Policy Iterations as Traditional Abstract Domains

Pierre Roux [1,2] **Pierre-Loïc Garoche** [1]

Synchron 2012

November 19

[1]ONERA, Toulouse, France

[2]ISAE, Toulouse, France

A Static Analysis Talk

This talk is about **static analysis** and may seem weakly related to synchronous programming. However:

- we analyze control command programs, typically written in the synchronous paradigm;
- we present an implementation analyzing Lustre code.

A Toy Imperative Language

Definition

$$
\begin{aligned}
stm ::= {} & v \mathrel{\mathbf{:=}} expr \mid v \mathrel{\mathbf{:=}} \mathbf{?(}r, r\mathbf{)} \mid stm;\ stm \\
& \mid \mathbf{if}\ expr \leq 0\ \mathbf{then}\ stm\ \mathbf{else}\ stm\ \mathbf{fi} \\
& \mid \mathbf{while}\ expr \leq 0\ \mathbf{do}\ stm\ \mathbf{od}
\end{aligned}
$$

$$
\begin{aligned}
expr ::= {} & v \mid r \\
& \mid expr + expr \mid expr - expr \mid expr \times expr \mid expr / expr
\end{aligned}
$$

$v \in \mathbb{V}$, $r \in \mathbb{R}$.

First step of analysis is to compile synchronous programs to this kind of language.

A Toy Imperative Language, Example

Example

```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
    in := ?(0, 1);
    if 0.9 − in ≤ 0 then
        x := 10×in − 9;
        y := 10×in − 9
    else
        t := x;
        x := 0.2×t − 0.7×y + 0.5×in;
        y := 0.7×t + 0.2×y + 0.5×in fi od
```

A Toy Imperative Language, Semantics

Definition (Collecting Semantics)

Expressions : $[\![e]\!] : (\mathbb{V} \to \mathbb{R}) \to \mathbb{R}$

$$
\begin{aligned}
[\![v]\!](\rho) &= \rho(v) \\
[\![r]\!](\rho) &= r \\
[\![e_1 \diamond e_2]\!](\rho) &= [\![e_1]\!](\rho) \diamond [\![e_2]\!](\rho) \text{ for } \diamond \in \{\,+, -, \times, /\,\}
\end{aligned}
$$

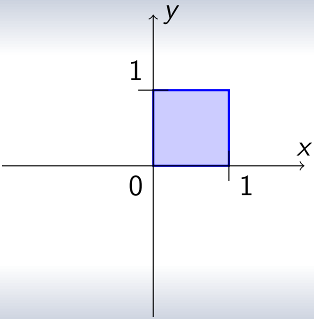
Statements : $[\![s]\!] : 2^{(\mathbb{V} \to \mathbb{R})} \to 2^{(\mathbb{V} \to \mathbb{R})}$

$$
\begin{aligned}
&[\![v := e]\!](R) = \{\rho[v \leftarrow [\![e]\!](\rho)] \mid \rho \in R\} \\
&[\![v := ?(\mathrm{r}_1, \mathrm{r}_2)]\!](R) = \{\rho[v \leftarrow r] \mid \rho \in R, r \in \mathbb{R}, r_1 \leq r \leq r_2\} \\
&[\![e \bowtie 0]\!](R) = \{\rho \in R \mid [\![e]\!](\rho) \bowtie 0\} \text{ for } \bowtie \in \{\,>, \geq, <, \leq\,\} \\
&[\![s_1; s_2]\!](R) = [\![s_2]\!]([\![s_1]\!](R)) \\
&[\![\textbf{if } \mathrm{e} \leq 0 \textbf{ then } \mathrm{s}_1 \textbf{ else } \mathrm{s}_2 \textbf{ fi}]\!](R) = [\![s_1]\!]([\![e \leq 0]\!](R)) \cup [\![s_2]\!]([\![e > 0]\!](R)) \\
&[\![\textbf{while } \mathrm{e} \leq 0 \textbf{ do } \mathrm{s} \textbf{ od}]\!](R) = [\![e > 0]\!](\mathrm{lfp}\,(X \mapsto R \cup X \cup [\![s]\!]([\![e \leq 0]\!](X))))
\end{aligned}
$$

Kleene Iterations

With interval domain:

```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```

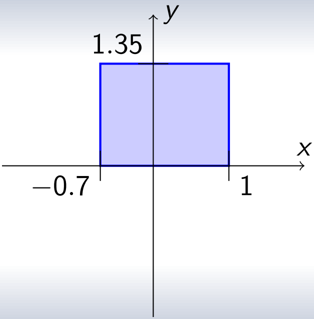


before entering the loop

Kleene Iterations

With interval domain:

```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```

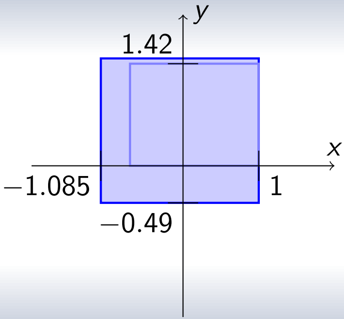


after a first iteration

Kleene Iterations

With interval domain:

```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```

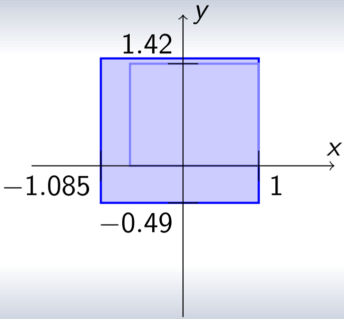


after a second iteration

Kleene Iterations

With interval domain:

```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



. . . does not converge

Widening

Basic Idea

Jump to ensure convergence in **finitely many** iterations.

Example (Simplest Widening)

Jump to $(-\infty, +\infty)$ as soon as bounds are not stable.

Widening

Basic Idea

Jump to ensure convergence in **finitely many** iterations.

Example (Simplest Widening)

Jump to $(-\infty, +\infty)$ as soon as bounds are not stable.

Leads to $x \in (-\infty, +\infty) \wedge y \in (-\infty, +\infty)$

Widening

Basic Idea

Jump to ensure convergence in **finitely many** iterations.

Example (Simplest Widening)

Jump to $(-\infty, +\infty)$ as soon as bounds are not stable.

Leads to $x \in (-\infty, +\infty) \wedge y \in (-\infty, +\infty)$: useless result.

Widening

Basic Idea

Jump to ensure convergence in **finitely many** iterations.

Example (Simplest Widening)

Jump to $(-\infty, +\infty)$ as soon as bounds are not stable.

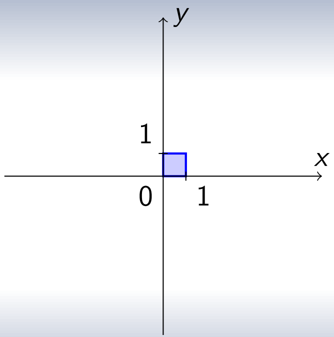
Leads to $x \in (-\infty, +\infty) \wedge y \in (-\infty, +\infty)$: useless result.

Example (Widening with Thresholds)

Stop on a finite number of thresholds T on the way to infinity.
For instance $T = \{-5, 5\}$.

Widening with Thresholds

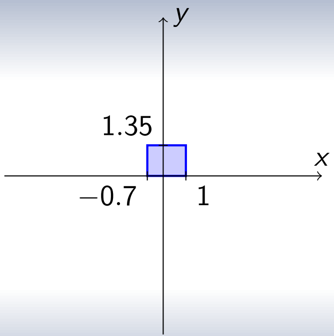
x **:=** **?(**0, 1**)**; y **:=** **?(**0, 1**)**;
while $-1 \le 0$ **do**
  in **:=** **?(**0, 1**)**;
  if 0.9 − in ≤ 0 **then**
    x **:=** 10×in − 9;
    y **:=** 10×in − 9
  else
    t **:=** x;
    x **:=** 0.2×t − 0.7×y + 0.5×in;
    y **:=** 0.7×t + 0.2×y + 0.5×in
  fi
od



before entering the loop

Widening with Thresholds

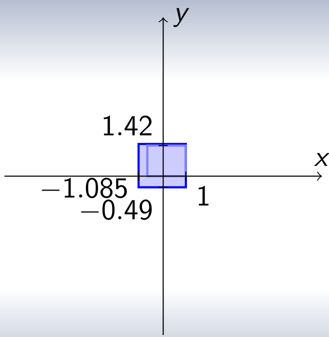
```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



after a first iteration

Widening with Thresholds

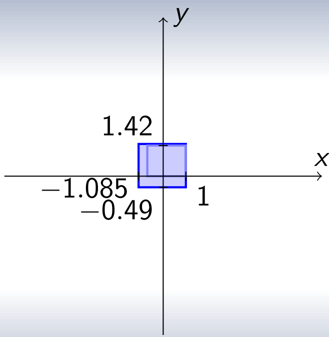
```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



after a second iteration

Widening with Thresholds

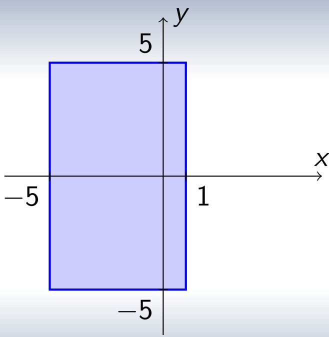
```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



not stable → widening

Widening with Thresholds

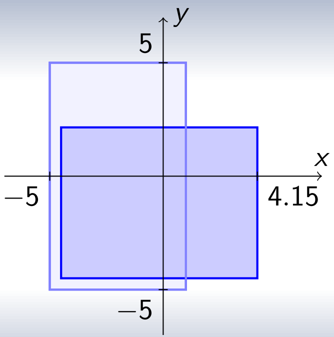
```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



after widening

Widening with Thresholds

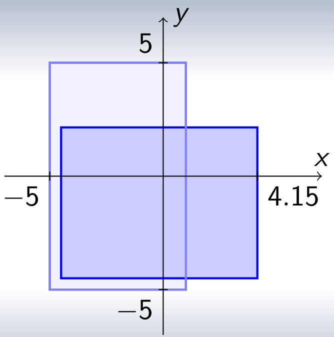
```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



after another iteration

Widening with Thresholds

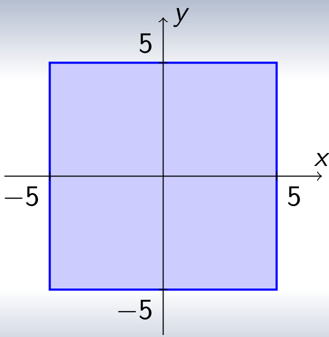
```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



not stable → widening

Widening with Thresholds

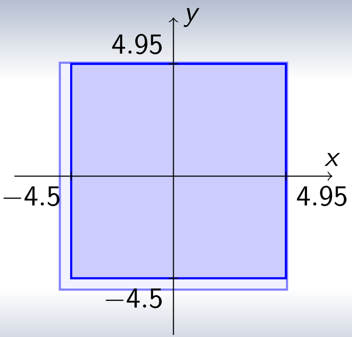
```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



after widening

Widening with Thresholds

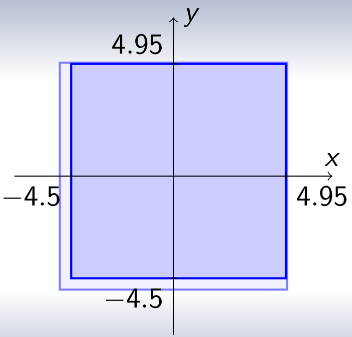
```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



after another iteration

Widening with Thresholds

```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



stable !

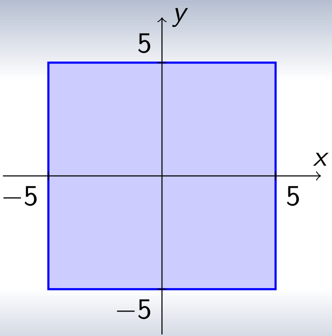
Descending Iterations with Narrowing

Perform a few descending iterations:

- allows to retrieve precision lost in widening;
- cannot always regain everything.

Descending Iterations with Narrowing

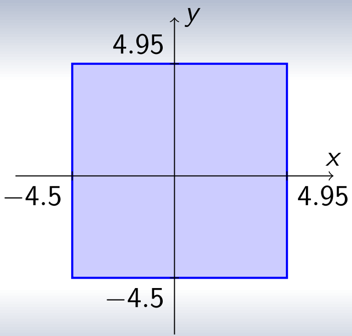
```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



fixpoint reached with widening

Descending Iterations with Narrowing

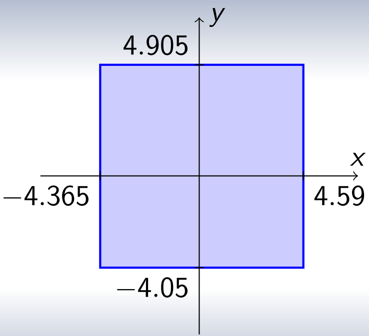
```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



after a descending iteration

Descending Iterations with Narrowing

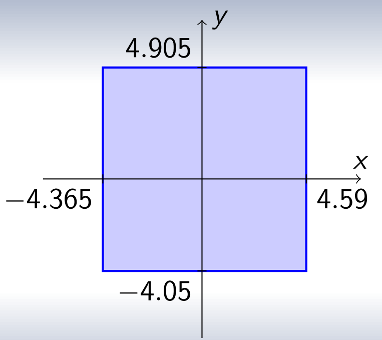
```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



after a second iteration

Descending Iterations with Narrowing

```
x := ?(0, 1); y := ?(0, 1);
while −1 ≤ 0 do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```



not converging, we stop here

Widening/Narrowing

- Often works well.
- But sometime incurs dramatic losses of precision.
- Lot of work to improve on this.
- Among which **policy iterations**.

Policy Iterations

Basic Idea

Use **numerical optimization tools** to compute bounds that are hard to guess for widening.

Policy Iterations

Basic Idea

Use **numerical optimization tools** to compute bounds that are hard to guess for widening.

Example

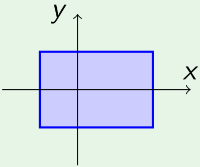
- linear programming
- semidefinite programming

Policy Iterations: Template Domains

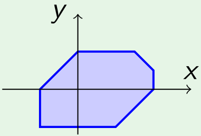
Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{ t_1, \ldots, t_n \}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n)$ $= \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

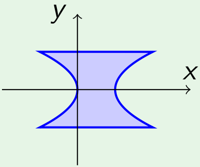
Example



intervals
$(\mathcal{T} = \{ x, -x, y, -y \})$
$(\beta = (2, 1, 1, 1))$



octagons
$(\mathcal{T} = \left\{ \begin{array}{c} x, -x, y, -y, \\ x+y, x-y, \\ -x+y, -x-y \end{array} \right\})$
$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



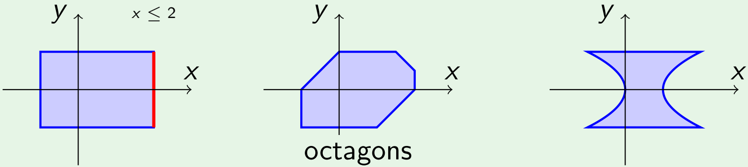
quadratic
$(\mathcal{T} = \{ y, -y, x - y^2, y^2 - x \})$
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{\, t_1, \ldots, t_n \,\}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n)$ $= \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \le b_1, \ldots, [\![t_n]\!](\rho) \le b_n\}$.

Example



intervals
$(\mathcal{T} = \{\, x, \, -x, y, \, -y \,\})$
$(\beta = (2, 1, 1, 1))$

octagons
$(\mathcal{T} = \left\{ \begin{array}{c} x, \, -x, y, \, -y, \\ x+y, x-y, \\ -x+y, \, -x-y \end{array} \right\})$
$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$

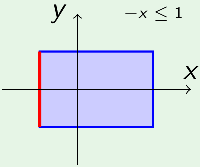
quadratic
$(\mathcal{T} = \{\, y, \, -y, x - y^2, y^2 - x \,\})$
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{ t_1, \ldots, t_n \}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n) = \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

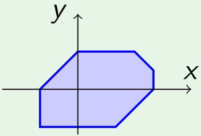
Example



intervals

$(\mathcal{T} = \{ x, \ -x, y, \ -y \})$

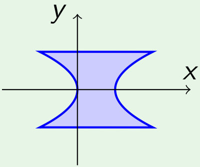
$(\beta = (2, 1, 1, 1))$



octagons

$$(\mathcal{T} = \left\{ \begin{array}{c} x, \ -x, y, \ -y, \\ x+y, x-y, \\ -x+y, \ -x-y \end{array} \right\})$$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{ y, \ -y, x-y^2, y^2-x \})$

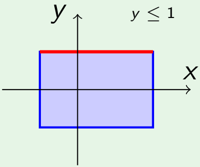
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{ t_1, \ldots, t_n \}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n)$ $= \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

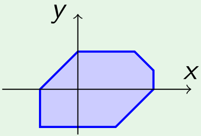
Example



intervals

$(\mathcal{T} = \{ x, -x, y, -y \})$

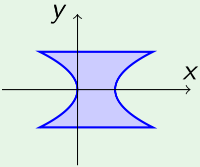
$(\beta = (2, 1, 1, 1))$



octagons

$(\mathcal{T} = \left\{ \begin{array}{c} x, -x, y, -y, \\ x+y, x-y, \\ -x+y, -x-y \end{array} \right\})$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{ y, -y, x - y^2, y^2 - x \})$

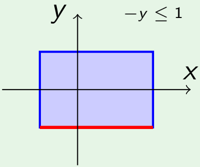
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{\, t_1, \ldots, t_n \,\}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n)$ $= \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

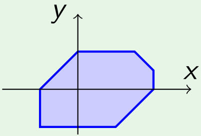
Example



intervals

$(\mathcal{T} = \{\, x, \, -x, y, \, -y \,\})$

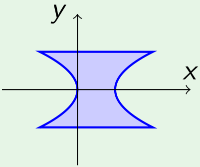
$(\beta = (2, 1, 1, 1))$



octagons

$(\mathcal{T} = \left\{ \begin{array}{c} x, \, -x, y, \, -y, \\ x+y, x-y, \\ -x+y, \, -x-y \end{array} \right\})$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{\, y, \, -y, x - y^2, y^2 - x \,\})$

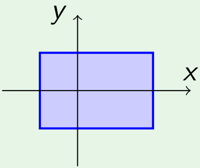
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{ t_1, \dots, t_n \}$, abstract values are tuples $\beta = (b_1, \dots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \dots, b_n) = \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \dots, [\![t_n]\!](\rho) \leq b_n\}$.

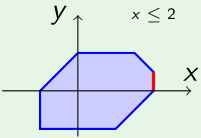
Example



intervals

$(\mathcal{T} = \{ x, -x, y, -y \})$

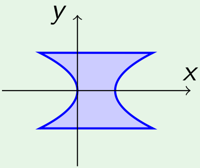
$(\beta = (2, 1, 1, 1))$



octagons

$(\mathcal{T} = \left\{ \begin{array}{c} x, -x, y, -y, \\ x + y, x - y, \\ -x + y, -x - y \end{array} \right\})$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{ y, -y, x - y^2, y^2 - x \})$

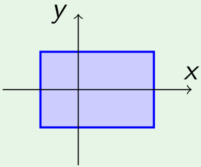
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{ t_1, \ldots, t_n \}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n)$ $= \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

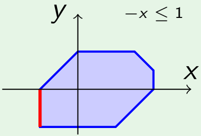
Example



intervals

$(\mathcal{T} = \{ x, -x, y, -y \})$

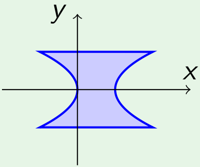
$(\beta = (2, 1, 1, 1))$



octagons

$(\mathcal{T} = \left\{ \begin{array}{c} x, -x, y, -y, \\ x+y, x-y, \\ -x+y, -x-y \end{array} \right\})$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{ y, -y, x - y^2, y^2 - x \})$

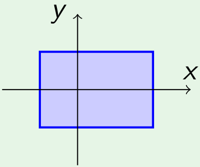
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{ t_1, \ldots, t_n \}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n)$ $= \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

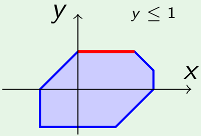
Example



intervals

$(\mathcal{T} = \{ x, -x, y, -y \})$

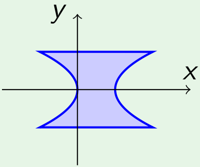
$(\beta = (2, 1, 1, 1))$



octagons

$(\mathcal{T} = \left\{ \begin{array}{c} x, -x, y, -y, \\ x+y, x-y, \\ -x+y, -x-y \end{array} \right\})$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{ y, -y, x - y^2, y^2 - x \})$

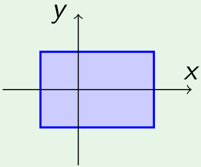
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{ t_1, \ldots, t_n \}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n)$ $= \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

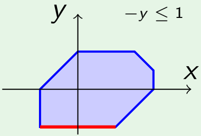
Example



intervals

$(\mathcal{T} = \{ x, -x, y, -y \})$

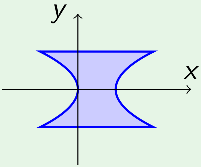
$(\beta = (2, 1, 1, 1))$



octagons

$$(\mathcal{T} = \left\{ \begin{array}{c} x, -x, y, -y, \\ x+y, x-y, \\ -x+y, -x-y \end{array} \right\})$$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{ y, -y, x - y^2, y^2 - x \})$

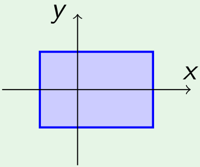
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{\, t_1, \ldots, t_n \,\}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n)$ $= \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

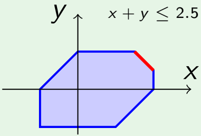
Example



intervals

$(\mathcal{T} = \{\, x, \; -x, y, \; -y \,\})$

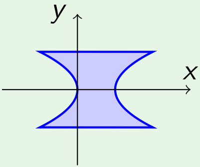
$(\beta = (2, 1, 1, 1))$



octagons

$(\mathcal{T} = \left\{ \begin{array}{c} x, \; -x, y, \; -y, \\ x + y, x - y, \\ -x + y, \; -x - y \end{array} \right\})$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{\, y, \; -y, x - y^2, y^2 - x \,\})$

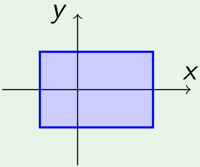
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{ t_1, \ldots, t_n \}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n) = \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \le b_1, \ldots, [\![t_n]\!](\rho) \le b_n\}$.

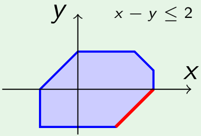
Example



intervals

$(\mathcal{T} = \{ x, -x, y, -y \})$

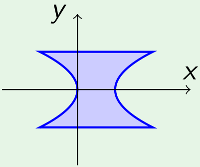
$(\beta = (2, 1, 1, 1))$



octagons

$(\mathcal{T} = \left\{ \begin{array}{c} x, -x, y, -y, \\ x+y, x-y, \\ -x+y, -x-y \end{array} \right\})$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{ y, -y, x - y^2, y^2 - x \})$

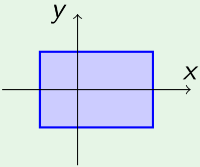
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{\, t_1, \ldots, t_n \,\}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n) = \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

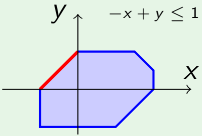
Example



intervals

$(\mathcal{T} = \{\, x, \, -x, y, \, -y \,\})$

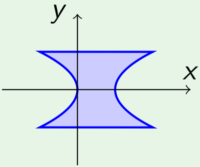
$(\beta = (2, 1, 1, 1))$



octagons

$$(\mathcal{T} = \left\{ \begin{array}{c} x, \, -x, y, \, -y, \\ x + y, x - y, \\ -x + y, \, -x - y \end{array} \right\})$$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{\, y, \, -y, x - y^2, y^2 - x \,\})$

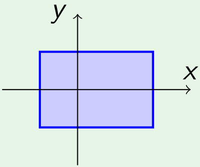
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{ t_1, \ldots, t_n \}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n) = \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

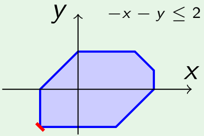
Example



intervals

$(\mathcal{T} = \{ x, -x, y, -y \})$

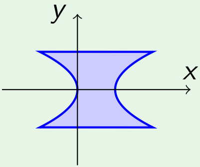
$(\beta = (2, 1, 1, 1))$



octagons

$(\mathcal{T} = \left\{ \begin{array}{c} x, -x, y, -y, \\ x + y, x - y, \\ -x + y, -x - y \end{array} \right\})$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{ y, -y, x - y^2, y^2 - x \})$

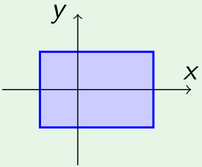
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{ t_1, \ldots, t_n \}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n) = \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

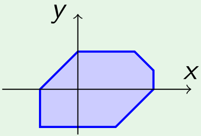
Example



intervals

$(\mathcal{T} = \{ x, -x, y, -y \})$

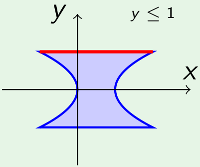
$(\beta = (2, 1, 1, 1))$



octagons

$(\mathcal{T} = \left\{ \begin{matrix} x, -x, y, -y, \\ x+y, x-y, \\ -x+y, -x-y \end{matrix} \right\})$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{ y, -y, x - y^2, y^2 - x \})$

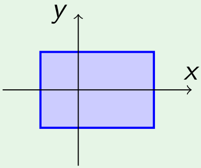
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{\, t_1, \ldots, t_n \,\}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n)$ $= \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \le b_1, \ldots, [\![t_n]\!](\rho) \le b_n\}$.

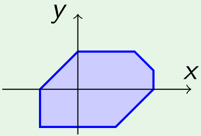
Example



intervals

$(\mathcal{T} = \{\, x, \; -x, y, \; -y \,\})$

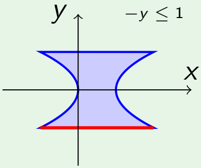
$(\beta = (2, 1, 1, 1))$



octagons

$(\mathcal{T} = \left\{ \begin{array}{c} x, \; -x, y, \; -y, \\ x+y, x-y, \\ -x+y, \; -x-y \end{array} \right\})$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{\, y, \; -y, x - y^2, y^2 - x \,\})$

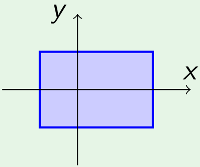
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{ t_1, \ldots, t_n \}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n)$ $= \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

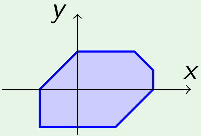
Example



intervals

$(\mathcal{T} = \{ x, -x, y, -y \})$

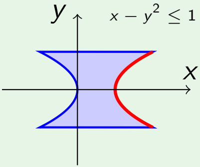
$(\beta = (2, 1, 1, 1))$



octagons

$(\mathcal{T} = \left\{ \begin{array}{c} x, -x, y, -y, \\ x + y, x - y, \\ -x + y, -x - y \end{array} \right\})$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{ y, -y, x - y^2, y^2 - x \})$

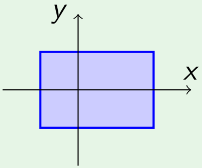
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{ t_1, \ldots, t_n \}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n) = \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

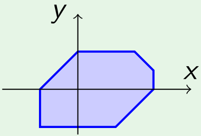
Example



intervals

$(\mathcal{T} = \{ x, -x, y, -y \})$

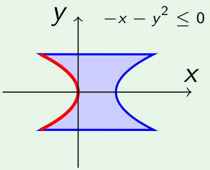
$(\beta = (2, 1, 1, 1))$



octagons

$$(\mathcal{T} = \left\{ \begin{array}{c} x, -x, y, -y, \\ x+y, x-y, \\ -x+y, -x-y \end{array} \right\})$$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



quadratic

$(\mathcal{T} = \{ y, -y, x - y^2, y^2 - x \})$

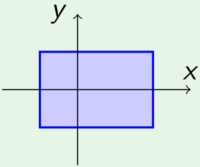
$(\beta = (1, 1, 1, 0))$

Policy Iterations: Template Domains

Definition (Template Domain)

Given a set of expressions $\mathcal{T} = \{ t_1, \ldots, t_n \}$, abstract values are tuples $\beta = (b_1, \ldots, b_n) \in \bar{\mathbb{R}}^n$, representing $\gamma_{\mathcal{T}}(b_1, \ldots, b_n) = \{\rho \in (\mathbb{V} \to \mathbb{R}) \mid [\![t_1]\!](\rho) \leq b_1, \ldots, [\![t_n]\!](\rho) \leq b_n\}$.

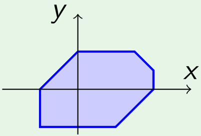
Example



intervals

$(\mathcal{T} = \{ x, -x, y, -y \})$

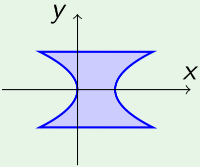
$(\beta = (2, 1, 1, 1))$



octagons

$$(\mathcal{T} = \left\{ \begin{array}{c} x, -x, y, -y, \\ x+y, x-y, \\ -x+y, -x-y \end{array} \right\})$$

$(\beta = (2, 1, 1, 1, 2.5, 2, 1, 2))$



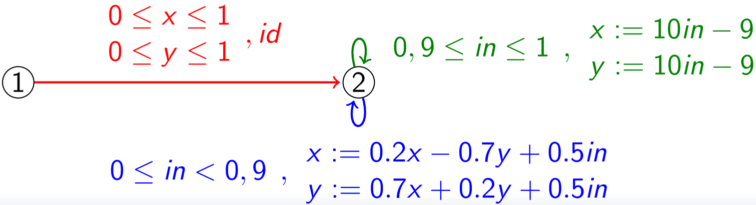
quadratic

$(\mathcal{T} = \{ y, -y, x-y^2, y^2-x \})$

$(\beta = (1, 1, 1, 0))$

Policy Iterations: System of Equations

First step: building the **control flow graph**.



Policy Iterations: System of Equations

Second step: deriving a system of equations
from the graph and the templates ($\mathcal{T} = \{ x, -x, y, -y \}$).

$$
\begin{cases}
b_{1,1} = +\infty \qquad b_{1,2} = +\infty \qquad b_{1,3} = +\infty \qquad b_{1,4} = +\infty \\
b_{2,1} = \max\{x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.2\,x - 0.7\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,2} = \max\{-x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.2\,x + 0.7\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,3} = \max\{y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.7\,x + 0.2\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,4} = \max\{-y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.7\,x - 0.2\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\}
\end{cases}
$$

with $\mathrm{be}(i)$ a shortcut for $x \leq b_{i,1} \wedge -x \leq b_{i,2} \wedge y \leq b_{i,3} \wedge -y \leq b_{i,4}$

Policy Iterations: Min or Max Strategy

Two main approaches to solve the equations:

- **Min-Strategy Iteration**
  descending iterations (like Newton-Raphson)
  - no guarantee to reach a fixpoint
  - can be stopped at any time leaving a sound result
  - convergence is usually fast
- **Max-Strategy Iteration**
  starts from bottom and iterates greatest fixpoint computations on max-strategies until a global fixpoint is reached
  - have to wait until the end for a sound result
  - terminates with a precise fixpoint

Policy Iterations: Min or Max Strategy

Two main approaches to solve the equations:

- **Min-Strategy Iteration**
  descending iterations (like Newton-Raphson)
    - no guarantee to reach a fixpoint
    - can be stopped at any time leaving a sound result
    - convergence is usually fast
- **Max-Strategy Iteration**
  starts from bottom and iterates greatest fixpoint computations on max-strategies until a global fixpoint is reached
    - have to wait until the end for a sound result
    - terminates with a precise fixpoint

We detail the second approach.

Policy Iterations: Max-Strategies

A **max-strategy** is the choice of one disjunct per equation:

$$
\begin{cases}
b_{1,1} = +\infty \quad b_{1,2} = +\infty \quad b_{1,3} = +\infty \quad b_{1,4} = +\infty \\
b_{2,1} = \max\{x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.2\,x - 0.7\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,2} = \max\{-x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.2\,x + 0.7\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,3} = \max\{y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.7\,x + 0.2\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,4} = \max\{-y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.7\,x - 0.2\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\}
\end{cases}
$$

with $\mathrm{be}(i)$ still a shortcut for $x \leq b_{i,1} \wedge -x \leq b_{i,2} \wedge y \leq b_{i,3} \wedge -y \leq b_{i,4}$

Policy Iterations: Max-Strategies

A **max-strategy** is the choice of one disjunct per equation:

$$
\begin{cases}
b_{1,1} = +\infty \qquad b_{1,2} = +\infty \qquad b_{1,3} = +\infty \qquad b_{1,4} = +\infty \\
b_{2,1} = \max\{x \mid 0 \leq x \leq 1 \land 0 \leq y \leq 1 \land \mathrm{be}(1)\} \\
\qquad \lor \max\{10\, in - 9 \mid 0.9 \leq in \leq 1 \land \mathrm{be}(2)\} \\
\qquad \lor \max\{0.2\, x - 0.7\, y + 0.5\, in \mid 0 \leq in \leq 0.9 \land \mathrm{be}(2)\} \\
b_{2,2} = \max\{-x \mid 0 \leq x \leq 1 \land 0 \leq y \leq 1 \land \mathrm{be}(1)\} \\
\qquad \lor \max\{-10\, in + 9 \mid 0.9 \leq in \leq 1 \land \mathrm{be}(2)\} \\
\qquad \lor \max\{-0.2\, x + 0.7\, y - 0.5\, in \mid 0 \leq in \leq 0.9 \land \mathrm{be}(2)\} \\
b_{2,3} = \max\{y \mid 0 \leq x \leq 1 \land 0 \leq y \leq 1 \land \mathrm{be}(1)\} \\
\qquad \lor \max\{10\, in - 9 \mid 0.9 \leq in \leq 1 \land \mathrm{be}(2)\} \\
\qquad \lor \max\{0.7\, x + 0.2\, y + 0.5\, in \mid 0 \leq in \leq 0.9 \land \mathrm{be}(2)\} \\
b_{2,4} = \max\{-y \mid 0 \leq x \leq 1 \land 0 \leq y \leq 1 \land \mathrm{be}(1)\} \\
\qquad \lor \max\{-10\, in + 9 \mid 0.9 \leq in \leq 1 \land \mathrm{be}(2)\} \\
\qquad \lor \max\{-0.7\, x - 0.2\, y - 0.5\, in \mid 0 \leq in \leq 0.9 \land \mathrm{be}(2)\}
\end{cases}
$$

with $\mathrm{be}(i)$ still a shortcut for $x \leq b_{i,1} \land -x \leq b_{i,2} \land y \leq b_{i,3} \land -y \leq b_{i,4}$

Policy Iterations: Max-Strategies

A **max-strategy** is the choice of one disjunct per equation:

$$\begin{cases} b_{1,1} = +\infty \quad b_{1,2} = +\infty \quad b_{1,3} = +\infty \quad b_{1,4} = +\infty \\ b_{2,1} = \\ \qquad \max\{10\, in - 9 \mid 0.9 \le in \le 1 \land \mathrm{be}(2)\} \\ b_{2,2} = \max\{-x \mid 0 \le x \le 1 \land 0 \le y \le 1 \land \mathrm{be}(1)\} \\ b_{2,3} = \max\{y \mid 0 \le x \le 1 \land 0 \le y \le 1 \land \mathrm{be}(1)\} \\ b_{2,4} = \\ \qquad \max\{-0.7\, x - 0.2\, y - 0.5\, in \mid 0 \le in \le 0.9 \land \mathrm{be}(2)\} \end{cases}$$

with $\mathrm{be}(i)$ still a shortcut for $x \le b_{i,1} \land -x \le b_{i,2} \land y \le b_{i,3} \land -y \le b_{i,4}$

Max-Strategies Iterations

Algorithm

- add a disjunct $-\infty$ to each equation
- initialize strategy σ_0 to all the $-\infty$ and β_0 to the tuple $(-\infty, \ldots, -\infty)$
- iterates on strategies until there is no more improving strategy
  - find an improving strategy σ_{i+1} (i.e. σ_{i+1} avaluated at β_i reaches values greater than β_i)
  - compute β_{i+1} the greatest fixpoint of σ_{i+1}

Max-Strategies Iterations: Example

We first add the $-\infty$ disjunct:

$$\begin{cases} b_{1,1} = -\infty \vee +\infty & b_{1,2} = -\infty \vee +\infty \\ b_{1,3} = -\infty \vee +\infty & b_{1,4} = -\infty \vee +\infty \\ b_{2,1} = -\infty \vee \max\{x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} & \\ \quad \vee \max\{10\, in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} & \\ \quad \vee \max\{0.2\, x - 0.7\, y + 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} & \\ b_{2,2} = -\infty \vee \max\{-x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} & \\ \quad \vee \max\{-10\, in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} & \\ \quad \vee \max\{-0.2\, x + 0.7\, y - 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} & \\ b_{2,3} = -\infty \vee \max\{y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} & \\ \quad \vee \max\{10\, in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} & \\ \quad \vee \max\{0.7\, x + 0.2\, y + 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} & \\ b_{2,4} = -\infty \vee \max\{-y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} & \\ \quad \vee \max\{-10\, in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} & \\ \quad \vee \max\{-0.7\, x - 0.2\, y - 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} & \end{cases}$$

with $\mathrm{be}(i)$ still a shortcut for $x \leq b_{i,1} \wedge -x \leq b_{i,2} \wedge y \leq b_{i,3} \wedge -y \leq b_{i,4}$

Max-Strategies Iterations: Example

Then initiate strategy σ_0 and $\beta_0 = (-\infty, \ldots, -\infty)$

$$
\begin{cases}
b_{1,1} = -\infty \vee +\infty \qquad\qquad b_{1,2} = -\infty \vee +\infty \\
b_{1,3} = -\infty \vee +\infty \qquad\qquad b_{1,4} = -\infty \vee +\infty \\
b_{2,1} = -\infty \vee \max\{x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.2\,x - 0.7\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,2} = -\infty \vee \max\{-x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.2\,x + 0.7\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,3} = -\infty \vee \max\{y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.7\,x + 0.2\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,4} = -\infty \vee \max\{-y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.7\,x - 0.2\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\}
\end{cases}
$$

with $\mathrm{be}(i)$ still a shortcut for $x \leq b_{i,1} \wedge -x \leq b_{i,2} \wedge y \leq b_{i,3} \wedge -y \leq b_{i,4}$

Max-Strategies Iterations: Example

We find an improving strategy σ_1:

$$\begin{cases}
b_{1,1} = -\infty \vee +\infty \qquad\qquad b_{1,2} = -\infty \vee +\infty \\
b_{1,3} = -\infty \vee +\infty \qquad\qquad b_{1,4} = -\infty \vee +\infty \\
b_{2,1} = -\infty \vee \max\{x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.2\,x - 0.7\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,2} = -\infty \vee \max\{-x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.2\,x + 0.7\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,3} = -\infty \vee \max\{y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.7\,x + 0.2\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,4} = -\infty \vee \max\{-y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.7\,x - 0.2\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\}
\end{cases}$$

with $\mathrm{be}(i)$ still a shortcut for $x \leq b_{i,1} \wedge -x \leq b_{i,2} \wedge y \leq b_{i,3} \wedge -y \leq b_{i,4}$

Max-Strategies Iterations: Example

And compute the greatest fixpoint
$\beta_1 = (+\infty, +\infty, +\infty, +\infty, -\infty, -\infty, -\infty, -\infty)$

$$\begin{cases}
b_{1,1} = -\infty \vee +\infty & b_{1,2} = -\infty \vee +\infty \\
b_{1,3} = -\infty \vee +\infty & b_{1,4} = -\infty \vee +\infty \\
b_{2,1} = -\infty \vee \max\{x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\quad \vee \max\{10\, in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\quad \vee \max\{0.2\, x - 0.7\, y + 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,2} = -\infty \vee \max\{-x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\quad \vee \max\{-10\, in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\quad \vee \max\{-0.2\, x + 0.7\, y - 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,3} = -\infty \vee \max\{y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\quad \vee \max\{10\, in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\quad \vee \max\{0.7\, x + 0.2\, y + 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,4} = -\infty \vee \max\{-y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\quad \vee \max\{-10\, in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\quad \vee \max\{-0.7\, x - 0.2\, y - 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\}
\end{cases}$$

with $\mathrm{be}(i)$ still a shortcut for $x \leq b_{i,1} \wedge -x \leq b_{i,2} \wedge y \leq b_{i,3} \wedge -y \leq b_{i,4}$

Max-Strategies Iterations: Example

We find an improving strategy σ_2:

$$
\begin{cases}
b_{1,1} = -\infty \vee +\infty & b_{1,2} = -\infty \vee +\infty \\
b_{1,3} = -\infty \vee +\infty & b_{1,4} = -\infty \vee +\infty \\
b_{2,1} = -\infty \vee \max\{x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\, in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.2\, x - 0.7\, y + 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,2} = -\infty \vee \max\{-x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\, in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.2\, x + 0.7\, y - 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,3} = -\infty \vee \max\{y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\, in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.7\, x + 0.2\, y + 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,4} = -\infty \vee \max\{-y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\, in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.7\, x - 0.2\, y - 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\}
\end{cases}
$$

with $\mathrm{be}(i)$ still a shortcut for $x \leq b_{i,1} \wedge -x \leq b_{i,2} \wedge y \leq b_{i,3} \wedge -y \leq b_{i,4}$

Max-Strategies Iterations: Example

And compute the greatest fixpoint $\beta_2 = (+\infty, +\infty, +\infty, +\infty, 1, 0, 1, 0)$

$$
\begin{cases}
b_{1,1} = -\infty \vee +\infty & b_{1,2} = -\infty \vee +\infty \\
b_{1,3} = -\infty \vee +\infty & b_{1,4} = -\infty \vee +\infty \\
b_{2,1} = -\infty \vee \max\{x \mid 0 \le x \le 1 \wedge 0 \le y \le 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \le in \le 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.2\,x - 0.7\,y + 0.5\,in \mid 0 \le in \le 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,2} = -\infty \vee \max\{-x \mid 0 \le x \le 1 \wedge 0 \le y \le 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \le in \le 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.2\,x + 0.7\,y - 0.5\,in \mid 0 \le in \le 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,3} = -\infty \vee \max\{y \mid 0 \le x \le 1 \wedge 0 \le y \le 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \le in \le 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.7\,x + 0.2\,y + 0.5\,in \mid 0 \le in \le 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,4} = -\infty \vee \max\{-y \mid 0 \le x \le 1 \wedge 0 \le y \le 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \le in \le 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.7\,x - 0.2\,y - 0.5\,in \mid 0 \le in \le 0.9 \wedge \mathrm{be}(2)\}
\end{cases}
$$

with $\mathrm{be}(i)$ still a shortcut for $x \le b_{i,1} \wedge -x \le b_{i,2} \wedge y \le b_{i,3} \wedge -y \le b_{i,4}$

Max-Strategies Iterations: Example

We find an improving strategy σ_3:

$$
\begin{cases}
b_{1,1} = -\infty \vee +\infty & b_{1,2} = -\infty \vee +\infty \\
b_{1,3} = -\infty \vee +\infty & b_{1,4} = -\infty \vee +\infty \\
b_{2,1} = -\infty \vee \max\{x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} & \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} & \\
\qquad \vee \max\{0.2\,x - 0.7\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} & \\
b_{2,2} = -\infty \vee \max\{-x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} & \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} & \\
\qquad \vee \max\{-0.2\,x + 0.7\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} & \\
b_{2,3} = -\infty \vee \max\{y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} & \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} & \\
\qquad \vee \max\{0.7\,x + 0.2\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} & \\
b_{2,4} = -\infty \vee \max\{-y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} & \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} & \\
\qquad \vee \max\{-0.7\,x - 0.2\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} &
\end{cases}
$$

with $\mathrm{be}(i)$ still a shortcut for $x \leq b_{i,1} \wedge -x \leq b_{i,2} \wedge y \leq b_{i,3} \wedge -y \leq b_{i,4}$

Max-Strategies Iterations: Example

And compute the greatest fixpoint
$\beta_3 = (+\infty, +\infty, +\infty, +\infty, 1, 1.12578125, 1.14375, 0)$

$$\begin{cases} b_{1,1} = -\infty \vee +\infty & b_{1,2} = -\infty \vee +\infty \\ b_{1,3} = -\infty \vee +\infty & b_{1,4} = -\infty \vee +\infty \\ b_{2,1} = -\infty \vee \max\{x \mid 0 \le x \le 1 \wedge 0 \le y \le 1 \wedge \mathrm{be}(1)\} \\ \qquad \vee \max\{10\,in - 9 \mid 0.9 \le in \le 1 \wedge \mathrm{be}(2)\} \\ \qquad \vee \max\{0.2\,x - 0.7\,y + 0.5\,in \mid 0 \le in \le 0.9 \wedge \mathrm{be}(2)\} \\ b_{2,2} = -\infty \vee \max\{-x \mid 0 \le x \le 1 \wedge 0 \le y \le 1 \wedge \mathrm{be}(1)\} \\ \qquad \vee \max\{-10\,in + 9 \mid 0.9 \le in \le 1 \wedge \mathrm{be}(2)\} \\ \qquad \vee \max\{-0.2\,x + 0.7\,y - 0.5\,in \mid 0 \le in \le 0.9 \wedge \mathrm{be}(2)\} \\ b_{2,3} = -\infty \vee \max\{y \mid 0 \le x \le 1 \wedge 0 \le y \le 1 \wedge \mathrm{be}(1)\} \\ \qquad \vee \max\{10\,in - 9 \mid 0.9 \le in \le 1 \wedge \mathrm{be}(2)\} \\ \qquad \vee \max\{0.7\,x + 0.2\,y + 0.5\,in \mid 0 \le in \le 0.9 \wedge \mathrm{be}(2)\} \\ b_{2,4} = -\infty \vee \max\{-y \mid 0 \le x \le 1 \wedge 0 \le y \le 1 \wedge \mathrm{be}(1)\} \\ \qquad \vee \max\{-10\,in + 9 \mid 0.9 \le in \le 1 \wedge \mathrm{be}(2)\} \\ \qquad \vee \max\{-0.7\,x - 0.2\,y - 0.5\,in \mid 0 \le in \le 0.9 \wedge \mathrm{be}(2)\} \end{cases}$$

with $\mathrm{be}(i)$ still a shortcut for $x \le b_{i,1} \wedge -x \le b_{i,2} \wedge y \le b_{i,3} \wedge -y \le b_{i,4}$

Max-Strategies Iterations: Example

We find an improving strategy σ_4:

$$
\begin{cases}
b_{1,1} = -\infty \vee +\infty & b_{1,2} = -\infty \vee +\infty \\
b_{1,3} = -\infty \vee +\infty & b_{1,4} = -\infty \vee +\infty \\
b_{2,1} = -\infty \vee \max\{x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.2\,x - 0.7\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,2} = -\infty \vee \max\{-x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.2\,x + 0.7\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,3} = -\infty \vee \max\{y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.7\,x + 0.2\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,4} = -\infty \vee \max\{-y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.7\,x - 0.2\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\}
\end{cases}
$$

with $\mathrm{be}(i)$ still a shortcut for $x \leq b_{i,1} \wedge -x \leq b_{i,2} \wedge y \leq b_{i,3} \wedge -y \leq b_{i,4}$

Max-Strategies Iterations: Example

And compute the greatest fixpoint
$\beta_4 = (+\infty, +\infty, +\infty, +\infty, 1, 1.12578125, 1.14375, 1.1005859375)$

$$
\begin{cases}
b_{1,1} = -\infty \vee +\infty & b_{1,2} = -\infty \vee +\infty \\
b_{1,3} = -\infty \vee +\infty & b_{1,4} = -\infty \vee +\infty \\
b_{2,1} = -\infty \vee \max\{x \mid 0 \le x \le 1 \wedge 0 \le y \le 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \le in \le 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.2\,x - 0.7\,y + 0.5\,in \mid 0 \le in \le 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,2} = -\infty \vee \max\{-x \mid 0 \le x \le 1 \wedge 0 \le y \le 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \le in \le 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.2\,x + 0.7\,y - 0.5\,in \mid 0 \le in \le 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,3} = -\infty \vee \max\{y \mid 0 \le x \le 1 \wedge 0 \le y \le 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \le in \le 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.7\,x + 0.2\,y + 0.5\,in \mid 0 \le in \le 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,4} = -\infty \vee \max\{-y \mid 0 \le x \le 1 \wedge 0 \le y \le 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \le in \le 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.7\,x - 0.2\,y - 0.5\,in \mid 0 \le in \le 0.9 \wedge \mathrm{be}(2)\}
\end{cases}
$$

with $\mathrm{be}(i)$ still a shortcut for $x \le b_{i,1} \wedge -x \le b_{i,2} \wedge y \le b_{i,3} \wedge -y \le b_{i,4}$

Max-Strategies Iterations: Example

We find an improving strategy σ_5:

$$
\begin{cases}
b_{1,1} = -\infty \vee +\infty & b_{1,2} = -\infty \vee +\infty \\
b_{1,3} = -\infty \vee +\infty & b_{1,4} = -\infty \vee +\infty \\
b_{2,1} = -\infty \vee \max\{x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\, in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.2\, x - 0.7\, y + 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,2} = -\infty \vee \max\{-x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\, in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.2\, x + 0.7\, y - 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,3} = -\infty \vee \max\{y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\, in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.7\, x + 0.2\, y + 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,4} = -\infty \vee \max\{-y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\, in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.7\, x - 0.2\, y - 0.5\, in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\}
\end{cases}
$$

with $\mathrm{be}(i)$ still a shortcut for $x \leq b_{i,1} \wedge -x \leq b_{i,2} \wedge y \leq b_{i,3} \wedge -y \leq b_{i,4}$

Max-Strategies Iterations: Example

And compute the greatest fixpoint
$\beta_5 \approx (+\infty, +\infty, +\infty, +\infty, 2.27, 2.23, 2.55, 1.95)$

$$
\begin{cases}
b_{1,1} = -\infty \vee +\infty \qquad\qquad b_{1,2} = -\infty \vee +\infty \\
b_{1,3} = -\infty \vee +\infty \qquad\qquad b_{1,4} = -\infty \vee +\infty \\
b_{2,1} = -\infty \vee \max\{x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.2\,x - 0.7\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,2} = -\infty \vee \max\{-x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.2\,x + 0.7\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,3} = -\infty \vee \max\{y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{0.7\,x + 0.2\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} \\
b_{2,4} = -\infty \vee \max\{-y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} \\
\qquad \vee \max\{-0.7\,x - 0.2\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\}
\end{cases}
$$

with $\mathrm{be}(i)$ still a shortcut for $x \leq b_{i,1} \wedge -x \leq b_{i,2} \wedge y \leq b_{i,3} \wedge -y \leq b_{i,4}$

Max-Strategies Iterations: Example

No more improving strategy, result:
$-2.24 \leq x \leq 2.27 \wedge -1.96 \leq y \leq 2.55$.

$$
\begin{cases}
b_{1,1} = -\infty \vee +\infty & b_{1,2} = -\infty \vee +\infty \\
b_{1,3} = -\infty \vee +\infty & b_{1,4} = -\infty \vee +\infty \\
b_{2,1} = -\infty \vee \max\{x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} & \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} & \\
\qquad \vee \max\{0.2\,x - 0.7\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} & \\
b_{2,2} = -\infty \vee \max\{-x \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} & \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} & \\
\qquad \vee \max\{-0.2\,x + 0.7\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} & \\
b_{2,3} = -\infty \vee \max\{y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} & \\
\qquad \vee \max\{10\,in - 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} & \\
\qquad \vee \max\{0.7\,x + 0.2\,y + 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} & \\
b_{2,4} = -\infty \vee \max\{-y \mid 0 \leq x \leq 1 \wedge 0 \leq y \leq 1 \wedge \mathrm{be}(1)\} & \\
\qquad \vee \max\{-10\,in + 9 \mid 0.9 \leq in \leq 1 \wedge \mathrm{be}(2)\} & \\
\qquad \vee \max\{-0.7\,x - 0.2\,y - 0.5\,in \mid 0 \leq in \leq 0.9 \wedge \mathrm{be}(2)\} &
\end{cases}
$$

with $\mathrm{be}(i)$ still a shortcut for $x \leq b_{i,1} \wedge -x \leq b_{i,2} \wedge y \leq b_{i,3} \wedge -y \leq b_{i,4}$

Max Strategies Iterations: Termination

- each strategy is considered at most once

Max Strategies Iterations: Termination

- each strategy is considered at most once
- there is an exponential number of strategies

Max Strategies Iterations: Termination

- each strategy is considered at most once
- there is an exponential number of strategies

$\Rightarrow$ bounds the number of iterations

Max Strategies Iterations: Termination

- each strategy is considered at most once
- there is an exponential number of strategies

$\Rightarrow$ bounds the number of iterations

- in practice, only a few strategies are considered

Policy Iterations vs Kleene Iterations

- policy iterations are an efficient alternative to widening

Policy Iterations vs Kleene Iterations

- policy iterations are an efficient alternative to widening
- however their use seems orthogonal to traditional Kleene iterations

Policy Iterations vs Kleene Iterations

- policy iterations are an efficient alternative to widening
- however their use seems orthogonal to traditional Kleene iterations
- this prevents an easy use in existing static analyzers
  and collaboration with existing domains through reduced products

Policy Iterations vs Kleene Iterations

- policy iterations are an efficient alternative to widening
- however their use seems orthogonal to traditional Kleene iterations
- this prevents an easy use in existing static analyzers
  and collaboration with existing domains through reduced products

$\Rightarrow$ we want to integrate policy iterations
into a traditional abstract domain

How to Integrate Policy Iterations into a Traditional Abstract Domain

- policy iterations require a system of equations
- system of equations is immediatly derived from control flow graph

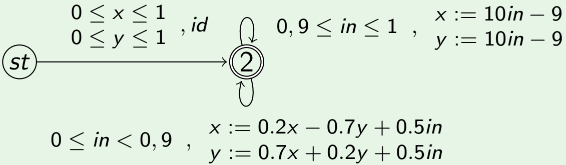
$\Rightarrow$ we will develop an abstract domain to build the graph

Lattice Structure

We use graphs with

- edges of the form *constraint*, *assignment*
- special vertices *st* and *fi*
  - *st* : starting point
  - *fi* : temporary final point
- a doubly circled vertex, indicating the "current point"

Example



Lattice Structure: Order

Basically, we define $g_1 \sqsubseteq^\sharp_{\mathcal{G}} g_2$ if

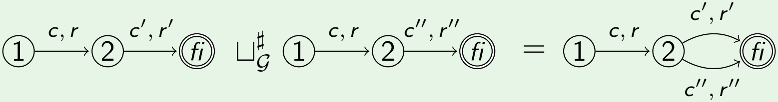
- for all edge from u to v in g_2
  - there is no edge from u to v with the same assignment
  - or all edges with the same assignment have stronger constraints
- and for all edge from u to v in g_1
  - there is at least one edge from u to v with the same assignment in g_2 having weaker constraints
- or we can reach the previous case by redirecting all edges from *fi* in g_2 to the "current point" of g_1 (see widening, later)

Lattice Structure: Order

Basically, we define $g_1 \sqsubseteq^\sharp_{\mathcal{G}} g_2$ if

- for all edge from u to v in g_2
  - there is no edge from u to v with the same assignment
  - or all edges with the same assignment have stronger constraints
- and for all edge from u to v in g_1
  - there is at least one edge from u to v with the same assignment in g_2 having weaker constraints
- or we can reach the previous case by redirecting all edges from *fi* in g_2 to the "current point" of g_1 (see widening, later)

Example (Join)



Abstract Operators: Guards

Example

$$[\![x \leq 0]\!]^\sharp(\top_\mathcal{G}) = \textcircled{st} \xrightarrow{x \leq 0,\, id} \textcircled{fi}$$

(a) case $g = \top_\mathcal{G}$

$$[\![x \leq 0]\!]^\sharp\left(\textcircled{st} \xrightarrow{y \leq 0,\, r} \textcircled{fi}\right) = \textcircled{st} \xrightarrow{\begin{smallmatrix} r(x) \leq 0 \\ y \leq 0 \end{smallmatrix},\, r} \textcircled{fi}$$

(b) case $g = (e, fi)$

$$[\![x \leq 0]\!]^\sharp\left(\textcircled{st} \xrightarrow{y \leq 0,\, r} \textcircled{t}\right) = \textcircled{st} \xrightarrow{y \leq 0,\, r} \textcircled{t} \xrightarrow{x \leq 0,\, id} \textcircled{fi}$$

(c) case $g = (e, t),\ t \neq fi$

Abstract Operators: Assignments

Example

$$[\![x := x+1]\!]^\sharp(\top_{\mathcal{G}}) = st \xrightarrow{\text{true},\, x := x+1} fi$$

(a) case $g = \top_{\mathcal{G}}$

$$[\![x := x+1]\!]^\sharp\left(st \xrightarrow{y \leq 0,\, r} fi \right) = st \xrightarrow{r[x := r(x)+1]} fi$$

(b) case $g = (e, fi)$

$$[\![x := x+1]\!]^\sharp\left(st \xrightarrow{y \leq 0,\, r} t \right) = st \xrightarrow{y \leq 0,\, r} t \xrightarrow{\text{true},\, x := x+1} fi$$

(c) case $g = (e, t)$, $t \neq fi$

Abstract Operators: Random Assignments

Definition

$\llbracket x := ?(r_1, r_2) \rrbracket^\sharp(g) := \llbracket r_1 - x_{ex} \leq 0 \rrbracket^\sharp \left(\llbracket x_{ex} - r_2 \leq 0 \rrbracket^\sharp \left(g'\right) \right)$
with $g' := \llbracket x := x_{ex} \rrbracket^\sharp(g)$
with x_{ex} an extra variable not appearing anywhere in g

Widening

- the domain $\mathcal{G}$ has infinite ascending chains
  $\Rightarrow$ a widening is required to enforce termination of the analyses

Widening

- the domain $\mathcal{G}$ has infinite ascending chains
  $\Rightarrow$ a widening is required to enforce termination of the analyses
- widening will
  - add nodes for loop heads
  - close the loops

Widening: Examples

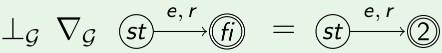
Example

$$\bot_{\mathcal{G}} \ \nabla_{\mathcal{G}} \ \ \textcircled{st} \xrightarrow{e,\, r} \textcircled{\textcircled{fi}} \ = \ \textcircled{st} \xrightarrow{e,\, r} \textcircled{\textcircled{2}}$$

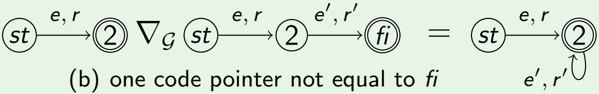
(a) both code pointers equal to *fi* ($\bot_{\mathcal{G}} = (\dot{\bot}_{\mathcal{C}}, fi)$)

Widening: Examples

Example



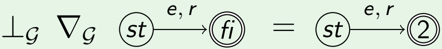
(a) both code pointers equal to *fi* ($\bot_{\mathcal{G}} = (\dot{\bot}_{\mathcal{C}}, fi)$)



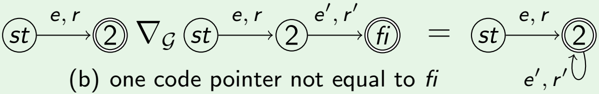
(b) one code pointer not equal to *fi*

Widening: Examples

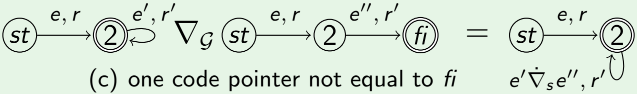
Example



(a) both code pointers equal to *fi* ($\bot_{\mathcal{G}} = (\dot{\bot}_{\mathcal{C}}, fi)$)



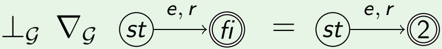
(b) one code pointer not equal to *fi*



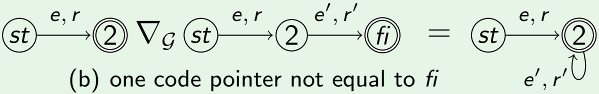
(c) one code pointer not equal to *fi*

Widening: Examples

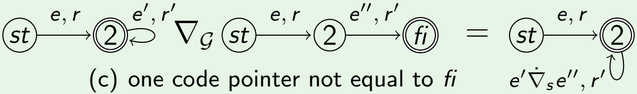
Example



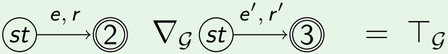
(a) both code pointers equal to *fi* ($\bot_{\mathcal{G}} = (\dot{\bot}_{\mathcal{C}}, fi)$)



(b) one code pointer not equal to *fi*



(c) one code pointer not equal to *fi*



(d) different code pointers

Example

```
(a) x := ?(0, 1);    y := ?(0, 1);
while    −1 ≤ 0    do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```

$\top_{\mathcal{G}}$

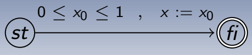
(a)

Example

$_{(a)}$ x **:=** **?(**0, 1**)**;$_{(b)}$ y **:=** **?(**0, 1**)**;
while $-1 \leq 0$ **do**
in **:=** **?(**0, 1**)**;
if $0.9 - \text{in} \leq 0$ **then**
x **:=** 10×in − 9;
y **:=** 10×in − 9
else
t **:=** x;
x **:=** 0.2×t − 0.7×y + 0.5×in;
y **:=** 0.7×t + 0.2×y + 0.5×in
fi
od

$\top_{\mathcal{G}}$

(a)



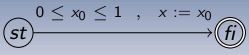
(b) $[\![x := ?(0,1)]\!]^\sharp(\mathrm{a})$

Example

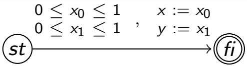
```
(a) x := ?(0, 1);(b) y := ?(0, 1);(c)
while    −1 ≤ 0    do
   in := ?(0, 1);
   if 0.9 − in ≤ 0 then
      x := 10×in − 9;
      y := 10×in − 9
   else
      t := x;
      x := 0.2×t − 0.7×y + 0.5×in;
      y := 0.7×t + 0.2×y + 0.5×in
   fi
od
```

$\top_{\mathcal{G}}$

(a)



(b) $[\![x := ?(0,1)]\!]^\sharp(\mathrm{a})$



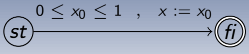
(c) $[\![y := ?(0,1)]\!]^\sharp(\mathrm{b})$

Example

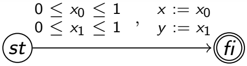
$_{(a)}$ **x := ?(0, 1);** $_{(b)}$ **y := ?(0, 1);** $_{(c)}$
while $_{(d)}$ $-1 \leq 0$ **do**
  in := ?(0, 1);
  if $0.9 - \text{in} \leq 0$ **then**
    x := 10×in − 9;
    y := 10×in − 9
  else
    t := x;
    x := 0.2×t − 0.7×y + 0.5×in;
    y := 0.7×t + 0.2×y + 0.5×in
  fi
od

$\top_{\mathcal{G}}$

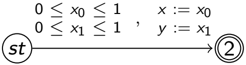
(a)



(b) $[\![x := ?(0,1)]\!]^\sharp(\mathrm{a})$



(c) $[\![y := ?(0,1)]\!]^\sharp(\mathrm{b})$



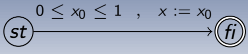
(d) $\perp \nabla_{\mathcal{G}} (\mathrm{c})$

Example

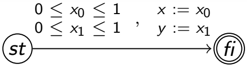
$_{(a)}$ **x :=** ?(0, 1);$_{(b)}$ **y :=** ?(0, 1);$_{(c)}$
while$_{(d)}$ $-1 \leq 0_{(e)}$ **do**
  in := ?(0, 1);
  if $0.9 - \text{in} \leq 0$ **then**
    x := 10×in − 9;
    y := 10×in − 9
  else
    t := x;
    x := 0.2×t − 0.7×y + 0.5×in;
    y := 0.7×t + 0.2×y + 0.5×in
  fi
od

$\top_{\mathcal{G}}$

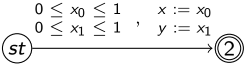
(a)



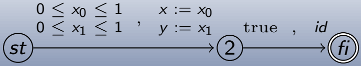
(b) $[\![x := ?(0,1)]\!]^\sharp(\mathrm{a})$



(c) $[\![y := ?(0,1)]\!]^\sharp(\mathrm{b})$



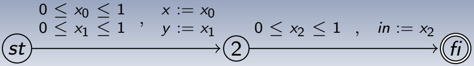
(d) $\bot \nabla_{\mathcal{G}} (\mathrm{c})$



(e) $[\![-1 \leq 0]\!]^\sharp(\mathrm{d})$

Example

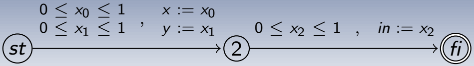
$_{(a)}$ x **:=** **?(**0, 1**)**;$_{(b)}$ y **:=** **?(**0, 1**)**;$_{(c)}$
while$_{(d)}$ $-1 \leq 0_{(e)}$ **do**
  in **:=** **?(**0, 1**)**;$_{(f)}$
  if 0.9 − in ≤ 0 **then**
    x **:=** 10×in − 9;
    y **:=** 10×in − 9
  else
    t **:=** x;
    x **:=** 0.2×t − 0.7×y + 0.5×in;
    y **:=** 0.7×t + 0.2×y + 0.5×in
  fi
od



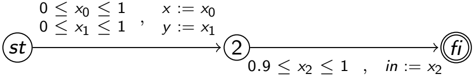
(f) $[\![in := ?(0,1)]\!]^\sharp(\mathrm{e})$

Example

$_{(a)}$ **x :=** **?(**0, 1**)**;$_{(b)}$ y **:=** **?(**0, 1**)**;$_{(c)}$
while$_{(d)}$ $-1 \leq 0_{(e)}$ **do**
  in **:=** **?(**0, 1**)**;$_{(f)}$
  if $0.9 - \text{in} \leq 0_{(g)}$ **then**
    x **:=** 10×in − 9;
    y **:=** 10×in − 9
  else
    t **:=** x;
    x **:=** 0.2×t − 0.7×y + 0.5×in;
    y **:=** 0.7×t + 0.2×y + 0.5×in
  fi
od



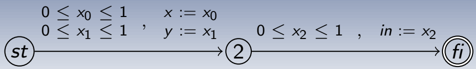
(f) $[\![in := ?(0,1)]\!]^\sharp(\mathrm{e})$



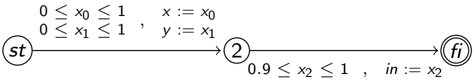
(g) $[\![0.9 - in \leq 0]\!]^\sharp(\mathrm{f})$

Example

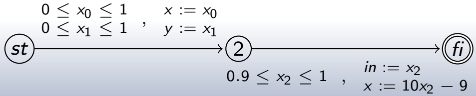
$_{(a)}$ x := ?(0, 1);$_{(b)}$ y := ?(0, 1);$_{(c)}$
while$_{(d)}$ $-1 \leq 0_{(e)}$ **do**
  in := ?(0, 1);$_{(f)}$
  if $0.9 - \text{in} \leq 0_{(g)}$ **then**
    x := 10×in − 9;$_{(h)}$
    y := 10×in − 9
  else
    t := x;
    x := 0.2×t − 0.7×y + 0.5×in;
    y := 0.7×t + 0.2×y + 0.5×in
  fi
od



(f) $[\![in := ?(0,1)]\!]^\sharp(\mathrm{e})$



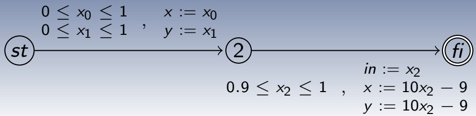
(g) $[\![0.9 - in \leq 0]\!]^\sharp(\mathrm{f})$



(h) $[\![x := 10in - 9]\!]^\sharp(\mathrm{g})$

Example

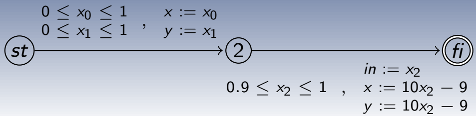
$_{(a)}$ x **:= ?(**0, 1**)**;$_{(b)}$ y **:= ?(**0, 1**)**;$_{(c)}$
while$_{(d)}$ $-1 \leq 0_{(e)}$ **do**
  in **:= ?(**0, 1**)**;$_{(f)}$
  if $0.9 - \text{in} \leq 0_{(g)}$ **then**
    x **:=** $10 \times \text{in} - 9;_{(h)}$
    y **:=** $10 \times \text{in} - 9_{(i)}$
  else
    t **:=** x;
    x **:=** $0.2 \times t - 0.7 \times y + 0.5 \times \text{in};$
    y **:=** $0.7 \times t + 0.2 \times y + 0.5 \times \text{in}$
  fi
od



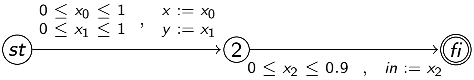
(i) $[\![y := 10in - 9]\!]^\sharp(\mathrm{h})$

Example

$_{(a)}$ x **:=** **?**(0, 1);$_{(b)}$ y **:=** **?**(0, 1);$_{(c)}$
while$_{(d)}$ $-1 \leq 0_{(e)}$ **do**
  in **:=** **?**(0, 1);$_{(f)}$
  if 0.9 − in $\leq 0_{(g)}$ **then**
    x **:=** 10×in − 9;$_{(h)}$
    y **:=** 10×in − $9_{(i)}$
  else$_{(j)}$
    t **:=** x;
    x **:=** 0.2×t − 0.7×y + 0.5×in;
    y **:=** 0.7×t + 0.2×y + 0.5×in
  fi
od



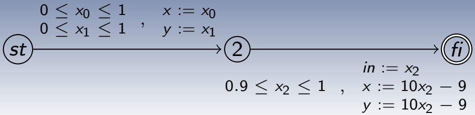
(i) $[\![y := 10in - 9]\!]^\sharp(\mathrm{h})$



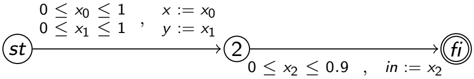
(j) $[\![0.9 - in \geq 0]\!]^\sharp(\mathrm{f})$

Example

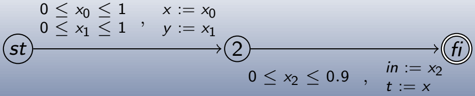
$_{(a)}$ **x :=** **?**(0, 1)**;**$_{(b)}$ **y :=** **?**(0, 1)**;**$_{(c)}$
while$_{(d)}$ $-1 \leq 0_{(e)}$ **do**
  in **:=** **?**(0, 1)**;**$_{(f)}$
  if $0.9 - \text{in} \leq 0_{(g)}$ **then**
    x **:=** $10\times\text{in} - 9;_{(h)}$
    y **:=** $10\times\text{in} - 9_{(i)}$
  else$_{(j)}$
    t **:=** x;$_{(k)}$
    x **:=** $0.2\times\text{t} - 0.7\times\text{y} + 0.5\times\text{in};$
    y **:=** $0.7\times\text{t} + 0.2\times\text{y} + 0.5\times\text{in}$
  fi
od



(i) $[\![y := 10in - 9]\!]^\sharp(\mathrm{h})$



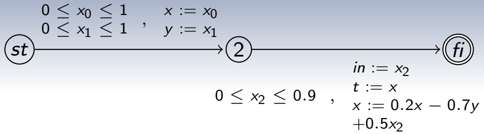
(j) $[\![0.9 - in \geq 0]\!]^\sharp(\mathrm{f})$



(k) $[\![t := x]\!]^\sharp(\mathrm{j})$

Example

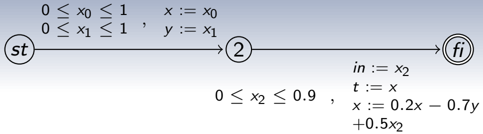
$_{(a)}$ x := ?(0, 1);$_{(b)}$ y := ?(0, 1);$_{(c)}$
while$_{(d)}$ $-1 \leq 0_{(e)}$ **do**
  in := ?(0, 1);$_{(f)}$
  if $0.9 - \text{in} \leq 0_{(g)}$ **then**
    x := 10×in − 9;$_{(h)}$
    y := 10×in − 9$_{(i)}$
  else$_{(j)}$
    t := x;$_{(k)}$
    x := 0.2×t − 0.7×y + 0.5×in;$_{(l)}$
    y := 0.7×t + 0.2×y + 0.5×in
  fi
od



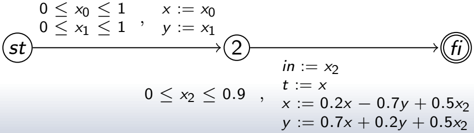
(l) $[\![x := 0.2t - 0.7y + 0.5in]\!]^\sharp(\mathrm{k})$

Example

$_{(a)}$ x **:=** **?**(0, 1);$_{(b)}$ y **:=** **?**(0, 1);$_{(c)}$
while$_{(d)}$ $-1 \leq 0_{(e)}$ **do**
  in **:=** **?**(0, 1);$_{(f)}$
  if $0.9 -$ in $\leq 0_{(g)}$ **then**
    x **:=** 10×in − 9;$_{(h)}$
    y **:=** 10×in − 9$_{(i)}$
  else$_{(j)}$
    t **:=** x;$_{(k)}$
    x **:=** 0.2×t − 0.7×y + 0.5×in;$_{(l)}$
    y **:=** 0.7×t + 0.2×y + 0.5×in$_{(m)}$
  fi
od

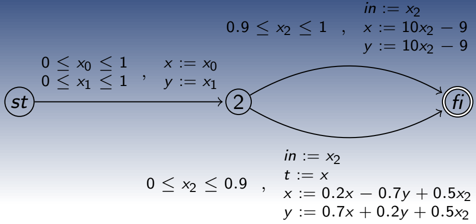


(l) $[\![x := 0.2t - 0.7y + 0.5in]\!]^\sharp(\mathrm{k})$



(m) $[\![y := 0.7t + 0.2y + 0.5in]\!]^\sharp(\mathrm{l})$

Example

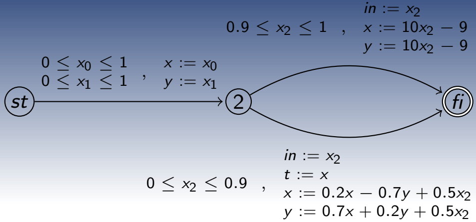


(n) (i) $\sqcup^{\sharp}_{\mathcal{G}}$ (m)

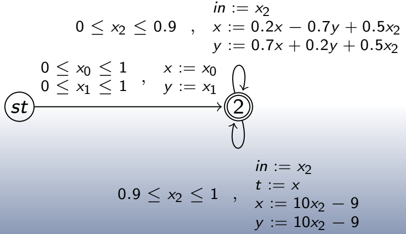
$_{(a)}$ x **:=** **?(**0, 1**)**;$_{(b)}$ y **:=** **?(**0, 1**)**;$_{(c)}$
while$_{(d)}$ $-1 \leq 0_{(e)}$ **do**
  in **:=** **?(**0, 1**)**;$_{(f)}$
  if $0.9 - \text{in} \leq 0_{(g)}$ **then**
    x **:=** 10×in − 9;$_{(h)}$
    y **:=** 10×in − 9$_{(i)}$
  else$_{(j)}$
    t **:=** x;$_{(k)}$
    x **:=** 0.2×t − 0.7×y + 0.5×in;$_{(l)}$
    y **:=** 0.7×t + 0.2×y + 0.5×in$_{(m)}$
  fi$_{(n)}$
od

Example

$_{(a)}$ x := ?(0, 1);$_{(b)}$ y := ?(0, 1);$_{(c)}$
while$_{(d),(o)}$ $-1 \leq 0_{(e)}$ **do**
  in := ?(0, 1);$_{(f)}$
  if $0.9 - \text{in} \leq 0_{(g)}$ **then**
    x := 10×in − 9;$_{(h)}$
    y := 10×in − 9$_{(i)}$
  else$_{(j)}$
    t := x;$_{(k)}$
    x := 0.2×t − 0.7×y + 0.5×in;$_{(l)}$
    y := 0.7×t + 0.2×y + 0.5×in$_{(m)}$
  fi$_{(n)}$
od



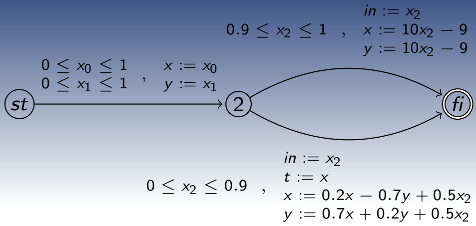
(n) $(i) \sqcup^{\sharp}_{\mathcal{G}} (m)$



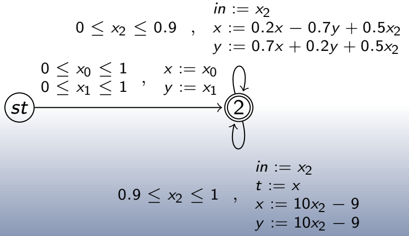
(o) $(d) \nabla_{\mathcal{G}} (n)$

Example

$_{(a)}$ **x := ?(0, 1)**;$_{(b)}$ **y := ?(0, 1)**;$_{(c)}$
while$_{(d),(o)}$ $-1 \leq 0_{(e)}$ **do**
  in := **?(0, 1)**;$_{(f)}$
  if $0.9 - \text{in} \leq 0_{(g)}$ **then**
    x := 10×in − 9;$_{(h)}$
    y := 10×in − 9$_{(i)}$
  else$_{(j)}$
    t := x;$_{(k)}$
    x := 0.2×t − 0.7×y + 0.5×in;$_{(l)}$
    y := 0.7×t + 0.2×y + 0.5×in$_{(m)}$
  fi$_{(n)}$
od



(n) $(\mathrm{i}) \sqcup^{\sharp}_{\mathcal{G}} (\mathrm{m})$



(o) $(\mathrm{d}) \nabla_{\mathcal{G}} (\mathrm{n})$

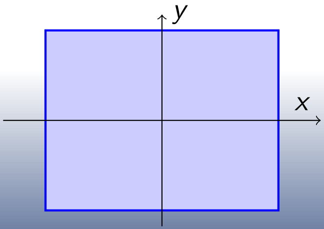
A second iteration proves that (o) is a **fixpoint**.

Quadratic invariants

- **Linear invariants** commonly used in static analysis are not well suited for linear reactive systems:
  - at best costly;
  - at worst no result.

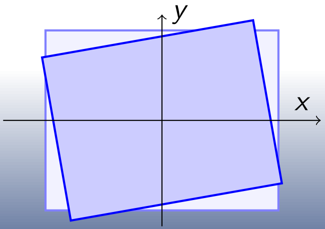
Quadratic invariants

- **Linear invariants** commonly used in static analysis are not well suited for linear reactive systems:
  - at best costly;
  - at worst no result.



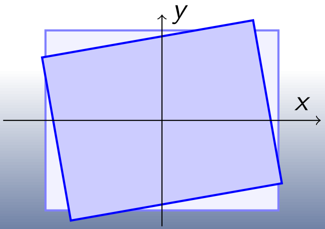
Quadratic invariants

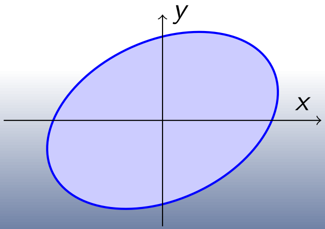
- **Linear invariants** commonly used in static analysis are not well suited for linear reactive systems:
  - at best costly;
  - at worst no result.



Quadratic invariants

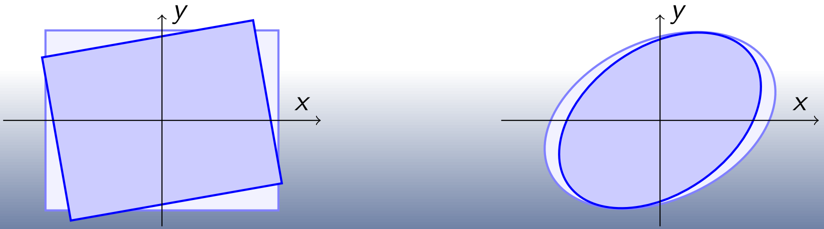
- **Linear invariants** commonly used in static analysis are not well suited for linear reactive systems:
  - at best costly;
  - at worst no result.
- Control theorists know for long that such systems are stable iff they admit **quadratic invariants**.





Quadratic invariants

- **Linear invariants** commonly used in static analysis are not well suited for linear reactive systems:
  - at best costly;
  - at worst no result.
- Control theorists know for long that such systems are stable iff they admit **quadratic invariants**.



Choosing Appropriate Templates

- from the control flow graph, we can extract
  subset of variables x and u and matrices A and B such that

$$x_{k+1} = Ax_k + Bu_k$$

- then we can extract a quadratic template
  by solving a Lyapunov equation
- we finally add templates y and $-y$
  for each variable y in the program

Implementation

- A new domain in our Lustre analyzer
- 4kloc of OCaml and 0.5kloc of C code
- use CSDP library for semidefinite programming
- soundness check of the result by Cholesky decompositions
- available under a GPL license at `http://cavale.enseeiht.fr/policy2013/`

Benchmarks (1/2)

	n	#*policies*	total	templates	iterations	check
Ex. 1	3	5	0.84	0.34	0.44	0.02
	4	5	1.81	0.36	1.30	0.07
	3	5	0.92	0.34	0.52	0.03
Ex. 2	5	5	2.76	0.58	2.04	0.05
	6	6	13.36	0.51	12.45	0.18
	5	5	2.78	0.51	2.08	0.06
Ex. 3 Discretized lead-lag controller	3	5	1.08	0.32	0.71	$\perp$ (0.01)
	4	5	3.98	0.31	3.57	$\perp$ (0.03)
	3	5	1.12	0.31	0.75	$\perp$ (0.02)
Ex. 4 Linear quadratic gaussian regulator	4	4	1.65	0.33	1.21	0.06
	5	4	7.42	0.33	6.94	$\perp$ (0.02)
	4	5	1.49	0.34	1.00	0.06

All times are in second.

Benchmarks (2/2)

	n	#*policies*	total	templates	iterations	check
Ex. 5 Observer based controller for a coupled mass system	6	4	3.16	0.59	2.31	0.11
	7	6	30.52	0.58	29.25	0.38
	6	5	4.50	0.60	3.60	0.13
Ex. 6 Butterworth low-pass filter	6	4	4.42	0.87	3.26	0.14
	7	5	24.80	0.88	23.10	0.47
	6	5	5.03	0.87	3.79	0.16
Ex. 7 Dampened oscillator	2	5	0.41	0.19	0.18	0.02
	3	5	1.07	0.20	0.72	0.07
	2	5	0.51	0.21	0.24	0.03
Ex. 8 Harmonic oscillator	2	5	0.41	0.19	0.17	0.02
	3	5	1.05	0.19	0.71	0.07
	2	5	0.48	0.19	0.23	0.03

All times are in second.

Conclusion and Future Work

- it works
- we should try min-strategy iteration
  which could bring performance improvements
- we should consider floating point semantic of programs

Questions

Thank you for your attention!

?