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.

```c
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.

```c
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 \((\alpha \circ \tau \circ \gamma)^* (X)\)
- \textsc{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
- \textsc{STING}

Ellipsoidal methods (Kurzhanski & Varaiya 2000, Feret 2004, ...)
- Restricted to stable loops
- \textsc{Astrée}

...
assume(temp\text{ext}=14 \land 16 \leq \text{temp} \leq 17);

\textbf{while}(1)
{
    \hspace{1em} \text{time} = 0;
    \hspace{1em} \textbf{while}(\text{temp} \leq 22.0)
    {
        \hspace{2em} \text{temp} = 1.0 + (15.0*\text{temp} - \text{temp}\text{ext})/16.0; \quad \text{heater on}
        \hspace{2em} \text{time}++; \\
    }
    \hspace{1em} \text{time} = 0;
    \hspace{1em} \textbf{while}(\text{temp} \geq 18.0)
    {
        \hspace{2em} \text{temp} = (15.0*\text{temp} - \text{temp}\text{ext})/16.0; \quad \text{heater off}
        \hspace{2em} \text{time}++; \\
    }
}
}
Thermostat

interproc
Thermostat

![Diagram showing temperature and time with different processes (Sting, Astrée, Interproc).]
Thermostat

```
2 4 6 8 10 12 14 16 18 20 22 24
12 14 16 18 20 22 24
```

```
temp
```

```
time
```

AstréeSting

our

approach

Interproc

Astrée
Existing Techniques

Standard polyhedral analysis (Cousot & Halbwachs 1978)
- Compute iteratively \((\alpha \circ \tau \circ \gamma)^*(X)\)
- Not precise
- \textbf{INTERPROC, ...}
- Thresholds, landmarks, abstract acceleration, ...
  improve precision for linear behavior

Non-linear constraint solving (Colon et al. 2003, Sankaranarayanan et al. 2004)
- Solving \(X = \tau(X)\) over convex polyhedra
- Relaxation and reduction to non-linear constraint solving
- \textbf{Sting} (Ellipsoid methods)
  - \textbf{ASTRÉE}
  - Restricted to stable loops
  - \textbf{All these methods compute inductive invariants in abstract domains too weak to express precise invariants of linear systems.}
Abstract Acceleration of Linear Loops
(Gonnord & Halbwachs 2006)

Approach

Given a linear loop

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

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
\[ \tau = (G\vec{x} \leq \vec{h} \implies \vec{x}' = A\vec{x} + \vec{b}) \]
provide an operator \( \otimes \) s.t.
\[ \tau \otimes \supseteq \alpha \circ \tau^* \circ \gamma \]
(vs. Kleene: \( (\alpha \circ \tau \circ \gamma)^* \))
in the convex polyhedra domain.
Abstract Acceleration of Linear Loops
(Gonnord&Halbwachs 2006)

Approach

Given a linear loop
\[ \tau = (G\vec{x} \leq \vec{h} \rightarrow \vec{x}' = A\vec{x} + \vec{b}) \]
provide an operator \( \otimes \) s.t.
\[ \tau \otimes \supseteq \alpha \circ \tau^* \circ \gamma \]
(vs. Kleene: \( (\alpha \circ \tau \circ \gamma)^* \))
in the convex polyhedra domain.

Then, given a polyhedron \( X \)
\[ \tau \otimes (X) \] is a precise over-approximation of \( \tau^*(X) \)
Contribution

Abstract acceleration of any linear loop $\tau$

Integrates smoothly into a static analyzer
Abstract acceleration of any linear loop $\tau$

Integrates smoothly into a static analyzer
Our Approach

Loops without guard

convex polyhedron

\[ X \xrightarrow{\text{true}} \vec{x}' = A\vec{x} \]

template polyhedron

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

\[ \mathcal{A}X \supseteq \mathcal{A}^*X = \{ A^k\vec{x} \mid k \geq 0, \vec{x} \in X \} \]
Loop without Guard: Example

Example

\[
x=x_0; \quad y=y_0;
\]
\[
\text{while true do}
\]
\[
\quad x=1.5x;
\]
\[
\quad y=y+1;
\]
\[
\text{done}
\]
Loop without Guard: Example

Example

\begin{verbatim}
x=x0; y=y0;
while true do
  x=1.5*x;
  y=y+1;
end
\end{verbatim}

Homogeneization: we introduce $\xi$ for constant coefficient

\begin{verbatim}
x=x0; y=y0; $\xi=1$;
while true do
  x=1.5*x;
  y=y+$\xi$;
end
\end{verbatim}
Loop without Guard: Example

Example

\[
x = x_0; \quad y = y_0;
\]

while true do
\[
\begin{align*}
x &= 1.5 * x; \\
y &= y + 1;
\end{align*}
\]
done

Homogeneization: we introduce $\xi$ for constant coefficient

\[
x = x_0; \quad y = y_0; \quad \xi = 1;
\]

while true do
\[
\begin{align*}
x &= 1.5 * x; \\
y &= y + \xi;
\end{align*}
\]
done

\[
\text{Loop: true} \rightarrow \begin{pmatrix} x' \\ y' \\ \xi' \end{pmatrix} = A \begin{pmatrix} x \\ y \\ \xi \end{pmatrix}
\]

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 $\varphi_A(m_1, m_2)$ such that

$$\mathcal{A} = \left\{ \begin{pmatrix} m_1 & 0 & 0 \\ 0 & 1 & m_2 \\ 0 & 0 & 1 \end{pmatrix} \, \mid \, \varphi_A(m_1, m_2) \right\} \supseteq A^*$$
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 $\varphi_A(m_1, m_2)$ such that

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

Let’s consider octagonal templates over $m_1$ and $m_2$:

$$\varphi_A = \begin{cases} m_1 \geq 1 \\ m_2 \geq 0 \\ m_1 - m_2 \geq 0.25 \end{cases}$$
Loop without Guard: Example

Approximating $AX$ with a convex polyhedron
Properly multiplying generators (vertices & rays) of $A$ by generators of $X$
Loop without Guard: Example

Approximating $AX$ with a convex polyhedron

Properly multiplying generators (vertices & rays) of $A$ by generators of $X$

Result on our example:
Loop Without Guard: General Case

convex polyhedron

\[ X \xrightarrow{\text{true}} \vec{x}' = A\vec{x} \]

template polyhedron

\[ A \supseteq A^* = \{ A^k \mid k \geq 0 \} \]

\[ AX \supseteq A^*X = \{ A^k \vec{x} \mid k \geq 0, \vec{x} \in X \} \]
Loop Without Guard: General Case

convex polyhedron

\[ X \rightarrow \text{true} \rightarrow \vec{x}' = A\vec{x} \]

template polyhedron

\[ A \supseteq A^* = \{A^k \mid k \geq 0\} \]

(1)

\[ AX \supseteq A^*X = \{A^k\vec{x} \mid k \geq 0, \vec{x} \in X\} \]

(2)
(1) Over-approximation of $A^*$ by $\mathcal{A}$

(a) Closed-Form Expression of $A^k$

real Jordan normal form

$$A = Q^{-1}JQ$$
(1) Over-approximation of $A^*$ by $A$

(a) Closed-Form Expression of $A^k$

real Jordan normal form

$$A^k = Q^{-1} J^k Q$$
(1) Over-approximation of $A^*$ by $A$

(a) Closed-Form Expression of $A^k$

real Jordan normal form

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

coefficients of the form

$$\binom{k}{p} \lambda^{k-p} \cos((k-p)\theta - r \pi)$$
(1) Over-approximation of $A^*$ by $A$

(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 \( A \)

(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 \( J^k \)

\[
\{ \tilde{m} \mid \ell_i \leq e_i(\tilde{m}) \leq u_i \}
\]
(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 $J^k$

\[ \left\{ \vec{m} \mid \ell_i \leq e_i(\vec{m}) \leq u_i \right\} \]

template expressions (e.g. logahedra)
(1) Over-approximation of $A^*$ by $A$

(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

\[ \binom{k}{p} \lambda^{k-p} \cos \left( (k-p)\theta - r \frac{\pi}{2} \right) \]

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

vector of coefficients of $J^k$

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

template expressions (e.g. logahedra)

lower and upper bounds computed by asymptotic analysis techniques
(1) Over-approximation of $A^*$ by $A$

(a) Closed-Form Expression of $A^k$

real Jordan normal form

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

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 $J^k$

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

template expressions (e.g. logahedra)

lower and upper bounds computed by asymptotic analysis techniques

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

$$\mathcal{AX} = \{A\vec{x} \mid A \in \mathcal{A} \land \vec{x} \in X\}$$
(2) Computing $\mathcal{AX}$

\[\mathcal{AX} = \{A\vec{x} \mid A \in \mathcal{A} \land \vec{x} \in X\}\]

In general: neither convex nor a polyhedron!
(2) Computing $\mathcal{AX}$

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

In general: neither convex nor a polyhedron!

Over-approximate $\mathcal{AX}$

- $\mathcal{AX} \subseteq \text{ConvexSpan}( \{ V_1, V_2 \}, \{ V_1R_2 \cup R_1V_2 \cup R_1R_2 \} )$

  vertices

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

\[ \mathcal{A}X = \{ A\vec{x} | A \in \mathcal{A} \land \vec{x} \in X \} \]

In general: neither convex nor a polyhedron!

Over-approximate $\mathcal{A}X$

- $\mathcal{A}X \subseteq \text{ConvexSpan}(\underbrace{V_1, V_2}_{\text{vertices}}, \underbrace{V_1R_2 \cup R_1, V_2 \cup R_1R_2}_{\text{rays}})$

- Best convex over-approximation if $R_1 = R_2 = \emptyset$
Loop without Guard: Example 2

\[ \tau = true \rightarrow \vec{x}' = \begin{pmatrix} 0.8 \cos \frac{\pi}{6} & -0.8 \sin \frac{\pi}{6} & 0 \\ 0.8 \sin \frac{\pi}{6} & 0.8 \cos \frac{\pi}{6} & 0 \\ 0 & 0 & 1 \end{pmatrix} \vec{x} \]

\[ A^* \subseteq A = \left\{ \begin{pmatrix} m_1 & -m_2 & 0 \\ m_2 & m_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mid \varphi_A(m_1, m_2) \right\} \]

\[ \tau \otimes (X) = AX \]

With octagonal template:
Loop with Guard

Convex polyhedron

\[ X \xrightarrow{G\vec{x} \leq 0} \vec{x}' = A\vec{x} \]

Maximum number of iterations

With \( N = \operatorname{argmin}_{k \geq 0} GA^k\vec{x} > 0 \)

Template polyhedron

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

\[ \mathcal{A}[0,N] X \supseteq \mathcal{A}[0,N] X \]

Convex polyhedron
Loop with Guard: Example

Program

We want to compute a loop invariant of

\[
\begin{align*}
x &= x_0; \\
y &= y_0; \\
\text{while } y \leq 3 \text{ do} \\
x &= 1.5 \times x; \\
y &= y + 1; \\
\text{done}
\end{align*}
\]

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

Computing the maximum number of iterations

\( y_0 \in [0, 2] \land y' = y + 1 \land y \leq 3 \implies \max \text{ nb. of iterations } N = 4 \)
Loop with Guard: Example

Exploiting the maximum number of iterations

We compute

\[ X \sqcup \tau(A^{[0,3]} \cdot (X \cap G)) \]
Influence of Template for Approximating $\mathcal{A}$

Program

\[
\begin{align*}
    x &= x_0; \quad y = y_0; \\
    \text{while } x + 2y &\leq 100 \text{ do} \\
    x &= 1.5x; \quad y = y + 1; \\
    \text{done}
\end{align*}
\]

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

Result

![Graph showing the influence of the template for approximating $\mathcal{A}$ with a shaded region indicating the 64 template expressions and another indicating the 8 template expressions, with $N = 11$.](image)
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

![Car Convoy Diagram](image)

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

Percentage of inferred bounds that are finite

Benchmarks
Experimental Results: Analysis Time

Benchmarks:

- **INTERPROC**
- **STING**
- **Our Approach**

Timed out:
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