Abstract Acceleration of General Linear Loops

Bertrand Jeannet, Peter Schrammel, Sriram Sankaranarayanan

Principles of Programming Languages, POPL'14
San Diego, CA

Motivation and Challenge

Motivation

- Inferring polyhedral invariants on programs containing non-trivial numerical loops (e.g. control programs).

Motivation and Challenge

Motivation

- Inferring polyhedral invariants on programs containing non-trivial numerical loops (e.g. control programs).

Challenge

- Linear loops generate non-linear trajectories.

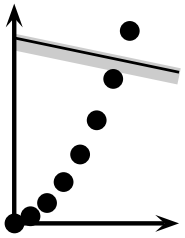
Motivation and Challenge

Motivation

- Inferring polyhedral invariants on programs containing non-trivial numerical loops (e.g. control programs).

Challenge

- Linear loops generate non-linear trajectories.



```
while(3*y+x<=10){
   x = x+1;
   y = y+x;
}
```

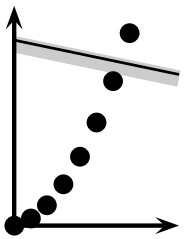
Motivation and Challenge

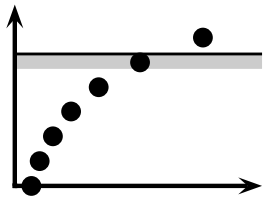
Motivation

- Inferring polyhedral invariants on programs containing non-trivial numerical loops (e.g. control programs).

Challenge

- Linear loops generate non-linear trajectories.





```
while(y<=3){
   x = 1.5*x;
   y = y+1;
}
```

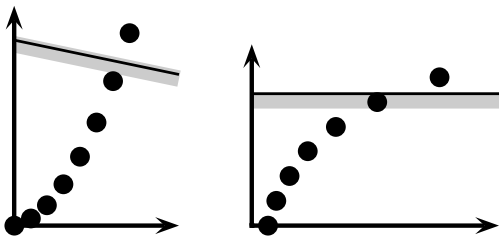
Motivation and Challenge

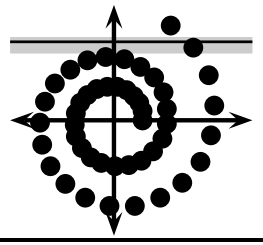
Motivation

- Inferring polyhedral invariants on programs containing non-trivial numerical loops (e.g. control programs).

Challenge

- Linear loops generate non-linear trajectories.





```
while(y<=10){
    x = 1.1*x - 0.05*y;
    y = 0.05*x + 1.1*y;
}
```

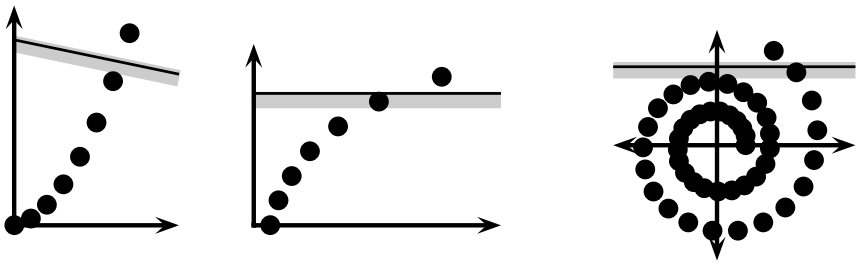
Motivation and Challenge

Motivation

- Inferring polyhedral invariants on programs containing non-trivial numerical loops (e.g. control programs).

Challenge

- Linear loops generate non-linear trajectories.



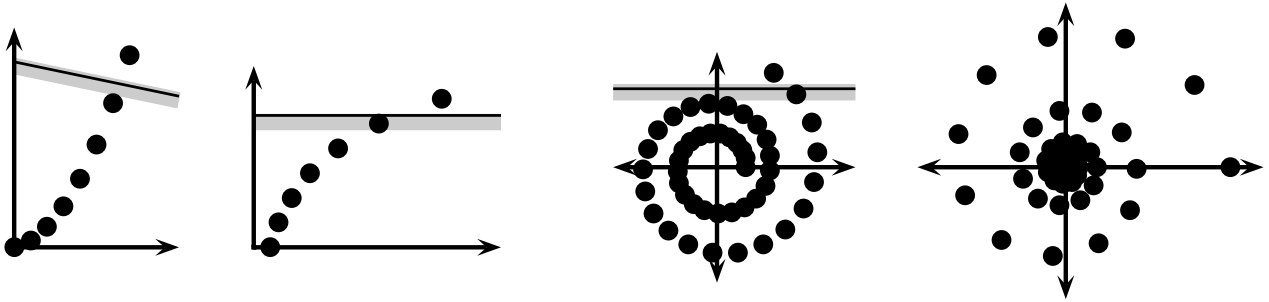
Motivation and Challenge

Motivation

- Inferring polyhedral invariants on programs containing non-trivial numerical loops (e.g. control programs).

Challenge

- Linear loops generate non-linear trajectories.



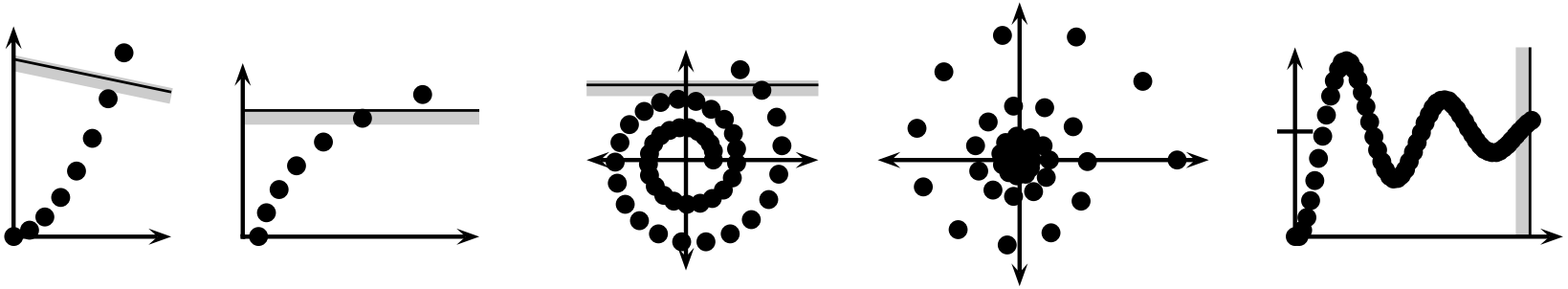
Motivation and Challenge

Motivation

- Inferring polyhedral invariants on programs containing non-trivial numerical loops (e.g. control programs).

Challenge

- Linear loops generate non-linear trajectories.



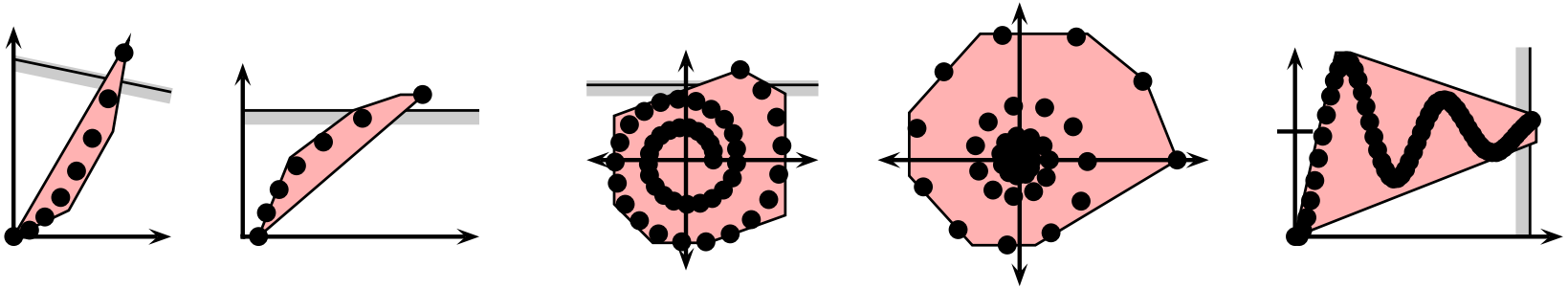
Motivation and Challenge

Motivation

- Inferring polyhedral invariants on programs containing non-trivial numerical loops (e.g. control programs).

Challenge

- Linear loops generate non-linear trajectories.



Goal

- Compute convex polyhedra containing all such trajectories.

Existing Techniques

Standard polyhedral analysis (Cousot&Halbwachs 1978)

- Compute iteratively $(\alpha \circ \tau \circ \gamma)^*(X)$
- INTERPROC, ...
- Not precise
- Thresholds, landmarks, abstract acceleration, ...
  improve precision for linear behavior

Existing Techniques

Standard polyhedral analysis (Cousot&Halbwachs 1978)

- Compute iteratively $(\alpha \circ \tau \circ \gamma)^*(X)$
- INTERPROC, ...
- Not precise
- Thresholds, landmarks, abstract acceleration, ...
  improve precision for linear behavior

Non-linear constraint solving (Colon et al 2003, Sankaranarayanan et al 2004)

- Solving directly $X = \tau(X)$ over convex polyhedra X
- Relaxation and reduction to non-linear constraint solving problem
- STING

Existing Techniques

Standard polyhedral analysis (Cousot&Halbwachs 1978)

- Compute iteratively $(\alpha \circ \tau \circ \gamma)^*(X)$
- INTERPROC, ...
- Not precise
- Thresholds, landmarks, abstract acceleration, ...
  improve precision for linear behavior

Non-linear constraint solving (Colon et al 2003, Sankaranarayanan et al 2004)

- Solving directly $X = \tau(X)$ over convex polyhedra X
- Relaxation and reduction to non-linear constraint solving problem
- STING

Ellipsoid methods (Kurzhanski&Varaiya 2000, Feret 2004, ...)

- Restricted to stable loops
- ASTRÉE

Existing Techniques

Standard polyhedral analysis (Cousot&Halbwachs 1978)

- Compute iteratively $(\alpha \circ \tau \circ \gamma)^*(X)$
- INTERPROC, ...
- Not precise
- Thresholds, landmarks, abstract acceleration, ...
  improve precision for linear behavior

Non-linear constraint solving (Colon et al 2003, Sankaranarayanan et al 2004)

- Solving directly $X = \tau(X)$ over convex polyhedra X
- Relaxation and reduction to non-linear constraint solving problem
- STING

Ellipsoid methods (Kurzhanski&Varaiya 2000, Feret 2004, ...)

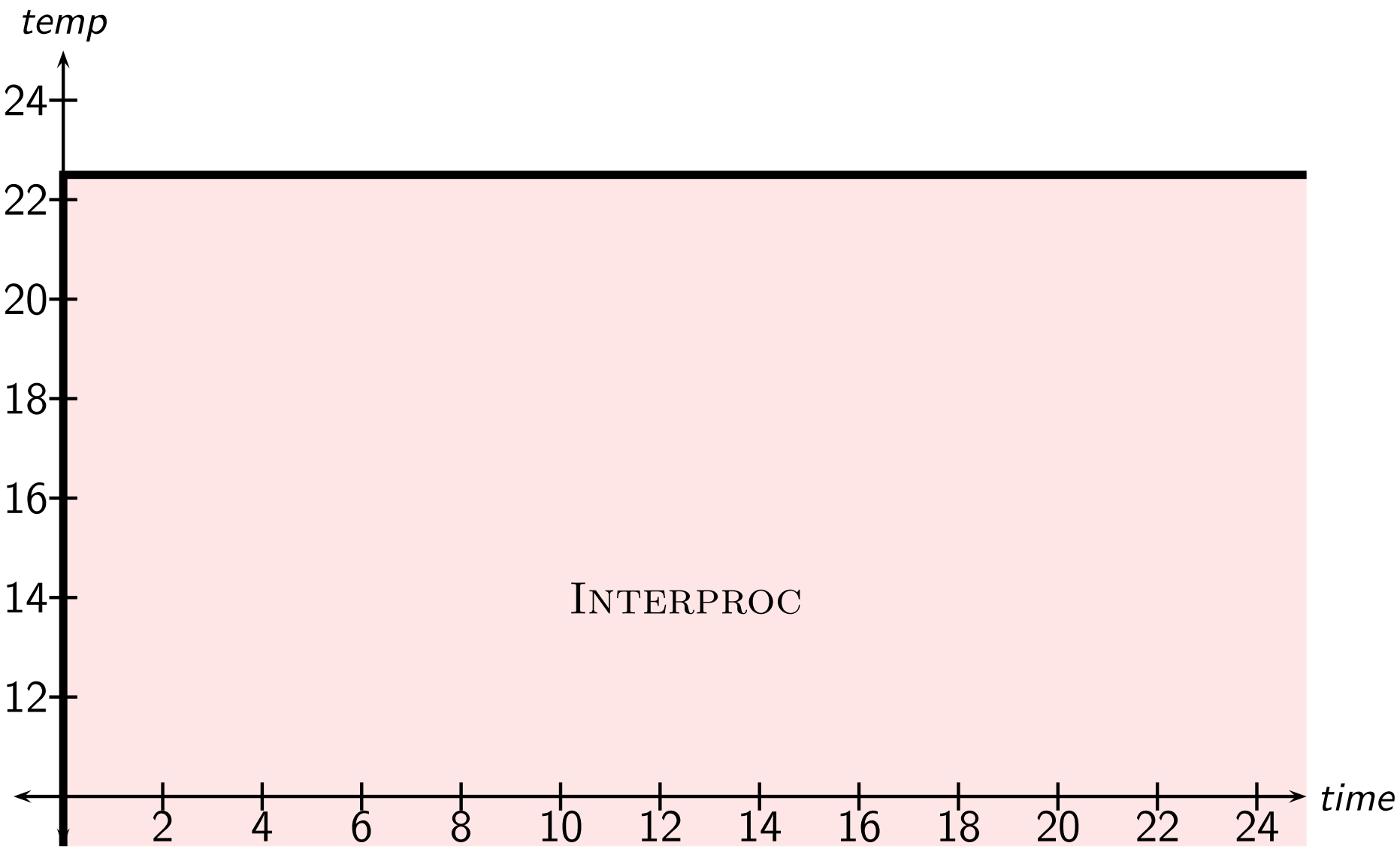
- Restricted to stable loops
- ASTRÉE

...

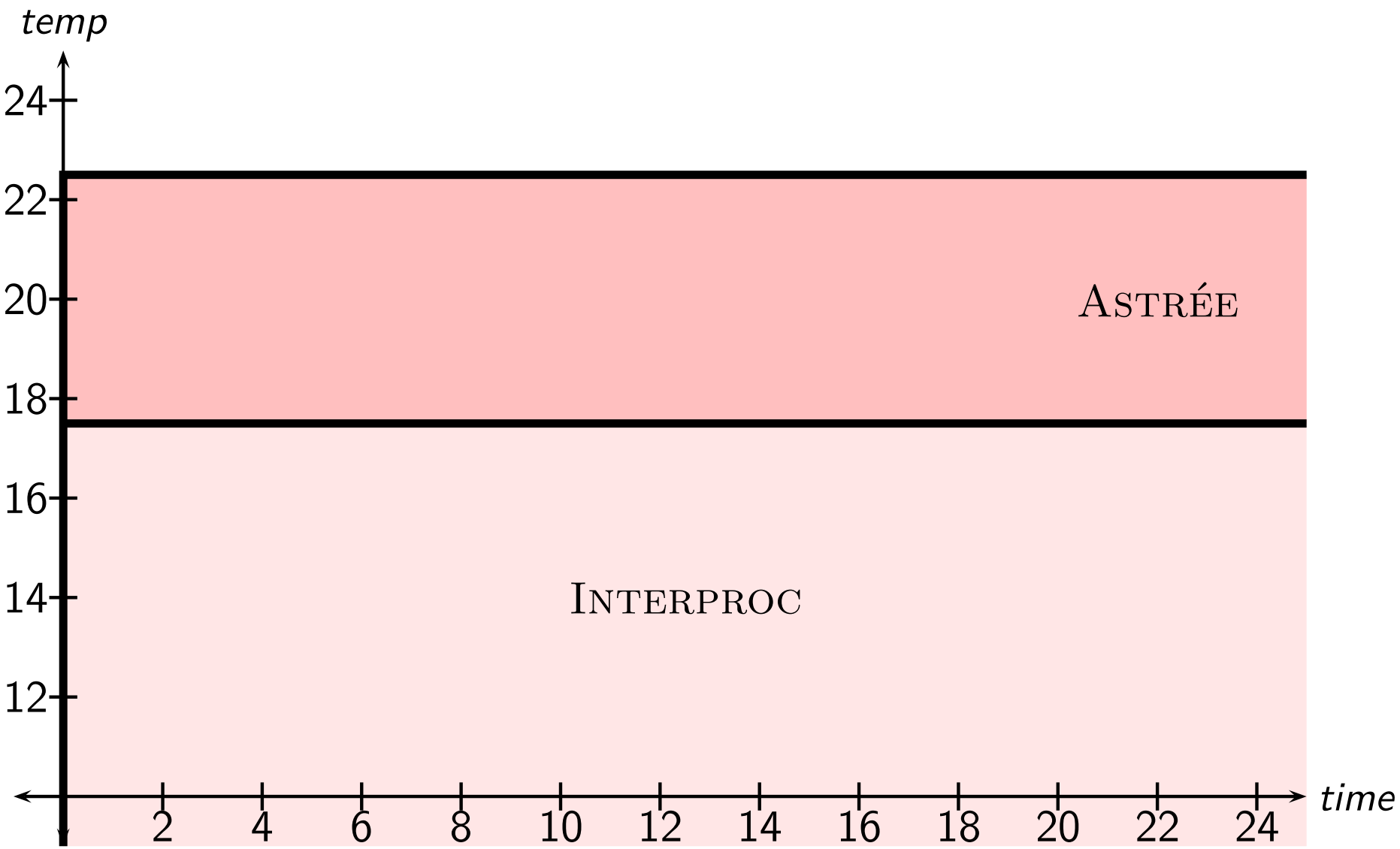
Thermostat

```
assume(temp_ext=14 && 16 ≤ temp ≤ 17);
while(1)
{
  time = 0;
  while(temp ≤ 22.0)                                  ⎫
  {                                                   ⎪
    temp = 1.0 + (15.0*temp - temp_ext)/16.0;         ⎬ heater on
    time++;                                           ⎪
  }                                                   ⎭
  time = 0;
  while(temp ≥ 18.0)                                  ⎫
  {                                                   ⎪
    temp = (15.0*temp - temp_ext)/16.0;               ⎬ heater off
    time++;                                           ⎪
  }                                                   ⎭
}
```

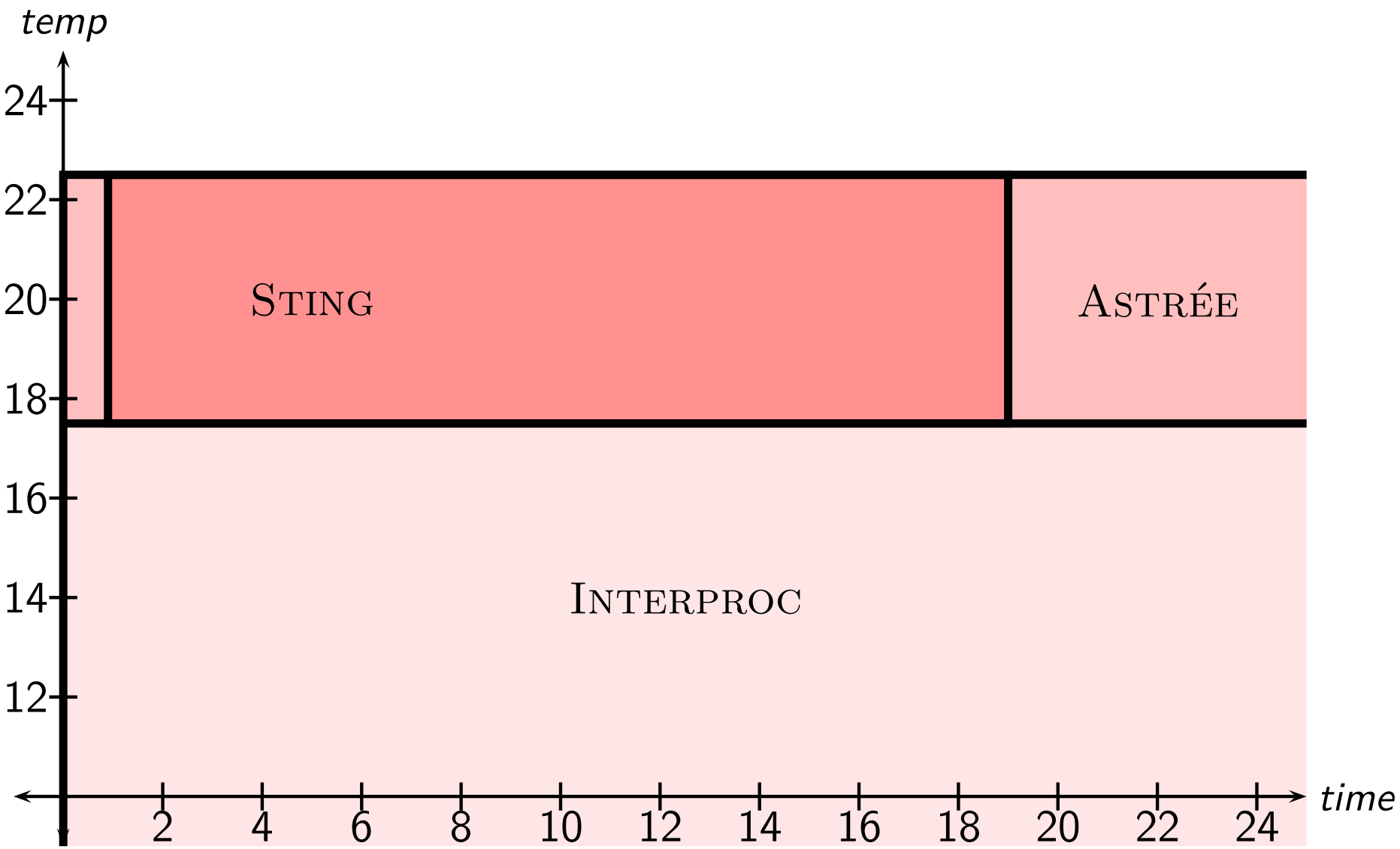
Thermostat


temp
24
22
20
18
16
14
12
Interproc
2
4
6
8
10
12
14
16
18
20
22
24
time

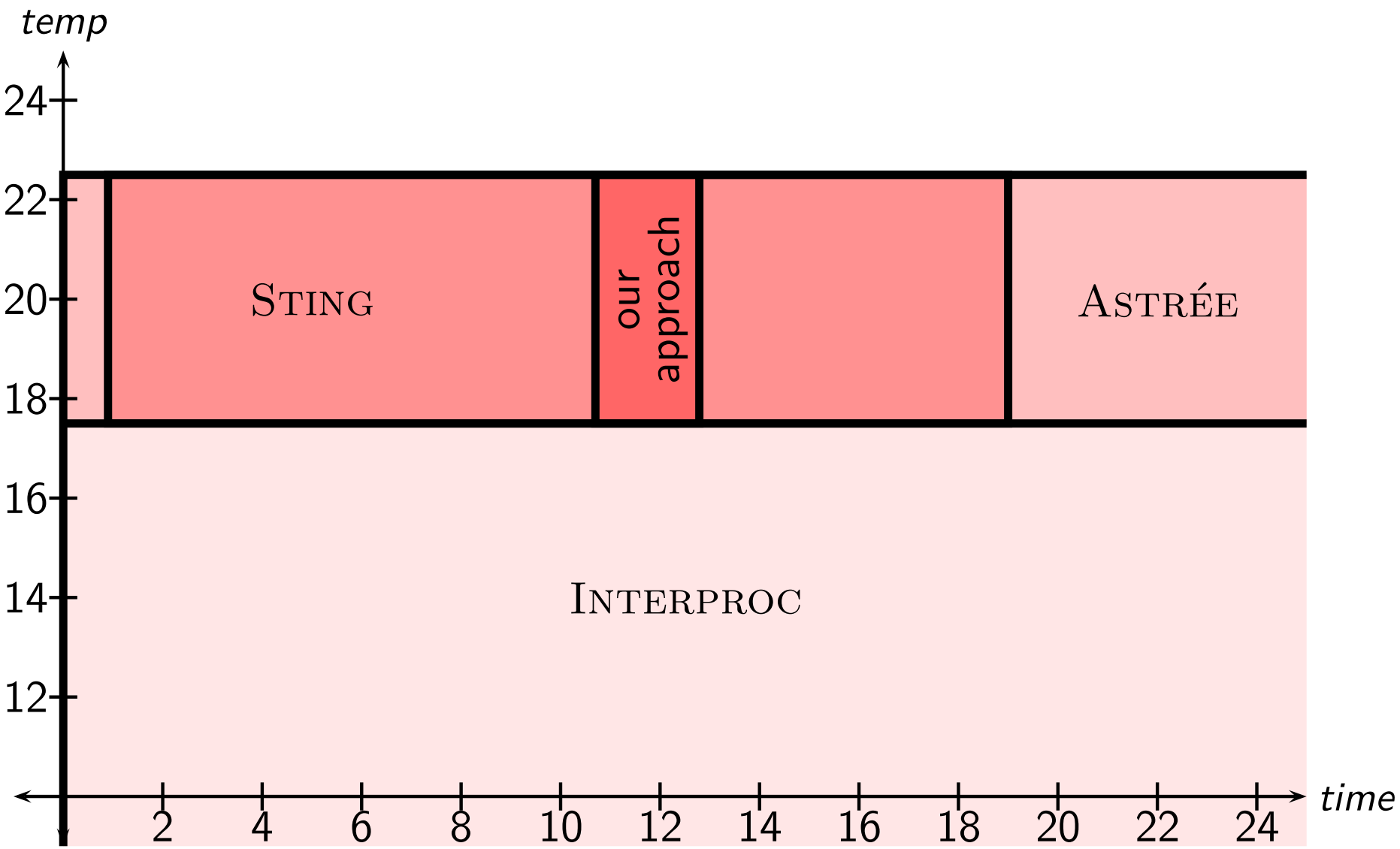
Thermostat


temp
24
22
20
18
16
14
12
Astrée
Interproc
2
4
6
8
10
12
14
16
18
20
22
24
time

Thermostat


temp
24
22
20
18
16
14
12
Sting
Astrée
Interproc
2
4
6
8
10
12
14
16
18
20
22
24
time

Thermostat


temp
24
22
20
18
16
14
12
Sting
our approach
Astrée
Interproc
2
4
6
8
10
12
14
16
18
20
22
24
time

Existing Techniques

Standard polyhedral analysis (Cousot&Halbwachs 1978)

- Compute iteratively $(\alpha \circ \tau \circ \gamma)^*(X)$
- Not precise
- INTERPROC, ...
- Thresholds, landmarks, abstract acceleration, ...
  improve precision for linear behavior

Non-line ... 004)

- Solvin
- Relaxa ... g
  proble
- STING

All these methods compute
inductive invariants
in abstract domains too weak
to express precise invariants of linear systems.

Ellipsoid methods (Kurzhanski&Varaiya 2000, Feret 2004, ...)

- Restricted to stable loops
- ASTRÉE

...

Abstract Acceleration of Linear Loops

(Gonnord&Halbwachs 2006)

Approach

Given a linear loop

$$\tau \;=\; \left(G\vec{x} \leq \vec{h} \quad \rightarrow \quad \vec{x'} = A\vec{x} + \vec{b}\right)$$

provide an operator $^{\otimes}$ s.t.

$$\tau^{\otimes} \supseteq \alpha \circ \tau^{*} \circ \gamma$$

in the convex polyhedra domain.

Abstract Acceleration of Linear Loops

(Gonnord&Halbwachs 2006)

Approach

Given a linear loop

$$\tau \;=\; \left(G\vec{x} \leq \vec{h} \quad \rightarrow \quad \vec{x'} = A\vec{x} + \vec{b}\right)$$

provide an operator $^{\otimes}$ s.t.

$$\tau^{\otimes} \supseteq \alpha \circ \tau^{*} \circ \gamma \qquad\qquad \text{(vs. Kleene: } (\alpha \circ \tau \circ \gamma)^{*}\text{)}$$

in the convex polyhedra domain.

Abstract Acceleration of Linear Loops

(Gonnord&Halbwachs 2006)

Approach

Given a linear loop

$$\tau \;=\; \left(G\vec{x} \leq \vec{h} \quad \rightarrow \quad \vec{x'} = A\vec{x} + \vec{b}\right)$$

provide an operator $^{\otimes}$ s.t.

$$\tau^{\otimes} \supseteq \alpha \circ \tau^{*} \circ \gamma \qquad\qquad \text{(vs. Kleene: } (\alpha \circ \tau \circ \gamma)^{*}\text{)}$$

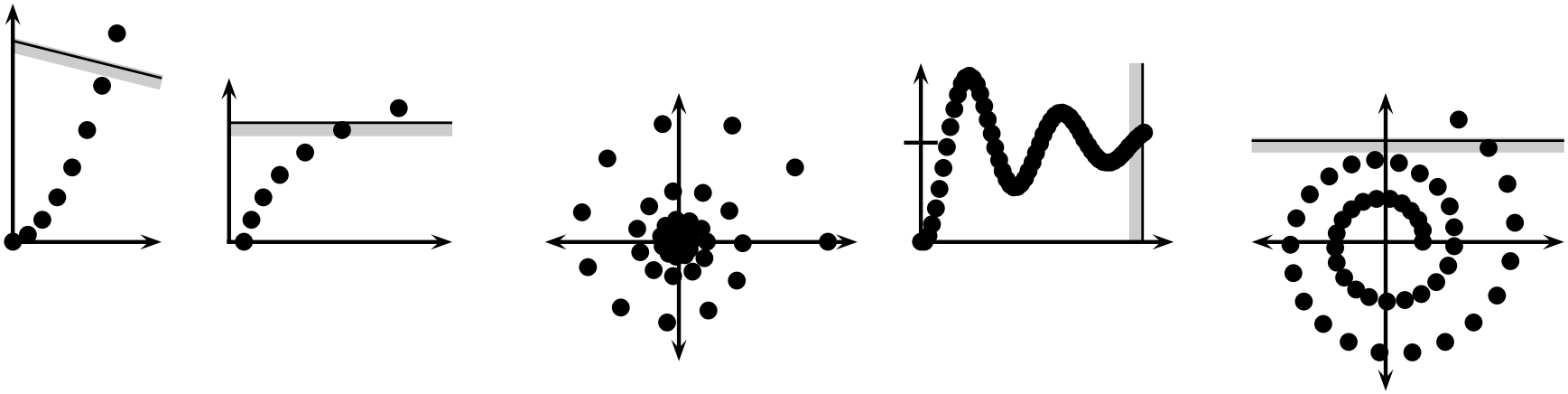
in the convex polyhedra domain.

Then, given a polyhedron X

$$\tau^{\otimes}(X) \text{ is a precise over-approximation of } \tau^{*}(X)$$

Contribution

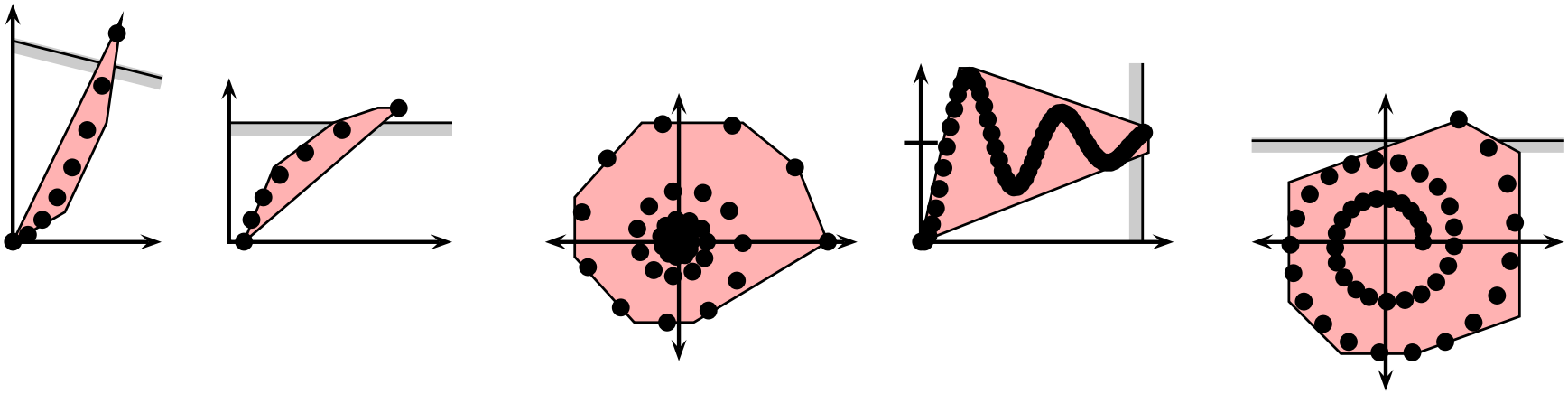
Abstract acceleration of any linear loop τ



Integrates smoothly into a static analyzer

Contribution

Abstract acceleration of any linear loop τ

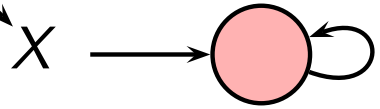


Integrates smoothly into a static analyzer

Our Approach

Loops without guard

convex polyhedron



$true \rightarrow \vec{x'} = A\vec{x}$

template polyhedron

$$\mathcal{A} \quad \supseteq \quad A^* \quad = \{A^k \mid k \geq 0\}$$

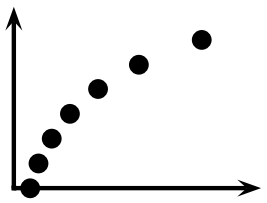
$$\underbrace{\mathcal{A}X} \quad \supseteq \quad A^*X \quad = \{A^k\vec{x} \mid k \geq 0, \vec{x} \in X\}$$

convex polyhedron

Loop without Guard: Example

Example

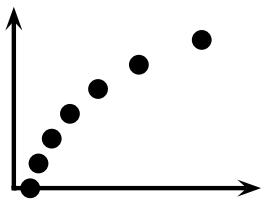
```
x=x0; y=y0;
while true do
  x=1.5*x;
  y=y+1;
done
```



Loop without Guard: Example

Example

```
x=x0; y=y0;
while true do
  x=1.5*x;
  y=y+1;
done
```



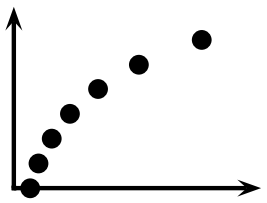
Homogeneization: we introduce ξ for constant coefficient

```
x=x0; y=y0; ξ=1;
while true do
  x=1.5*x;
  y=y+ξ;
done
```

Loop without Guard: Example

Example

```
x=x0; y=y0;
while true do
  x=1.5*x;
  y=y+1;
done
```



Homogeneization: we introduce ξ for constant coefficient

```
x=x0; y=y0; ξ=1;
while true do
  x=1.5*x;
  y=y+ξ;
done
```

Loop: true $\rightarrow \begin{pmatrix} x' \\ y' \\ \xi' \end{pmatrix} = A \begin{pmatrix} x \\ y \\ \xi \end{pmatrix}$ with $A = \begin{pmatrix} 1.5 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$

Loop without Guard: Example

Approximating A^*

$$A^k = \begin{pmatrix} 1.5^k & 0 & 0 \\ 0 & 1 & k \\ 0 & 0 & 1 \end{pmatrix}$$

Loop without Guard: Example

Approximating A^*

$$A^k = \begin{pmatrix} 1.5^k & 0 & 0 \\ 0 & 1 & k \\ 0 & 0 & 1 \end{pmatrix}$$

We look for a template polyhedron $\varphi_{\mathcal{A}}(m_1, m_2)$ such that

$$\mathcal{A} = \left\{ \begin{pmatrix} m_1 & 0 & 0 \\ 0 & 1 & m_2 \\ 0 & 0 & 1 \end{pmatrix} \middle| \varphi_{\mathcal{A}}(m_1, m_2) \right\} \supseteq A^*$$

Loop without Guard: Example

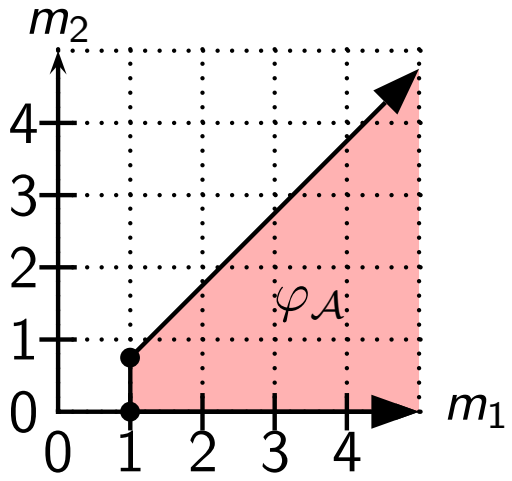
Approximating A^*

$$A^k = \begin{pmatrix} 1.5^k & 0 & 0 \\ 0 & 1 & k \\ 0 & 0 & 1 \end{pmatrix}$$

We look for a template polyhedron $\varphi_{\mathcal{A}}(m_1, m_2)$ such that

$$\mathcal{A} = \left\{ \begin{pmatrix} m_1 & 0 & 0 \\ 0 & 1 & m_2 \\ 0 & 0 & 1 \end{pmatrix} \middle| \varphi_{\mathcal{A}}(m_1, m_2) \right\} \supseteq A^*$$

Let's consider octagonal templates over m_1 and m_2:



$$\varphi_{\mathcal{A}} = \begin{cases} m_1 \geq 1 \\ m_2 \geq 0 \\ m_1 - m_2 \geq 0.25 \end{cases}$$

Loop without Guard: Example

Approximating $\mathcal{A}X$ with a convex polyhedron

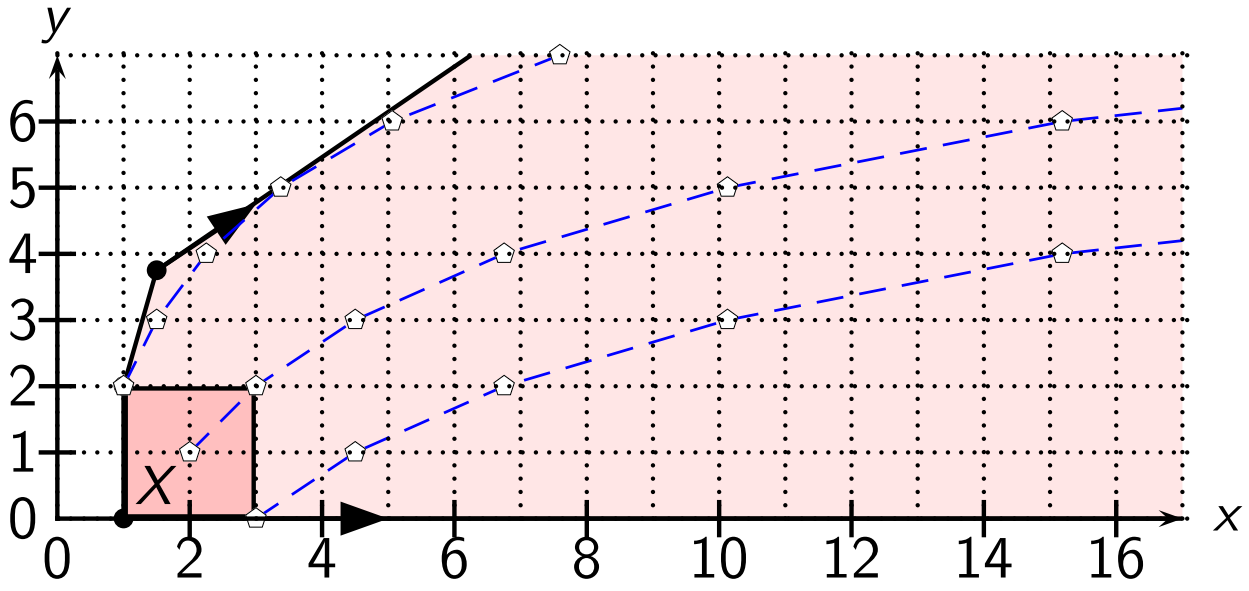
Properly multiplying generators (vertices & rays) of $\mathcal{A}$ by generators of X

Loop without Guard: Example

Approximating $\mathcal{A}X$ with a convex polyhedron

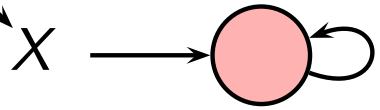
Properly multiplying generators (vertices & rays) of $\mathcal{A}$ by generators of X

Result on our example:



Loop Without Guard: General Case

convex polyhedron



$true \rightarrow \vec{x'} = A\vec{x}$

template polyhedron

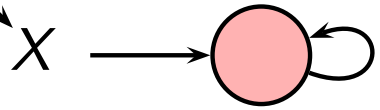
$$\mathcal{A} \quad \supseteq \quad A^* \quad = \{A^k \mid k \geq 0\}$$

$$\underbrace{\mathcal{A}X} \quad \supseteq \quad A^*X \quad = \{A^k\vec{x} \mid k \geq 0, \vec{x} \in X\}$$

convex polyhedron

Loop Without Guard: General Case

convex polyhedron



$true \rightarrow \vec{x'} = A\vec{x}$

template polyhedron

(1)

$$\mathcal{A} \supseteq A^* = \{A^k \mid k \geq 0\}$$

(2)

$$\underbrace{\mathcal{A}X} \supseteq A^*X = \{A^k\vec{x} \mid k \geq 0, \vec{x} \in X\}$$

convex polyhedron

(1) Over-approximation of A^* by $\mathcal{A}$

(a) Closed-Form Expression of A^k

real Jordan
normal form

$$A = Q^{-1} J Q$$

(1) Over-approximation of A^* by $\mathcal{A}$

(a) Closed-Form Expression of A^k

real Jordan
normal form

$$A^k = Q^{-1} J^k Q$$

(1) Over-approximation of A^* by $\mathcal{A}$

(a) Closed-Form Expression of A^k

real Jordan normal form

$$A^k = Q^{-1} J^k Q$$

coefficients of the form $\binom{k}{p} \lambda^{k-p} \cos((k-p)\theta - r\frac{\pi}{2})$

(1) Over-approximation of A^* by $\mathcal{A}$

(a) Closed-Form Expression of A^k

real Jordan normal form

$$A^k = Q^{-1} J^k Q$$

at most n distinct coefficients

coefficients of the form $\binom{k}{p} \lambda^{k-p} \cos((k-p)\theta - r\frac{\pi}{2})$

(1) Over-approximation of A^* by $\mathcal{A}$

(a) Closed-Form Expression of A^k

real Jordan normal form

$$A^k = Q^{-1} J^k Q$$

at most n distinct coefficients

coefficients of the form $\binom{k}{p} \lambda^{k-p} \cos((k-p)\theta - r\frac{\pi}{2})$

(b) Over-approximation of J^* by a template polyhedron $\mathcal{J}$

vector of coefficients of J^k

$$\{\vec{m} \mid \ell_i \leq e_i(\vec{m}) \leq u_i\}$$

(1) Over-approximation of A^* by $\mathcal{A}$

(a) Closed-Form Expression of A^k

real Jordan normal form

$$A^k = Q^{-1} J^k Q$$

at most n distinct coefficients

coefficients of the form $\binom{k}{p} \lambda^{k-p} \cos((k-p)\theta - r\frac{\pi}{2})$

(b) Over-approximation of J^* by a template polyhedron $\mathcal{J}$

vector of coefficients of J^k

template expressions (e.g. logahedra)

$$\{\vec{m} \mid \ell_i \leq e_i(\vec{m}) \leq u_i\}$$

(1) Over-approximation of A^* by $\mathcal{A}$

(a) Closed-Form Expression of A^k

real Jordan normal form

$$A^k = Q^{-1} J^k Q$$

at most n distinct coefficients

coefficients of the form $\binom{k}{p} \lambda^{k-p} \cos((k-p)\theta - r\frac{\pi}{2})$

(b) Over-approximation of J^* by a template polyhedron $\mathcal{J}$

vector of coefficients of J^k

template expressions (e.g. logahedra)

$$\{\vec{m} \mid \ell_i \leq e_i(\vec{m}) \leq u_i\}$$

lower and upper bounds computed by asymptotic analysis techniques

(1) Over-approximation of A^* by $\mathcal{A}$

(a) Closed-Form Expression of A^k

real Jordan normal form

$$A^k = Q^{-1} J^k Q$$

at most n distinct coefficients

coefficients of the form $\binom{k}{p} \lambda^{k-p} \cos((k-p)\theta - r\frac{\pi}{2})$

(b) Over-approximation of J^* by a template polyhedron $\mathcal{J}$

vector of coefficients of J^k

template expressions (e.g. logahedra)

$$\{\vec{m} \mid \ell_i \leq e_i(\vec{m}) \leq u_i\}$$

lower and upper bounds computed by asymptotic analysis techniques

(c) $\mathcal{A} = Q^{-1}\mathcal{J}Q$

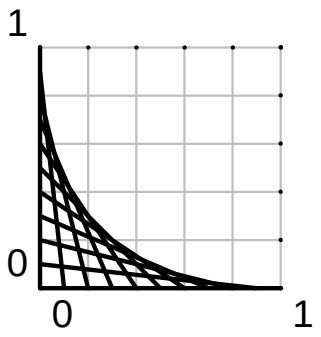
(2) Computing $\mathcal{A}X$

$$\mathcal{A}X = \{A\vec{x} \mid A \in \mathcal{A} \wedge \vec{x} \in X\}$$

(2) Computing $\mathcal{A}X$

$$\mathcal{A}X = \{A\vec{x} \mid A \in \mathcal{A} \wedge \vec{x} \in X\}$$

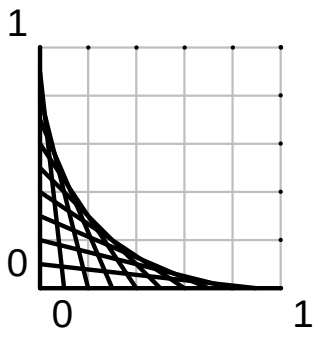
In general: neither convex nor a polyhedron!



(2) Computing $\mathcal{A}X$

$$\mathcal{A}X = \{A\vec{x} \mid A \in \mathcal{A} \wedge \vec{x} \in X\}$$

In general: neither convex nor a polyhedron!



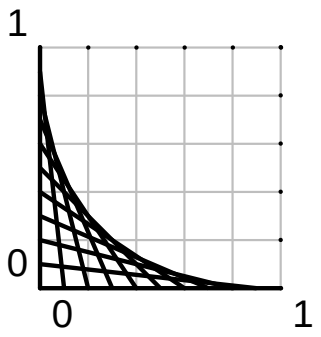
Over-approximate $\mathcal{A}X$

- $\mathcal{A}X \subseteq \text{ConvexSpan}(\underbrace{V_1 V_2}_{\text{vertices}}, \underbrace{V_1 R_2 \cup R_1 V_2 \ \cup R_1 R_2}_{\text{rays}})$

(2) Computing $\mathcal{A}X$

$$\mathcal{A}X = \{A\vec{x} \mid A \in \mathcal{A} \wedge \vec{x} \in X\}$$

In general: neither convex nor a polyhedron!



Over-approximate $\mathcal{A}X$

- $\mathcal{A}X \subseteq \text{ConvexSpan}(\underbrace{V_1 V_2}_{\text{vertices}}, \underbrace{V_1 R_2 \cup R_1 V_2 \ \cup R_1 R_2}_{\text{rays}})$

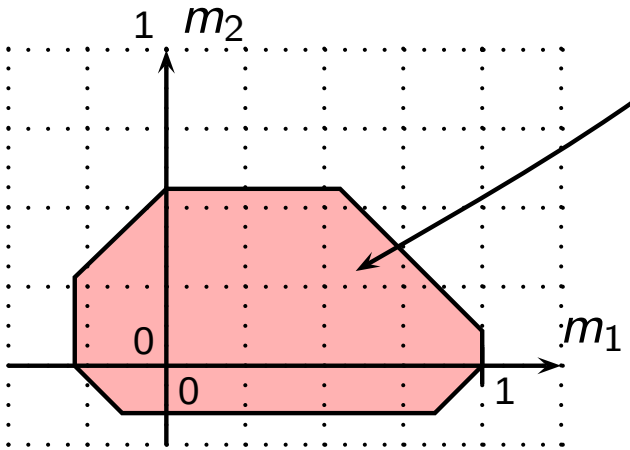
$\star$ Best convex over-approximation if $R_1 = R_2 = \emptyset$

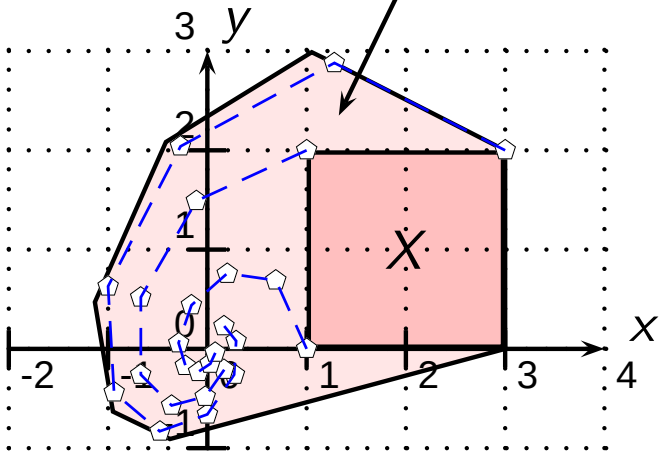
Loop without Guard: Example 2

$$\tau \quad = \quad \mathit{true} \;\rightarrow\; \vec{x'} = \begin{pmatrix} 0.8\cos\frac{\pi}{6} & -0.8\sin\frac{\pi}{6} & 0 \\ 0.8\sin\frac{\pi}{6} & 0.8\cos\frac{\pi}{6} & 0 \\ 0 & 0 & 1 \end{pmatrix} \vec{x}$$

$$A^* \subseteq \mathcal{A} = \left\{ \begin{pmatrix} m_1 & -m_2 & 0 \\ m_2 & m_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \middle| \; \varphi_{\mathcal{A}}(m_1, m_2) \right\} \qquad \tau^{\otimes}(X) = \mathcal{A}X$$

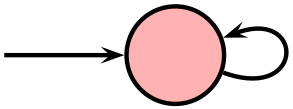
With octagonal template:





Loop with Guard

convex polyhedron

X  $G\vec{x} \leq 0 \quad \rightarrow \quad \vec{x'} = A\vec{x}$

maximum number of iterations

with $N = \text{argmin}_{k \geq 0} GA^k\vec{x} > 0$

template polyhedron

$$\mathcal{A}^{[0,N]} \supseteq A^{[0,N]} \qquad = \{A^k \mid 0 \leq k \leq N\}$$

$$\underbrace{\mathcal{A}^{[0,N]}X}_{} \supseteq A^{[0,N]}X$$

convex polyhedron

Loop with Guard: Example

Program

We want to compute a loop invariant of

```
x=x0; y=y0;
while y<=3 do
   x = 1.5*x;
   y = y+1;
done
```

with $X(x_0, y_0) = x_0 \in [1, 3] \wedge y_0 \in [0, 2]$

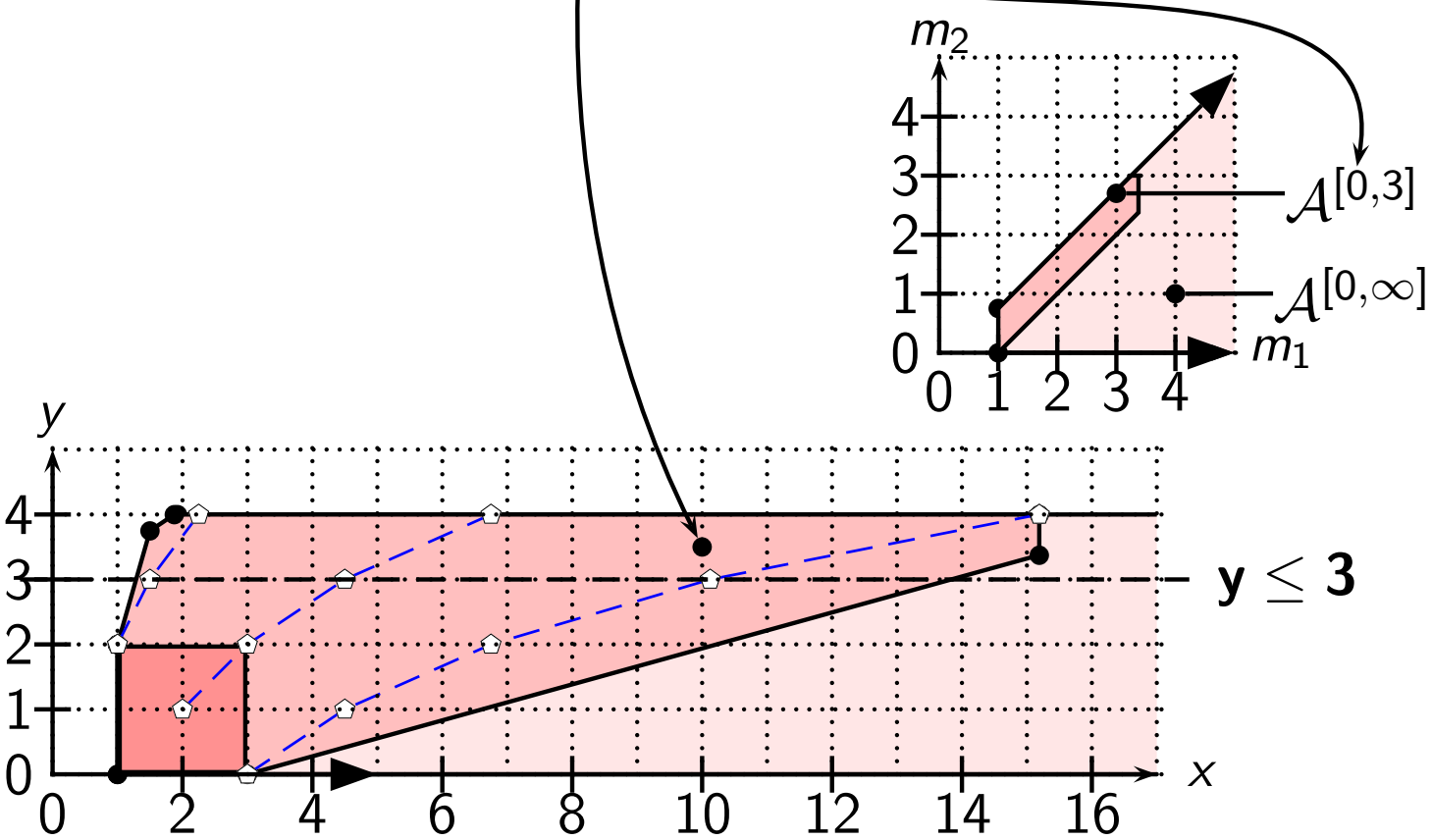
Computing the maximum number of iterations

$y_0 \in [0, 2] \wedge y' = y + 1 \wedge y \leq 3 \implies$ max nb. of iterations $N = 4$

Loop with Guard: Example

Exploiting the maximum number of iterations

We compute $\underbrace{X \sqcup \tau(\mathcal{A}^{[0,3]} \cdot (X \sqcap G))}$



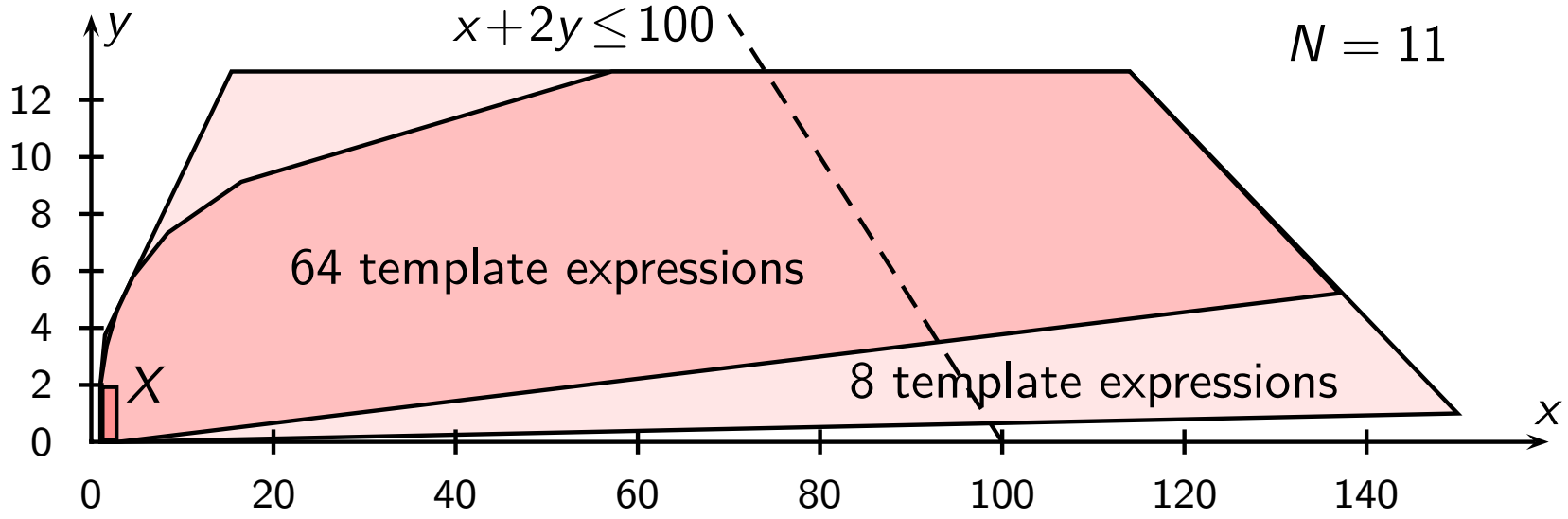
Influence of Template for Approximating $\mathcal{A}$

Program

```
x=x0; y=y0;
while x+2*y <= 100 do
  x = 1.5*x; y = y+1;
done
```

with $X(x_0, y_0) \;=\; x_0 \in [1,3] \wedge y_0 \in [0,2]$

Result



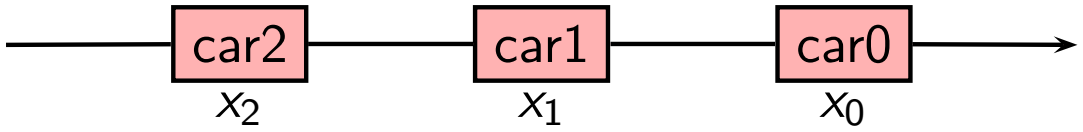
Implementation and Experiments

Implementation

- Based on
  - APRON for polyhedral computations
  - SAGE for computation of Jordan normal form
- Loops with conditionals transformed into multiple self-loops
- Acceleration of inner loops, widening for outer loops

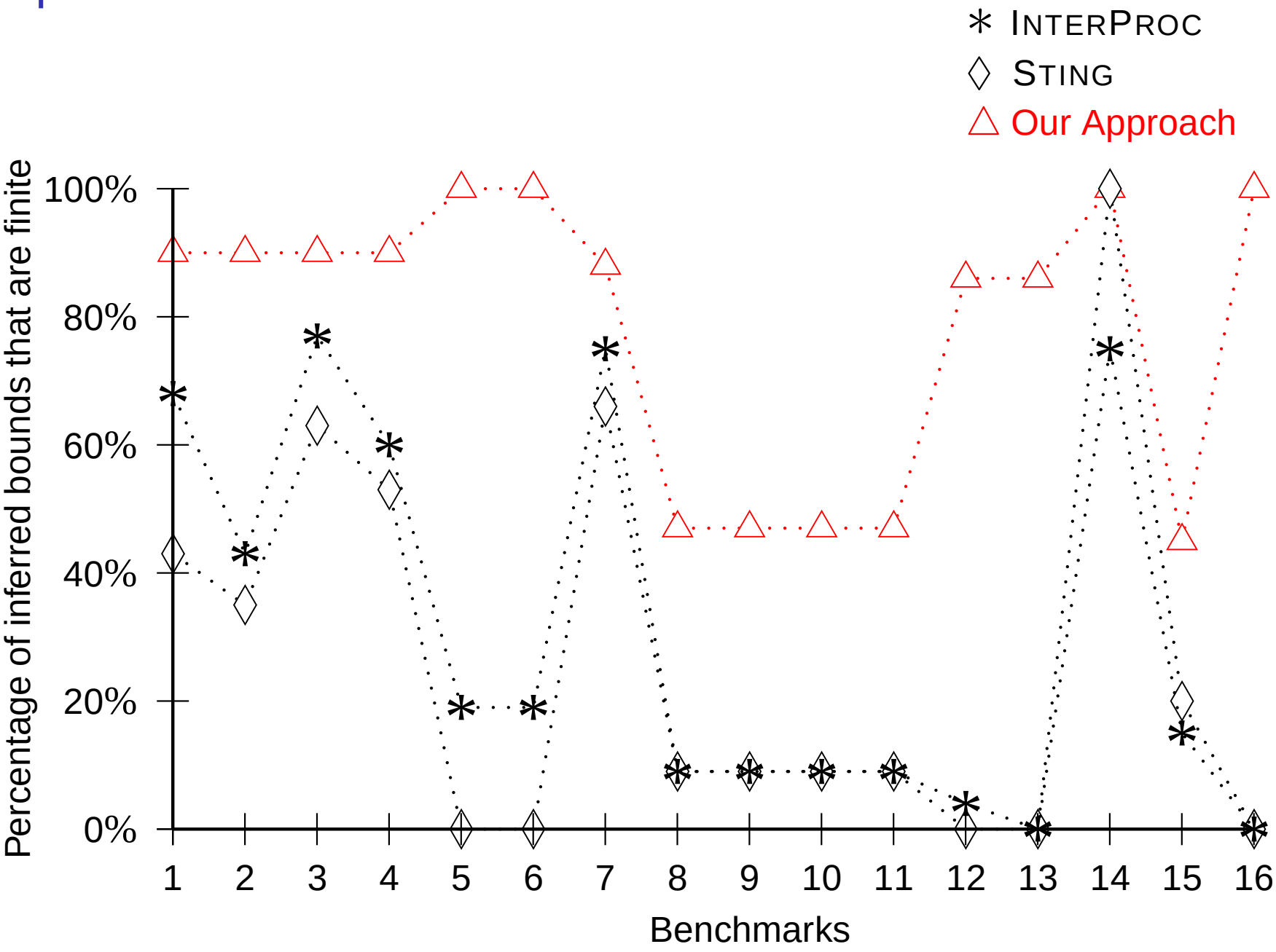
Benchmarks

- Small filters (single loops)
- Forced oscillator with stable and unstable switching modes (nested loops)
- Thermostat system (nested loops)
- Car convoy

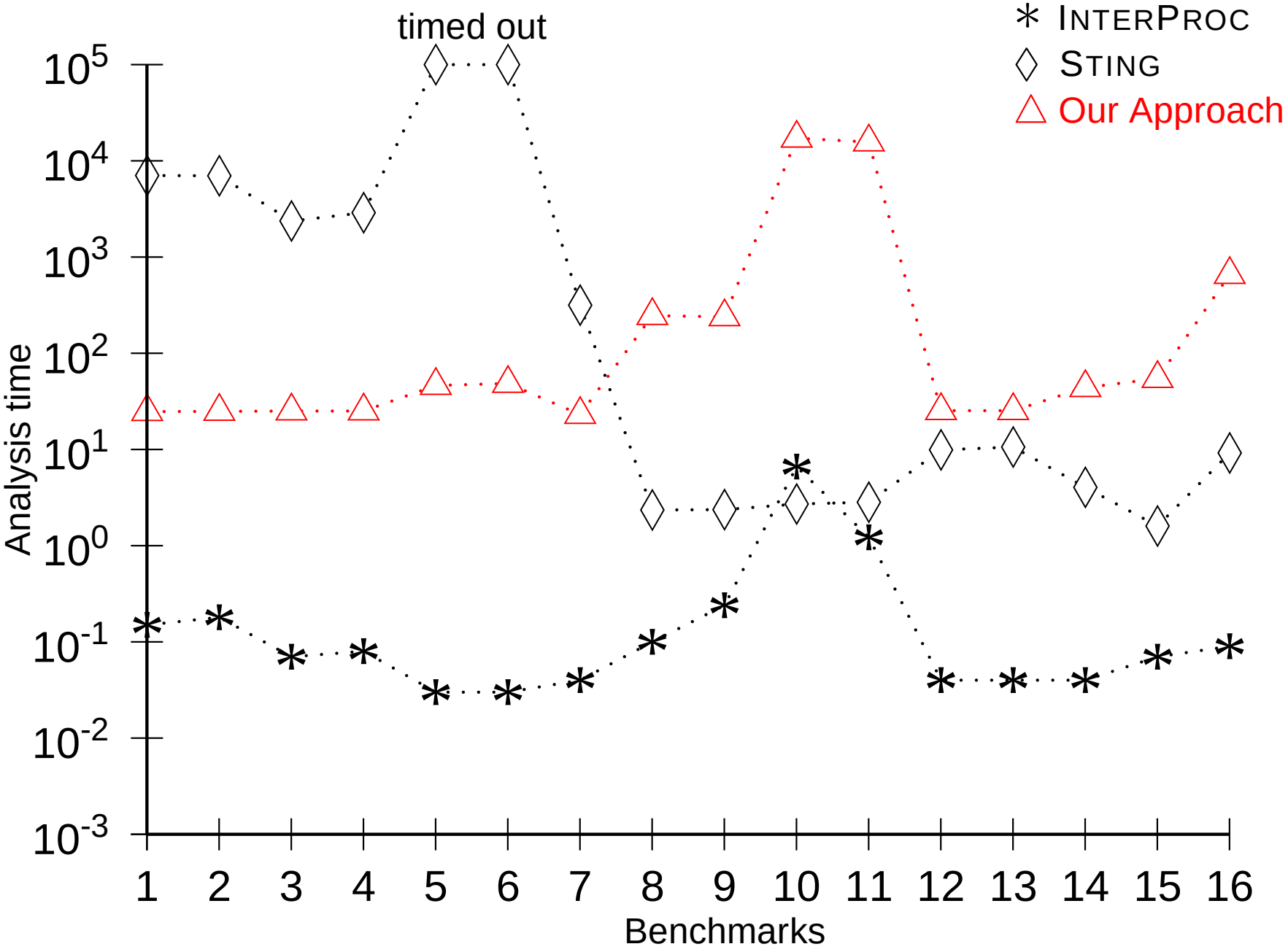


$$\text{for all } i \in [1, N-1] : \ddot{x}_i = c(\dot{x}_{i-1} - \dot{x}_i) + d(x_{i-1} - x_i - 50)$$

Experimental Results: Inferred Bounds



Experimental Results: Analysis Time



Summary and Prospects

Contribution

- Precise polyhedral overapproximation of general linear loops
  - Matrix abstract domain
  - Precise loop bound estimation
- Integrates smoothly into static analyzer

Summary and Prospects

Contribution

- Precise polyhedral overapproximation of general linear loops
  - Matrix abstract domain
  - Precise loop bound estimation
- Integrates smoothly into static analyzer

Rationale of the Approach

- Use non-linear deductions within polyhedra analysis

Summary and Prospects

Contribution

- Precise polyhedral overapproximation of general linear loops
  - Matrix abstract domain
  - Precise loop bound estimation
- Integrates smoothly into static analyzer

Rationale of the Approach

- Use non-linear deductions within polyhedra analysis

Ongoing Work

- Open systems (with inputs)

Applications

- Hybrid systems with linear differential equations
- Stability/termination analysis