Using non-convex Approximations for Efficient Analysis of Timed Automata

F. Herbreteau\textsuperscript{1}, D. Kini\textsuperscript{2}, B. Srivathsan\textsuperscript{1} and I. Walukiewicz\textsuperscript{1}

LaBRI, Université de Bordeaux 1

Indian Institute of Technology Bombay, India

ACTS III, January 2011 - Chennai, India
Run: finite sequence of transitions,

\[
(s_0, 0, 0) \xrightarrow{0.4, a} (s_1, 0.4, 0) \xrightarrow{0.5, c} (s_3, 0.9, 0.5)
\]

▶ A run is accepting if it ends in a green state.
The problem we are interested in ...
The problem we are interested in ... 

Given a TA, does there exist an accepting run?

Theorem [AD94]

This problem is PSPACE-complete
First solution to this problem

Key idea: Partition the space of valuations into a finite number of regions
First solution to this problem

**Key idea:** Partition the space of valuations into a **finite** number of **regions**

- **Region:** set of valuations satisfying the **same** guards w.r.t. time
- **Finiteness:** Parametrized by **maximal constant**
First solution to this problem

Key idea: Partition the space of valuations into a finite number of regions

- Region: set of valuations satisfying the same guards w.r.t. time
- Finiteness: Parametrized by maximal constant
First solution to this problem

Key idea: Partition the space of valuations into a finite number of regions

- **Region**: set of valuations satisfying the same guards w.r.t. time

- **Finiteness**: Parametrized by maximal constant

Sound and complete

**Region graph** preserves state reachability
First solution to this problem

Key idea: Partition the space of valuations into a finite number of regions

- **Region**: set of valuations satisfying the same guards w.r.t. time
- **Finiteness**: Parametrized by maximal constant $O(|X|! \cdot M^{|X|})$ many regions!

Sound and complete

**Region graph** preserves state reachability
A more efficient solution...

Key idea: Maintain all valuations reaching a state via a path
A more efficient solution...

Key idea: Maintain **all valuations** reaching a state via a path

\[
x = y \geq 0
\]

\[
x = y \geq 7
\]

\[
y - x \geq 7 \quad (x \leq 5)
\]

\[
y - x \geq 7 \quad (y \geq 7)
\]

\[
x := 0
\]
A more efficient solution...

Key idea: Maintain **all valuations** reaching a state via a path

![Diagram](image)
A more efficient solution...

Key idea: Maintain **all valuations** reaching a state via a path

\[ \begin{align*}
(x &\leq 5) \\
(y &\geq 7) \\
x &:= 0
\end{align*} \]
A more efficient solution...

Key idea: **Maintain all valuations** reaching a state via a path.
A more efficient solution...

Key idea: Maintain all valuations reaching a state via a path
A more efficient solution...

Key idea: Maintain all valuations reaching a state via a path
A more efficient solution...

Key idea: Maintain all valuations reaching a state via a path
A more efficient solution...

**Key idea:** Maintain **all valuations** reaching a state via a path.

\[
x = y \geq 0 \quad x = y \geq 5 \quad y - x \geq 7
\]

\[
(x \leq 5) \quad (y \geq 7) \quad x := 0
\]
A more efficient solution...

**Key idea:** Maintain **all valuations** reaching a state via a path

\[
\begin{align*}
 x &\leq 5 \\
 y &\geq 7 \\
 x &:= 0
\end{align*}
\]
A more efficient solution...

Key idea: Maintain all valuations reaching a state via a path
A more efficient solution...

**Key idea:** Maintain **all valuations** reaching a state via a path

\[
x \leq 5 \quad (x \leq 5)
\]

\[
y \geq 7 \quad (y \geq 7)
\]

\[
x := 0
\]
A more efficient solution...

Key idea: Maintain all valuations reaching a state via a path
A more efficient solution...

Key idea: Maintain **all valuations** reaching a state via a path
Zones and zone graph

- **Zone**: set of valuations defined by conjunctions of constraints:
  - \( x \sim c \)
  - \( x - y \sim c \)
  - e.g. \( (x - y \geq 1) \land y < 2 \)

- **Representation**: by DBM
Zones and zone graph

Zone: set of valuations defined by conjunctions of constraints:
- \( x \sim c \)
- \( x - y \sim c \)
- e.g. \((x - y \geq 1) \land y < 2\)

Representation: by DBM
Zones and zone graph

- **Zone**: set of valuations defined by conjunctions of constraints:
  - $x \sim c$
  - $x - y \sim c$
  - e.g. $(x - y \geq 1) \land y < 2$

- **Representation**: by DBM

---

**Sound and complete**

**Zone graph** preserves state **reachability**
But the zone graph could be infinite ...
But the zone graph could be infinite ...
But the zone graph could be infinite ...
But the zone graph could be infinite ...

(y = 1)

$y := 0$

$q_0 \xrightarrow{x := 0} q_1$

$q_1 \xrightarrow{y := 0} q_1$
But the zone graph could be infinite ...
But the zone graph could be infinite ...
Use finite abstractions

Key idea: **Extrapolate** each zone in a **sound** manner

\[(q_0, Z_0) \rightarrow (q_1, Z_1) \rightarrow (q_2, Z_2)\]
Use finite abstractions

Key idea: **Extrapolate** each zone in a **sound** manner

\[
(q_0, Z_0) \quad \rightarrow \quad (q_1, Z_1) \quad \rightarrow \quad (q_2, Z_2) \quad \rightarrow \quad (q_0, a(Z_0))
\]

- Number of extrapolated zones is finite
- Bigger extrapolated zones $\rightarrow$ smaller simulation graph
Use finite abstractions

Key idea: **Extrapolate** each zone in a **sound** manner

\[(q_0, Z_0) \quad \rightarrow \quad (q_1, Z_1) \quad \rightarrow \quad (q_2, Z_2)\]

\[(q_0, a(Z_0)) \quad \rightarrow \quad \quad \quad \quad \quad \quad \quad \quad \quad \]

- Number of extrapolated zones is finite
- Bigger extrapolated zones $\rightarrow$ smaller simulation graph
Use finite abstractions

Key idea: Extrapolate each zone in a sound manner

\[(q_0, Z_0) \quad \quad (q_1, Z_1) \quad \quad (q_2, Z_2)\]

\[(q_0, \alpha(Z_0)) \quad \quad (q_1, Z')\]
Use finite abstractions

Key idea: **Extrapolate** each zone in a **sound** manner

\[(q_0, Z_0) \quad \rightarrow \quad (q_1, Z_1) \quad \rightarrow \quad (q_2, Z_2)\]

\[(q_0, \alpha(Z_0)) \quad \rightarrow \quad (q_1, \alpha(Z'))\]
Use finite abstractions

Key idea: **Extrapolate** each zone in a **sound** manner

\[
(q_0, Z_0) \quad \rightarrow \quad (q_1, Z_1) \quad \rightarrow \quad (q_2, Z_2)
\]

\[
(q_0, \alpha(Z_0)) \quad \rightarrow \quad (q_1, \alpha(Z')) \quad \rightarrow \quad (q_2, Z'')
\]
Use finite abstractions

Key idea: **Extrapolate** each zone in a **sound** manner

\[(q_0, Z_0) \rightarrow (q_1, Z_1) \rightarrow (q_2, Z_2)\]

\[(q_0, a(Z_0)) \rightarrow (q_1, a(Z')) \rightarrow (q_2, a(Z''))\]
Use finite abstractions

Key idea: **Extrapolate** each zone in a **sound** manner

\[(q_0, Z_0) \xrightarrow{\pm} (q_1, Z_1) \xrightarrow{\pm} (q_2, Z_2)\]

\[(q_0, a(Z_0)) \xrightarrow{\pm} (q_1, a(Z')) \xrightarrow{\pm} (q_2, a(Z''))\]

- **Number of extrapolated zones is finite**
- **Bigger** extrapolated zones → **smaller simulation graph**
Abstractions in literature [Bou04, BBLP06]
Abstractions in literature [Bou04, BBLP06]

All the above abstractions preserve state reachability
Abstractions in literature [Bou04, BBLP06]

\[
\begin{align*}
\text{Closure}_\alpha & \quad \preceq_{LU} \quad \text{Extra}^+_{LU} \\
\text{Extra}^+_{\alpha} & \quad \text{Extra}^+_{\alpha} \\
\text{Extra}^+_{\alpha} & \quad \text{Extra}^+_{\alpha}
\end{align*}
\]

Sound and complete

All the above abstractions preserve state reachability

But for implementation extrapolated zone should be a zone
Abstractions in literature [Bou04, BBLP06]

Only convex abstractions in implementations!
Efficient use of the non-convex Closure abstraction!
What is Closure_α?
What is Closure$_{\alpha}$?
What is Closure_\(\alpha\)?

Closure_\(\alpha(Z)\): set of regions that \(Z\) intersects
Using non-convex Approximations for Efficient Analysis of Timed Automata - 12/23
Using non-convex Approximations for Efficient Analysis of Timed Automata - 12/23
Schema

\[ q_3 = q_1 \land Z_3 \subseteq \text{Closure}_\alpha(Z_1) ? \]
Reduction to 2 clocks

\[ Z \subseteq \text{Closure}_\alpha(Z') \text{ iff for all pairs of clocks } x, y, \text{ we have} \]

\[ \text{Proj}_{xy}(Z) \subseteq \text{Closure}_\alpha(\text{Proj}_{xy}(Z')) \]

Complexity: \( \mathcal{O}(|X|^2) \) where \( X \) is the set of clocks
$Z \subseteq \text{Closure}_\alpha(Z')$

**Reduction to 2 clocks**

$Z \subseteq \text{Closure}_\alpha(Z')$ iff for all pairs of clocks $x, y$, we have

$$\text{Proj}_{xy}(Z) \subseteq \text{Closure}_\alpha(\text{Proj}_{xy}(Z'))$$

**Complexity:** $O(|X|^2)$ where $X$ is the set of clocks

**Same complexity as** $Z \subseteq Z'$!
So what do we have now...

- **No need** to restrict to **convex** abstractions
- Compute $ZG(A)$: $Z \subseteq \text{Closure}_\alpha(Z')$ to **terminate**
So what do we have now...

- **No need** to restrict to convex abstractions
- Compute $ZG(A)$: $Z \subseteq \text{Closure}_{\alpha}(Z')$ to terminate

Coming next: **prune the bound function $\alpha$**!
Bound function $\alpha$

Naive: $\alpha(x) = 14, \alpha(y) = 10^6$
Static analysis: bound function for every $q$ [BBFL03]

Naive: $\alpha(x) = 14$, $\alpha(y) = 10^6$
Static analysis: bound function for every $q$ [BBFL03]

Naive: $\alpha(x) = 14$, $\alpha(y) = 10^6$
Static analysis: bound function for every $q$

[BBFL03]

\[ y \geq 10^6 \]
\[ x \leq 5 \]
\[ y \leq 14 \]
\[ y := 0 \]
\[ x := 0 \]

- $q_0$ to $q_1$: $x \leq 5$
- $q_0$ to $q_3$: $y \geq 10^6$
- $q_1$ to $q_2$: $y \geq 5$
- $q_3$ to $q_2$: $x \leq 14$

$\nu(y) = 10$
$\nu'(y) = 10^7$

Naive: $\alpha(x) = 14$, $\alpha(y) = 10^6$
Static analysis: bound function for every $q$

[BBFL03]

$\nu(y) = 10$

$\nu'(y) = 10^7$

Naive: $\alpha(x) = 14$, $\alpha(y) = 10^6$
Static analysis: bound function for every $q$

[BBFL03]

Naive: $\alpha(x) = 14$, $\alpha(y) = 10^6$

$$y \geq 10^6$$

$$x \leq 5$$

$$y \geq 5$$

$$x \leq 14$$

$$y := 0$$

$$x := 0$$

$$\nu(y) = 10$$

$$\nu'(y) = 10^7$$
Static analysis: bound function for every $q$

[BBFL03]
Static analysis: bound function for every $q$

[BBFL03]

But this is not enough!
Need to look at semantics...

\[ x = 1 \]
\[ x := 0 \]

\[ q_0 \quad \rightarrow \quad q_1 \quad \text{if} \quad x \geq 2 \]

\[ q_1 \quad \downarrow \quad q_2 \quad \text{if} \quad x < 1 \]

\[ q_2 \quad \rightarrow \quad q_3 \quad \text{if} \quad y = 10^6 \]
Need to look at semantics...

\[ x = 1 \]
\[ x := 0 \]

\[ 10^6 \]

\[ q_0 \quad x \geq 2 \quad q_1 \]

\[ q_3 \quad y = 10^6 \quad q_2 \]

More than 10^6 zones at \( q_0 \) not necessary!
Need to look at semantics...

More than $10^6$ zones at $q_0$ not necessary!
Bound function for every \((q, Z)\) in \(ZG(A)\)
Bound function for every \((q, Z)\) in \(ZG(A)\)

constants at
Constant propagation

\[ \alpha(x) = -\infty \]

\[ (q, Z, \alpha) \]

\[ * \]
Constant propagation

\[ \alpha(x) = -\infty \]

\((q, Z, \alpha)\)

\[ x \leq 3 \]
Constant propagation

\[ \alpha(x) = 3 \]

\((q, Z, \alpha)\)

\[ x \leq 3 \]
Constant propagation

\[ \alpha(x) = 3 \]

\[ (q, Z, \alpha) \]

\[ x \leq 3 \]
Constant propagation

\[ \alpha(x) = 5 \]

\[ (q, Z, \alpha) \]

\[ x \leq 3 \]
Constant propagation

\[ \alpha(x) = 5 \]

\((q, Z, \alpha)\)

\[ x \leq 3 \]
Constant propagation

\[ \alpha(x) = 5 \]

\((q, Z, \alpha)\)

\[ x \leq 3 \]

\[ x > 6 \]
Constant propagation

\[ \alpha(x) = 6 \]

\((q, Z, \alpha)\)

- \(x \leq 3\)
- \(x > 6\)
Constant propagation

\[ \alpha(x) = 6 \]

\[(q, Z, \alpha)\]

- \[x \leq 3\]
- \[x > 6\]

All tentative nodes consistent → No more exploration → Terminate!
Constant propagation

\[ \alpha(x) = 6 \]

\[ (q, Z, \alpha) \]

\[ x \leq 3 \]

\[ x > 6 \]
$$\alpha(x) = 6$$

Using non-convex Approximations for Efficient Analysis of Timed Automata - 18/23
Constant propagation

\[ \alpha(x) = 11 \]

\[(q, Z, \alpha)\]

All tentative nodes consistent + No more exploration → Terminate!

Using non-convex Approximations for Efficient Analysis of Timed Automata - 18/23
Constant propagation

\[ \alpha(x) = 11 \]

\((q, Z, \alpha)\)

\[ x \leq 3 \]

\[ x > 6 \]

\[ x \geq 11 \]
Constant propagation

\[ \alpha(x) = 11 \]

\[(q, Z, \alpha)\]

\[ x \leq 3 \]
\[ x > 6 \]
\[ x \geq 11 \]
Constant propagation

\[ \alpha(x) = 11 \]

\((q, Z, \alpha)\)

- \(x \leq 3\)
- \(x > 6\)
- \(x \geq 11\)
- \(x := 0\)

All tentative nodes consistent → No more exploration → Terminate!
Constant propagation

\[ \alpha(x) = 11 \]

\[ (q, Z, \alpha) \]

\[ x \leq 3 \]
\[ x > 6 \]
\[ x > 6 \]
\[ x \geq 11 \]

All tentative nodes consistent

+ No more exploration

\[ x := 0 \]

\[ \rightarrow \text{Terminate!} \]
Invariants on the bounds

- Non tentative nodes: $\alpha = \max\{\alpha_{\text{succ}}\}$ (modulo resets)
- Tentative nodes: $\alpha = \alpha_{\text{master}}$
Invariants on the bounds

- Non tentative nodes: \( \alpha = \max\{\alpha_{\text{succ}}\} \) (modulo resets)
- Tentative nodes: \( \alpha = \alpha_{\text{master}} \)

**Theorem (Correctness)**
An accepting state is reachable in \( ZG(A) \) iff the algorithm reaches a node with an accepting state and a non-empty zone.
Overall algorithm

- Compute $ZG(A)$: $Z \subseteq \text{Closure}_{\alpha'}(Z')$ for termination
- **Bounds $\alpha$** calculated on-the-fly
- Extra $^+_LU$ can also be handled:
  - $O(|X|^2)$ procedure for $Z \subseteq \text{Closure}_{\alpha'}(Extra^+_LU'(Z'))$
## Benchmarks

<table>
<thead>
<tr>
<th>Model</th>
<th>$E_{LU}^+, sa$</th>
<th>$CI_{LU}^+, sa$</th>
<th>$CI_{LU}^+, otf$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nodes</td>
<td>s.</td>
<td>nodes</td>
</tr>
<tr>
<td>Fi7</td>
<td>48535</td>
<td>6.24</td>
<td>48535</td>
</tr>
<tr>
<td>Fi8</td>
<td>229890</td>
<td>63.97</td>
<td>229890</td>
</tr>
<tr>
<td>Fi9</td>
<td>1024697</td>
<td>558.90</td>
<td>1024697</td>
</tr>
<tr>
<td>Fi10</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>C7</td>
<td>23137</td>
<td>6.07</td>
<td>23137</td>
</tr>
<tr>
<td>C8</td>
<td>86157</td>
<td>40.45</td>
<td>86157</td>
</tr>
<tr>
<td>C9</td>
<td>317326</td>
<td>283.56</td>
<td>317326</td>
</tr>
<tr>
<td>FD10</td>
<td>726</td>
<td>1.89</td>
<td>640</td>
</tr>
<tr>
<td>FD20</td>
<td>2846</td>
<td>70.13</td>
<td>2430</td>
</tr>
<tr>
<td>FD30</td>
<td>6366</td>
<td>670.31</td>
<td>5370</td>
</tr>
</tbody>
</table>

$E_{LU}^+, sa$ is currently used in **UPPAAL**

- Closure can be **efficiently implemented**
- **Both** Closure and otf bounds help
Conclusions & Perspectives

- Efficient implementation of a non-convex approximation that subsumes current ones in use

- On-the-fly learning of bounds that is better than the current static analysis

- More sophisticated non-convex approximations

- Strategies for constraint propagation
R. Alur and D.L. Dill.
A theory of timed automata.

Static guard analysis in timed automata verification.

Lower and upper bounds in zone-based abstractions of timed automata.

P. Bouyer.
Forward analysis of updatable timed automata.