Widening Operators for Weakly-Relational Numeric Abstractions

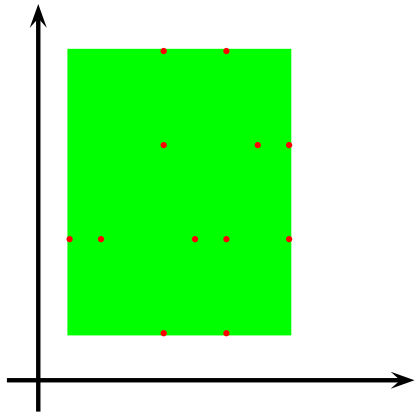
Roberto BAGNARA, Elena MAZZI, Enea ZAFFANELLA
University of Parma, Italy

Patricia M. HILL,
University of Leeds, United Kingdom

http://www.cs.unipr.it/ppl/

WIDENINGS OPERATORS FOR *What?*

Intervals, Polyhedra, and Things in Between (I)



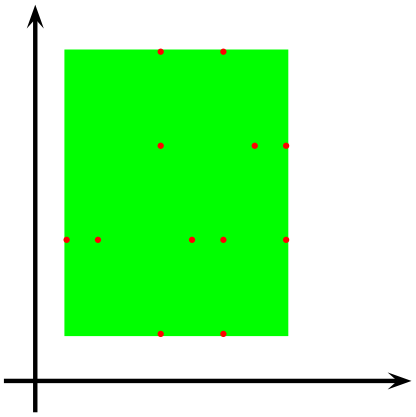
Interval

A non-relational domain.

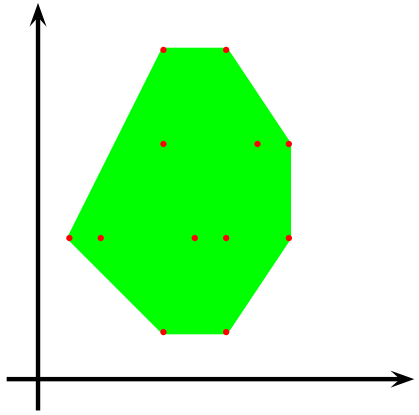
Very efficient.

May be imprecise.

Intervals, Polyhedra, and Things in Between (II)



Interval



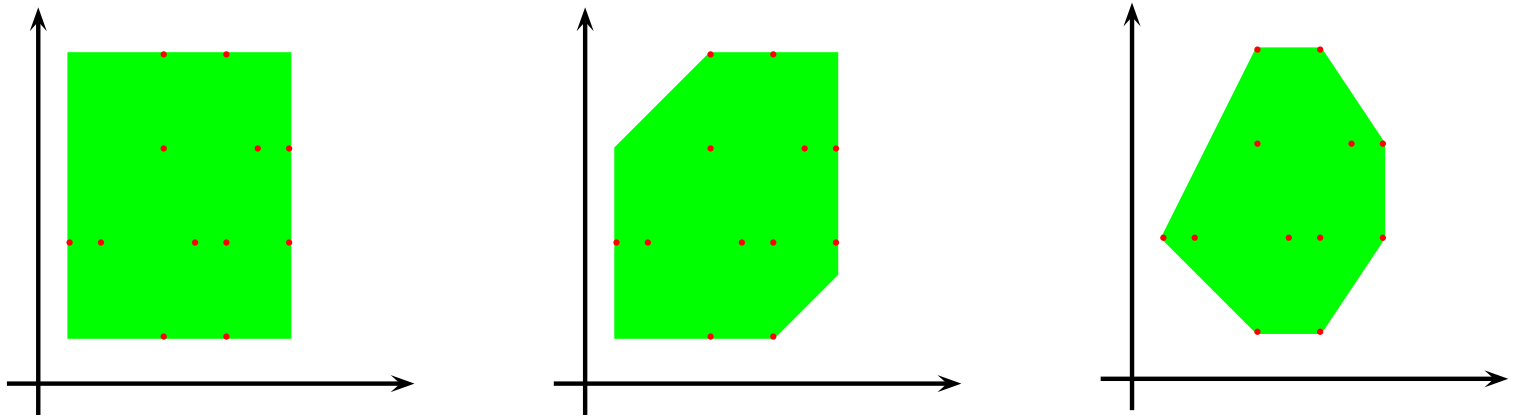
Polyhedron

A fully-relational domain.

Very precise.

May be inefficient.

Intervals, Polyhedra, and Things in Between (III)



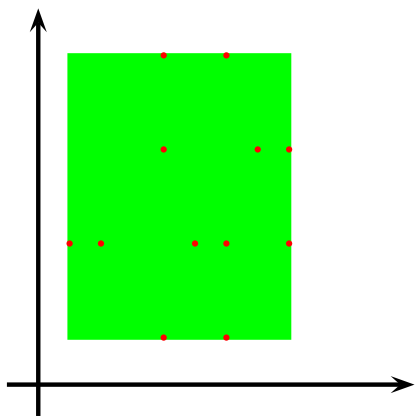
Interval | Bounded differences | Polyhedron

A weakly-relational domain.

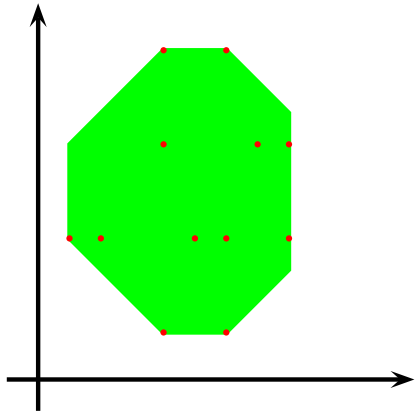
More precise than intervals.

More efficient than polyhedra.

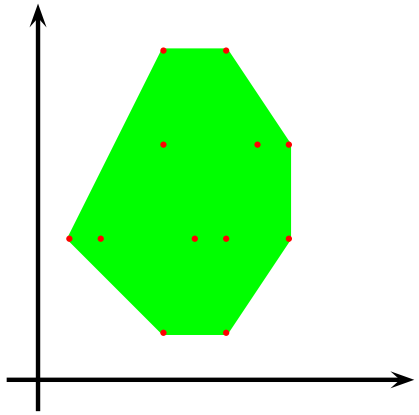
Intervals, Polyhedra, and Things in Between (iv)



Interval



Octagon



Polyhedron

A weakly-relational domain.

More precise than intervals.

More efficient than polyhedra.

OUR GOAL

- ➜ Weakly-relational numeric domains are useful for tuning the effi ciency/precision trade-off of static analyses.
- ➜ Many examples in the literature: bounded differences (Bagnara, PhD th., ’97; Shaham *et al.*, CC’00; Miné, PADO’01); octagons (Miné, WCRE’01); two variables per inequality (Simon *et al.*, LOPSTR’02); octahedra (Clarisó and Cortadella, SAS’04); template constraints (Sankaranarayanan *et al.*, VMCAI’05).
- ➜ Other domains not formally related to intervals/polyhedra: zone congruences (Miné, SAS’02); bounded quotients (Bagnara, PhD’97).

OUR GOAL

- ➜ Weakly-relational numeric domains are useful for tuning the effi ciency/precision trade-off of static analyses.
- ➜ Many examples in the literature: bounded differences (Bagnara, PhD th., ’97; Shaham *et al.*, CC’00; Miné, PADO’01); octagons (Miné, WCRE’01); two variables per inequality (Simon *et al.*, LOPSTR’02); octahedra (Clarisó and Cortadella, SAS’04); template constraints (Sankaranarayanan *et al.*, VMCAI’05).
- ➜ Other domains not formally related to intervals/polyhedra: zone congruences (Miné, SAS’02); bounded quotients (Bagnara, PhD’97).
- ➜ These domains have been typically defi ned to have a syntactic nature: different domain elements may encode the same concrete object.
  - ➜ Why? In order to avoid a convergence issue.
- ➜ We will provide a more natural solution to the convergence issue, arguing for the adoption of more abstract, semantic domains.

Plan of the Talk

① The problem on the simplest domain: Bounded Differences

② The (syntactic) solution adopted up to now

③ Argue for and propose an alternative (semantic) solution:
   ➜ Technical results already available from the literature

④ Instantiate the same approach on the Octagon domain:
   ➜ An effi cient algorithm removing redundancies
   ➜ A simpler and more effi cient strong closure algorithm

Plan of the Talk

① The problem on the simplest domain: Bounded Differences

② The (syntactic) solution adopted up to now

③ Argue for and propose an alternative (semantic) solution:
   ➜ Technical results already available from the literature

④ Instantiate the same approach on the Octagon domain:
   ➜ An effi cient algorithm removing redundancies
   ➜ A simpler and more effi cient strong closure algorithm

Bounded Differences a.k.a. Zones a.k.a. Potential Constraint Networks a.k.a. Difference-Bound Matrices

$\Rightarrow$ Bounded Difference Graphs (BDGs)

Bounded Differences a.k.a. Zones a.k.a. Potential Constraint Networks a.k.a. Difference-Bound Matrices

⇒ Bounded Difference Graphs (BDGs)

➜ They encode systems of constraints of the form $x - y \leq c$ and $\pm x \leq c$.

➜ For n variables, a weighted graph G with $n + 1$ nodes is used:

$$\begin{array}{rcccl} x_i - x_j \leq c & \iff & w(x_i, x_j) = c & \iff & M[i, j] = c \\ x_i - \mathbf{0} \leq c & \iff & w(x_i, \mathbf{0}) = c & \iff & M[i, 0] = c \\ \mathbf{0} - x_j \leq c & \iff & w(\mathbf{0}, x_j) = c & \iff & M[0, j] = c \end{array}$$

Bounded Differences a.k.a. Zones a.k.a. Potential Constraint Networks a.k.a. Difference-Bound Matrices

⇒ Bounded Difference Graphs (BDGs)

➜ They encode systems of constraints of the form $x - y \leq c$ and $\pm x \leq c$.

➜ For n variables, a weighted graph G with $n + 1$ nodes is used:

$$\begin{array}{rcccl} x_i - x_j \leq c & \iff & w(x_i, x_j) = c & \iff & M[i,j] = c \\ x_i - \mathbf{0} \leq c & \iff & w(x_i, \mathbf{0}) = c & \iff & M[i,0] = c \\ \mathbf{0} - x_j \leq c & \iff & w(\mathbf{0}, x_j) = c & \iff & M[0,j] = c \end{array}$$

➜ The shortest-path closure of the graph can be seen to implement closure by transitivity: it provides a canonical form for domain elements.

Shortest-Path Closure Algorithm

➜ Transitivity.

$$\frac{x - y \leq c \quad y - z \leq d}{x - z \leq c + d}$$

➜ The Floyd-Warshall algorithm for dense graphs: complexity is $\mathrm{O}(n^3)$.

```
for k := 0 to n
  for i := 0 to n
    for j := 0 to n
      M[i,j] := min(M[i,j], M[i,k] + M[k,j])
```

➜ For sparse graphs, Johnson's algorithm is (theoretically) better, achieving $\mathrm{O}(nm + n^2 \log n)$.

BDGs and Abstract Interpretation

- The first application of BDGs in the field of Abstract Interpretation is in (Shaham *et al.*, CC'00). A domain of (shortest-path) closed BDGs is considered and all the required abstract operators are specified.

BDGs and Abstract Interpretation

- The first application of BDGs in the field of Abstract Interpretation is in (Shaham *et al.*, CC'00). A domain of (shortest-path) closed BDGs is considered and all the required abstract operators are specified.
- The proposed widening for BDGs, which is reminiscent of the widenings defined on intervals and polyhedra, is defined by:

$$\frac{(x_i - x_j \leq c_1) \in G_1 \quad (x_i - x_j \leq c_2) \in G_2 \quad c_1 \geq c_2}{(x_i - x_j \leq c_1) \in G_1 \nabla G_2}$$

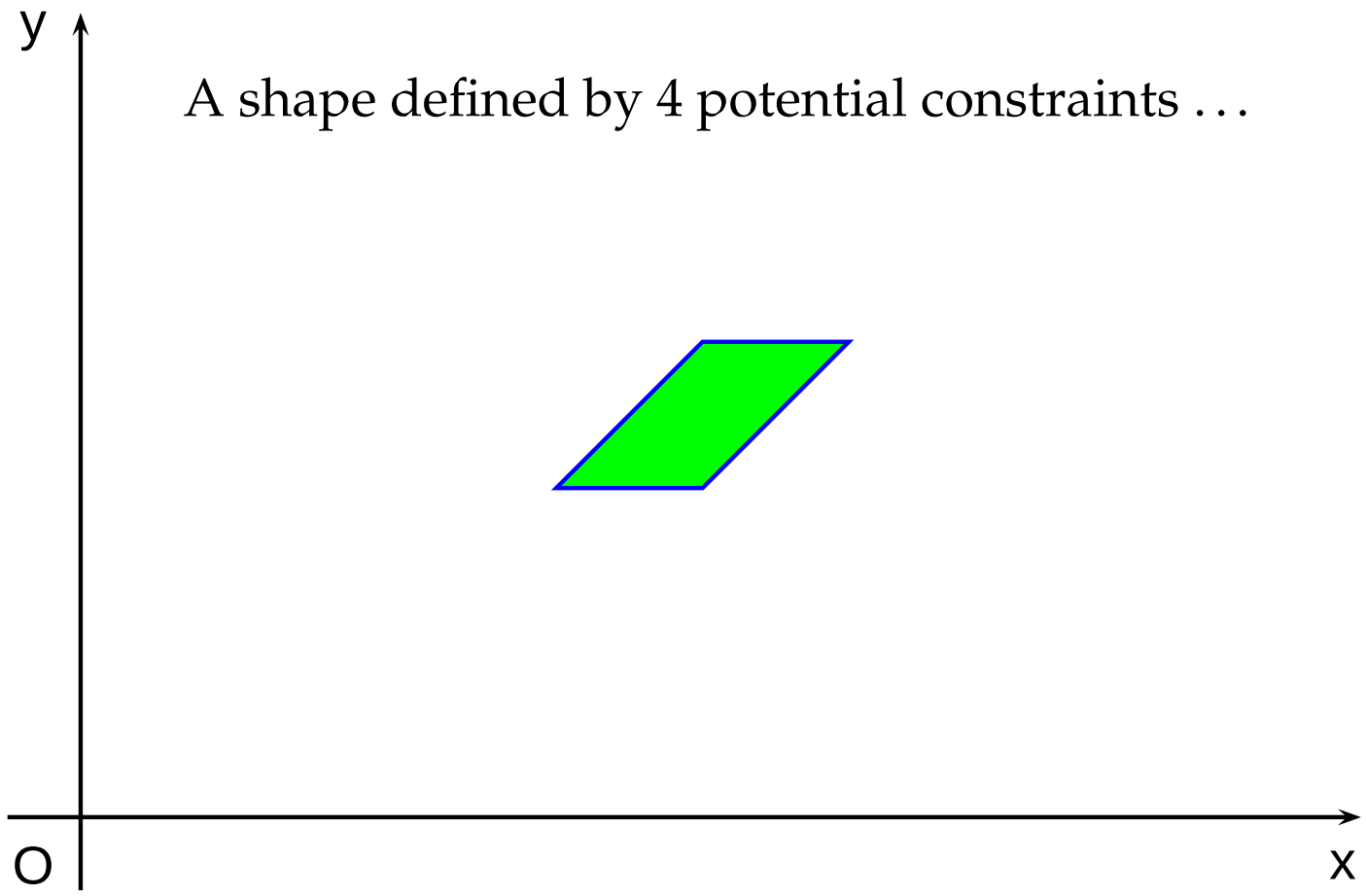
BDGs and Abstract Interpretation

- The first application of BDGs in the field of Abstract Interpretation is in (Shaham *et al.*, CC'00). A domain of (shortest-path) closed BDGs is considered and all the required abstract operators are specified.
- The proposed widening for BDGs, which is reminiscent of the widenings defined on intervals and polyhedra, is defined by:

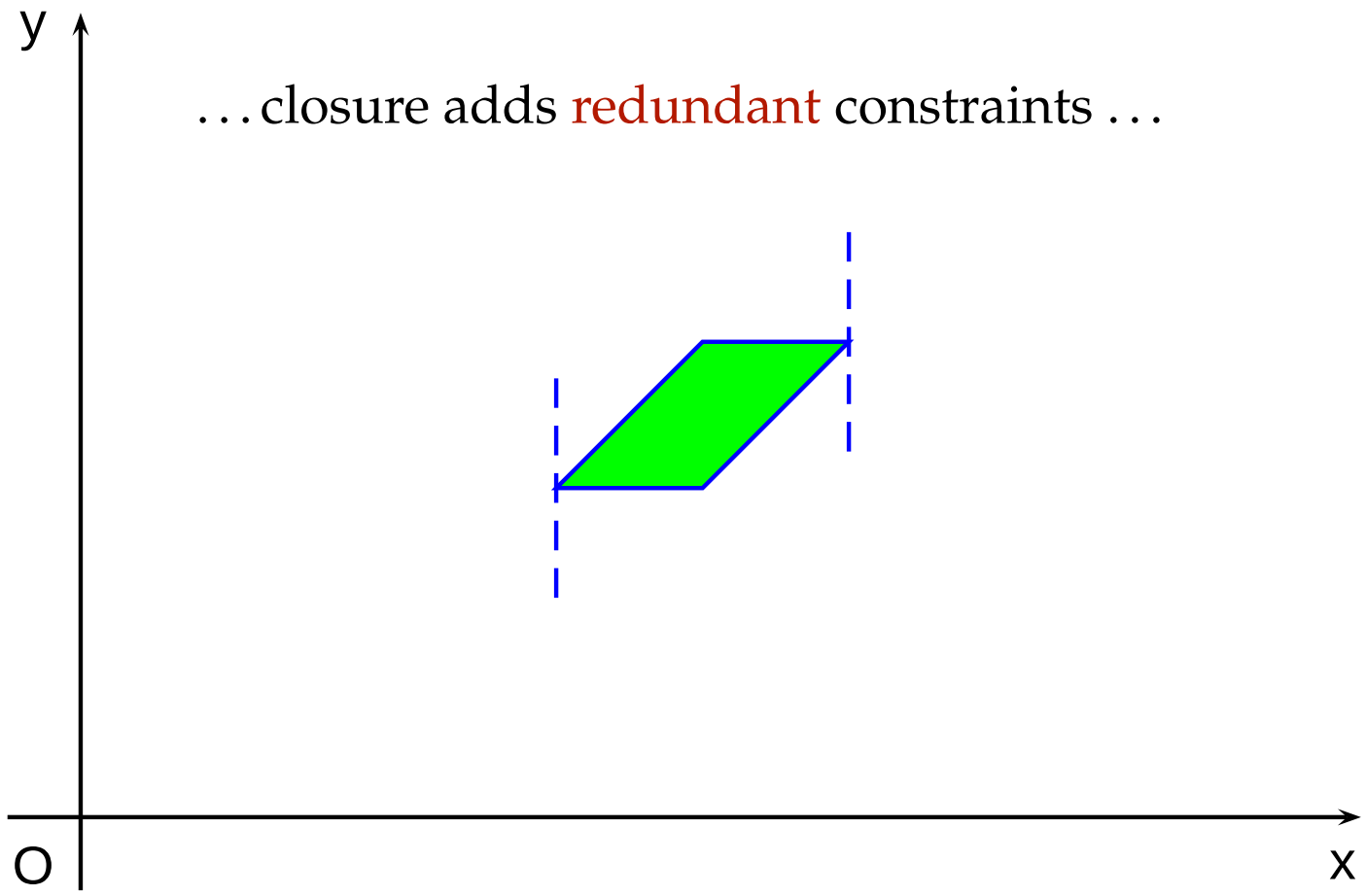
$$\frac{(x_i - x_j \leq c_1) \in G_1 \quad (x_i - x_j \leq c_2) \in G_2 \quad c_1 \geq c_2}{(x_i - x_j \leq c_1) \in G_1 \nabla G_2}$$

- Unfortunately, the operator above cannot ensure convergence, due to its interaction with closure.

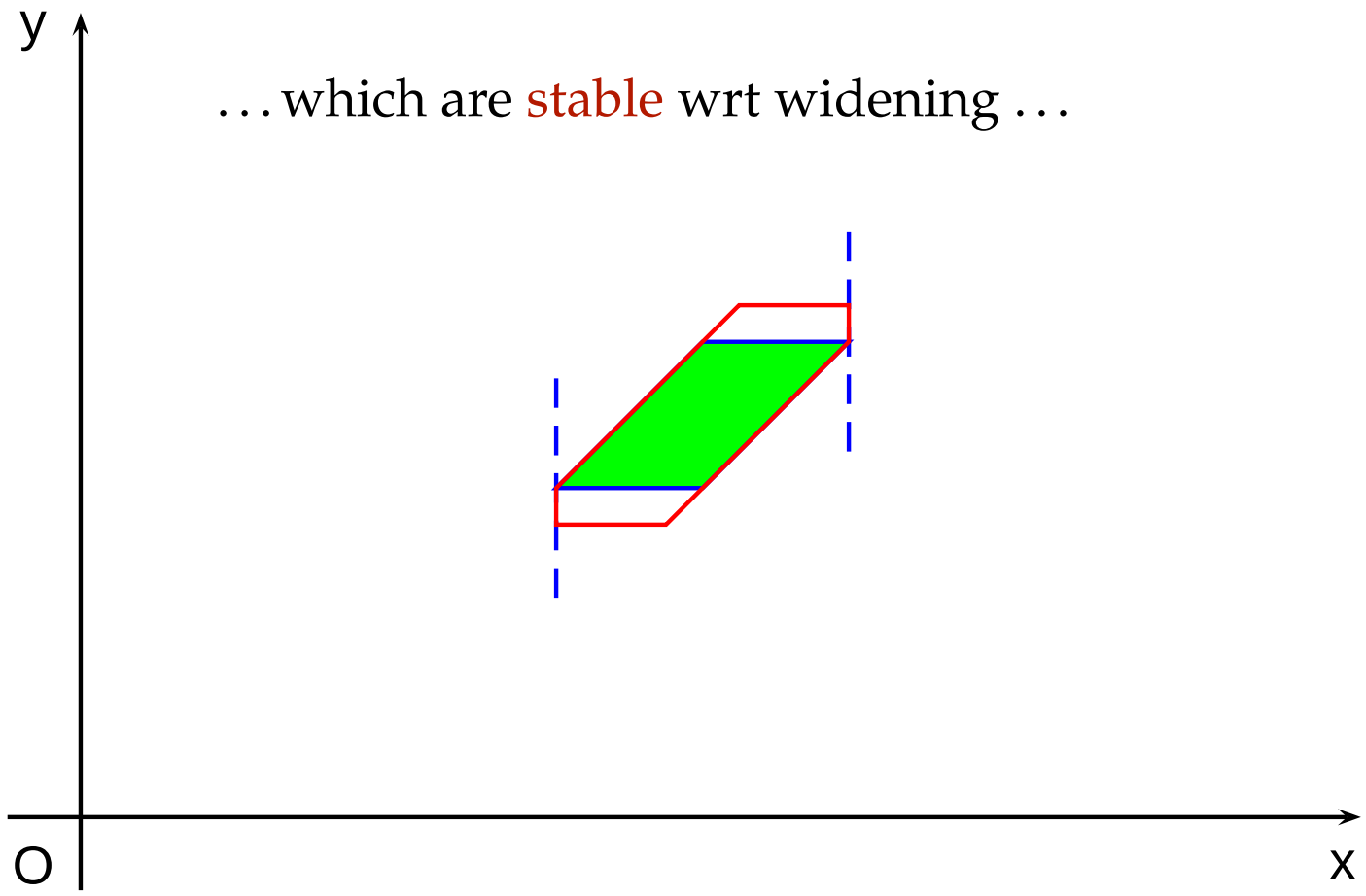
An Example of Divergence (I)



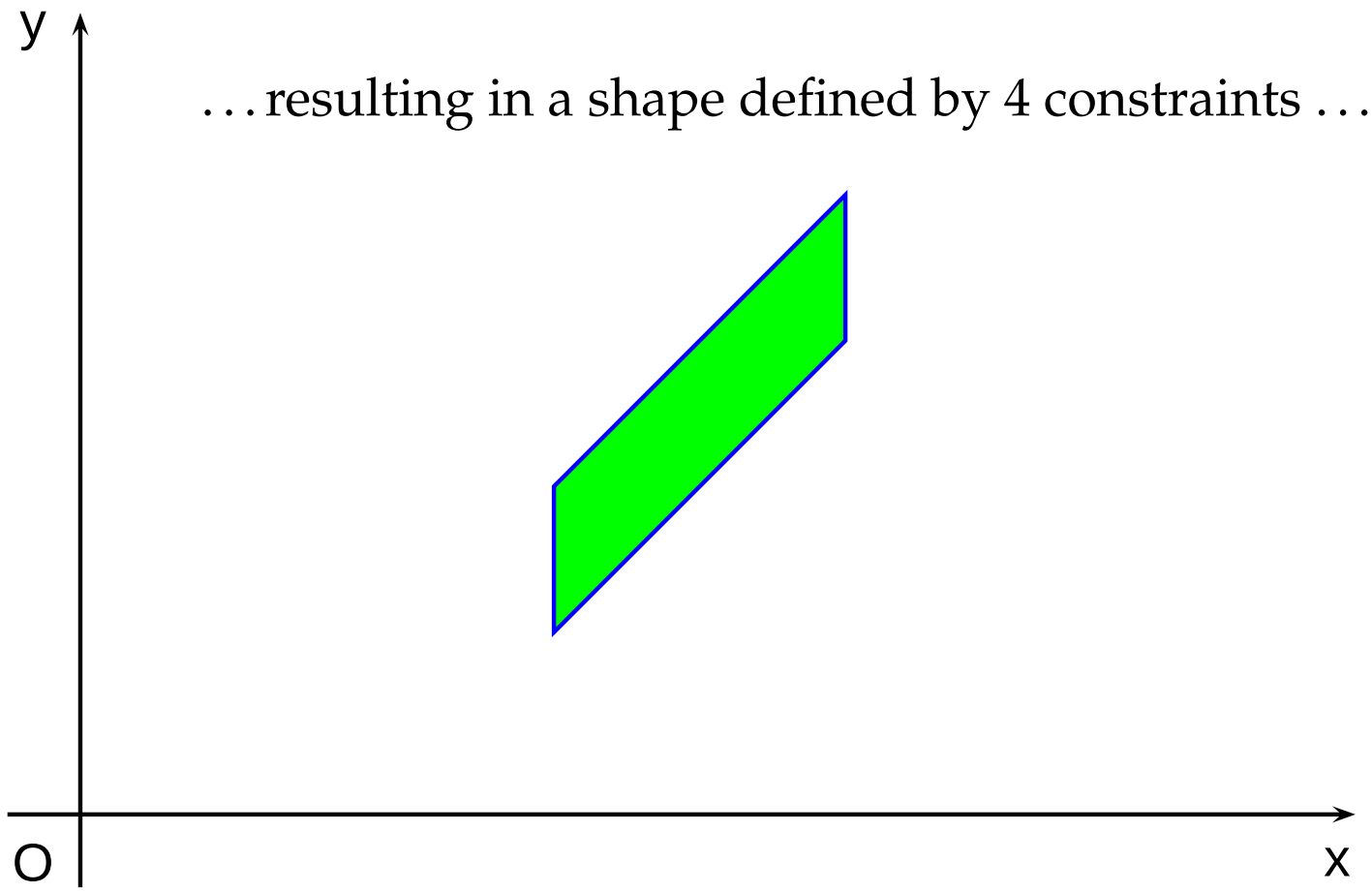
An Example of Divergence (II)



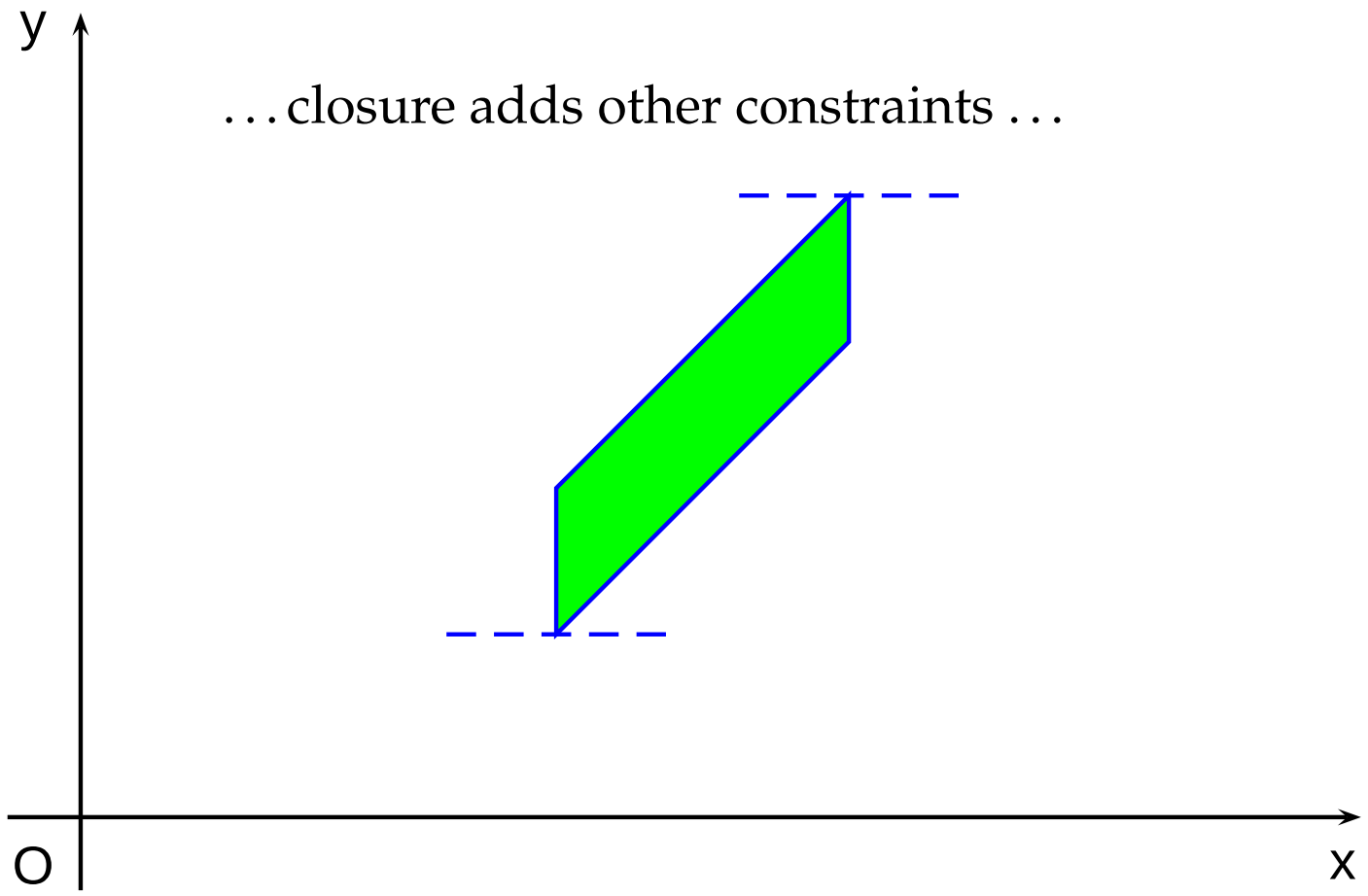
An Example of Divergence (III)



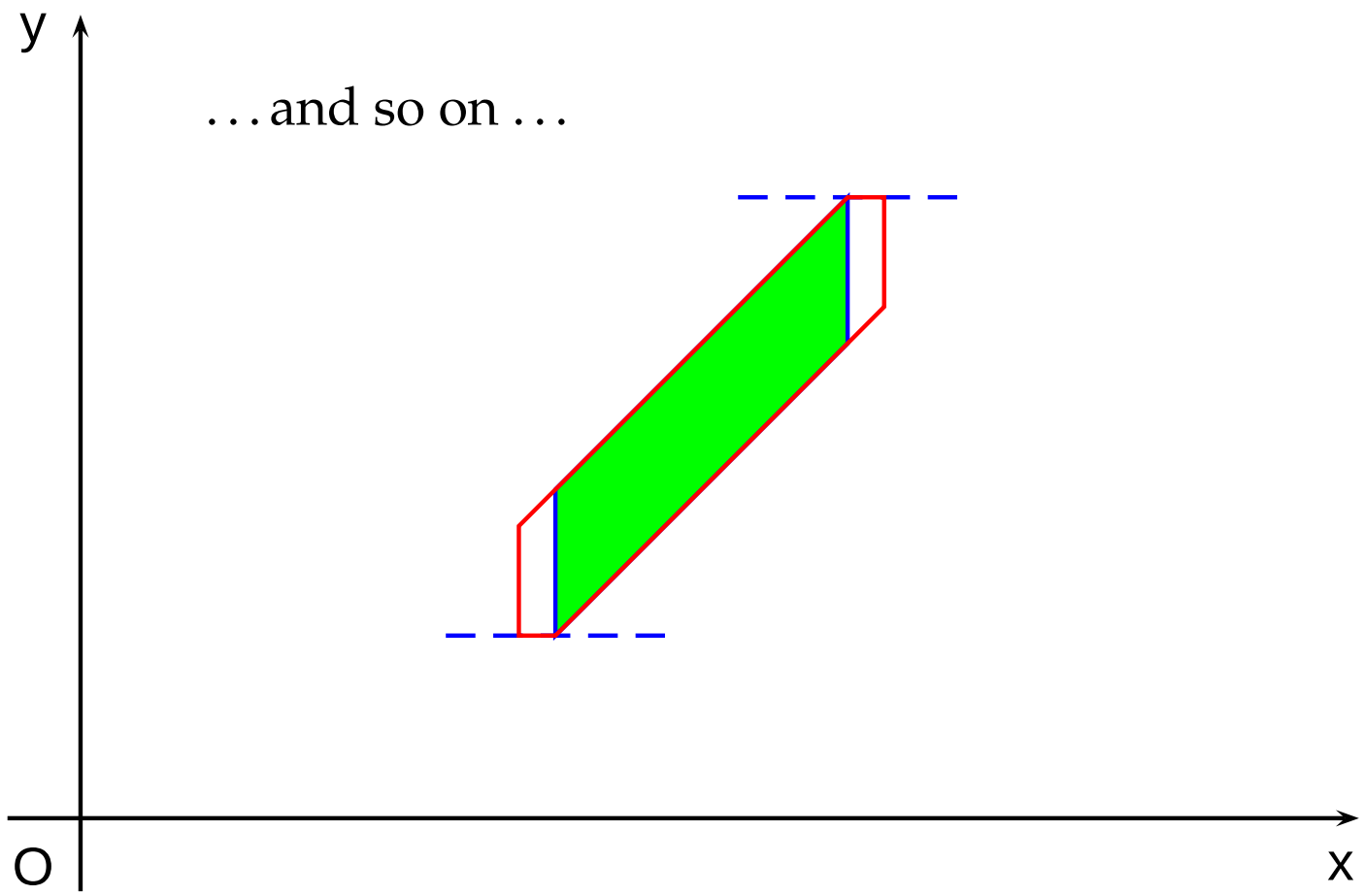
An Example of Divergence (iv)



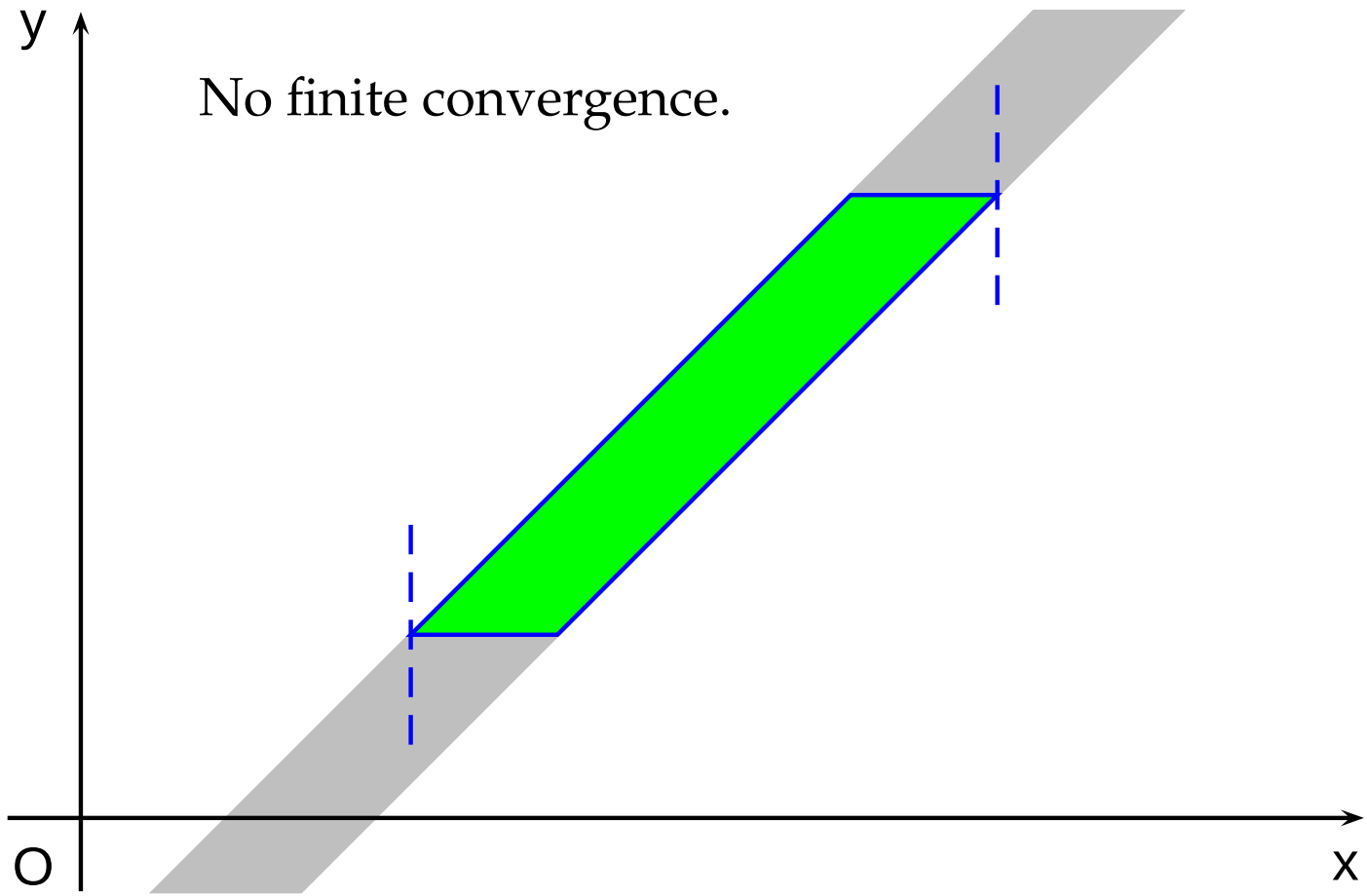
An Example of Divergence (v)



An Example of Divergence (vi)



An Example of Divergence (vii)



PLAN OF THE TALK

① The problem on the simplest domain: Bounded Differences

② The (syntactic) solution adopted up to now

③ Argue for and propose an alternative (semantic) solution:
   ➜ Technical results already available from the literature

④ Instantiate the same approach on the Octagon domain:
   ➜ An effi cient algorithm removing redundancies
   ➜ A simpler and more effi cient strong closure algorithm

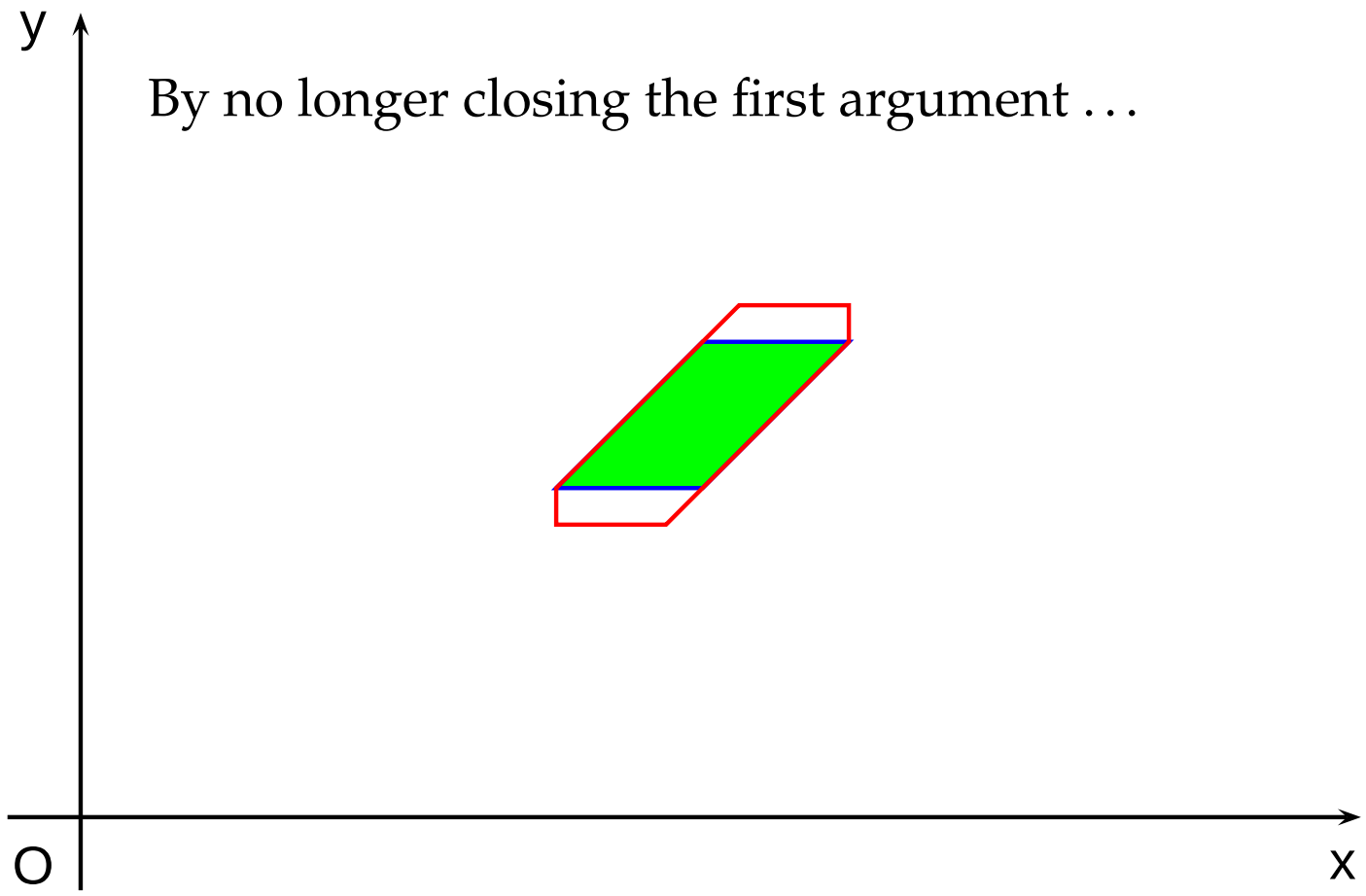
A Syntactic Solution

- The domain of (not necessarily shortest-path closed) BDGs is considered in (Miné, PADO'01). On this syntactic domain, different elements may encode the same geometric shape.
- Closure is applied only when needed (typically, to improve precision).
- It is a kernel operator, mapping a BDG into the most precise BDG encoding the same geometric shape.

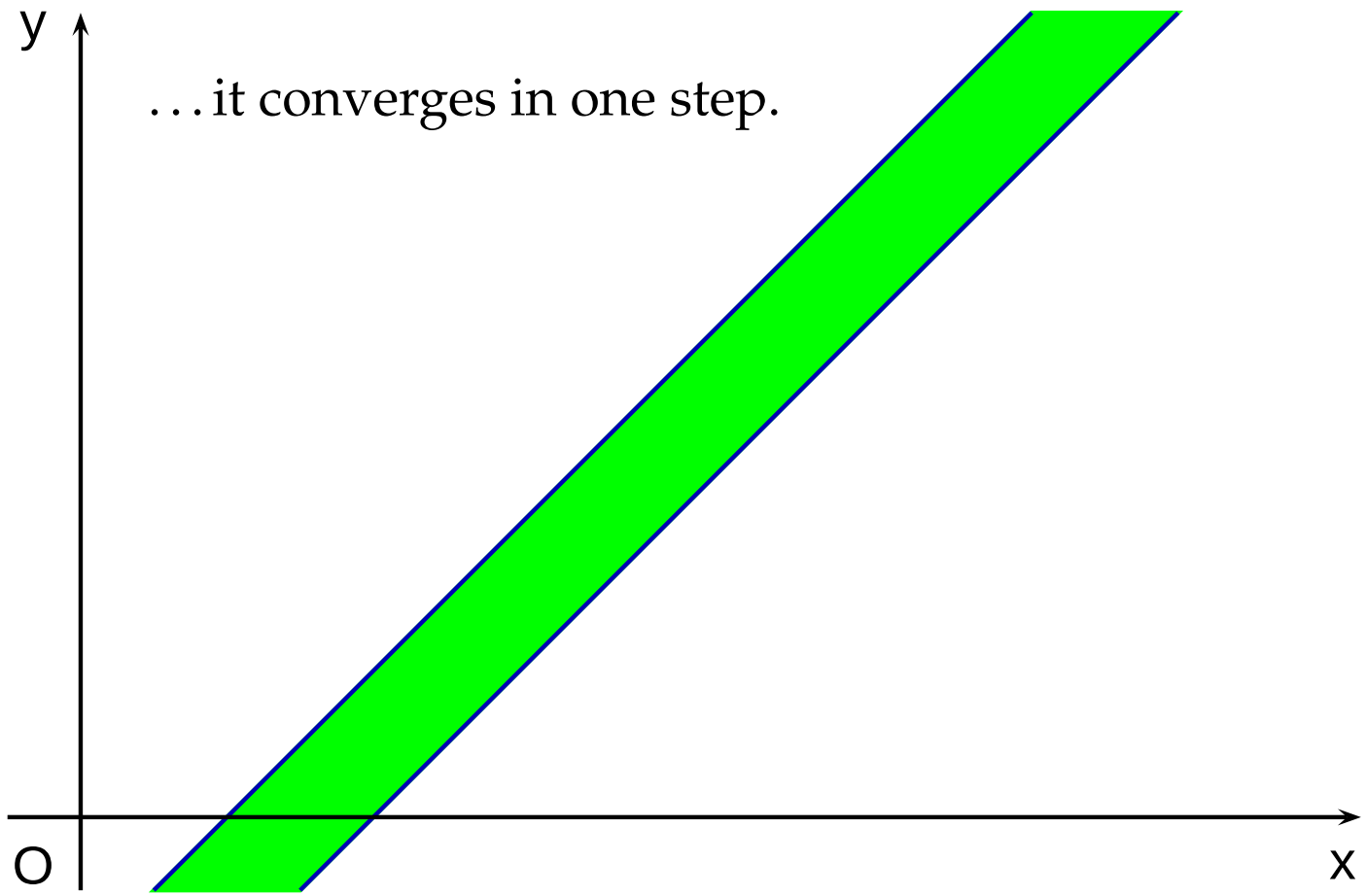
A Syntactic Solution

- The domain of (not necessarily shortest-path closed) BDGs is considered in (Miné, PADO'01). On this syntactic domain, different elements may encode the same geometric shape.
- Closure is applied only when needed (typically, to improve precision).
- It is a kernel operator, mapping a BDG into the most precise BDG encoding the same geometric shape.
- To solve the convergence problem faced in (Shaham *et al.*, CC'00), the fi rst argument of the widening is not closed.
- The discussions in (Miné, PADO'01, WCRE'01) make clear that the solution of the convergence problem was the one and only motivation for the adoption of this more concrete, syntactic domain.

DIVERGENCE IS NOW AVOIDED (I)



DIVERGENCE IS NOW AVOIDED (II)



Plan of the Talk

① The problem on the simplest domain: Bounded Differences

② The (syntactic) solution adopted up to now

③ Argue for and propose an alternative (semantic) solution:

➜ Technical results already available from the literature

④ Instantiate the same approach on the Octagon domain:

➜ An effi cient algorithm removing redundancies

➜ A simpler and more effi cient strong closure algorithm

Towards a Semantic Solution: Bounded Difference Shapes

➜ Resorting to syntactic domains has negative sides:

① less elegant formalization of operators and meaning functions;

② more complex user interfaces (need to explain implementation details, such as closure operators);

③ more complex application of domain refi nement operators; e.g., the fi nite powerset refi nement of (Bagnara et al., VMCAI'04).

Towards a Semantic Solution: Bounded Difference Shapes

- Resorting to syntactic domains has negative sides:
  1. less elegant formalization of operators and meaning functions;
  2. more complex user interfaces (need to explain implementation details, such as closure operators);
  3. more complex application of domain refi nement operators; e.g., the fi nite powerset refi nement of (Bagnara et al., VMCAI'04).
- An element of the abstract domain should be a geometric shape, rather than (any) one of its graph representations: we will call it a Bounded Difference Shape (BDS).

Towards a Semantic Solution: Bounded Difference Shapes

- Resorting to syntactic domains has negative sides:
  1. less elegant formalization of operators and meaning functions;
  2. more complex user interfaces (need to explain implementation details, such as closure operators);
  3. more complex application of domain refi nement operators; e.g., the fi nite powerset refi nement of (Bagnara et al., VMCAI'04).
- An element of the abstract domain should be a geometric shape, rather than (any) one of its graph representations: we will call it a Bounded Difference Shape (BDS).
- A BDS can also be seen as the equivalence class of all BDGs representing it. At the implementation level, one can freely switch between equivalent representations.
- On the semantic abstract domain of BDSs, shortest-path closure is the identity function.

No Convergence Problems for BDSs (I)

A result based on two simple observations:

① A BDS is a polyhedron.

② The set of BDSs (interpreted as polyhedra) is closed under the application of the standard widening of (Cousot and Halbwachs, POPL'78).

Therefore, no convergence problems can be incurred when applying the standard widening to an increasing sequence of BDSs.

No Convergence Problems for BDSs (II)

- The operator in (Shaham *et al.*, CC'00) is not the standard widening.
- The implementation of the standard widening requires that the first argument polyhedron is described by a non-redundant constraint system: in contrast, shortest-path closure typically adds a lot of redundant constraints.

No Convergence Problems for BDSs (II)

- The operator in (Shaham *et al.*, CC'00) is not the standard widening.
- The implementation of the standard widening requires that the first argument polyhedron is described by a non-redundant constraint system: in contrast, shortest-path closure typically adds a lot of redundant constraints.
- What is needed is a procedure for eliminating redundancies in a BDG: shortest-path reduction (Larsen *et al.*, RTSS'97).
- Reduction is just an implementation detail: on the domain of BDSs, it is the identity function.

SHORTEST-PATH REDUCTION FOR BDGS

- A (very sketchy) description of the algorithm in (Larsen *et al.*, RTSS'97):
  1. Compute the shortest-path closure of the graph;
  2. Partition the nodes into equivalence classes based on equality constraints;
  3. Split the graph into two subgraphs E and I:
     - subgraph E containing the arcs inside the equivalence classes (*equalities*);
     - subgraph I containing the arcs linking leaders of different equivalence classes (*inequalities*);
  4. Reduce subgraph E exploiting transitivity;
  5. Reduce subgraph I exploiting transitivity;
  6. Merge the results of steps ④ and ⑤.

Equivalence Classes for BDGs

- Equivalence classes encode equality constraints.
- $x_i \equiv x_j$ if they lie on the same cycle of weight 0.

EQUIVALENCE CLASSES FOR BDGS

- ➜ Equivalence classes encode equality constraints.
- ➜ $x_i \equiv x_j$ if they lie on the same cycle of weight 0.

Singular class:

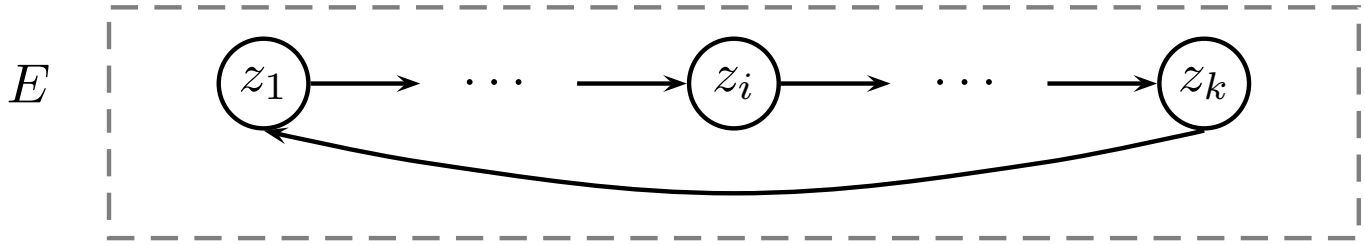
- ➜ Equivalence class E_0 containing the special variable $\mathbf{0}$ is singular: it encodes all the unary equality constraints (e.g., $x = c$).

Non-singular classes:

- ➜ Other equivalence classes can only encode binary equality constraints (e.g., $x - y = d$).

Shortest-Path Reduction of Equivalence Classes for BDGs

- Let $E = \{z_1, \ldots, z_k\}$ be (any) class, where $z_1 < \cdots < z_k$.
- Remove all arcs but the 0-cycle following the ordering between nodes.



Shortest-Path Reduction Rule For the Inequality Subgraph

① Reduction by transitivity.

$$\frac{x - y \leq c \quad y - z \leq d \quad x - z \leq e \quad c + d \leq e}{x - z \leq e \text{ is redundant}}$$

On the Precision of the Widening

- If used without any precaution, the standard widening on BDSs could provide imprecise results.
- For improving precision, (Miné, PADO'01, WCRE'01) suggest:
  1. to close the second argument;
  2. to close the first BDG G_0 of the upward iteration sequence.

On the Precision of the Widening

- If used without any precaution, the standard widening on BDSs could provide imprecise results.
- For improving precision, (Miné, PADO'01, WCRE'01) suggest:
  1. to close the second argument;
  2. to close the first BDG G_0 of the upward iteration sequence.
- Both improvements can be also obtained on the domain of BDSs:
  1. the first one can be applied as is;
  2. the second one can be subsumed by the 'widening up to' technique (Halbwachs *et al.*, SAS'94) or its variation called 'staged widening with thresholds' (Blanchet *et al.*, PLDI'03): one simply adds to the set of thresholds the constraints of the closure of G_0.

On the Precision of the Widening

- If used without any precaution, the standard widening on BDSs could provide imprecise results.
- For improving precision, (Miné, PADO'01, WCRE'01) suggest:
  1. to close the second argument;
  2. to close the first BDG G_0 of the upward iteration sequence.
- Both improvements can be also obtained on the domain of BDSs:
  1. the first one can be applied as is;
  2. the second one can be subsumed by the 'widening up to' technique (Halbwachs *et al.*, SAS'94) or its variation called 'staged widening with thresholds' (Blanchet *et al.*, PLDI'03): one simply adds to the set of thresholds the constraints of the closure of G_0.
- Further improvements can be achieved by applying any delay strategy or the widening framework of (Bagnara *et al.*, SAS'03).

What We Have Achieved

- A proper widening operator on the semantic abstract domain of BDSs. This can be made as precise as the widening on (syntactic) BDGs.

What We Have Achieved

- A proper widening operator on the semantic abstract domain of BDSs. This can be made as precise as the widening on (syntactic) BDGs.
- Both the syntactic and the semantic abstract domains are well-defi ned. The adoption of the semantic abstract domain solves all the issues highlighted before.

What We Have Achieved

- A proper widening operator on the semantic abstract domain of BDSs. This can be made as precise as the widening on (syntactic) BDGs.
- Both the syntactic and the semantic abstract domains are well-defi ned. The adoption of the semantic abstract domain solves all the issues highlighted before.
- Most of the other weakly-relational domains have a syntactic nature. For many of them, the semantic version can be defi ned.
- The key requirement is a reasonably effi cient procedure that removes redundancies from the considered constraint description.

Plan of the Talk

① The problem on the simplest domain: Bounded Differences

② The (syntactic) solution adopted up to now

③ Argue for and propose an alternative (semantic) solution:

➜ Technical results already available from the literature

④ Instantiate the same approach on the Octagon domain:

➜ An effi cient algorithm removing redundancies

➜ A simpler and more effi cient strong closure algorithm

OCTAGONS AS OCTAGONAL GRAPHS

➜ Octagons are defi ned by constraints of the form $\pm x \pm y \leq c$ and $\pm x \leq c$.

Octagons As Octagonal Graphs

➜ Octagons are defi ned by constraints of the form $\pm x \pm y \leq c$ and $\pm x \leq c$.

➜ Octagons can be implemented using BDGs. Each variable x is split into two forms: $x^+ \equiv x$ and $x^- \equiv -x$.

Difference constraint	$x - y \leq c$	becomes	$x^+ - y^+ \leq c$;
Sum constraint	$x + y \leq c$	becomes	$x^+ - y^- \leq c$;
Unary constraint	$x \leq c$	becomes	$x^+ - x^- \leq 2c$;
Unary constraint	$-x \leq c$	becomes	$x^- - x^+ \leq 2c$.

➜ An octagon is thus encoded into a BDG having $2n$ nodes.

Octagons As Octagonal Graphs

➜ Octagons are defi ned by constraints of the form $\pm x \pm y \leq c$ and $\pm x \leq c$.

➜ Octagons can be implemented using BDGs. Each variable x is split into two forms: $x^+ \equiv x$ and $x^- \equiv -x$.

Difference constraint	$x - y \leq c$	becomes	$x^+ - y^+ \leq c$;
Sum constraint	$x + y \leq c$	becomes	$x^+ - y^- \leq c$;
Unary constraint	$x \leq c$	becomes	$x^+ - x^- \leq 2c$;
Unary constraint	$-x \leq c$	becomes	$x^- - x^+ \leq 2c$.

➜ An octagon is thus encoded into a BDG having $2n$ nodes.

➜ Octagonal Graphs are coherent BDGs:

$$\frac{x^+ - y^+ \leq c}{y^- - x^- \leq c} \qquad \frac{x^+ - y^- \leq c}{y^+ - x^- \leq c} \qquad \frac{x^- - y^+ \leq c}{y^- - x^+ \leq c} \qquad \frac{x^- - y^- \leq c}{y^+ - x^+ \leq c}$$

STRONG CLOSURE

- A strong closure procedure is defined (Miné, WCRE'01) that takes into account, besides transitivity and coherence, also the following inference rule:
  - Strong coherence.

$$\frac{x \leq c \quad y \leq d}{x + y \leq c + d} \qquad \frac{x^+ - x^- \leq 2c \quad y^+ - y^- \leq 2d}{x^+ - y^- \leq c + d}$$

Miné's Strong Closure Algorithm

```
for k := 1 to n
begin /* Modified Floyd-Warshall: n steps */
  for i := 1 to 2*n
    for j := 1 to 2*n
      M[i,j] := min(M[i,j],
                    M[i,k] + M[k,j], M[i,c(k)] + M[c(k),j],
                    M[i,k] + M[k,c(k)] + M[c(k),j],
                    M[i,c(k)] + M[c(k),k] + M[k,j])
  /* Strong coherence: n steps */
  for i := 1 to 2*n
    for j := 1 to 2*n
      M[i,j] := min(M[i,j], (M[i,c(i)]+M[c(j),j])/2)
end
```

Strong Closure and Widening

- Again, widening and strong closure interact negatively.
- (Miné, WCRE'01) adopts the syntactic domain of octagonal graphs.
- The first argument of the widening should not be strongly closed.

Strong Closure and Widening

- Again, widening and strong closure interact negatively.
- (Miné, WCRE'01) adopts the syntactic domain of octagonal graphs.
- The first argument of the widening should not be strongly closed.

Strong Reduction $\Longrightarrow$ Octagonal Shapes

- The semantic domain of Octagonal Shapes can be adopted (together with the standard widening) provided we define a strong reduction procedure for octagonal graphs.

Plan of the Talk

① The problem on the simplest domain: Bounded Differences

② The (syntactic) solution adopted up to now

③ Argue for and propose an alternative (semantic) solution:

➜ Technical results already available from the literature

④ Instantiate the same approach on the Octagon domain:

➜ An effi cient algorithm removing redundancies

➜ A simpler and more effi cient strong closure algorithm

STRONG REDUCTION FOR OCTAGONAL GRAPHS

➜ A (very sketchy) description of the new algorithm:

① Compute the strong closure of the graph;

② Partition the nodes into equivalence classes based on equality constraints;

③ Split the graph into two subgraphs E and I:
- ➜ subgraph E containing the arcs inside the equivalence classes (*equalities*);
- ➜ subgraph I containing the arcs linking leaders of different equivalence classes (*inequalities*);

④ Reduce subgraph E exploiting transitivity;

⑤ Reduce subgraph I exploiting transitivity, strong coherence and singularity;

⑥ Merge the results of steps ④ and ⑤.

Equivalence Classes for Octagonal Graphs

Singular class:

- ➜ There no longer is the special variable $\mathbf{0}$.
- ➜ There may still be a singular equivalence class E_0: the only class containing both the positive x^+ and the negative x^- form of a variable.
- ➜ The singular class still encodes the set of unary equality constraints.

Non-singular classes:

- ➜ Non-singular classes still encode binary equality constraints.

Equivalence Classes for Octagonal Graphs

Singular class:

- ➜ There no longer is the special variable $\mathbf{0}$.
- ➜ There may still be a singular equivalence class E_0: the only class containing both the positive x^+ and the negative x^- form of a variable.
- ➜ The singular class still encodes the set of unary equality constraints.

Non-singular classes:

- ➜ Non-singular classes still encode binary equality constraints.

Strong Reduction of Equivalence Classes:

- ➜ Computed (almost) as before.

Strong Reduction Rules For the Inequality Subgraph

① Reduction by transitivity.

$$\frac{x - y \leq c \quad y - z \leq d \quad x - z \leq e \quad c + d \leq e}{x - z \leq e \text{ is redundant}}$$

Strong Reduction Rules For the Inequality Subgraph

① Reduction by transitivity.

$$\frac{x - y \leq c \quad y - z \leq d \quad x - z \leq e \quad c + d \leq e}{x - z \leq e \text{ is redundant}}$$

② Reduction by strong coherence.

$$\frac{x \leq c \quad y \leq d \quad x + y \leq e \quad c + d \leq e}{x + y \leq e \text{ is redundant}}$$

③ Reduction by singularity.

$$\frac{\pm x \pm y \leq c \quad x \in E_0}{\pm x \pm y \leq c \text{ is redundant}}$$

Plan of the Talk

① The problem on the simplest domain: Bounded Differences

② The (syntactic) solution adopted up to now

③ Argue for and propose an alternative (semantic) solution:

➜ Technical results already available from the literature

④ Instantiate the same approach on the Octagon domain:

➜ An effi cient algorithm removing redundancies

➜ A simpler and more effi cient strong closure algorithm

Miné's Strong Closure Algorithm

➜ Transitivity and strong coherence are interleaved `n` times.

```
for k := 1 to n
begin /* Modified Floyd-Warshall: n steps */
  for i := 1 to 2*n
    for j := 1 to 2*n
      M[i,j] := min(M[i,j],
                    M[i,k] + M[k,j], M[i,c(k)] + M[c(k),j],
                    M[i,k] + M[k,c(k)] + M[c(k),j],
                    M[i,c(k)] + M[c(k),k] + M[k,j])
  /* Strong coherence: n steps */
  for i := 1 to 2*n
    for j := 1 to 2*n
      M[i,j] := min(M[i,j], (M[i,c(i)]+M[c(j),j])/2)
end
```

An Improved Strong Closure Algorithm

➜ A shortest-path closed octagonal graph can be strongly closed by means of a single strong coherence step.

```
/* Classical Floyd-Warshall: 2n steps */
for k := 1 to 2*n
  for i := 1 to 2*n
    for j := 1 to 2*n
      M[i,j] := min(M[i,j], M[i,k] + M[k,j])

/* Strong coherence: 1 step */
for i := 1 to 2*n
  for j := 1 to 2*n
    M[i,j] := min(M[i,j], (M[i,c(i)] + M[c(j),j])/2)
```

EFFICIENCY COMPARISON (I)

➜ On the theoretical side, for sparse octagonal graphs, the complexity can be reduced from $\mathrm{O}(n^3)$ to $\mathrm{O}(nm + n^2 \log n)$ by computing shortest-path closure using Johnson's algorithm.

EFFICIENCY COMPARISON (I)

- On the theoretical side, for sparse octagonal graphs, the complexity can be reduced from $\mathrm{O}(n^3)$ to $\mathrm{O}(nm + n^2 \log n)$ by computing shortest-path closure using Johnson's algorithm.
- On the more practical side, even for dense octagonal graphs, we obtain a sensible improvement in the number of additions and comparisons between coeffi cients.
- When strongly closing a consistent octagonal graph:

	Full closure	Incremental closure
oct-lib-0.9.6	$20n^3 + 24n^2$	$68n^2 + 24n$
New algo	$16n^3 + 4n^2 + 4n$	$52n^2 - 44n$

EFFICIENCY COMPARISON (II)

➜ Speed-up:

	oct-lib-0.9.6 / New algo	
Dim. n	Full closure	Incremental closure
2	1.68	2.35
4	**1.51**	**1.76**
10	1.36	1.47
25	1.30	1.37
50	1.27	1.34
$n \rightarrow \infty$	1.25	1.31

➜ Measuring the number of additions and comparisons. Timing experiments confi rm the theoretical speed-up.

Work In Progress: Tight Closure Algorithm for Octagons on $\mathbb{Z}$

➜ A shortest-path closed octagonal graph can be tightly closed by means of a single tight coherence step. From $\mathrm{O}(n^4)$ to $\mathrm{O}(n^3)$.

```
/* Classical Floyd-Warshall: 2n steps */
for k := 1 to 2*n
  for i := 1 to 2*n
    for j := 1 to 2*n
      M[i,j] := min(M[i,j], M[i,k] + M[k,j])

/* Tight coherence: 1 step */
for i := 1 to 2*n
  for j := 1 to 2*n
    M[i,j] := min(M[i,j], M[i,c(i)] Div 2 + M[c(j),j] Div 2)
```

CONCLUSION

- ➜ Several syntactic weakly-relational numeric domains can be further abstracted into their semantic counterparts.
- ➜ All the rest being equal, semantic domains should be preferred to their syntactic counterparts.
- ➜ We have provided a new strong reduction procedure, as well as an improved strong closure procedure, for the Octagon domain. Formal proofs are available at `http://www.cs.unipr.it/ppl/`.

CONCLUSION

- ➜ Several syntactic weakly-relational numeric domains can be further abstracted into their semantic counterparts.
- ➜ All the rest being equal, semantic domains should be preferred to their syntactic counterparts.
- ➜ We have provided a new strong reduction procedure, as well as an improved strong closure procedure, for the Octagon domain. Formal proofs are available at `http://www.cs.unipr.it/ppl/`.
- ➜ Implementation work is in progress. The domains of Bounded Difference Shapes and Octagonal Shapes will be made available in future releases of the Parma Polyhedra Library (GPL).
- ➜ Code, documentation and useful links about the Parma Polyhedra Library are available at `http://www.cs.unipr.it/ppl/`.

On Floating-Point Computations

- The stated results are valid if the implementation uses unbounded precision rationals as coeffi cients.
- If bounded precision floating-point coeffi cients are used, the shortest-path/strong closure procedure will not provide a normal form. Neither shortest-path/strong reduction will provide a true (i.e., semantic) minimization procedure.
- Hence, a semantic abstract domain is diffi cult to obtain when using floating-point computations only.

On Floating-Point Computations

- The stated results are valid if the implementation uses unbounded precision rationals as coeffi cients.
- If bounded precision floating-point coeffi cients are used, the shortest-path/strong closure procedure will not provide a normal form. Neither shortest-path/strong reduction will provide a true (i.e., semantic) minimization procedure.
- Hence, a semantic abstract domain is diffi cult to obtain when using floating-point computations only.
- Shortest-path/strong reduction will still be able to remove most, even though not all, of the redundancies in the syntactic domain, therefore mitigating most of its negative sides.
- The standard widening using shortest-path/strong reduction is still a proper widening operator.