V}_i \equiv c_j \; [k_j] \right)$.

Algorithms:

- there exists minimal forms (but not unique), computed using an extension of Euclide's algorithm,
- there is a dual representation: $\{ \mathbf{G} \times \vec{\lambda} + \vec{O} \mid \vec{\lambda} \in \mathbb{Z}^m \}$, and passage algorithms,
- see [Gran91].

Analysis example

Program example:

```
X:=0; Y:=0;
while •[0,1]=0 do
  if [0,1]=0 then X:=X+4
  else X:=X+12 fi;
  Y:=Y+4
done
```

At •, we find: $(X \equiv 0\ [4]) \wedge (Y \equiv 0\ [4]) \wedge (X \equiv Y\ [8])$.

Polyhedron domain

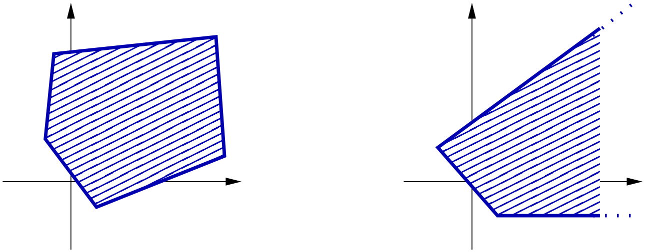
The polyhedron domain

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

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 polyhedron domain proposed by [Cous78]:

$\mathcal{D}^\sharp \stackrel{\text{def}}{=} \{$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 [Schr86])

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: $\vec{R}_1, \ldots, \vec{R}_r$

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

Origin of duality

Dual $A^* \stackrel{\text{def}}{=} \{ \vec{x} \in \mathbb{I}^n \mid \forall \vec{a} \in A,\ \vec{a} \cdot \vec{x} \leq 0 \}$

- $\{\vec{a}\}^*$ and $\{\lambda \vec{r} \mid \lambda \geq 0\}^*$ are half-spaces,
- $(A \cup B)^* = A^* \cap B^*$,
- if A is convex, closed, and $\vec{0} \in A$, then $A^{**} = A$.

Duality on polyhedral cones:

Cone: $C = \{\vec{V} \mid \mathbf{M} \times \vec{V} \geq \vec{0}\}$ or $C = \{ \sum_{j=1}^{r} \beta_j \vec{R}_j \mid \forall j,\ \beta_j \geq 0\}$

- $C^{**} = C$,
- C^* is also a polyhedral cone,
- a ray of C corresponds to a constraint of C^*,
- a constraint of C corresponds to a ray of C^*.

extended to polyhedra by homogenisation to polyhedral codes:

$C(P) \stackrel{\text{def}}{=} \{ \lambda \vec{V} \mid \lambda \geq 0,\ (V_1, \ldots, V_n) \in \gamma(P),\ V_{n+1} = 1 \} \subseteq \mathbb{I}^{n+1}$

Polyhedra representation (cont.)

Minimal representations

- A constraint system is minimal if no constraint can be omitted without changing the concretization.
- A generator system is minimal if no generator can be omitted without changing the concretization.

Remarks:

- most operators are easier on one representation;
- minimal representations are not unique;
- there is no memory bound on the representations (even minimal ones);
- equality constraints, as well as lines (pairs of opposed rays) may be handled separately and more efficiently.

Chernikova's algorithm

Switch from a constraint system to an equivalent generator system.

Algorithm introduced by [Cher68].

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. a $n-$dimensional hyper-cube has $2n$ constraints and 2^n vertices)

Algorithm: incrementally add constraints one by one

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

Chernikova's algorithm (cont.)

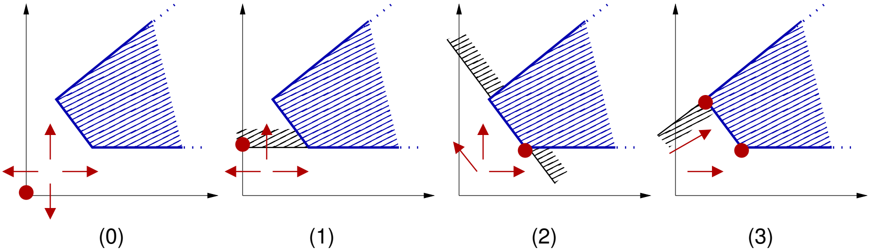
Update $[\mathbf{P}_{k-1}, \mathbf{R}_{k-1}]$ to $[\mathbf{P}_k, \mathbf{R}_k]$
by adding one constraint $\vec{M}_k \cdot \vec{V} \geq C_k \in \langle \mathbf{M}, \vec{C} \rangle$:

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 $\mathbf{P}_k$:
  $\frac{C_k - \vec{M}_k \cdot \vec{Q}}{\vec{M}_k \cdot \vec{P} - \vec{M}_k \cdot \vec{Q}} \vec{P} - \frac{C_k - \vec{M}_k \cdot \vec{P}}{\vec{M}_k \cdot \vec{P} - \vec{M}_k \cdot \vec{Q}} \vec{Q}$
- 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 $\mathbf{P}_k$:
  $\vec{P} + \frac{C_k - \vec{M}_k \cdot \vec{P}}{\vec{M}_k \cdot \vec{R}} \vec{R}$
- 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 $\mathbf{R}_k$: $(\vec{M}_k \cdot \vec{S})\vec{R} - (\vec{M}_k \cdot \vec{R})\vec{S}$

Chernikova's algorithm (example)

Example:



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

Redudancy removal

Goal: only introduce non-redundant points and rays during Chernikova's algorithm.

Definitions (for rays in polyhedral cones)

Given $C = \{\vec{V} \mid \mathbf{M} \times \vec{V} \geq \vec{0}\} = \{\, \mathbf{R} \times \vec{\beta} \mid \vec{\beta} \geq \vec{0}\}$.

$\vec{R}$ saturates $\vec{M}_k \cdot \vec{V} \geq 0 \overset{\text{def}}{\iff} \vec{M}_k \cdot \vec{R} = 0$.

$S(\vec{R}, C) \overset{\text{def}}{=} \{k \mid \vec{M}_k \cdot \vec{R} = 0\}$.

Theorem:

assume C has no line ($\nexists \vec{L} \neq \vec{0}$ s.t. $\forall \alpha, \alpha\vec{L} \in C$)

$\vec{R}$ is non-redundant wrt. $\mathbf{R} \iff \nexists \vec{R}_i \in \mathbf{R}, S(\vec{R}, C) \subseteq S(\vec{R}_i, C)$

- $S(\vec{R}_i, C)$, $\vec{R}_i \in \mathbf{R}$ is maintained during Chernikova's algorithm in a saturation matrix,
- extension possible to polyhedra and lines,
- various improvements exist [LeVe92].

Operators on polyhedra

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

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

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

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

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

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.)

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

$$C^\sharp[\![\, \sum_i \alpha_i \mathtt{V}_i + \beta = 0 \,]\!]\, \mathcal{X}^\sharp \stackrel{\text{def}}{=}$$
$$(C^\sharp[\![\, \sum_i \alpha_i \mathtt{V}_i + \beta \geq 0 \,]\!] \circ C^\sharp[\![\, \sum_i (-\alpha_i) \mathtt{V}_i - \beta \geq 0 \,]\!])\mathcal{X}^\sharp$$

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

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

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:

- $C^\sharp[\![\, \sum_i \alpha_i \mathtt{V}_i + \beta \geq 0 \,]\!]$, $C^\sharp[\![\, \mathtt{V}_j := \sum_i \alpha_i \mathtt{V}_i + \beta \,]\!]\, \mathcal{X}$ and $C^\sharp[\![\, \mathtt{V}_j :=]-\infty, +\infty[\,]\!]$ are exact.
- We can also define $C^\sharp[\![\, \mathtt{V}_j := \sum_i \alpha_i \mathtt{V}_i + \beta \,]\!]$ on a generator system.

Operators on polyhedra (cont.)

Backward assignments:

$$C^\sharp[\![\, \overleftarrow{V_j :=]-\infty, +\infty[} \,]\!] (\mathcal{X}^\sharp, \mathcal{R}^\sharp) \stackrel{\text{def}}{=} \mathcal{X}^\sharp \cap^\sharp (C^\sharp[\![\, V_j :=]-\infty, +\infty[\,]\!]\, \mathcal{R}^\sharp)$$

$$C^\sharp[\![\, \overleftarrow{V_j := \sum_i \alpha_i V_i + \beta} \,]\!] (\mathcal{X}^\sharp, \mathcal{R}^\sharp) \stackrel{\text{def}}{=}$$
$\mathcal{X}^\sharp \cap^\sharp (\mathcal{R}^\sharp$ where V_j is replaced with $(\sum_i \alpha_i V_i + \beta))$

$$C^\sharp[\![\, \overleftarrow{V_j := e} \,]\!] (\mathcal{X}^\sharp, \mathcal{R}^\sharp) \stackrel{\text{def}}{=} C^\sharp[\![\, \overleftarrow{V_j :=]-\infty, +\infty[} \,]\!] (\mathcal{X}^\sharp, \mathcal{R}^\sharp)$$
for other assignments

Note: identical to the case of linear equalities.

Polyhedra widening

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

Definition:

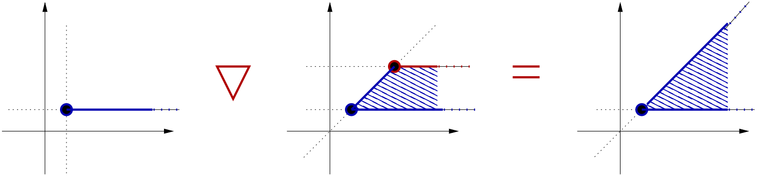
Take $\mathcal{X}^\sharp$ and $\mathcal{Y}^\sharp$ in minimal constraint-set form.

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

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

However, we 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:

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

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

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

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 [Bagn03].

Strict inequalities

The polyhedron domain can be extended to allow strict constraints: $\{ \vec{V} \mid \mathbf{M} \times \vec{V} \geq \vec{C} \text{ and } \mathbf{M}' \times \vec{V} > \vec{C}' \}$

Idea:

A non-closed polyhedron on $\mathbb{V}$ is represented as a closed polyhedron P on $\mathbb{V}' \stackrel{\text{def}}{=} \mathbb{V} \cup \{\mathtt{V}_\epsilon\}$.

$$\begin{array}{lll} \alpha_1 \mathtt{V}_1 + \cdots + \alpha_n \mathtt{V}_n + 0\mathtt{V}_\epsilon \geq 0 & \text{represents} & \alpha_1 \mathtt{V}_1 + \cdots + \alpha_n \mathtt{V}_n \geq 0 \\ \alpha_1 \mathtt{V}_1 + \cdots + \alpha_n \mathtt{V}_n - c\mathtt{V}_\epsilon \geq 0,\ c > 0 & \text{represents} & \alpha_1 \mathtt{V}_1 + \cdots + \alpha_n \mathtt{V}_n > 0 \end{array}$$

P represents the non necessarily closed polyhedron:
$\gamma_\epsilon(P) \stackrel{\text{def}}{=} \{(\mathtt{V}_1, \ldots, \mathtt{V}_n) \mid \exists \mathtt{V}_\epsilon > 0,\ (\mathtt{V}_1, \ldots, \mathtt{V}_n, \mathtt{V}_\epsilon) \in \gamma(P)\}$.

Notes:

- The minimal form needs some adaptation [Bagn02].
- Chernikova's algorithm, $\cap^\sharp$, $\cup^\sharp$, $C^\sharp[\![c]\!]$, and $C^\sharp[\![\overleftarrow{c}]\!]$ can be easily reused.

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 $\mathcal{X}^\sharp$:

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

we then have:

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

Constraint-only polyhedron domain (cont.)

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

Redundancy removal: using linear programming [Schr86]

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

If $c = (\vec{\alpha} \cdot \vec{\mathrm{V}} \geq \beta) \in \mathcal{X}^\sharp$ and $\beta \leq$ *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:

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

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

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

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

For $\cup^\sharp$, we introduce temporaries $\mathtt{V}_j^{\mathcal{X}}$, $\mathtt{V}_j^{\mathcal{Y}}$, $\sigma^{\mathcal{X}}$, $\sigma^{\mathcal{Y}}$:

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

(see [Beno96])

Integer polyhedra

How can we deal with $\mathbb{I} = \mathbb{Z}$?

Issue: integer linear programming is difficult.

Example: satsfiability of conjunctions of linear constraints:

- polynomial cost in $\mathbb{Q}$,
- NP-complete cost in $\mathbb{Z}$.

Possible solutions:

- Use some complete integer algorithms.
  (e.g. Presburger arithmetics)
  Costly, and we do not have any abstract domain structure.
- Keep $\mathbb{Q}-$polyhedra as representation, and change the concretization into: $\gamma_{\mathbb{Z}}(\mathcal{X}^\sharp) \stackrel{\text{def}}{=} \gamma(\mathcal{X}^\sharp) \cap \mathbb{Z}^n$.
  However, operators are no longer exact / optimal.

Weakly relational domains

Zone domain

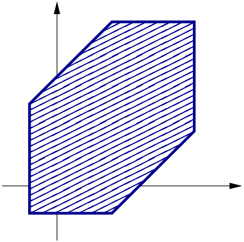
The zone domain

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**.



[Mine01a]

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:

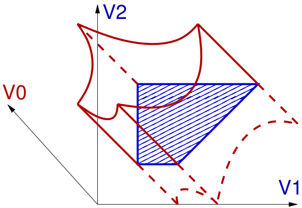
$$\gamma(\mathbf{m}) \stackrel{\text{def}}{=} \{ (v_1, \ldots, v_n) \in \mathbb{I}^n \mid \forall i, j,\ v_j - v_i \leq m_{ij} \}.$$

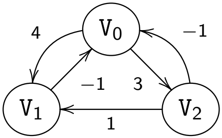
Machine representation (cont.)

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

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

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

Remarks:

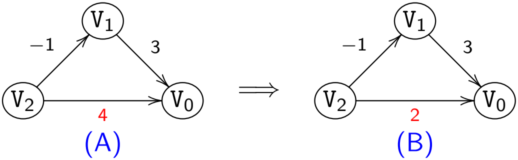
- $\mathcal{D}^\sharp$ is complete if $\leq$ is ($\mathbb{I} = \mathbb{R}$ or $\mathbb{Z}$, but not $\mathbb{Q}$),
- $\mathbf{m} \subseteq^\sharp \mathbf{n} \implies \gamma_0(\mathbf{m}) \subseteq \gamma_0(\mathbf{n})$, but not the converse,
- $\mathbf{m} =^\sharp \mathbf{n} \implies \gamma_0(\mathbf{m})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

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



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

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

- If $\gamma_0(\mathbf{m}) \neq \emptyset$, then $\mathbf{m}^* = \mathbf{m}^{n+1}$, (normal form)
- $\gamma_0(\mathbf{m}) = \emptyset \iff \exists i,\ m^{n+1}_{ii} < 0$, (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 $\perp^\sharp$, 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:

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

$$C^\sharp[\![\, V_{j_0} - V_{i_0} = [a,b] \,]\!]\, \mathbf{m} \stackrel{\text{def}}{=} (C^\sharp[\![\, V_{j_0} - V_{i_0} \leq b \,]\!] \circ C^\sharp[\![\, V_{i_0} - V_{j_0} \leq -a \,]\!])\mathbf{m}$$

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

(not optimal on non-closed arguments)

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

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

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

These transfer functions are exact.

Abstract operators (cont.)

Backward assignment:

$$C^\sharp[\![\overleftarrow{V_{j_0} :=]-\infty,+\infty[}\,]\!](\mathbf{m},\mathbf{r}) \stackrel{\text{def}}{=} \mathbf{m} \cap^\sharp (C^\sharp[\![\, V_{j_0} :=]-\infty,+\infty[\,]\!]\,\mathbf{r})$$

$$C^\sharp[\![\overleftarrow{V_{j_0} := V_{j_0} + [a,b]}\,]\!](\mathbf{m},\mathbf{r}) \stackrel{\text{def}}{=} \mathbf{m} \cap^\sharp (C^\sharp[\![\, V_{j_0} := V_{j_0} + [-b,-a]\,]\!]\,\mathbf{r})$$

$$\left[C^\sharp[\![\overleftarrow{V_{j_0} := V_{i_0} + [a,b]}\,]\!](\mathbf{m},\mathbf{r})\right]_{ij} \stackrel{\text{def}}{=}$$

$$\mathbf{m} \cap^\sharp \begin{cases} \min(\mathbf{r}^*_{ij}, \mathbf{r}^*_{j_0 j} + b) & \text{if } i = i_0 \text{ and } j \neq i_0, j_0 \\ \min(\mathbf{r}^*_{ij}, \mathbf{r}^*_{i j_0} - a) & \text{if } j = i_0 \text{ and } i \neq i_0, j_0 \\ +\infty & \text{if } i = j_0 \text{ or } j = j_0 \\ \mathbf{r}^*_{ij} & \text{otherwise.} \end{cases}$$

Abstract operators (cont.)

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

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

Possible solution:

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

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

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

Quadratic total cost (plus the cost of closure).

Abstract operators (cont.)

Example:

Argument

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

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

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

Intervals

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

Approximate solution

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

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 $\triangledown$

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

Unstable constraints are deleted.

Narrowing $\triangle$

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

Only $+\infty$ bounds are refined.

Remarks:

- We can construct widenings with thresholds.
- $\triangledown$ (resp. $\triangle$) can be seen as a point-wise extension of an interval widening (resp. narrowing).

Interaction between closure and widening

Widening $\triangledown$ and closure $*$ cannot always be mixed safely:

- $\mathbf{m}_{i+1} \stackrel{\text{def}}{=} \mathbf{m}_i \triangledown (\mathbf{n}_i^*)$ OK
- $\mathbf{m}_{i+1} \stackrel{\text{def}}{=} (\mathbf{m}_i^*) \triangledown \mathbf{n}_i$ wrong!
- $\mathbf{m}_{i+1} \stackrel{\text{def}}{=} (\mathbf{m}_i \triangledown \mathbf{n}_i)^*$ wrong

otherwise the sequence $(\mathbf{m}_i)$ may be infinite!

Example:

```
X:=0; Y:=[-1,1];
while • 1=1 do
  R:=[-1,1];
  if X=Y then Y:=X+R
  else X:=Y+R fi
done
```

$\mathcal{X}_\bullet^{\sharp 2j}$	$\mathcal{X}_\bullet^{\sharp 2j+1}$
$\mathtt{X} \in [-2j, 2j]$ $\mathtt{Y} \in [-2j-1, 2j+1]$ $\mathtt{X}-\mathtt{Y} \in [-1, 1]$	$\mathtt{X} \in [-2j-2, 2j+2]$ $\mathtt{Y} \in [-2j-1, 2j+1]$ $\mathtt{X}-\mathtt{Y} \in [-1, 1]$

Applying the closure after the widening at • prevents convergence.
Without the closure, we would find in finite time $\mathtt{X}-\mathtt{Y} \in [-1, 1]$.
Note: this situation also occurs in reduced products
(here, $\mathcal{D}^\sharp \simeq$reduced product of $n \times n$ intervals, $* \simeq$reduction)

Octagon domain

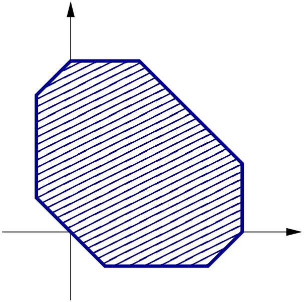
The octagon domain

Now, $\mathbb{I} \in \{\mathbb{Q}, \mathbb{R}\}$.

We look for invariants of the form: $\bigwedge \quad \pm \mathtt{V}_i \pm \mathtt{V}_j \leq c, \quad c \in \mathbb{I}$

A subset of $\mathbb{I}^n$ defined by such constraints is called an octagon.

It is a generalisation of zones (more symmetric).



[Mine01b]

Machine representation

Idea: use a variable change to get back to potential constraints.

Let $\mathbb{V}' \stackrel{\text{def}}{=} \{\mathtt{V'}_1, \ldots, \mathtt{V'}_{2n}\}$.

the constraint:	is encoded as:
$\mathtt{V}_i - \mathtt{V}_j \leq c \quad (i \neq j)$	$\mathtt{V'}_{2i-1} - \mathtt{V'}_{2j-1} \leq c$ and $\mathtt{V'}_{2j} - \mathtt{V'}_{2i} \leq c$
$\mathtt{V}_i + \mathtt{V}_j \leq c \quad (i \neq j)$	$\mathtt{V'}_{2i-1} - \mathtt{V'}_{2j} \leq c$ and $\mathtt{V'}_{2j-1} - \mathtt{V'}_{2i} \leq c$
$-\mathtt{V}_i - \mathtt{V}_j \leq c \quad (i \neq j)$	$\mathtt{V'}_{2j} - \mathtt{V'}_{2i-1} \leq c$ and $\mathtt{V'}_{2i} - \mathtt{V'}_{2j-1} \leq c$
$\mathtt{V}_i \leq c$	$\mathtt{V'}_{2i-1} - \mathtt{V'}_{2i} \leq 2c$
$\mathtt{V}_i \geq c$	$\mathtt{V'}_{2i} - \mathtt{V'}_{2i-1} \leq -2c$

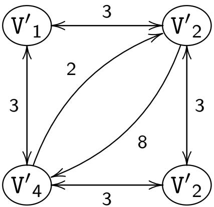
We use a matrix $\mathbf{m}$ of size $(2n) \times (2n)$ with elements in $\mathbb{I} \cup \{+\infty\}$ and $\gamma_{\pm}(\mathbf{m}) \stackrel{\text{def}}{=} \{\, (v_1, \ldots, v_n) \mid (v_1, -v_1, \ldots, v_n, -v_n) \in \gamma(\mathbf{m}) \,\}$.

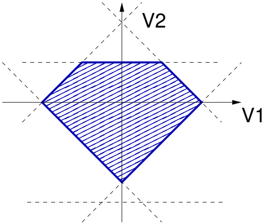
Note:
Two distinct $\mathbf{m}$ elements can represent the same constraint on $\mathbb{V}$.
To avoid this, we impose that $\forall i, j,\ m_{ij} = m_{\bar{\jmath}\,\bar{\imath}}$ where $\bar{\imath} = i \oplus 1$.

Machine representation (cont.)

Example:

$$\begin{cases} V_1 + V_2 \leq 3 \\ V_2 - V_1 \leq 3 \\ V_1 - V_2 \leq 3 \\ -V_1 - V_2 \leq -3 \\ 2V_2 \leq 2 \\ -2V_2 \leq 8 \end{cases}$$





Lattice

Constructed by point-wise extension of $\leq$ on $\mathbb{I} \cup \{+\infty\}$.

Algorithms

$\mathbf{m}^*$ is not a normal form for $\gamma_\pm$.

Idea use two local transformations instead of one:

$$\left\{\begin{array}{l} \mathtt{V'}_i - \mathtt{V'}_k \leq c \\ \mathtt{V'}_k - \mathtt{V'}_j \leq d \end{array}\right. \implies \mathtt{V'}_i - \mathtt{V'}_j \leq c + d$$

and

$$\left\{\begin{array}{l} \mathtt{V'}_i - \mathtt{V'}_{\bar\imath} \leq c \\ \mathtt{V'}_{\bar\jmath} - \mathtt{V'}_j \leq d \end{array}\right. \implies \mathtt{V'}_i - \mathtt{V'}_j \leq (c + d)/2$$

Modified Floyd–Warshall algorithm

$\mathbf{m}^\bullet \stackrel{\text{def}}{=} S(\mathbf{m}^{2n+1})$

where:

(A) $$\left\{\begin{array}{l} \mathbf{m}^1 \stackrel{\text{def}}{=} \mathbf{m} \\ [\mathbf{m}^{k+1}]_{ij} \stackrel{\text{def}}{=} \min(n_{ij}, n_{ik} + n_{kj}),\ 1 \leq k \leq 2n \end{array}\right.$$

(B) $$[S(\mathbf{n})]_{ij} \stackrel{\text{def}}{=} \min(n_{ij}, (n_{i\,\bar\imath} + n_{\bar\jmath j})/2)$$

Algorithms (cont.)

Applications

- $\gamma_\pm(\mathbf{m}) = \emptyset \iff \exists i,\ \mathbf{m}^\bullet_{ii} < 0$,
- if $\gamma_\pm(\mathbf{m}) \neq \emptyset$, $\mathbf{m}^\bullet$ is a normal form:
  $$\mathbf{m}^\bullet = \min_{\subseteq^\sharp} \{\, \mathbf{n} \mid \gamma_\pm(\mathbf{n}) = \gamma_\pm(\mathbf{m}) \,\},$$
- $(\mathbf{m}^\bullet) \cup^\sharp (\mathbf{n}^\bullet)$ is the best abstraction for the set-union $\gamma_\pm(\mathbf{m}) \cup \gamma_\pm(\mathbf{n})$.

Widening and narrowing

- The zone widening and narrowing can be used on octagons.
- The widened iterates should not be closed.
  (prevents convergence)

Abstract transfer functions are similar to the case of the zone domain.

Analysis example

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 $|\mathtt{R}|$

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.

Integer octagons

Recall that zones work equally well on $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{Z}$.

Issue:

The octagon domain we have presented is not complete on $\mathbb{Z}$:

- the algorithm for $\mathbf{m}^\bullet$ uses divisions by 2,
- when replacing $x \mapsto x/2$ with $\mapsto \lfloor x/2 \rfloor$, we get:
  $\mathbf{m}^\bullet \neq \min_{\subseteq^\sharp} \{\, \mathbf{o} \mid \gamma_\pm(\mathbf{o}) = \gamma_\pm(\mathbf{m}) \,\}$.

Possible solutions:

- Use $\mathbf{m}^\bullet$ with $\lfloor x/2 \rfloor$ instead of $/2$.
  All computations remain sound on integers.
  The best-precision results are no longer valid.
- See [Bagn08] for a $\mathcal{O}(n^3)$ time "tight closure" for integer octagons.

Summary

Summary of numerical domains

domain	non-relational	linear equalities	polyhedra	octagons
invariants	$\mathtt{V} \in D_b^\sharp$	$\sum_i \alpha_i \mathtt{V_i} = \beta$	$\sum_i \alpha_i \mathtt{V_i} \leq \beta$	$\pm \mathtt{V_i} \pm \mathtt{V_j} \leq c$
memory cost	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(2^n)$	$\mathcal{O}(n^2)$
time cost	$\mathcal{O}(n)$	$\mathcal{O}(n^3)$	$\mathcal{O}(2^n)$	$\mathcal{O}(n^3)$

Linearization

Abstraction framework

Issue:

Most relational domains can only deal with linear expressions.
How can we abstract non-linear assignments such as $\texttt{X} := \texttt{Y} \times \texttt{Z}$?

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

Framework:

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

$$R \models e_1 \preceq e_2 \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \forall \rho \in R,\ \mathsf{E}[\![\, e_1 \,]\!]\, \rho \subseteq \mathsf{E}[\![\, e_2 \,]\!]\, \rho.$$

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

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

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

Linearization

In practice, we put expressions into affine interval form:

$$\texttt{exp}_\ell : [a_0, b_0] + \sum_k [a_k, b_k] \times \texttt{V}_\texttt{k}$$

Advantages:

- affine expressions are easy to manipulate,
- interval coefficients allow non-determinism in expressions, hence, the opportunity for abstraction,
- we can easily construct abstract transfer functions for affine interval expressions.

Linearization (cont.)

Operations on affine interval forms

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

Noting i_k the interval $[a_k, b_k]$ and using interval operations $+^\sharp_b$, $-^\sharp_b$, $\times^\sharp_b$, $/^\sharp_b$ (e.g., $[a,b] +^\sharp_b [c,d] = [a+c, b+d]$):

- $(i_0 + \sum_k i_k \times \mathtt{V_k}) \boxplus (i'_0 + \sum_k i'_k \times \mathtt{V_k}) \stackrel{\text{def}}{=} (i_0 +^\sharp_b i'_0) + \sum_k (i_k +^\sharp_b i'_k) \times \mathtt{V_k}$
- $i \boxtimes (i_0 + \sum_k i_k \times \mathtt{V_k}) \stackrel{\text{def}}{=} (i \times^\sharp_b i_0) + \sum_k (i \times^\sharp_b i_k) \times \mathtt{V_k}$
- ...

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 (cont.)

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

Flattens the expression into a single interval:

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

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

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 $e \in \texttt{exp}$, 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., $\ell(\texttt{V}/\texttt{V}, \texttt{V} \mapsto [1, +\infty]) = [0, 1] \times \texttt{V} \not\preceq 1$)

Application to the octagon domain

```
Y:=[0,+∞];
T:=[-1,1];
X:=T×Y
```

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

Linearization application (cont.)

Application to the interval domain

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

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:
  $E^\sharp[\![\, 2 \times \mathtt{V} - \mathtt{V} \,]\!]\, (\mathcal{X}^\sharp) = 2 \times_b^\sharp [a, b] -_b^\sharp [a, b] = [2a - b, 2b - a]$,
- after linearization $\ell(2 \times \mathtt{V} - \mathtt{V}, \mathcal{X}^\sharp) = \mathtt{V}$, so
  $$E^\sharp[\![\, \ell(2 \times \mathtt{V} - \mathtt{V}, \mathcal{X}^\sharp) \,]\!]\, \mathcal{X}^\sharp = [a, b]$$
  strictly more precise than $[2a - b, 2b - a]$ when $a \neq b$.

Bibliography

Bibliography

[Anco10] **C. Ancourt, F. Coelho & F. Irigoin**. *A modular static analysis approach to affine loop invariants detection.* In Proc. NSAD'10, ENTCS, Elsevier, 2010.

[Bagn02] **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.

[Bagn03] **R. Bagnara, P. Hill, E. Ricci, E. Zaffanella**. *Precise widening operators for convex polyhedra.* In Proc. SAS'03, LNCS 2694, 337-354, Springer, 2003.

[Bagn08] **R. Bagnara, P. M. Hill & E. Zaffanella**. *An improved tight closure algorithm for integer octagonal constraints.* In Proc. VMCAI'08, LNCS 4905, 8–21, Springer, 2008.

[Beno96] **F. Benoy & A. King**. *Inferring argument size relationships with CLP(R).* In In Proc. of LOPSTR'96, LNCS 1207, 204–223. Springer, 1996.

Bibliography (cont.)

[Chen08] **L. Chen, A. Miné & P. Cousot**. *A sound floating-point polyhedra abstract domain.* In Proc. APLAS'08, LNCS 5356, 3–18, Springer, 2008.

[Cher68] **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.

[Cous78] **P. Cousot & N. Halbwachs**. *Automatic discovery of linear restraints among variables of a program.* In Proc. POPL'78, 84–96, ACM, 1978.

[Gran91] **P. Granger**. *Static analysis of linear congruence equalities among variables of a program.* In Proc. TAPSOFT'91, LNCS 49, 169–192. Springer, 1991.

[Jean09] **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`.

Bibliography (cont.)

[Karr76] **M. Karr**. *Affine relationships among variables of a program.* In Acta Informatica, 6:133–151, 1976.

[LeVe92] **H. Le Verge**. *A note on Chernikova's algorithm.* In Research Report 1662, INRIA Rocquencourt, 1992.

[Mine01a] **A. Miné**. *A new numerical abstract domain based on difference-bound matrices.* In Proc. PADO II, LNCS 2053, 155–172, Springer, 2001.

[Mine01b] **A. Miné**. *The octagon abstract domain.* In Proc. AST'01, 310–319, IEEE, 2001.

[Schr86] **A. Schrijver**. *Theory of linear and integer programming.* In John Wiley & Sons, Inc., 1986.