Efficient Reachability in Timed Automata

F. Herbreteau

Univ. Bordeaux, LaBRI, CNRS, UMR 5800

Séminaire DTIM Onera, Toulouse
January 28th, 2013

Joint work with: D. Kini, B. Srivathsan, I. Walukiewicz
Timed Automata and the Reachability Problem

Symbolic semantics and abstractions

Bounds based abstractions

Small bounds for abstractions

Conclusions and future work
**Timed Automata [AD94]**

Run: finite sequence of transitions,

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

Accepting run (reachability): ends in a green state.
Example #1: the CSMA/CD protocol

**Property to check:** detection of **collisions**

Reachability of a state with collision and \( \text{wait}_1 \) or \( \text{wait}_2 \)?
Example #2: scheduling jobs (1/2)

- Jobs **compete** to execute tasks on machines

\[ J_1 : (m_1, 2)(m_2, 1)(m_3, 3) \quad J_2 : (m_1, 1)(m_3, 3) \]

- Can the jobs be **scheduled within** 7s?
Example #2: scheduling jobs (2/2)

\[ J_1 : (m_1, 2) (m_2, 1) (m_3, 3) \]
Example #2: scheduling jobs (2/2)

$J_1 : (m_1, 2)(m_2, 1)(m_3, 3)$  $J_2 : (m_1, 1)(m_3, 3)$ within 7s.

Reachability of the green state?
The problem we are interested in ...

**Problem (Emptiness/State reachability)**

Given a TA and a state $q$, is $q$ **reachable**?
The problem we are interested in ... 

Problem (Emptiness/State reachability)

Given a TA and a state $q$, is $q$ reachable?

**Restriction:** guards only involve integer constants

**Theorem ([AD94, CY92])**

This problem is PSPACE-complete
The problem we are interested in ...

<table>
<thead>
<tr>
<th>Problem (Emptiness/State reachability)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given a TA and a state $q$, is $q$ reachable?</td>
</tr>
</tbody>
</table>

Restriction: guards only involve integer constants

<table>
<thead>
<tr>
<th>Theorem ([AD94, CY92])</th>
</tr>
</thead>
<tbody>
<tr>
<td>This problem is PSPACE-complete</td>
</tr>
</tbody>
</table>

This talk: challenges and advances for solving reachability in Timed Automata efficiently
Solving the reachability problem

Search space = reachability tree
Uncountable branching due to density of time
Solution: tree over sets of valuations instead of valuations
Solving the reachability problem

Search space = **reachability tree**
Solving the reachability problem

Uncountable branching due to density of time

Solution: tree over sets of valuations instead of valuations

Search space = reachability tree
Outline

Timed Automata and the Reachability Problem

Symbolic semantics and abstractions

Bounds based abstractions

Small bounds for abstractions

Conclusions and future work
Symbolic reachability tree

- Initial valuations
- Valuations reachable from $Z$
- Zone: set of valuations with efficient symbolic representation by DBMs
- $\text{e.g. } (x - y \leq 1) \land (y < 2)$
Symbolic reachability tree

- **Zone:** set of valuations with efficient symbolic representation by DBMs
  - e.g. \((x - y \leq 1) \land (y < 2)\)
- **Covering tree** (⊆ wrt zones)
Symbolic reachability tree

The tree may be **infinite!**
The tree may be infinite

\[(y = 1), y := 0\]

\[(s_0, x - y = 0)\]
The tree may be infinite

\[ s_0 \xrightarrow{x, y := 0} s_1 \]

\[(y = 1), y := 0\]

\[(s_0, x - y = 0)\]

\[(s_1, x - y = 0)\]
The tree may be infinite

\[(y = 1), y := 0\]

\[x, y := 0\]

\[s_0 \rightarrow s_1\]

\[y\]

\[x\]

\[(s_0, x - y = 0)\]

\[(s_1, x - y = 0)\]

\[(s_1, x - y = 1)\]
The tree may be infinite

$$(y = 1), y := 0$$

\[ x, y := 0 \]

\[ s_0 \rightarrow s_1 \]

$$\begin{align*}
(s_0, x - y = 0) \\
(s_1, x - y = 0) \\
(s_1, x - y = 1) \\
(s_1, x - y = 2)
\end{align*}$$
The tree may be infinite

\( (y = 1), y := 0 \)

\[ x, y := 0 \]

\[ s_0 \]

\[ s_1 \]

\[ (s_0, x - y = 0) \]

\[ (s_1, x - y = 0) \]

\[ (s_1, x - y = 1) \]

\[ (s_1, x - y = 2) \]

\[ (s_1, x - y = 3) \]
Introducing abstractions

Don’t explore \((s_1, Z'_1)\): all its runs are possible from \((s_1, Z_1)\)
Introducing abstractions

Don’t explore \((s_1, Z'_1)\): all its runs are possible from \((s_1, Z_1)\)

**Correctness:** abstractions preserve runs, only add “equivalent” valuations
Introducing abstractions

Don’t explore \((s_1, Z'_1)\): all its runs are possible from \((s_1, Z_1)\)

Correctness: abstractions preserve runs, only add “equivalent” valuations

Termination: ensure finitely many abstracted zones
Outline

Timed Automata and the Reachability Problem

Symbolic semantics and abstractions

Bounds based abstractions

Small bounds for abstractions

Conclusions and future work
Regions [AD94]

**Bound** $M$: guards $x \leq c$, $x \geq c$ only use constants $c \leq M(x)$

**Region:** set of valuations that enable the **same** sequences of transitions
Bound $M$: guards $x \leq c$, $x \geq c$ only use constants $c \leq M(x)$

Region: set of valuations that enable the same sequences of transitions

Correct:

$v_1 \xrightarrow{x \geq 2} y \leq 1$

$v_2 \xrightarrow{x \geq 2} y \leq 1$

$v_3 \xrightarrow{x \geq 2} y \leq 1$
Bound $M$: guards $x \leq c$, $x \geq c$ only use constants $c \leq M(x)$

Region: set of valuations that enable the same sequences of transitions

Correct:

$v_1 \xrightarrow{x \geq 2} y \leq 1$
$v_2 \xrightarrow{x \geq 2} y \leq 1$
$v_3 \xrightarrow{x \geq 2} y \leq 1$

Incorrect:

$v_4 \xrightarrow{x \leq 4}$
$v_5 \xrightarrow{x \leq 4}$
Region based abstraction

$M(M(x))$ may not be convex: how to check inclusion?
Region based abstraction

\[ M(x) \times M(y) \]

\[ Z \]

\[ M(y) \]

\[ M(x) \]

0

x

y

\[ Z \text{ may not be convex: how to check inclusion?} \]
Region based abstraction

\[ \alpha_M(Z) \] is the union of regions that \( Z \) intersects.
Region based abstraction

\[ a_M(Z) \text{ is the union of regions that } Z \text{ intersects} \]

- **Correctness:** \( Z \) and \( a_M(Z) \) have the same executions

- **Termination:** finitely many regions

\[ a_M(Z) = \bigcup \{ M(x) \mid x \in Z \} \]
Region based abstraction

Early termination: $Z' \not\subseteq Z$ but $Z' \subseteq a_M(Z)$
Region based abstraction

Early termination: $Z' \not\subseteq Z$ but $Z' \subseteq \alpha_M(Z)$

$\alpha_M(Z)$ may not be convex: how to check inclusion?
Abstractions [DT98, BBLP06]

**Standard restriction:** use abstractions such that $a(Z)$ is a zone (inclusion in $O(|X|^2)$)
Abstractions [DT98, BBLP06]

**Standard restriction:** use abstractions such that $a(Z)$ is a zone (inclusion in $O(|X|^2)$)

Can we check $Z \subseteq a_{LU}(Z')$ efficiently?
Efficient algorithm for $\alpha_{LU}$ and $\alpha_M$

<table>
<thead>
<tr>
<th>Theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z \subseteq \alpha_{LU}(Z')$ is decided in $O(</td>
</tr>
</tbody>
</table>

| Idea: | do not compute $\alpha_{LU}(Z)$ |
|------------------|
| ▶ define $\subseteq_{\alpha_{LU}}$ s.t. $Z \subseteq_{\alpha_{LU}} Z'$ iff $Z \subseteq \alpha_{LU}(Z')$ |
| ▶ $\subseteq_{\alpha_{LU}}$ is easy for 2 clocks |
| ▶ $n$ clocks: check all pairs of clocks |

<table>
<thead>
<tr>
<th>Theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_{LU}$ is the <strong>coarsest abstraction</strong> if bounds are the only parameter</td>
</tr>
</tbody>
</table>
Timed Automata and the Reachability Problem

Symbolic semantics and abstractions

Bounds based abstractions

Small bounds for abstractions

Conclusions and future work
Back to regions

\[Z' \not\subseteq a_M(Z)\]

\[Z' \subseteq a_{M'}(Z)\]
Back to regions

![Graph showing regions]

\[ Z' \not\subseteq a_M(Z) \quad \text{and} \quad Z' \subseteq a_{M'}(Z) \]

Take the **smallest bounds** you can!

**Recall:** \( M \) preserve the sequence of transitions with guards \( x \leq c, x \geq c \) that only use constants \( c \leq M(x) \)
Global bounds

\[ s_0, y := 0 \]

\[ s_1, a, (y < 3), y := 0 \]

\[ s_2, b, (y \leq 1), c, (x < 1) \]

\[ s_3, d, (x > 2) \]

[LU]-bounds [BBLP06]:

\[ x -\infty \leq y \leq 3, \quad -\infty \leq x \leq 2 \]

M -bounds [AD94]:

\[ x -\infty \leq y \leq 3, \quad -\infty \leq x \leq 2 \]
Global bounds

\begin{align*}
(y & \leq 1) \quad (x < 1) \quad (x < 1) \\
(y & < 3) \quad (x > 2)
\end{align*}

\textbf{M-bounds [AD94]:}

\begin{array}{ccc}
x & y & \\
M & 2 & 3
\end{array}

\textbf{LU-bounds [BBLP06]:}

\begin{array}{ccc}
x & y & \\
L & 2 & -\infty \\
U & 1 & 3
\end{array}
Global bounds

\[(y \leq 1) \quad (x < 1) \quad (x < 1)\]

\[(y < 3) \quad (x > 2)\]

\(M\)-bounds [AD94]:

\[
\begin{array}{ccc}
  x & y & \\
  M & 2 & 3 \\
\end{array}
\]

\(LU\)-bounds [BBLP06]:

\[
\begin{array}{ccc}
  x & y & \\
  L & 2 & -\infty \\
  U & 1 & 3 \\
\end{array}
\]

\((s, v) \sim_M (s, v')\) iff they enable the same sequences of transitions
Idea: bounds are local to each state in the automaton

\[ M(x) = 2 \quad M(x) = 2 \]

\[ s_0 \xrightarrow{x \geq 1} s_1 \quad x \leq 2 \]

\[ (x = 1.5) \not\sim (x = 3) \]

\[ M(x) = 1 \quad M(x) = 2 \]

\[ s_0 \xrightarrow{x \geq 1} s_1 \quad x \leq 2 \]

\[ (x = 1.5) \sim (x = 3) \]
**Local bounds [BBFL03]**

**Idea:** bounds are **local to each state** in the automaton

\[
M(x) = 2
\]

\[
M(x) = 2
\]

\[
M(x) = 1
\]

\[
M(x) = 2
\]

\[
(x = 1.5) \not\sim (x = 3)
\]

\[
(x = 1.5) \sim (x = 3)
\]

\[
(s, v) \sim_M (s, v') \text{ iff they enable the same sequences of transitions}
\]
Local bounds [BBFL03]

**Idea:** bounds are **local to each state** in the automaton

\[
M(x) = 2 \quad M(x) = 2 \quad M(x) = 1 \quad M(x) = 2
\]

\[
\begin{array}{c}
s_0 \xrightarrow{x \geq 1} s_1 \\
(x = 1.5) \not\sim (x = 3)
\end{array}
\quad
\begin{array}{c}
s_0 \xrightarrow{x \leq 2} s_1 \\
(x = 1.5) \sim (x = 3)
\end{array}
\quad
\begin{array}{c}
s_0 \xrightarrow{x \geq 1} s_1 \\
M(x) = 1
\end{array}
\quad
\begin{array}{c}
s_0 \xrightarrow{x = 0} s_1 \\
M(x) = 2
\end{array}
\]

\[(s, v) \sim_M (s, v') \text{ iff they enable the same sequences of transitions}\]

**Computation:** static analysis on the automaton
Better abstraction: look at semantics

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

\[ s_0 \rightarrow s_1 \quad x \geq 2 \]

\[ s_1 \rightarrow s_3 \quad x < 1 \]

\[ s_3 \rightarrow s_2 \quad y = 10^6 \]
Better abstraction: look at semantics

Static analysis: $M(y) = 10^6$

$\begin{align*}
x &= 1 \\
x &:= 0
\end{align*}$
Better abstraction: look at semantics

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

Static analysis: \( M(y) = 10^6 \)

More than 10^6 zones at \( s_0 \) not necessary!
On-the-fly bounds

- Bounds $M$ local to the nodes in the reachability tree
- $M$ are updated on-the-fly by propagation
- Abstraction using local bounds: $Z_2 \subseteq a_{M_1}(Z_1)$

Bounds can be updated as abstractions are not stored
On-the-fly bounds propagation

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

\[ (q, Z), M \]

All tentative nodes consistent → No more exploration → Terminate!
On-the-fly bounds propagation

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

\[ (q, Z), M \]

\[ x \leq 3 \]
On-the-fly bounds propagation

\[ M(x) = 3 \]

\[ (q, Z), M \]

\[ x \leq 3 \]
On-the-fly bounds propagation

\[ M(x) = 3 \]

\[ (q, Z), M \]

\[ x \leq 3 \]
On-the-fly bounds propagation

\[ M(x) = 5 \]

\[ (q, Z), M \]

\[ x \leq 3 \]
On-the-fly bounds propagation

\[ M(x) = 5 \]

\[ Z' \subseteq a_M(Z) \]

\[ (q, Z), M \]

\[ x \leq 3 \]

\[ (q, Z'), M' \]
On-the-fly bounds propagation

\[ M(x) = 5 \]

\[ Z' \subseteq a_M(Z) \]

\[ (q, Z), M \]

\[ x \leq 3 \]

\[ x > 6 \]
On-the-fly bounds propagation

\[ M(x) = 6 \]

\[ (q, Z), M \]

\[ Z' \subseteq a_M(Z) \]

\[ x \leq 3 \]

\[ x > 6 \]
On-the-fly bounds propagation

\[ M(x) = 6 \]

\[ (q, Z), M \]

\[ Z' \subseteq a_M(Z) \]

\[ x \leq 3 \]

\[ x > 6 \]
On-the-fly bounds propagation

\[ M(x) = 6 \]

\[ (q, Z), M \]

\[ Z' \subseteq a_M(Z) \]

\[ M'(x) = 6 \]

\[ (q, Z'), M' \]

\[ x \leq 3 \]

\[ x > 6 \]
On-the-fly bounds propagation

\[ M(x) = 6 \]

\[ (q, Z), M \]

\[ x \leq 3 \]

\[ x \geq 11 \]

\[ Z' \subseteq a_M(Z) \]

\[ (q, Z'), M' \]
On-the-fly bounds propagation

\[ M(x) = 11 \]

\[ Z' \subseteq a_M(Z) \]

\[ (q, Z), M \]

\[ x \leq 3 \]

\[ x > 6 \]

\[ x \geq 11 \]
On-the-fly bounds propagation

\[ M(x) = 11 \]

\[ (q, Z), M \]

\[ Z' \subseteq a_M(Z) \]

\[ x \leq 3 \]

\[ x > 6 \]

\[ x \geq 11 \]

\[ x \geq 6 \]
On-the-fly bounds propagation

\[ M(x) = 11 \]

\[ (q, Z), M \]

\[ Z' \subseteq \alpha_M(Z) \]

\[ x \leq 3 \]

\[ x \geq 11 \]

\[ x > 6 \]
On-the-fly bounds propagation

\[ M(x) = 11 \]

\[ (q, Z), M \]

\[ x := 0 \]

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

\[ x > 6 \]

\[ Z' \subseteq a_M(Z) \]
On-the-fly bounds propagation

\[ M(x) = 11 \]

\[ (q, Z), M \]

\[ Z' \subseteq a_M(Z) \]

All tentative nodes consistent

\[ + \text{ No more exploration} \]

\[ \rightarrow \text{Terminate!} \]
On-the-fly bounds propagation (cont’d)

Theorem
The algorithm is correct and it terminates

- **Non tentative nodes:** $M = \max\{M_{\text{succ}}\}$ (resets)
- **Tentative nodes:** $M = M_{\text{covering}}$
- $M$ only increases and is bounded by [BBFL03]

$(s, v) \sim_M (s, v')$ iff they enable the **same sequences of transitions**
Experiments I

\[ A_1 \]

\[
\begin{array}{c}
q_0 \\
x := 0 \\
y \geq 20 && x = 2 \\
y = 10000 \\
x := 1 \\
x := 5 \\
q_1 \\
q_3 \\
y = 10000 \\
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
A_1 & \text{nodes} & \text{s.} \\
\hline
\text{Extra}^+_L, \text{sa} & 4001 & 6.16 \\
\alpha_L, \text{otf} & 9 & 0.00 \\
\hline
\end{array}
\]
Experiments II

\( A_2 \)

<table>
<thead>
<tr>
<th>( A_2 )</th>
<th>nodes</th>
<th>s.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Extra}^+_\text{LU, sa} )</td>
<td>10014</td>
<td>95.62</td>
</tr>
<tr>
<td>( \alpha_{LU, otf} )</td>
<td>3</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Experiments III

\[ A_3 \]

<table>
<thead>
<tr>
<th></th>
<th>nodes</th>
<th>s.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_3 )</td>
<td>20006</td>
<td>99.26</td>
</tr>
<tr>
<td>( \text{Extra}_{LU, \text{sa}}^+ )</td>
<td>4</td>
<td>0.00</td>
</tr>
</tbody>
</table>
But we can do even better...

**Idea:** only the **disabled edges** matter!

- Take bounds from the **disabled edges** only
- **Optimize** propagation
Outline

Timed Automata and the Reachability Problem

Symbolic semantics and abstractions

Bounds based abstractions

Small bounds for abstractions

Conclusions and future work
## Benchmarks

<table>
<thead>
<tr>
<th>Model</th>
<th>nb. of clocks</th>
<th>UPPAAL (-C) nodes</th>
<th>Extra$_{LU}$,sa nodes</th>
<th>$\alpha_{LU}$,otf nodes</th>
<th>$\alpha_{LU}$,dis. nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sched$_7$</td>
<td>14</td>
<td>18654</td>
<td>18654</td>
<td>213</td>
<td>72</td>
</tr>
<tr>
<td>Sched$_8$</td>
<td>16</td>
<td>311310</td>
<td>311309</td>
<td>198669</td>
<td>123915</td>
</tr>
<tr>
<td>Sched$_{70}$</td>
<td>140</td>
<td>786447</td>
<td>786446</td>
<td>493582</td>
<td>294924</td>
</tr>
<tr>
<td>CSMA/CD 10</td>
<td>11</td>
<td>120845</td>
<td>120844</td>
<td>78604</td>
<td>51210</td>
</tr>
<tr>
<td>CSMA/CD 11</td>
<td>12</td>
<td>311210</td>
<td>311309</td>
<td>198669</td>
<td>123915</td>
</tr>
<tr>
<td>CSMA/CD 12</td>
<td>13</td>
<td>786447</td>
<td>786446</td>
<td>493582</td>
<td>294924</td>
</tr>
<tr>
<td>FDDI 50</td>
<td>151</td>
<td>12605</td>
<td>12606</td>
<td>5448</td>
<td>401</td>
</tr>
<tr>
<td>FDDI 70</td>
<td>211</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDDI 140</td>
<td>421</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fischer 9</td>
<td>9</td>
<td>135485</td>
<td>135485</td>
<td>135485</td>
<td>135485</td>
</tr>
<tr>
<td>Fischer 10</td>
<td>10</td>
<td>447598</td>
<td>447598</td>
<td>447598</td>
<td>447598</td>
</tr>
<tr>
<td>Fischer 11</td>
<td>11</td>
<td>1464971</td>
<td>1464971</td>
<td>1464971</td>
<td>1464971</td>
</tr>
<tr>
<td>Stari 2</td>
<td>7</td>
<td>7870</td>
<td>6993</td>
<td>5779</td>
<td>4202</td>
</tr>
<tr>
<td>Stari 3</td>
<td>10</td>
<td>136632</td>
<td>113958</td>
<td>82182</td>
<td>29964</td>
</tr>
<tr>
<td>Stari 4</td>
<td>13</td>
<td>1323193</td>
<td>983593</td>
<td>602762</td>
<td>278120</td>
</tr>
</tbody>
</table>

Both non-convex abstractions $\alpha_M/\alpha_{LU}$ and on-the-fly bounds computation help.
Conclusions

- New reachability algorithm with **non-convex abstractions** and **on-the-fly computation** of bounds
- **Optimal abstraction** when only bounds are considered
- **Tightening** of bounds
Future Work

\[ (s_0, Z_0) \alpha_0 \]

\[ (s, Z_1) \alpha_1 \]

\[ (s, Z_2) \alpha_2 \]

\[ Z_1 \] defines the subtree, which in turn defines \( \alpha_1 \)
Future Work

$Z_1$ defines the subtree, which in turn defines $a_1$

- **Optimal** bounds: better propagation

- Beyond bounds: define $a$ from constraints, ...
- Extend to infinite runs (beyond reachability)
- Prototype tool
Future Work

\( (s, Z_0) \ a_0 \)

\( (s, Z_1) \ a_1 \)

\( (s, Z_2) \ a_2 \)

\( Z_1 \) defines the subtree, which in turn defines \( a_1 \)

- **Optimal** bounds: better propagation
- **Beyond bounds**: define \( a \) from constraints, . . .
Future Work

$Z_1$ defines the subtree, which in turn defines $a_1$

- **Optimal** bounds: better propagation
- **Beyond bounds**: define $a$ from constraints, . . .
- Extend to **infinite runs** (beyond reachability)
Future Work

\[(s_0, Z_0) \ a_0\]

\[(s, Z_1) \ a_1\]

\[(s, Z_2) \ a_2\]

\(Z_1\) defines the subtree, which in turn defines \(a_1\)

- **Optimal** bounds: better propagation
- **Beyond bounds**: define \(a\) from constraints, . . .
- Extend to **infinite runs** (beyond reachability)
- Prototype **tool**
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.

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.

François Laroussinie and Philippe Schnoebelen.
The state explosion problem from trace to bisimulation equivalence.
When is $Z' \subseteq \alpha_M(Z)$?

**Recall:** $\alpha_M(Z)$ is the union of regions that intersect $Z$.
When is $Z' \subseteq a_M(Z)$?

**Recall:** $a_M(Z)$ is the union of regions that intersect $Z$.
When is $Z' \subseteq a_M(Z)$?

Recall: $a_M(Z)$ is the union of regions that intersect $Z$.
When is $Z' \subseteq a_M(Z)$?

Recall: $a_M(Z)$ is the union of regions that intersect $Z$

$Z' \not\subseteq a_M(Z)$ if and only if there exist 2 clocks $x, y$ s.t.

$\text{Proj}_{xy}(Z') \not\subseteq a_M(\text{Proj}_{xy}(Z))$

iff

$\exists R. R$ intersects $Z'$, but $R$ does not intersect $Z$
When is \( Z' \subseteq a_M(Z) \)?

Recall: \( a_M(Z) \) is the union of regions that intersect \( Z \)

\[ Z' \not\subseteq a_M(Z) \] iff

\( \exists R. \ R \) intersects \( Z' \), but \( R \) does not intersect \( Z \)

**Theorem**

\( Z' \not\subseteq a_M(Z) \) if and only if there exist 2 clocks \( x, y \) s.t.

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