Abstractions for timed automata

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

B. Srivathsan

Ph.D. defence

Jury
Ahmed Bouajjani
Patricia Bouyer
Bruno Courcelle
Frédéric Herbreteau Advisor
Joost-Pieter Katoen
Igor Walukiewicz Advisor
James Worrell
Reachability: Does something **bad** happen?

Liveness: Does something **good** happen **repeatedly**?
Reachability: Does something **bad** happen?

**UPPAAL, KRONOS, RED, IF, PAT, Rabbit ...**

Liveness: Does something **good** happen **repeatedly**?

**PROFOUNDER, CTAV ...**

*A THEORY OF TIMED AUTOMATA*
R. Alur and D.L. Dill, *TCS’94*
In this thesis...

We revisit **reachability** and **liveness** problems for Alur-Dill timed automata.
Reachability

Liveness

Reachability

Liveness
Reachability

Liveness

Reachability

Liveness
Timed Automata

Run: finite sequence of transitions

- accepting if ends in green state
Reachability problem

Given a TA, does it have an accepting run

Theorem [AD94]

This problem is PSPACE-complete

first solution based on Regions
Key idea: Maintain **sets of valuations** reachable along a path.
Key idea: Maintain sets of valuations reachable along a path

\[ x = y \geq 0 \]

\[ x = y \geq 0 \]

\[ y - x \geq 7 \]

\[ y - x \geq 7 \]

Easy to describe convex sets
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 [Dil89]

---

**Sound and complete [DT98]**

**Zone graph preserves state reachability**
Problem of non-termination

\begin{align*}
q_0 \xrightarrow{x} q_1
\end{align*}
Abstractions

potentially infinite...

Zone graph
Abstractions

potentially infinite...
Abstractions

potentially infinite...

Zone graph

a(Zₐ)

q₀, Z₀

q₁, Z₁

q₂, Z₂

q₃, Z₃

...
Abstractions

Zone graph

potentially infinite...

$Z_0, Z_1, Z_2, Z_3, \ldots$

$a(Z_0)$
Abstractions

potentially infinite...

Zone graph

\[ q_0, \quad Z_0 \]
\[ q_1, \quad Z_1 \]
\[ q_2, \quad Z_2 \]
\[ q_3, \quad Z_3 \]
\[ \ldots \]

\[ a(Z_0) \]
\[ a(W_1) \]

\[ q_0, \quad Z_0 \]
\[ q_1, \quad Z_1 \]
\[ q_2, \quad Z_2 \]
\[ q_3, \quad Z_3 \]
\[ \ldots \]

\[ q_0, \quad Z_0 \]
\[ q_1, \quad Z_1 \]
\[ q_2, \quad Z_2 \]
\[ q_3, \quad Z_3 \]
\[ \ldots \]

\[ a(Z_0) \]
\[ a(W_1) \]
Abstractions

potentially infinite...

Zone graph

\[ q_0, \quad Z_0 \]
\[ q_1, \quad Z_1 \]
\[ q_2, \quad Z_2 \]
\[ q_3, \quad Z_3 \]
\[ \vdots \]
\[ \vdots \]

\[ a(Z_0) \]

\[ q_0, \quad Z_0 \]
\[ q_1, \quad Z_1 \]
\[ q_2, \quad Z_2 \]
\[ q_3, \quad Z_3 \]
\[ \vdots \]
\[ \vdots \]

\[ a(W_1) \]
Abstractions

potentially infinite...

Zone graph

\[ a(Z_0) \]

\[ a(W_1) \]

\[ a(W_2) \]

\[ a(W_3) \]
Abstractions

potentially infinite...

Zone graph
Abstractions

Find $a$ such that number of \textbf{abstracted} sets is \textbf{finite}
Abstractions

Coarser the abstraction, smaller the abstracted graph
Condition 1: Abstractions should have \textbf{finite range}

Condition 2: Abstractions should be sound $\Rightarrow \alpha(W)$ can contain only valuations \textbf{simulated} by $W$
**Condition 1:** Abstractions should have **finite range**

**Condition 2:** Abstractions should be sound $\Rightarrow \alpha(W)$ can contain only valuations simulated by $W$

**Question:** Why not add all the valuations simulated by $W$?
Bounds and abstractions

Theorem [LS00]

Coarsest simulation relation is EXPTIME-hard
Theorem [LS00]

Coarsest simulation relation is EXPTIME-hard

\( M(x) = 6, \quad M(y) = 3 \)

\( v \preceq M v' \)

\( (y \leq 3) \quad (x < 4) \)

\( (x < 1) \quad (x > 6) \)

\( (y < 1) \)
Bounds and abstractions

Theorem [LS00]

Coarsest simulation relation is EXPTIME-hard

\[ (y \leq 3)\]
\[ (x < 4)\]
\[ (x < 1)\]
\[ (x > 6)\]
\[ (y < 1)\]

\textbf{M-bounds} [AD94]

\[ M(x) = 6,\ M(y) = 3\]

\[ \nu \preceq_M \nu' \]
Bounds and abstractions

**Theorem [LS00]**

Coarsest simulation relation is **EXPTIME-hard**

\[
\begin{align*}
(y \leq 3) & & (x < 4) & & (x < 1) \\
(x < 1) & & (x > 6) & & (y < 1)
\end{align*}
\]

**M-bounds [AD94]**

\[
M(x) = 6, \ M(y) = 3 \\
\nu \leq_M \nu'
\]

**LU-bounds [BBLP04]**

\[
L(x) = 6, \ L(y) = -\infty \\
U(x) = 4, \ U(y) = 3 \\
\nu \leq_{LU} \nu'
\]
Abstractions in literature [BBLP04, Bou04]
Abstractions in literature [BBLP04, Bou04]

\[(\preceq_{LU}) \leadsto a \preceq_{LU} \leadsto \text{Closure}_M \]

Non-convex
Abstractions in literature [BBLP04, Bou04]

\[ \alpha \preceq_{LU} \text{Closure}_M \preceq_{LU} \text{Extra}_{LU}^{+} \]

Non-convex

\[ \preceq_{LU} \]

Extra_{LU}^{+}

Extra_{M}^{+}

Extra_{M}

Convex

Only convex abstractions used in implementations!
Non-convex abstr.

Reachability

Liveness

Liveness
Step 1: We can use abstractions **without storing** them
Using non-convex abstractions

Standard algorithm: covering tree

$q_3 = q_1 \land \alpha(W_3) \subseteq \alpha(W_1)$?
Using non-convex abstractions

Pick simulation based $\alpha$

$q_3 = q_1 \land \alpha(W_3) \subseteq \alpha(W_1)$?
Using non-convex abstractions

Pick simulation based $\alpha$
Using non-convex abstractions

Pick simulation based $\alpha$
Using non-convex abstractions

Pick simulation based $\alpha$
Using non-convex abstractions

Pick simulation based $\alpha$
Using non-convex abstractions

Standard algorithm: covering tree

Pick simulation based \( \alpha \)

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

\[ q_0 \]

\[ q_1 \]

\[ q_2 \]

\[ q_3 \]

\[ q_4 \]

\[ q_5 \]

\[ Z_0 \]

\[ Z_1 \]

\[ Z_2 \]

\[ Z_3 \]

\[ Z_4 \]

\[ Z_5 \]

\[ a(Z_0) \]

\[ a(Z_1) \]

\[ a(Z_2) \]

\[ a(Z_3) \]

\[ a(Z_4) \]

\[ a(Z_5) \]
Using non-convex abstractions

Standard algorithm: covering tree
Pick simulation based
Need to store only concrete semantics

$q_3 = q_1 \land \alpha(Z_3) \subseteq \alpha(Z_1)$?
Using non-convex abstractions

Standard algorithm:

Pick simulation based

Need to store only concrete semantics

Use $Z \subseteq a(Z')$ for termination

$q_3 = q_1 \land Z_3 \subseteq a(Z_1)$?
Step 1: We can use abstractions **without storing** them

Step 2: We can do the **inclusion** test **efficiently**
Efficient inclusion testing

Main result

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

\[ \text{Proj}_{xy}(Z) \not\subseteq a \preceq_{LU} (\text{Proj}_{xy}(Z')) \]
Efficient inclusion testing

Main result

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

\[ \text{Proj}_{xy}(Z) \not\subseteq a_{\preceq_{LU}}(\text{Proj}_{xy}(Z')) \]

Complexity: \( \mathcal{O}(|X|^2) \), where \( X \) is the set of clocks
Efficient inclusion testing

Main result

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

\[ \text{Proj}_{xy}(Z) \not\subseteq a \preceq_{LU} (\text{Proj}_{xy}(Z')) \]

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

Same complexity as \( Z \subseteq Z'! \)
Efficient inclusion testing

Main result

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

\[ \text{Proj}_{xy}(Z) \not\subseteq a \preceq_{LU} (\text{Proj}_{xy}(Z')) \]

Complexity: \( \Theta(|X|^2) \), where \( X \) is the set of clocks

Same complexity as \( Z \subseteq Z'! \)

Slightly modified comparison works!
Step 1: We can use abstractions \textbf{without storing} them

Step 2: We can do the \textbf{inclusion} test \textbf{efficiently}

$\Rightarrow$ \textbf{new algorithm} for reachability
Can we do better than $a \preceq_{LU}$?
Question: Can we do better than $a \preceq_{LU}$?
**Optimality**

**LU-automata:** automata with guards **determined by** $L$ and $U$

**Theorem**

The $\alpha_{LU}$ abstraction is the **biggest abstraction** that is **sound** and **complete** for all LU-automata.
Non-convex abstr.
  Efficient use
  Optimality

Reachability

Liveness

Liveness
Non-convex abstr.
- Efficient use
- Optimality

Reachability

Liveness

Liveness
Question: If $a_{LU}$ is the best, can we do better?
Question: If $a_{\leq LU}$ is the best, can we do better?

Get better LU-bounds!
Global LU-bounds

Naive: $L_x = U_x = 10^6$, $L_y = U_y = 10^6$

Size of graph $\sim 10^6$
Static analysis: bounds for every $q$

[BBFL03]

Size of graph $< 10$
Static analysis: bounds for every $q$

[BBFL03]

$x = 1$

\{x\}

$x \geq 2$

$x \leq 1$

$x = 10^6$

$y = 10^6$

Size of graph $\sim 10^6$

Need to look at semantics...
LU bounds for every \((q, Z)\) in zone graph

constants at node depend on the subtree

\[
\frac{25}{43}
\]
Constant propagation

**Contribution:** A new on-the-fly algorithm to learn constants during exploration

$$x = 1 \quad \{x\} \quad q_0$$

$$x \geq 2 \quad q_3$$

$$x \leq 1$$

$$x = 10^6 \quad \{x, y\} \quad q_1$$

$$y = 10^6 \quad q_2$$

**Theorem (Correctness)**

An accepting state is reachable in $A$ iff the constant propagation algorithm reaches a node with accepting state and a non-empty zone.
# 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>5046</td>
<td>0.39</td>
<td>5923</td>
</tr>
<tr>
<td>CSMA/CD8</td>
<td>16609</td>
<td>0.75</td>
<td>19017</td>
</tr>
<tr>
<td>CSMA/CD9</td>
<td>54467</td>
<td>9.40</td>
<td>60783</td>
</tr>
<tr>
<td>FDDI10</td>
<td>459</td>
<td>0.04</td>
<td>525</td>
</tr>
<tr>
<td>FDDI20</td>
<td>1719</td>
<td>0.41</td>
<td>2045</td>
</tr>
<tr>
<td>FDDI30</td>
<td>3779</td>
<td>1.70</td>
<td>4565</td>
</tr>
<tr>
<td>Fischer7</td>
<td>7737</td>
<td>0.40</td>
<td>18353</td>
</tr>
<tr>
<td>Fischer8</td>
<td>25080</td>
<td>1.50</td>
<td>85409</td>
</tr>
<tr>
<td>Fischer9</td>
<td>81035</td>
<td>5.70</td>
<td>397989</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
- **a**\(_{LU}^-\) and **otf** bounds in our algorithm
Non-convex abstr.

Efficient use
Optimality

Bounds

On-the-fly

Liveness

Liveness
Timed Büchi automata

Run: infinite sequence of transitions

- **accepting** if infinitely often green state
- **non-Zeno** if time diverges ($\sum_{i \geq 0} \delta_i \rightarrow \infty$)
Büchi non-emptiness problem

Given a TBA, does it have a non-Zeno accepting run

Theorem [AD94]
This problem is PSPACE-complete
$ZG^a(\mathcal{A}) : \ (q_0, Z_0) \rightarrow (q_1, Z_1) \rightarrow (q_2, Z_2) \rightarrow \ldots$

$\mathcal{A} : \ (q_0, v_0) \rightarrow (q_1, v_1) \rightarrow (q_2, v_2) \rightarrow \ldots$

**Sound and complete [Tri09, Li09]**

All the above abstractions preserve repeated state reachability
Sound and complete [Tri09, Li09]
All the above abstractions preserve repeated state reachability

What about non-Zenoness?
Adding a clock for non-Zenoness [TYB05]

\[ A' : \text{strongly non-Zeno TBA} \]

\[ |X| + 1 \text{ clocks and at most } 2 \cdot |Q| \text{ states} \]

**Theorem [TYB05]**

A has a non-Zeno accepting run iff \( ZG^a(A') \) has an **accepting** run
Adding a clock for non-Zenoness [TYB05]

\[ A' : \text{ strongly non-Zeno TBA } \]
\[ |X| + 1 \text{ clocks and at most } 2 \cdot |Q| \text{ states } \]

**Theorem [TYB05]**

A has a non-Zeno accepting run iff \( ZG^a(A') \) has an accepting run

**Question:** Is this good enough?
Adding a clock for non-Zenoness [TYB05]

A': strongly non-Zeno TBA

\[ |X| + 1 \text{ clocks and at most } 2 \cdot |Q| \text{ states} \]

Theorem [TYB05]
A has a non-Zeno accepting run iff \( ZG^a(A') \) has an accepting run

Contribution: The construction can give exponential blowup

Theorem
There exists an automaton \( \mathcal{A}_n \) with \( n \) clocks for which

\[
|ZG^a(\mathcal{A}')| = \Theta(2^n) \cdot |ZG^a(\mathcal{A}_n)|
\]
Non-convex abstr.
Efficient use
Optimality

Bounds
On-the-fly

Non-Zenoness
Adding 1 clock is costly

Liveness
Coming next: A **new construction** for non-Zenoness
New construction

When does a path in $ZG^a(\mathscr{A})$ yield only Zeno runs?

**Blocking clocks**

$x$ never reset but checked for upper bound

**Zero-checks**

$x$ and $y$ should be 0 all along the path
Zero-checks

Can time elapse here?
Zero-checks

Time can elapse at a node if every zero-check is \textbf{preceded} by a reset
Zero-checks

\begin{align*}
\checkmark \quad \{x\} \quad (x = 0)
\end{align*}

Time can elapse at a node if every zero-check is \textbf{preceded} by a reset

\textbf{Guessing Zone Graph} (\(GZG^a(\mathcal{A})\)):

\begin{align*}
(q, Z, Y) & \xrightarrow{\{x\}} (q', Z', Y \cup \{x\}) \\
(q, Z, Y) & \xrightarrow{(x=0)} \text{enabled only if } x \in Y \\
(q, Z, Y) & \xrightarrow{\tau} (q, Z, \emptyset)
\end{align*}
Algorithm

Theorem

A has a non-Zeno run iff there is an unblocked path in GZG$^a(A)$ with infinitely many nodes that have $Y = \emptyset$.

Complexity: $|GZG^a(A)| \cdot (|X| + 1)$
$2^{|X|}$ more nodes in $GZG^a(A)$ than in $ZG^a(A)$ due to $Y$ sets?
Theorem

- For each reachable node \((q, Z)\), \(Z\) entails a total order on \(X\).
- \(\text{Extra}_M, \text{Extra}^+_M\) preserve the order.
- \(Y\) respects this order; only \(|X| + 1\) sets needed.
Theorem

- For each reachable node \((q, Z)\), \(Z\) entails a \textbf{total order} on \(X\).
- \(\text{Extra}_M\), \(\text{Extra}^+_M\) \textbf{preserve the order}.
- \(Y\) \textbf{respects} this order; only \(|X| + 1\) sets needed.

Extra\(_{LU}\), Extra\(_{LU}^+\) \textbf{do not preserve order}

Theorem

Non-Zenoness from LU-abstract zone graphs is \textbf{NP-complete}

Theorem

A slight \textbf{weakening} of Extra\(_{LU}\), Extra\(_{LU}^+\) \textbf{preserves order}
Non-convex abstr.
- Efficient use
- Optimality

Bounds
- On-the-fly

Non-Zenoness
- Adding 1 clock is costly
- New construction
- NP-complete for LU

Liveness
## Benchmarks

<table>
<thead>
<tr>
<th>A</th>
<th>ZG^a(A)</th>
<th>ZG^a(A')</th>
<th>GZG^a(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>size</td>
<td>size</td>
<td>otf</td>
</tr>
<tr>
<td>Train-Gate2 (mutex)</td>
<td>134</td>
<td>194</td>
<td>194</td>
</tr>
<tr>
<td>Train-Gate2 (bound. resp.)</td>
<td>988</td>
<td>227482</td>
<td>352</td>
</tr>
<tr>
<td>Train-Gate2 (liveness)</td>
<td>100</td>
<td>217</td>
<td>35</td>
</tr>
<tr>
<td>Fischer3 (mutex)</td>
<td>1837</td>
<td>3859</td>
<td>3859</td>
</tr>
<tr>
<td>Fischer4 (mutex)</td>
<td>46129</td>
<td>96913</td>
<td>96913</td>
</tr>
<tr>
<td>Fischer3 (liveness)</td>
<td>1315</td>
<td>4962</td>
<td>52</td>
</tr>
<tr>
<td>Fischer4 (liveness)</td>
<td>33577</td>
<td>147167</td>
<td>223</td>
</tr>
<tr>
<td>FDDI3 (liveness)</td>
<td>508</td>
<td>1305</td>
<td>44</td>
</tr>
<tr>
<td>FDDI5 (liveness)</td>
<td>6006</td>
<td>15030</td>
<td>90</td>
</tr>
<tr>
<td>FDDI3 (bound. resp.)</td>
<td>6252</td>
<td>41746</td>
<td>59</td>
</tr>
<tr>
<td>CSMA/CD4 (collision)</td>
<td>4253</td>
<td>7588</td>
<td>7588</td>
</tr>
<tr>
<td>CSMA/CD5 (collision)</td>
<td>45527</td>
<td>80776</td>
<td>80776</td>
</tr>
<tr>
<td>CSMA/CD4 (liveness)</td>
<td>3038</td>
<td>9576</td>
<td>1480</td>
</tr>
<tr>
<td>CSMA/CD5 (liveness)</td>
<td>32751</td>
<td>120166</td>
<td>8437</td>
</tr>
</tbody>
</table>

- Combinatorial explosion may **occur** in practice
- **Optimized** use of $GZG^a(A)$ gives best results
Non-convex abstr.

Efficient use
Optimality
LICS'12, FSTTCS'11

Bounds

On-the-fly
FSTTCS'11

Non-Zenoness

Adding 1 clock is costly
New construction
NP-complete for LU
CAV'10 + ATVA'10 (FMSD'12), CONCUR’11

Zenoness

First complete algorithm
NP-complete for LU
CONCUR’11
Perspectives

- More than LU
- Automata with diagonal constraints
- Probabilistic timed automata, priced timed automata
- Non-Zeno strategies for timed games
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.

D. Dill.
Timing assumptions and verification of finite-state concurrent systems.

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

Guangyuan Li.
Checking timed büchi automata emptiness using lu-abstractions.
François Laroussinie and Ph. Schnoebelen.
The state explosion problem from trace to bisimulation equivalence.  

S. Tripakis.
Checking timed Büchi emptiness on simulation graphs.  
*ACM Transactions on Computational Logic*, 10(3):??–??, 2009.

S. Tripakis, S. Yovine, and A. Bouajjani.  
Checking timed Büchi automata emptiness efficiently.  