Revisiting Reachability in Timed Automata

Karin Quaas ¹, Mahsa Shirmohammadi ², James Worrell ²

¹ Universität Leipzig, ² University of Oxford

LICS 2017
Timed Automata (TA)

\[0 < x_1 < 1 \quad x_1 \leftarrow 0 \quad x_2 = 1\]

\[(\begin{array}{c} 0.5 \\ 0.2 \end{array}) \rightarrow (\begin{array}{c} 0 \\ 0.6 \end{array}) \rightarrow (\begin{array}{c} 0.4 \\ 1.0 \end{array}) \rightarrow (\begin{array}{c} 0.7 \\ 1.7 \end{array}) \ldots\]

Parametric TCTL

\[\phi = \exists \theta \exists \Diamond = \theta (p_1 \land \exists \Diamond = \theta p_2)\]

Does there exist some value \(\theta\) such that there exists a run of duration \(\theta\) to a location in which \(p_1\) holds and from which there exists a run of duration \(\theta\) to a location in which \(p_2\) holds?

Model Checking

Does \((\ell_0, (0.6, 0.2))\) hold? Yes, but only if we allow real-valued parameters.

\[\ast (0.5, 0.2) \text{ and } (0.6, 0.7) \text{ are region-equivalent, but } (\ell_0, (0.6, 0.7)) \neq \phi : (0.7, 1.7) \rightarrow (0.8, 1.8) \ldots\]

Classical region-based decision procedures do not work.

\[\ast \ast \text{As opposed to Bruyère et al, 2003/08, where only integer-valued parameters are allowed}\]
Timed Automata (TA)

\[
\begin{align*}
0 < x_1 < 1 & \quad x_1 \leftarrow 0 \quad x_2 = 1
\end{align*}
\]

\[
\begin{pmatrix}
0.5 \\ 0.2
\end{pmatrix}
\rightarrow
\begin{pmatrix}
0.6 \\ 0
\end{pmatrix}
\rightarrow
\begin{pmatrix}
0.4 \\ 1.0
\end{pmatrix}
\rightarrow
\begin{pmatrix}
0.7 \\ 1.1
\end{pmatrix}
\rightarrow
\begin{pmatrix}
1.1 \\ 1.7
\end{pmatrix}
\rightarrow
\]

Parametric TCTL

\[
\varphi = \exists \theta \exists \Diamond = \theta (p_1 \land \exists \Diamond = \theta p_2)
\]

Does there exist some value \( \theta \) such that there exists a run of duration \( \theta \) to a location in which \( p_1 \) holds and from which there exists a run of duration \( \theta \) to a location in which \( p_2 \) holds?
Timed Automata (TA)

\[
\begin{align*}
0 < x_1 < 1 & \quad x_1 \leftarrow 0 \quad x_2 = 1 \\
(0.5, 0.2) & \rightarrow (0.6, 0.4) \rightarrow (1.0, 0.4) \rightarrow (1.7, 0.7) \rightarrow \cdots
\end{align*}
\]

Parametric TCTL

\[\varphi = \exists \theta \exists \diamond \theta (p_1 \land \exists \diamond \theta p_2)\]

Does there exist some value \(\theta\) such that there exists a run of duration \(\theta\) to a location in which \(p_1\) holds and from which there exists a run of duration \(\theta\) to a location in which \(p_2\) holds?
**Model Checking Timed Automata & Parametric TCTL**

**Timed Automata (TA)**

\[ 0 < x_1 < 1 \quad x_1 \leftarrow 0 \quad x_2 = 1 \]

\[
\begin{pmatrix}
0.5 \\
0.2
\end{pmatrix}
\overset{0.4}{\rightarrow}
\begin{pmatrix}
0 \\
0.6
\end{pmatrix}
\overset{0.4}{\rightarrow}
\begin{pmatrix}
0.4 \\
1.0
\end{pmatrix}
\overset{0.7}{\rightarrow}
\begin{pmatrix}
1.1 \\
1.7
\end{pmatrix} \ldots
\]

**Parametric TCTL**

\[
\varphi = \exists \theta \exists \Diamond = \theta (p_1 \land \exists \Diamond = \theta p_2)
\]

Does there exist some value \( \theta \) such that there exists a run of duration \( \theta \) to a location in which \( p_1 \) holds and from which there exists a run of duration \( \theta \) to a location in which \( p_2 \) holds?
Timed Automata (TA)

\[ 0 < x_1 < 1 \quad x_1 \leftarrow 0 \quad x_2 = 1 \]

Parametric TCTL

\[ \varphi = \exists \theta \exists \diamond = \theta (p_1 \land \exists \diamond = \theta p_2) \]

Does there exist some value \( \theta \) such that there exists a run of duration \( \theta \) to a location in which \( p_1 \) holds and from which there exists a run of duration \( \theta \) to a location in which \( p_2 \) holds?
Timed Automata (TA)

\[ 0 < x_1 < 1 \quad x_1 \leftarrow 0 \quad x_2 = 1 \]

\[ (\frac{0.5}{0.2}) \xrightarrow{0.4} (\frac{0}{0.6}) \xrightarrow{0.4} (\frac{0.4}{1.0}) \xrightarrow{0.7} (\frac{1.1}{1.7}) \ldots \]

Parametric TCTL

\[ \varphi = \exists \theta \exists \Diamond_{=\theta} (p_1 \land \exists \Diamond_{=\theta} p_2) \]

Does there exist some value \( \theta \) such that there exists a run of duration \( \theta \) to a location in which \( p_1 \) holds and from which there exists a run of duration \( \theta \) to a location in which \( p_2 \) holds?
Model Checking Timed Automata & Parametric TCTL

Timed Automata (TA)

\[0 < x_1 < 1 \quad x_1 \leftarrow 0 \quad x_2 = 1\]

\[
\begin{pmatrix}
0.5 \\
0.2
\end{pmatrix}
\rightarrow
\begin{pmatrix}
0.4 \\
0.6
\end{pmatrix}
\rightarrow
\begin{pmatrix}
0.4 \\
1.0
\end{pmatrix}
\rightarrow
\begin{pmatrix}
0.7 \\
1.7
\end{pmatrix}
\rightarrow
\cdots
\]

Parametric TCTL

\[\varphi = \exists \theta \exists \diamond \equiv \theta (p_1 \land \exists \diamond \equiv \theta p_2)\]

Does there exist some value \(\theta\) such that there exists a run of duration \(\theta\) to a location in which \(p_1\) holds and from which there exists a run of duration \(\theta\) to a location in which \(p_2\) holds?

As opposed to Bruyère et al, 2003/08, where only integer-valued parameters are allowed.
Timed Automata (TA)

Parametric TCTL

$$\varphi = \exists \theta \exists \diamond = \theta (p_1 \land \exists \diamond = \theta p_2)$$

Does there exist some value \(\theta\) such that there exists a run of duration \(\theta\) to a location in which \(p_1\) holds and from which there exists a run of duration \(\theta\) to a location in which \(p_2\) holds?
Timed Automata (TA)

\[ 0 < x_1 < 1 \quad x_1 \leftarrow 0 \quad x_2 = 1 \]

\[
\begin{pmatrix} 0.5 \\ 0.2 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 0.6 \end{pmatrix} \rightarrow \begin{pmatrix} 0.4 \\ 1.0 \end{pmatrix} \rightarrow \begin{pmatrix} 1.1 \\ 1.7 \end{pmatrix} \ldots
\]

Parametric TCTL

\[ \varphi = \exists \theta \exists \Diamond = \theta (p_1 \land \exists \Diamond = \theta p_2) \]

Does there exist some value \( \theta \) such that there exists a run of duration \( \theta \) to a location in which \( p_1 \) holds and from which there exists a run of duration \( \theta \) to a location in which \( p_2 \) holds?

Model Checking

Does \( (\ell_0, \begin{pmatrix} 0.5 \\ 0.2 \end{pmatrix}) \models \varphi \) hold?
Timed Automata (TA)

\[
\begin{align*}
0 < x_1 < 1 & \quad x_1 \leftarrow 0 \\
x_2 = 1 & \quad \rightarrow p_1 \\
& \quad \rightarrow p_2
\end{align*}
\]

\[
\begin{align*}
\begin{pmatrix} 0.5 \\ 0.2 \end{pmatrix} & \xrightarrow{0.4} \begin{pmatrix} 0.6 \\ 0 \end{pmatrix} & \xrightarrow{0.4} \begin{pmatrix} 0.4 \\ 1 \end{pmatrix} & \xrightarrow{0.7} \begin{pmatrix} 1 \frac{1}{7} \end{pmatrix} & \ldots
\end{align*}
\]

Parametric TCTL

\[
\phi = \exists \theta \exists \diamond \equiv \theta (p_1 \land \exists \diamond \equiv \theta p_2)
\]

Does there exist some value \( \theta \) such that there exists a run of duration \( \theta \) to a location in which \( p_1 \) holds and from which there exists a run of duration \( \theta \) to a location in which \( p_2 \) holds?

Model Checking

Does \((\ell_0, \begin{pmatrix} 0.5 \\ 0.2 \end{pmatrix})\) \models \phi hold? Yes, but only if we allow real-valued parameters.*
Timed Automata (TA)

\[ 0 < x_1 < 1 \quad x_1 \leftarrow 0 \quad x_2 = 1 \]

\[ \begin{align*}
(\begin{pmatrix} 0.5 \\ 0.2 \end{pmatrix}) & \xrightarrow{0.4} (\begin{pmatrix} 0.6 \\ 0 \end{pmatrix}) & \xrightarrow{0.4} (\begin{pmatrix} 0.4 \\ 1.0 \end{pmatrix}) & \xrightarrow{0.7} (\begin{pmatrix} 1.1 \\ 1.7 \end{pmatrix}) & \ldots
\end{align*} \]

Parametric TCTL

\[ \varphi = \exists \theta \exists \Diamond = \theta (p_1 \land \exists \Diamond = \theta p_2) \]

Does there exist some value \( \theta \) such that there exists a run of duration \( \theta \) to a location in which \( p_1 \) holds and from which there exists a run of duration \( \theta \) to a location in which \( p_2 \) holds?

Model Checking

Does \((\ell_0, (\begin{pmatrix} 0.5 \\ 0.2 \end{pmatrix})) \models \varphi\) hold? Yes, but only if we allow real-valued parameters.*

\((\begin{pmatrix} 0.5 \\ 0.2 \end{pmatrix}) \) and \((\begin{pmatrix} 0.7 \\ 0.2 \end{pmatrix}) \) are region-equivalent, but \((\ell_0, (\begin{pmatrix} 0.7 \\ 0.2 \end{pmatrix})) \not\models \varphi\)
Timed Automata (TA)

\[ 0 < x_1 < 1 \quad x_1 \leftarrow 0 \quad x_2 = 1 \]

\[ \begin{align*}
    (0.5, 0.2) &\xrightarrow{0.4} (0.6, 0.2) &\xrightarrow{0.4} (0.4, 1.0) &\xrightarrow{0.7} (1.1, 1.7) &\ldots
\end{align*} \]

Parametric TCTL

\[ \varphi = \exists \theta \, \exists \Diamond = \theta (p_1 \land \exists \Diamond = \theta p_2) \]

Does there exist some value \( \theta \) such that there exists a run of duration \( \theta \) to a location in which \( p_1 \) holds and from which there exists a run of duration \( \theta \) to a location in which \( p_2 \) holds?

Model Checking

Does \( (\ell_0, (0.5, 0.2)) \models \varphi \) hold? Yes, but only if we allow real-valued parameters.*

\[ (0.5, 0.2) \text{ and } (0.7, 0.2) \text{ are region-equivalent, but } (\ell_0, (0.7, 0.2)) \not\models \varphi: (0.7, 0.2) \xrightarrow{0.4} \]
Timed Automata (TA)

\[
\begin{align*}
0 < x_1 < 1 & \quad \xrightarrow{x_1 \leftarrow 0} \quad p_1 \\
\text{ } & \quad \xrightarrow{x_2 = 1} \quad p_2
\end{align*}
\]

\[
\begin{array}{c}
(0.5, 0.2) \\
\downarrow 0.4 \quad \downarrow 0.4 \quad \downarrow 0.7
\end{array}
\xrightarrow{\text{...}}
(0.7, 0.2)
\]

Parametric TCTL

\[
\varphi = \exists \theta \exists ! = \theta (p_1 \land \exists ! = \theta p_2)
\]

Does there exist some value \(\theta\) such that there exists a run of duration \(\theta\) to a location in which \(p_1\) holds and from which there exists a run of duration \(\theta\) to a location in which \(p_2\) holds?

Model Checking

Does \((\ell_0, (0.5, 0.2)) \models \varphi\) hold? Yes, but only if we allow real-valued parameters.

\((0.5, 0.2)\) and \((0.7, 0.2)\) are region-equivalent, but \((\ell_0, (0.7, 0.2)) \not\models \varphi: (0.7, 0.2) \xrightarrow{0.4} \\
\]

Classical region-based decision procedures do not work.

* As opposed to Bruyère et al, 2003/08, where only integer-valued parameters are allowed.
Goal

Computing the Reachability Relation

**Instance:** A timed automaton and locations $\ell, \ell'$

**Output:** First-order formula $\varphi_{\ell,\ell'}(y, y')$ such that $(v, v') \models \varphi_{\ell,\ell'}$ iff $(\ell, v) \rightarrow^* (\ell', v')$. 

**Historical Remarks**

Alur and Dill, 1990: control-state reachability without clock values


Our Logic

Existential fragment of mix of real-arithmetic and Presburger arithmetic
Computing the Reachability Relation

**Instance:** A timed automaton and locations $\ell, \ell'$

**Output:** First-order formula $\varphi_{\ell,\ell'}(y, y')$ such that $(v, v') \models \varphi_{\ell,\ell'}$ iff $(\ell, v) \rightarrow^* (\ell', v')$.

**Historical Remarks**
Alur and Dill, 1990: control-state reachability without clock values
Goal

Computing the Reachability Relation

Instance: A timed automaton and locations $\ell, \ell'$

Output: First-order formula $\varphi_{\ell, \ell'}(y, y')$ such that

$$ (v, v') \models \varphi_{\ell, \ell'} \iff (\ell, v) \rightarrow^* (\ell', v'). $$

Historical Remarks

Alur and Dill, 1990: control-state reachability without clock values
Goal

Computing the Reachability Relation

**Instance:** A timed automaton and locations $\ell, \ell'$

**Output:** First-order formula $\varphi_{\ell,\ell'}(y, y')$ such that $(v, v') \models \varphi_{\ell,\ell'}$ iff $(\ell, v) \rightarrow^* (\ell', v')$.

Historical Remarks

Alur and Dill, 1990: control-state reachability without clock values

Our Logic

Existential fragment of mix of real-arithmetic and Presburger arithmetic
Given a timed automaton with \( n \) clocks:

- Clock valuations \( \mathbf{v} = (v_1, \ldots, v_n) \)
Given a timed automaton with $n$ clocks:

- Clock valuations $\mathbf{v} = (v_1, \ldots, v_n)$
- Integer parts $\lfloor \mathbf{v} \rfloor = (\lfloor v_1 \rfloor, \ldots, \lfloor v_n \rfloor)$

Note: $\mathbf{v} = \lfloor \mathbf{v} \rfloor + \text{fr}(\mathbf{v})$.
Given a timed automaton with \( n \) clocks:

- Clock valuations \( \mathbf{v} = (v_1, \ldots, v_n) \)
- Integer parts \( \lfloor \mathbf{v} \rfloor = (\lfloor v_1 \rfloor, \ldots, \lfloor v_n \rfloor) \)
- Fractional parts \( \text{fr}(\mathbf{v}) = (\text{fr}(v_1), \ldots, \text{fr}(v_n)) \)
Notation

Given a timed automaton with \( n \) clocks:

- Clock valuations \( \mathbf{v} = (v_1, \ldots, v_n) \)
- Integer parts \( \lfloor \mathbf{v} \rfloor = (\lfloor v_1 \rfloor, \ldots, \lfloor v_n \rfloor) \)
- Fractional parts \( \text{fr}(\mathbf{v}) = (\text{fr}(v_1), \ldots, \text{fr}(v_n)) \)

Note: \( \mathbf{v} = \lfloor \mathbf{v} \rfloor + \text{fr}(\mathbf{v}) \).
Notation

Given a timed automaton with $n$ clocks:

- Clock valuations $\mathbf{v} = (v_1, \ldots, v_n)$
- Integer parts $\lfloor \mathbf{v} \rfloor = (\lfloor v_1 \rfloor, \ldots, \lfloor v_n \rfloor)$
- Fractional parts $\text{fr}(\mathbf{v}) = (\text{fr}(v_1), \ldots, \text{fr}(v_n))$

Note: $\mathbf{v} = \lfloor \mathbf{v} \rfloor + \text{fr}(\mathbf{v})$.

- Integer-valued variables $\mathbf{z} = (z_1, \ldots, z_n)$
Given a timed automaton with $n$ clocks:

- Clock valuations $\mathbf{v} = (v_1, \ldots, v_n)$
- Integer parts $\lfloor \mathbf{v} \rfloor = (\lfloor v_1 \rfloor, \ldots, \lfloor v_n \rfloor)$
- Fractional parts $\text{fr}(\mathbf{v}) = (\text{fr}(v_1), \ldots, \text{fr}(v_n))$

Note: $\mathbf{v} = \lfloor \mathbf{v} \rfloor + \text{fr}(\mathbf{v})$.

- Integer-valued variables $\mathbf{z} = (z_1, \ldots, z_n)$
- Real-valued variables $\mathbf{r} = (r_1, \ldots, r_n)$
Solution: Step I

**Instance:** Locations $\ell, \ell'$, concrete valuation $v$

**Output:** $\varphi_{\ell,\ell',v}(z, r, z', r')$ such that

$\left(\lfloor v \rfloor, \text{fr}(v), \lfloor v' \rfloor, \text{fr}(v')\right) \models \varphi_{\ell,\ell',v}$ iff $(\ell, v) \rightarrow^* (\ell', v')$
Solution: Step I

**Instance:** Locations $\ell, \ell'$, concrete valuation $v$

**Output:** $\varphi_{\ell, \ell', v}(z, r, z', r')$ such that 

$\left(\lfloor v \rfloor, \text{fr}(v), \lfloor v' \rfloor, \text{fr}(v') \right) \models \varphi_{\ell, \ell', v}$ iff $(\ell, v) \rightarrow^* (\ell', v')$

To obtain $\varphi_{\ell, \ell', v}$, we construct a monotonic counter machine $M_{\ell, \ell', v}$
Solution: Step 1

Instance: Locations ℓ, ℓ′, concrete valuation v
Output: \( \varphi_{\ell, \ell', v}(z, r, z', r') \) such that 
\[
(\lfloor v \rfloor, \text{fr}(v), \lfloor v' \rfloor, \text{fr}(v')) \models \varphi_{\ell, \ell', v} \text{ iff } (\ell, v) \rightarrow^* (\ell', v')
\]

To obtain \( \varphi_{\ell, \ell', v} \), we construct a monotonic counter machine \( M_{\ell, \ell', v} \)

- sound and complete with respect to reachability
Solution: Step I

**Instance:** Locations $\ell, \ell'$, concrete valuation $v$

**Output:** $\varphi_{\ell, \ell', v}(z, r, z', r')$ such that

$\left(\lfloor v \rfloor, \text{fr}(v), \lfloor v' \rfloor, \text{fr}(v')\right) \models \varphi_{\ell, \ell', v}$ iff $(\ell, v) \rightarrow^* (\ell', v')$

To obtain $\varphi_{\ell, \ell', v}$, we construct a monotonic counter machine $M_{\ell, \ell', v}$

- sound and complete with respect to reachability
- control states $q = (\ell, D)$, where $D$ is a DBM representing a *zone* over $[0, 1]^n$ representing the fractional parts of the clock values
Solution: Step I

**Instance:** Locations \( \ell, \ell' \), concrete valuation \( v \)

**Output:** \( \varphi_{\ell, \ell', v}(z, r, z', r') \) such that

\[
([v], \text{fr}(v), [v'], \text{fr}(v')) \models \varphi_{\ell, \ell', v} \text{ iff } (\ell, v) \rightarrow^* (\ell', v')
\]

To obtain \( \varphi_{\ell, \ell', v} \), we construct a monotonic counter machine \( M_{\ell, \ell', v} \)

- sound and complete with respect to reachability

- control states \( q = (\ell, D) \), where \( D \) is a DBM representing a zone over \([0, 1]^n\) representing the fractional parts of the clock values

- counter \( c_i \) stores the integer part of clock \( x_i \)
Solution: Step I, Example

$0 < x_1 < 1 \quad x_1 \leftarrow 0 \quad x_2 = 1 \quad \ell = \circ \quad \mathbf{v} = \begin{pmatrix} 0.5 \\ 0.2 \end{pmatrix} \quad \ell' = \circ$

\[ x_1 \]
\[ x_2 \]

- \[ x_2 \]
- \[ x_1 \]
- \[ c_1 < 1 \quad c_1 \leftarrow 0 \]
- \[ c_2 < 1 \]
- \[ \text{inc } c_2 \]
- \[ \text{inc } c_1 \]
- \[ \text{inc } c_2 \]

\[ x_1 \]
\[ x_2 \]
Solution: Step I

**Instance:** Locations $\ell, \ell'$, concrete valuation $v$

**Output:** $\varphi_{\ell,\ell',v}(z, r, z', r')$ such that

$([v], \text{fr}(v), [v'], \text{fr}(v')) \models \varphi_{\ell,\ell',v}$ iff $(\ell, v) \rightarrow^* (\ell', v')$

To obtain $\varphi_{\ell,\ell',v}$, we construct a monotonic counter machine $M_{\ell,\ell',v}$

- sound and complete with respect to reachability
- control states $q = (\ell, D)$, where $D$ is a DBM representing a zone over $[0, 1]^n$ representing the fractional parts of the clock values

![Diagram of a monotonic counter machine]

- counter $c_i$ stores the integer part of clock $x_i$
Solution: Step I

**Instance:** Locations $\ell, \ell'$, concrete valuation $v$

**Output:** $\varphi_{\ell,\ell',v}(z, r, z', r')$ such that $(\lfloor v \rfloor, \text{fr}(v), \lfloor v' \rfloor, \text{fr}(v')) \models \varphi_{\ell,\ell',v}$ iff $(\ell, v) \rightarrow^* (\ell', v')$

To obtain $\varphi_{\ell,\ell',v}$, we construct a monotonic counter machine $M_{\ell,\ell',v}$

- sound and complete with respect to reachability
- control states $q = (\ell, D)$, where $D$ is a DBM representing a *zone* over $[0, 1]^n$ representing the fractional parts of the clock values

The reachability relation of monotonic counter machines is definable in Presburger arithmetic.
Solution: Step I, Example

\[ 0 < x_1 < 1 \quad \xrightarrow{x_1 \leftarrow 0} \quad x_2 = 1 \]

\[ \ell = \quad v = \left( \begin{array}{c} 0.5 \\ 0.2 \end{array} \right) \quad \ell' = \]

\[ \varphi_{\ell, \ell', v} = \left( (z_2' - z_1' = 1) \land (0.3 < r_1' \leq 1) \land (0 \leq r_2' < 0.7) \land (0.3 < r_1' - r_2' \leq 0.8) \right) \lor \left( (z_2' - z_1' = 0) \land (0 \leq r_1' \leq 0.8) \land (0.3 < r_2' \leq 1) \land (0.3 < 1 + r_1' - r_2' \leq 0.8) \right) \]
Solution: Step I

**Instance:** Locations $\ell, \ell'$, concrete valuation $v$

**Output:** $\varphi_{\ell,\ell',v}(z, r, z', r')$ such that

$([v], \text{fr}(v), [v'], \text{fr}(v')) \models \varphi_{\ell,\ell',v}$ iff $(\ell, v) \rightarrow^* (\ell', v')$

To obtain $\varphi_{\ell,\ell',v}$, we construct a monotonic counter machine $M_{\ell,\ell',v}$

- sound and complete with respect to reachability
- control states $q = (\ell, D)$, where $D$ is a DBM representing a zone over $[0, 1]^n$ representing the fractional parts of the clock values

$$
\begin{pmatrix}
  x_0 \\
  x_1 \\
  x_2
\end{pmatrix}
\begin{bmatrix}
  \leq 0 & < -0.3 & \leq 0 \\
  \leq 1 & \leq 0 & \leq 0.8 \\
  < 0.7 & < -0.3 & \leq 0
\end{bmatrix}
(0.3 < r_1 \leq 1) \land (0 \leq r_2 < 0.7)
\land (0.3 < r_1 - r_2 \leq 0.8)
$$

- counter $c_i$ stores the integer part of clock $x_i$

The reachability relation of monotonic counter machines is definable in Presburger arithmetic.
Solution: Step II

**Instance:** Locations \( \ell, \ell' \), concrete valuation \( v \)

**Output:** \( \varphi_{\ell, \ell', \text{type}(u)}(z, r, z', r') \) such that

\[
([u], \text{fr}(u), [v'], \text{fr}(v')) \models \varphi_{\ell, \ell', \text{type}(v)} \text{ iff } (\ell, u) \rightarrow^* (\ell', v')
\]

for all \( u \) with \( \text{type}(u) = \text{type}(v) \).
Solution: Step II

Instance: Locations $\ell, \ell'$, concrete valuation $v$
Output: $\varphi_{\ell,\ell',\text{type}(u)}(z, r, z', r')$ such that
$$(\lfloor u \rfloor, \text{fr}(u), \lfloor v' \rfloor, \text{fr}(v')) \models \varphi_{\ell,\ell',\text{type}(v)}$$
iff $$(\ell, u) \to^* (\ell', v')$$
for all $u$ with $\text{type}(u) = \text{type}(v)$.

Difference Types

$\text{type}(v)$ is collection of formulas of the form
$$c + r_i - r_j \leq c' + r_{i'} - r_{j'}$$
that are satisfied by $v$, where $c, c' \in \{0, 1, -1\}$, $i, j, i', j' \in \{0, \ldots, n\}$. 

Solution: Step II

**Instance:** Locations \(\ell, \ell'\), concrete valuation \(v\)

**Output:** \(\varphi_{\ell,\ell',\text{type}(u)}(z, r, z', r')\) such that

\[(\lfloor u \rfloor, \text{fr}(u), \lfloor v' \rfloor, \text{fr}(v')) \models \varphi_{\ell,\ell',\text{type}(v)} \text{ iff } (\ell, u) \rightarrow^* (\ell', v')\]

for all \(u\) with \(\text{type}(u) = \text{type}(v)\).

**Difference Types**

\(\text{type}(v)\) is collection of formulas of the form

\[c + r_i - r_j \leq c' + r'_i - r'_j\]

that are satisfied by \(v\), where \(c, c' \in \{0, 1, -1\}, i, j, i', j' \in \{0, \ldots, n\}\).

**Example:** \(r_1 - r_2 \leq 1 + r_0 - r_1\) is in \(\text{type}\left(\frac{0.5}{0.2}\right)\), but it is not in \(\text{type}\left(\frac{0.7}{0.2}\right)\)
Solution: Step II

Instance: Locations $\ell, \ell'$, concrete valuation $v$
Output: $\varphi_{\ell,\ell',\text{type}(u)}(z, r, z', r')$ such that
\[
([u], fr(u), [v'], fr(v')) \models \varphi_{\ell,\ell',\text{type}(v)} \iff (\ell, u) \rightarrow^* (\ell', v')
\]
for all $u$ with $\text{type}(u) = \text{type}(v)$.

Difference Types

type($v$) is collection of formulas of the form
\[
c + r_i - r_j \leq c' + r'_i - r'_j
\]
that are satisfied by $v$, where $c, c' \in \{0, 1, -1\}$, $i, j, i', j' \in \{0, \ldots, n\}$.

Example: $r_1 - r_2 \leq 1 + r_0 - r_1$ is in type($\frac{0.5}{0.2}$), but it is not in type($\frac{0.7}{0.2}$).

Note: there are only finitely many distinct different types.
Solution: Step II ctd.

To obtain $\varphi_{\ell,\ell',\text{type}(v)}$, we construct a monotonic counter machine $M_{\ell,\ell',\text{type}(v)}$

- sound and complete with respect to reachability
- control state $q$ is a *parametric DBM* over equivalence classes of difference types (representing the fractional parts of the clock values)

\[
\begin{align*}
  x_0 &\quad (\leq, [0]) & (\leq, [0 - r_1]) & (\leq, [0]) \\
  x_1 &\quad (\leq, [1]) & (\leq, [0]) & (\leq, [1]) \\
  x_2 &\quad (<, [1 - r_1]) & (<, [0 - r_1]) & (\leq, [0])
\end{align*}
\]

- counter $c_i$ stores the integer part of clock $x_i$
To obtain $\varphi_{\ell,\ell',\text{type}(u)}$, we construct a monotonic counter machine $M_{\ell,\ell',\text{type}(v)}$

- sound and complete with respect to rachability
- control state $q$ is a parametric DBM over equivalence classes of difference types (representing the fractional parts of the clock values)

\[
\begin{bmatrix}
    x_0 & (\leq, [0]) & (\leq, [0 - r_1]) & (\leq, [0]) \\
    x_1 & (\leq, [1]) & (\leq, [0]) & (\leq, [1]) \\
    x_2 & (<, [1 - r_1]) & (<, [0 - r_1]) & (\leq, [0]) \\
\end{bmatrix}
\]

- counter $c_i$ stores the integer part of clock $x_i$

The reachability relation of monotonic counter machines is definable in Presburger arithmetic.
Solution: Final Step

Instance: Locations $\ell, \ell'$
Output: $\varphi_{\ell,\ell'}(z, r, z', r')$ such that
$$(\lfloor v \rfloor, \text{fr}(v), \lfloor v' \rfloor, \text{fr}(v')) \models \varphi_{\ell,\ell'} \text{ iff } (\ell, v) \rightarrow^* (\ell', v')$$

To obtain $\varphi_{\ell,\ell'}$, we
- construct for each of the finitely many difference types $T$
- the corresponding formula $\varphi_{\ell,\ell', T}$
- define $\varphi_{\ell,\ell'}$ as the disjunction of all these formulas.

Formula $\varphi_{\ell,\ell'}$ can be computed in time exponential in the size of the timed automaton.
Solution: Final Step

**Instance:** Locations $\ell, \ell'$

**Output:** $\varphi_{\ell,\ell'}(z, r, z', r')$ such that

\[
(\lfloor v \rfloor, \text{fr}(v), \lfloor v' \rfloor, \text{fr}(v')) \models \varphi_{\ell,\ell'} \iff (\ell, v) \rightarrow^*(\ell', v')
\]

To obtain $\varphi_{\ell,\ell'}$, we

- construct for each of the finitely many difference types $T$ the corresponding formula $\varphi_{\ell,\ell', T}$, and
Solution: Final Step

**Instance:** Locations $\ell, \ell'$

**Output:** $\varphi_{\ell,\ell'}(z, r, z', r')$ such that

$$([v], \text{fr}(v), [v'], \text{fr}(v')) \models \varphi_{\ell,\ell'} \iff (\ell, v) \rightarrow^* (\ell', v')$$

To obtain $\varphi_{\ell,\ell'}$, we

- construct for each of the finitely many difference types $T$ the corresponding formula $\varphi_{\ell,\ell',T}$, and

- define $\varphi_{\ell,\ell'}$ as the disjunction of all these formulas.
Solution: Final Step

Instance: Locations \( \ell, \ell' \)

Output: \( \varphi_{\ell,\ell'}(z, r, z', r') \) such that
\[
(\lfloor v \rfloor, \text{fr}(v), \lfloor v' \rfloor, \text{fr}(v')) \models \varphi_{\ell,\ell'} \text{ iff } (\ell, v) \rightarrow^* (\ell', v')
\]

To obtain \( \varphi_{\ell,\ell'} \), we

- construct for each of the finitely many difference types \( T \) the corresponding formula \( \varphi_{\ell,\ell',T} \), and

- define \( \varphi_{\ell,\ell'} \) as the disjunction of all these formulas.

Formula \( \varphi_{\ell,\ell'} \) can be computed in time exponential in the size of the timed automaton.
Back to Model Checking

- reachability fragment of parametric TCTL (without until modality)
- we reduce model checking to deciding the truth of formula in a fragment of first-order logic
- the formula contains reachability formulas
- yields a EXPSPACE-decision procedure for model checking
Model Checking Example

\[ \varphi = \exists \theta \exists \Diamond = \theta (p_1 \land \exists \Diamond = \theta p_2) \]

\[
\begin{aligned}
\begin{array}{c}
0 < x_1 < 1 \\
x_1 \leftarrow 0
\end{array}
\rightarrow
\begin{array}{c}
x_2 = 1
\end{array}
\rightarrow
\begin{array}{c}
\varphi
\end{array}
\end{aligned}
\]

\[
\begin{aligned}
\begin{array}{c}
(0.5) \models \varphi
\end{array}
\rightarrow
\begin{array}{c}
\begin{array}{c}
x_2
\end{array}
\end{array}
\rightarrow
\begin{array}{c}
\begin{array}{c}
x_2
\end{array}
\end{array}
\rightarrow
\begin{array}{c}
\begin{array}{c}
x_2
\end{array}
\end{array}
\rightarrow
\begin{array}{c}
\begin{array}{c}
x_2
\end{array}
\end{array}
\end{aligned}
\]

\[
\begin{aligned}
\begin{array}{c}
(0.7) \not\models \varphi
\end{array}
\rightarrow
\begin{array}{c}
\begin{array}{c}
x_2
\end{array}
\end{array}
\rightarrow
\begin{array}{c}
\begin{array}{c}
x_2
\end{array}
\end{array}
\rightarrow
\begin{array}{c}
\begin{array}{c}
x_2
\end{array}
\end{array}
\rightarrow
\begin{array}{c}
\begin{array}{c}
x_2
\end{array}
\end{array}
\end{aligned}
\]

Add a variable \( r_3 \) measuring the yellow time duration.
Model Checking Example

\[ \varphi = \exists \theta \exists \diamondsuit = \theta (p_1 \land \exists \diamondsuit = \theta p_2) \]

\[
\begin{align*}
(0.5, 0.2) & \models \varphi \\
(0.7, 0.2) & \not\models \varphi
\end{align*}
\]
Model Checking Example

\[
\varphi = \exists \theta \exists \Box = \theta (p_1 \land \exists \Box = \theta p_2)
\]

\[
(0.5, 0.2) \models \varphi
\]

\[
(0.7, 0.2) \not\models \varphi
\]
Model Checking Example

$\varphi = \exists \theta \exists \Box = \theta (p_1 \land \exists \Box = \theta p_2)$

$(\frac{0.5}{0.2}) \models \varphi$

$(\frac{0.7}{0.2}) \not\models \varphi$

Add a variable $r_3$ measuring the yellow time duration.
Future Research

- close complexity gap model checking for reachability fragment (NEXPTIME-hard, EXPSPACE-membership)
- model checking full parametric TCTL
- priced timed automata