Using non-convex approximations for efficient analysis of timed automata

B. Srivathsan

Joint work with F. Herbreteau, D. Kini and I. Walukiewicz

LaBRI, Université Bordeaux 1
Indian Institute of Technology Bombay, India

Verification Seminar

Université Libre de Bruxelles
Timed Automata [AD94]

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 ... 

Given a TA, does there exist an accepting run?
The problem we are interested in ...

Given a TA, does there exist an accepting run?

Theorem [AD94, CY92]
This problem is PSPACE-complete
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 [AD94]
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

\( \mathcal{O}(|X| \cdot M^{|X|}) \) many regions!

Sound and complete [AD94]

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

**Key idea:** Maintain all *valuations* reachable along a path
A more efficient solution...

**Key idea:** Maintain all *valuations* reachable along a path
A more efficient solution...

Key idea: Maintain all valuations reachable along a path
A more efficient solution...

Key idea: Maintain all valuations reachable along a path
A more efficient solution...

**Key idea:** Maintain all valuations reachable along a path

\[ x \leq 5 \]

\[ (y \geq 7) \]

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

**Key idea:** Maintain **all valuations** reachable along a path
A more efficient solution...

Key idea: Maintain all valuations reachable along a path
A more efficient solution…

Key idea: Maintain all valuations reachable along a path

\( x = y \geq 0 \)

\( y - x \geq 7 \)

\( x \leq 5 \)

\( y \geq 7 \)

\( x := 0 \)
A more efficient solution...

**Key idea:** Maintain all valuations reachable along a path.
A more efficient solution...

Key idea: Maintain **all valuations** reachable along a path

\[ x = y \geq 0 \]

\[ y - x \geq 7 \]

\[ (x \leq 5) \]

\[ (y \geq 7) \]

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

Key idea: Maintain all valuations reachable along a path

\[ x \leq 5 \]
\[ y \geq 7 \]
\[ x := 0 \]
A more efficient solution…

**Key idea:** Maintain **all valuations** reachable along a path

\[
x = y \geq 0
\]

\[
x = y \geq 0
\]

\[
y - x \geq 7
\]

\[
y - x \geq 7
\]

\[
(x \leq 5)
\]

\[
(y \geq 7)
\]

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

Key idea: Maintain all valuations reachable along a path
A more efficient solution...

Key idea: Maintain all valuations reachable along a path

\[
\begin{align*}
  x &= y \geq 0 \\
  x &= y \geq 0 \\
  y - x &\geq 7 \\
  y - x &\geq 7
\end{align*}
\]
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 [DT98]

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

\[
\begin{align*}
q_0 & \quad x := 0 \\
y & := 0
\end{align*}
\]

\[
\begin{align*}
(y = 1) & \\
y & := 0
\end{align*}
\]

\[
\begin{align*}
q_1 & \quad y := 0
\end{align*}
\]
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 ...
But the zone graph could be infinite ...

\[
\begin{align*}
q_0 & \xrightarrow{x=0} q_1 \\
y := 0 & \quad y := 0
\end{align*}
\]

\[(y = 1)\]
Use finite abstractions

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

\[(q_0, Z_0) \rightarrow (q_1, Z_1) \rightarrow (q_2, Z_2)\]
Key idea: **Abstract** each zone in a **sound** manner

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

\[
(q_1, Z_1) \quad \quad \quad (q_2, Z_2)
\]
Use finite abstractions

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

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

\[(q_0, a(Z_0))\]  \[\rightarrow\]  \[\text{Number of abstracted zones is finite}\]

\[\text{Coarser abstraction} \rightarrow \text{fewer abstracted zones}\]
Use finite abstractions

Key idea: Abstract each zone in a sound manner

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

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

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

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

\[(q_0, a(Z_0)) \rightarrow (q_1, a(Z'))\]

- Number of abstracted zones is finite
- Coarser abstraction → fewer abstracted zones
Use finite abstractions

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

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

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

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

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

\[(q_0, a(Z_0)) \quad (q_1, a(Z')) \quad (q_2, a(Z'')) \]

- Number of abstracted zones is finite
- Coarser abstraction → fewer abstracted zones
Abstractions in literature [Bou04, BBLP06]
Abstractions in literature [Bou04, BBLP06]

Sound and complete

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

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

Sound and complete

All the above abstractions preserve state reachability

But for implementation abstracted 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$_\alpha$?
What is Closure$_\alpha$?
What is $\text{Closure}_\alpha$?

$\text{Closure}_\alpha(Z)$: set of regions that $Z$ intersects
Using Closure \( \alpha \) for reachability

\[ q_3 = q_1 \land a(Z_3) \subseteq a(Z_1)? \]

Standard algorithm: covering tree
Using Closure\(\alpha\) for reachability

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

\[ (q_1, a(Z_1)) \]

\[ (q_2, a(Z_2)) \]

\[ (q_3, a(Z_3)) \]

\[ (q_4, a(Z_4)) \]

\[ (q_5, a(Z_5)) \]

Closure\(\alpha(Z)\) cannot be efficiently stored
Using Closure$_\alpha$ for reachability

\[ q_3 = q_1 \land a(Z_3) \subseteq a(Z_1)? Z_3 \subseteq \text{Closure}_\alpha(Z_1)? \]

Do not store abstracted zones!
Using Closure_\alpha for reachability

(q_0, Z_0)

(q_1, Z_1)

(q_2, Z_2)

(q_3, Z_3)

(q_4, Z_4)

(q_5, Z_5)

Use Closure for termination!
$Z \subseteq \text{Closure}_{\alpha}(Z')$?
$Z \subseteq \text{Closure}_\alpha(Z')$?
\( Z \subseteq \text{Closure}_\alpha(Z')? \)

**Diagram:**
- \( Z \)
- \( Z' \)
- \( \text{Closure}_\alpha(Z') \)

**Equation:**
\[
\alpha(x) \quad \alpha(y) \\
Z \quad Z' \\
\text{Closure}_\alpha(Z')
\]

**Text:**
\[
\begin{align*}
Z & \subseteq \text{Closure}_\alpha(Z')? \\
& \iff \exists R. R \text{ intersects } Z, R \text{ does not intersect } Z'
\end{align*}
\]

**Coming next:** An efficient algorithm for \( Z \not\subseteq \text{Closure}_\alpha(Z') \)

Using non-convex approximations for efficient analysis of timed automata - 13/30
$Z \subseteq \text{Closure}_{\alpha}(Z')$?

$Z \not\subseteq \text{Closure}_{\alpha}(Z') \iff \exists R. \text{ R intersects } Z, \text{ R does not intersect } Z'$
$Z \subseteq \text{Closure}_{\alpha}(Z')$?

$Z \not\subseteq \text{Closure}_{\alpha}(Z') \iff \exists R. \text{ R intersects } Z, \text{ R does not intersect } Z'$

Coming next: An efficient algorithm for $Z \not\subseteq \text{Closure}_{\alpha}(Z')$
Step 1: Representing regions and zones

\[\begin{align*}
0 &< x < 3 \\
0 &< y < \infty
\end{align*}\]
Step 1: Representing regions and zones

\[ x < 3 \quad y < \infty \]
\[ x > 2 \quad y > 2 \]
Step 1: Representing regions and zones

\[ x < 3 \quad y < \infty \]
\[ x > 2 \quad y > 2 \]
Step 1: Representing regions and zones

$$x < 3 \quad y < \infty$$

$$x > 2 \quad y > 2$$
Step 1: Representing regions and zones

\[ x - 0 < 3 \quad y < \infty \]
\[ x > 2 \quad y > 2 \]
Step 1: Representing regions and zones

\[ x - 0 < 3 \quad y < \infty \]
\[ x > 2 \quad y > 2 \]
Step 1: Representing regions and zones

\[ x - 0 < 3 \quad y < \infty \]
\[ 0 - x < -2 \quad y > 2 \]
Step 1: Representing regions and zones

\[ x - 0 < 3 \]
\[ 0 - x < -2 \]
\[ y < \infty \]
\[ y > 2 \]
Step 1: Representing regions and zones

\[
\begin{align*}
  x - 0 &< 3 & y &< \infty \\
  0 - x &< -2 & y &> 2
\end{align*}
\]
Step 1: Representing regions and zones

\[ x - 0 < 3 \quad y - 0 < \infty \]
\[ 0 - x < -2 \quad 0 - y < -2 \]
Step 1: Representing regions and zones

\[ x - 0 < 3 \quad y - 0 < \infty \]
\[ 0 - x < -2 \quad 0 - y < -2 \]
Step 1: Representing regions and zones

\[
\begin{align*}
x - 0 &< 3 \\
y - 0 &< \infty \\
0 - x &< -2 \\
0 - y &< -2
\end{align*}
\]
Step 1: Representing regions and zones

\[ x - 0 < 3 \quad y - 0 < \infty \]
\[ 0 - x < -2 \quad 0 - y < -2 \]
Step 1: Representing regions and zones

Need a **canonical** representation
Step 1: Representing regions and zones

Shortest path should be given by the direct edge
Step 1: Representing regions and zones

![Diagram showing regions and zones with inequalities]

- \[ x - 0 < 3 \]
- \[ y - 0 < \infty \]
- \[ 0 - x < -2 \]
- \[ 0 - y < -2 \]

**Shortest path** should be given by the **direct edge**
Step 1: Representing regions and zones

For every zone $Z$, **canonical distance graph** $G_Z$

$\begin{align*}
    x &- 0 < 3 \\
    0 - x &< -2 \\
    y &- 0 < \infty \\
    0 - y &< -2
\end{align*}$
Step 2: When is $R \cap Z'$ empty?
Step 2: When is $R \cap Z'$ empty?
Step 2: When is $R \cap Z'$ empty?

$G_R$

$G_{Z'}$

$\min(G_R, G_{Z'})$
Step 2: When is $R \cap Z'$ empty?

\[ \text{Lemma} \]

$R \cap Z'$ is empty $\iff$ min($G_R, G_{Z'}$) has a negative cycle

Using non-convex approximations for efficient analysis of timed automata - 15/30
Step 2: When is $R \cap Z'$ empty?

Lemma [Bou04]

$R \cap Z'$ is empty $\iff$ min$(G_R, G_{Z'})$ has a negative cycle involving 2 clocks!
Step 2: When is $R \cap Z'$ empty?

Lemma

$R \cap Z'$ is empty $\iff$ min$(G_R, G_{Z'})$ has a negative cycle involving 2 clocks!
Step 2: When is $R \cap Z'$ empty?

Lemma

$R \cap Z'$ is empty $\iff$ $\min(G_R, G_{Z'})$ has a negative cycle involving 2 clocks!
Step 2: When is $R \cap Z'$ empty?

$G_{\text{Proj}_{x_2x_3}}(R)$

$G_{\text{Proj}_{x_2x_3}}(Z')$

$\min(G_{\text{Proj}_{x_2x_3}}(R), G_{\text{Proj}_{x_2x_3}}(Z'))$

Lemma

$R \cap Z'$ is empty $\iff$ $\min(G_R, G_{Z'})$ has a negative cycle involving 2 clocks!
Step 2: When is $R \cap Z'$ empty?

\[ G_{\text{Proj}_{x_2x_3}}(R) \]

\[ G_{\text{Proj}_{x_2x_3}}(Z') \]

\[ \min(G_{\text{Proj}_{x_2x_3}}(R), G_{\text{Proj}_{x_2x_3}}(Z')) \]

**Lemma**

$R \cap Z'$ is empty $\iff \exists x, y. \text{Proj}_{xy}(R) \cap \text{Proj}_{xy}(Z')$ is empty
Step 3: Reduction to two clocks

Recall: $\exists R \text{. } R \text{ intersects } Z\text{, } R \text{ does not intersect } Z'$
Step 3: Reduction to two clocks

Recall: \( Z \not\subseteq \text{Closure}_\alpha(Z') \iff \exists R. R \text{ intersects } Z, R \text{ does not intersect } Z' \)
Step 3: Reduction to two clocks

Recall: $Z \not\subseteq \text{Closure}_\alpha(Z') \iff \exists R. R \text{ intersects } Z, R \text{ does not intersect } Z'$
Step 3: Reduction to two clocks

Recall: \( Z \not\subseteq \text{Closure}_\alpha(Z') \iff \exists R. \ R \text{ intersects } Z, \ R \text{ does not intersect } Z' \)
Step 3: Reduction to two clocks

Recall: \( Z \not\subset \text{Closure}_{\alpha}(Z') \iff \exists R. R \text{ intersects } Z, \ R \text{ does not intersect } Z' \)
Step 3: Reduction to two clocks

Recall: \( Z \not\subseteq \text{Closure}_\alpha(Z') \iff \exists R. R \text{ intersects } Z, \text{ } R \text{ does not intersect } Z' \)

Using non-convex approximations for efficient analysis of timed automata - 16/30
Step 3: Reduction to two clocks

Recall: $Z \not\subseteq \text{Closure}_{\alpha}(Z') \iff \exists R. R \text{ intersects } Z, R \text{ does not intersect } Z'$
Step 3: Reduction to two clocks

Recall: \( Z \not\subseteq \text{Closure}_{\alpha}(Z') \iff \exists R. R \text{ intersects } Z, R \text{ does not intersect } Z' \)
Step 3: Reduction to two clocks

Recall: $Z \not\subseteq \text{Closure}_{\alpha}(Z') \iff \exists R. R \text{ intersects } Z, \ R \text{ does not intersect } Z'$

Using non-convex approximations for efficient analysis of timed automata - 16/30
Step 3: Reduction to two clocks

Recall: \( Z \not\subset \text{Closure}_\alpha(Z') \iff \exists R. \ R \text{ intersects } Z, \ R \text{ does not intersect } Z' \)

Theorem

\[ Z \not\subset \text{Closure}_\alpha(Z') \text{ if and only if there exist 2 clocks } x, y \text{ s.t.} \]

\[ \text{Proj}_{xy}(Z) \not\subset \text{Closure}_\alpha(\text{Proj}_{xy}(Z')) \]
Step 3: Reduction to two clocks

Theorem

\[ Z \notin \text{Closure}_\alpha(Z') \text{ if and only if there exist 2 clocks } x, y \text{ s.t.} \]

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

Slightly modified edge-edge comparison is enough
Step 3: Reduction to two clocks

Theorem

\[ Z \not\subseteq \text{Closure}_\alpha(Z') \text{ if and only if there exist 2 clocks } x, y \text{ s.t. } \]

\[ \text{Proj}_{xy}(Z) \not\subseteq \text{Closure}_\alpha(\text{Proj}_{xy}(Z')) \]

Complexity: \( \mathcal{O}(|X|^2) \), where \( X \) is the set of clocks
Step 3: Reduction to two clocks

Theorem

\[ Z \not\subseteq \text{Closure}_\alpha(Z') \text{ if and only if there exist 2 clocks } x, y \text{ s.t. } \]
\[ \text{Proj}_{xy}(Z) \not\subseteq \text{Closure}_\alpha(\text{Proj}_{xy}(Z')) \]

Same complexity as \( Z \subseteq Z' \)!
So what do we have now...

$q_3 = q_1 \land Z_3 \subseteq \text{Closure}_\alpha(Z_1)$?

Efficient algorithm for $Z \subseteq \text{Closure}_\alpha(Z')$
So what do we have now...

$q_3 = q_1 \land Z_3 \subseteq \text{Closure}_\alpha(Z_1)$?

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

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

Size of graph $\sim 10^5$
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: $\alpha(x) = 14$, $\alpha(y) = 10^6$

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

Static analysis: $\alpha(y) = 10^6$

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

Constants at depend on subtree
Constant propagation

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

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

\[
\begin{align*}
&\text{All tentative nodes consistent} \\
&\text{No more exploration} \\
&\to \text{Terminate!}
\end{align*}
\]
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

\[ \alpha(x) = 5 \]

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

\[ x \leq 3 \]
Constant propagation

\[ \alpha(x) = 5 \]

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

\[ x \leq 3 \]

\[ x > 6 \]

\[ Z' \subseteq \text{Closure}_\alpha(Z) \]
Constant propagation

\[ \alpha(x) = 6 \]

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

\[ x \leq 3 \]

\[ x > 6 \]

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

\[(q', Z', \alpha') \]
Constant propagation

\[ \alpha(x) = 6 \]

\[ x \leq 3 \]

\[ x > 6 \]

\[ Z' \subseteq \text{Closure}_\alpha(Z) \]
Constant propagation

\[ \alpha(x) = 6 \]

\[ Z' \subseteq \text{Closure}_\alpha(Z) \]
Constant propagation

\[ \alpha(x) = 6 \]

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

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

\[ x \leq 3 \quad x \geq 11 \]

\[ x > 6 \]

\[ (q', Z', \alpha') \]
Constant propagation

\[ \alpha(x) = 11 \]

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

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

\[ x \leq 3 \]

\[ x > 6 \]

\[ x \geq 11 \]
Constant propagation

\[ \alpha(x) = 11 \]

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

\[ x \leq 3 \]

\[ x > 6 \]

\[ x \geq 11 \]

\[ Z' \subseteq \text{Closure}_\alpha(Z) \]
Constant propagation

\[ \alpha(x) = 11 \]

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

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

\[ x \leq 3 \]

\[ x > 6 \]

\[ x \geq 11 \]
Constant propagation

\[ \alpha(x) = 11 \]

\[ x \leq 3 \]

\[ x > 6 \]

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

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

\[ x := 0 \]

\[ x \geq 11 \]

\[ (q', Z', \alpha') \]
Constant propagation

\[ \alpha(x) = 11 \]

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

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

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

All tentative nodes consistent + No more exploration → Terminate!
Invariants on the bounds

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

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

Theorem (Correctness)
An accepting state is reachable in $ZG(\mathcal{A})$ 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
- Abstraction $\text{Extra}_{LU}^+$ can also be handled:

An efficient $O(|X|^2)$ procedure for $Z \subseteq \text{Closure}_{\alpha}(\text{Extra}_{LU}^+(Z'))$!
### Benchmarks

<table>
<thead>
<tr>
<th>Model</th>
<th>Our algorithm</th>
<th>UPPAAL’s algorithm</th>
<th>UPPAAL 4.1.3 (-n4 -C -o1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nodes</td>
<td>s.</td>
<td>nodes</td>
</tr>
<tr>
<td>CSMA/CD7</td>
<td>5031</td>
<td>0.32</td>
<td>5923</td>
</tr>
<tr>
<td>CSMA/CD8</td>
<td>16588</td>
<td>1.36</td>
<td>19017</td>
</tr>
<tr>
<td>CSMA/CD9</td>
<td>54439</td>
<td>6.01</td>
<td>60783</td>
</tr>
<tr>
<td>FDDI10</td>
<td>459</td>
<td>0.02</td>
<td>525</td>
</tr>
<tr>
<td>FDDI20</td>
<td>1719</td>
<td>0.29</td>
<td>2045</td>
</tr>
<tr>
<td>FDDI30</td>
<td>3779</td>
<td>1.29</td>
<td>4565</td>
</tr>
<tr>
<td>Fischer7</td>
<td>7737</td>
<td>0.42</td>
<td>18374</td>
</tr>
<tr>
<td>Fischer8</td>
<td>25080</td>
<td>1.55</td>
<td>85438</td>
</tr>
<tr>
<td>Fischer9</td>
<td>81035</td>
<td>5.90</td>
<td>398685</td>
</tr>
<tr>
<td>Fischer10</td>
<td>—</td>
<td>T.O.</td>
<td>—</td>
</tr>
</tbody>
</table>

- **Extra**$_{LU}^+$ and **static** analysis bounds in UPPAAL

- **Closure**$_{\alpha}(\text{Extra}^{+}_{LU})$ and **otf** bounds in our algorithm
Experiments I

\[ A_1 \]

- \( q_0 \)
  - \( x = 0 \)
  - \( y \geq 20 \) \&\& \( x = 2 \)

- \( q_1 \)
  - \( x = 1 \)
  - \( y = 10000 \)

- \( q_3 \)
  - \( x = 5 \)

- \( q_2 \)
  - \( x = 0 \)
  - \( y = 10000 \)

<table>
<thead>
<tr>
<th>( A_1 )</th>
<th>nodes</th>
<th>s.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our algorithm</td>
<td>7</td>
<td>0.0</td>
</tr>
<tr>
<td>UPPAAL’s algorithm</td>
<td>2003</td>
<td>0.60</td>
</tr>
<tr>
<td>UPPAAL 4.1.3</td>
<td>2003</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Experiments II

\[ A_2 \]

\[
\begin{align*}
x &= 1 \\
 x &= 0 \\
 q_0 &\xrightarrow{x \leq 1} q_1 \\
 q_0 &\xrightarrow{a!} q_1 \\
 q_1 &\xrightarrow{y \geq 10000} q_2 \\
 q_0 &\xrightarrow{y \leq 10} q_2
\end{align*}
\]

<table>
<thead>
<tr>
<th>( A_2 )</th>
<th>nodes</th>
<th>s.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our algorithm</td>
<td>2</td>
<td>0.0</td>
</tr>
<tr>
<td>UPPAAL's algorithm</td>
<td>10003</td>
<td>0.07</td>
</tr>
<tr>
<td>UPPAAL 4.1.3</td>
<td>10003</td>
<td>0.07</td>
</tr>
</tbody>
</table>
Experiments II

\[ A_2 \]

\begin{align*}
&x = 1 \\
&x := 0 \\
&x <= 1 \\
&y >= 10000 \\
&a! \quad q_1 \\
&y <= 10 \quad q_2
\end{align*}

<table>
<thead>
<tr>
<th>( A_2 )</th>
<th>nodes</th>
<th>s.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our algorithm</td>
<td>2</td>
<td>0.0</td>
</tr>
<tr>
<td>UPPAAL’s algorithm</td>
<td>10003</td>
<td>0.07</td>
</tr>
<tr>
<td>UPPAAL 4.1.3</td>
<td>10003</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Occurs in **CSMA/CD**!
Experiments III

\[ \mathcal{A}_3 \]

\[
\begin{align*}
&x = 1 \quad q_0 \\
x = 0 \\
x = 1 \\
n = 10 \land y \geq 10000 \\
n = 10 \land y \leq 200 \\
x = 1 \\
q_1 \quad y \leq 10 \\
q_2 \\
n = 10 \land y \leq 200
\end{align*}
\]

<table>
<thead>
<tr>
<th>( \mathcal{A}_3 )</th>
<th>nodes</th>
<th>s.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our algorithm</td>
<td>3</td>
<td>0.0</td>
</tr>
<tr>
<td>UPPAAL’s algorithm</td>
<td>10004</td>
<td>0.37</td>
</tr>
<tr>
<td>UPPAAL 4.1.3</td>
<td>10004</td>
<td>0.32</td>
</tr>
</tbody>
</table>
Experiments III

Our algorithm | 3 | 0.0
UPPAAL’s algorithm | 10004 | 0.37
UPPAAL 4.1.3 | 10004 | 0.32

Occurs in Fischer!
Experiments IV

\[ Z : x - y \geq 1 \]
\[ Z' : x > \alpha(x) \]
Experiments IV

\[ Z' : x - y \geq 1 \]
\[ Z : x > \alpha(x) \]

Occurs in FDDI!
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
- Propagating **more** than constants
- Automata with **diagonal** constraints
References

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.

C. Courcoubetis and M. Yannakakis.
Minimum and maximum delay problems in real-time systems.

C. Daws and S. Tripakis.
Model checking of real-time reachability properties using abstractions.