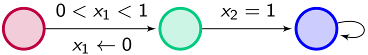
# Revisiting Reachability in Timed Automata

Karin Quaas [1], Mahsa Shirmohammadi [2], James Worrell [2]

[1] Universität Leipzig, [2] University of Oxford

LICS 2017

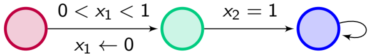
## Timed Automata (TA)




$$\textstyle\binom{0.5}{0.2} \xrightarrow{0.4} \binom{0}{0.6} \xrightarrow{0.4} \binom{0.4}{1.0} \xrightarrow{0.7} \binom{1.1}{1.7} \ldots$$

# Model Checking Timed Automata & Parametric TCTL

## Timed Automata (TA)




$$\binom{0.5}{0.2} \xrightarrow{0.4} \binom{0}{0.6} \xrightarrow{0.4} \binom{0.4}{1.0} \xrightarrow{0.7} \binom{1.1}{1.7} \ldots$$

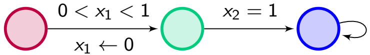
## Parametric TCTL

$$\varphi = \exists\theta\, \exists\Diamond_{=\theta}\,(p_1 \wedge \exists\Diamond_{=\theta}\, p_2)$$

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)




$$\begin{pmatrix} 0.5 \\ 0.2 \end{pmatrix} \xrightarrow{0.4} \begin{pmatrix} 0 \\ 0.6 \end{pmatrix} \xrightarrow{0.4} \begin{pmatrix} 0.4 \\ 1.0 \end{pmatrix} \xrightarrow{0.7} \begin{pmatrix} 1.1 \\ 1.7 \end{pmatrix} \ldots$$

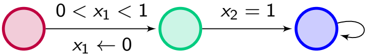
## Parametric TCTL

$$\varphi = \exists\theta\, \exists\Diamond_{=\theta}\, (p_1 \wedge \exists\Diamond_{=\theta}\, p_2)$$

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)




$$\binom{0.5}{0.2} \xrightarrow{0.4} \binom{0}{0.6} \xrightarrow{0.4} \binom{0.4}{1.0} \xrightarrow{0.7} \binom{1.1}{1.7} \ldots$$

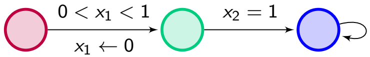
## Parametric TCTL

$$\varphi = \exists\theta\ \exists\Diamond_{=\theta}\,(p_1 \wedge \exists\Diamond_{=\theta}\, p_2)$$

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)




$$\binom{0.5}{0.2} \xrightarrow{0.4} \binom{0}{0.6} \xrightarrow{0.4} \binom{0.4}{1.0} \xrightarrow{0.7} \binom{1.1}{1.7} \dots$$

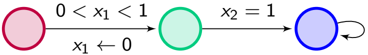
## Parametric TCTL

$$\varphi = \exists\theta\, \exists\Diamond_{=\theta}\, (p_1 \wedge \exists\Diamond_{=\theta}\, p_2)$$

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)




$$\binom{0.5}{0.2} \xrightarrow{0.4} \binom{0}{0.6} \xrightarrow{0.4} \binom{0.4}{1.0} \xrightarrow{0.7} \binom{1.1}{1.7} \ldots$$

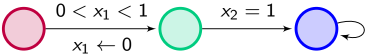
## Parametric TCTL

$$\varphi = \exists\theta\, \exists\Diamond_{=\theta}\, (p_1 \wedge \exists\Diamond_{=\theta} p_2)$$

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)




$$\begin{pmatrix}0.5\\0.2\end{pmatrix} \xrightarrow{0.4} \begin{pmatrix}0\\0.6\end{pmatrix} \xrightarrow{0.4} \begin{pmatrix}0.4\\1.0\end{pmatrix} \xrightarrow{0.7} \begin{pmatrix}1.1\\1.7\end{pmatrix} \dots$$

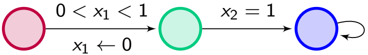
## Parametric TCTL

$$\varphi = \exists\theta\, \exists\Diamond_{=\theta}\, (p_1 \wedge \exists\Diamond_{=\theta}\, p_2)$$

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)




$\binom{0.5}{0.2} \xrightarrow{0.4} \binom{0}{0.6} \xrightarrow{0.4} \binom{0.4}{1.0} \xrightarrow{0.7} \binom{1.1}{1.7} \ldots$

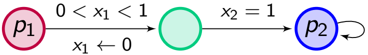
## Parametric TCTL

$$\varphi = \exists\theta\, \exists\Diamond_{=\theta}\, (p_1 \wedge \exists\Diamond_{=\theta}\, p_2)$$

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)




$$\binom{0.5}{0.2} \xrightarrow{0.4} \binom{0}{0.6} \xrightarrow{0.4} \binom{0.4}{1.0} \xrightarrow{0.7} \binom{1.1}{1.7} \ldots$$

## Model Checking

Does $(\ell_0, \binom{0.5}{0.2}) \models \varphi$ hold?

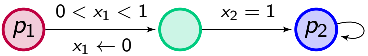
## Parametric TCTL

$$\varphi = \exists\theta\, \exists\Diamond_{=\theta}\, (p_1 \wedge \exists\Diamond_{=\theta}\, p_2)$$

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)




$\binom{0.5}{0.2} \xrightarrow{0.4} \binom{0}{0.6} \xrightarrow{0.4} \binom{0.4}{1.0} \xrightarrow{0.7} \binom{1.1}{1.7} \ldots$

## Parametric TCTL

$$\varphi = \exists\theta\, \exists\Diamond_{=\theta}\, (p_1 \wedge \exists\Diamond_{=\theta}\, p_2)$$

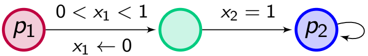
Does there exist some value $\theta$ such that there exists a run of duration $\theta$ to a location in which $p_1$ holds and from which there exists a run of duration $\theta$ to a location in which $p_2$ holds?

## Model Checking

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

# Model Checking Timed Automata & Parametric TCTL

## Timed Automata (TA)




$\binom{0.5}{0.2} \xrightarrow{0.4} \binom{0}{0.6} \xrightarrow{0.4} \binom{0.4}{1.0} \xrightarrow{0.7} \binom{1.1}{1.7} \ldots$

## Parametric TCTL

$$\varphi = \exists\theta\, \exists\Diamond_{=\theta}\, (p_1 \wedge \exists\Diamond_{=\theta}\, p_2)$$

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?

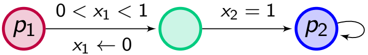
## Model Checking

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

$\binom{0.5}{0.2}$ and $\binom{0.7}{0.2}$ are region-equivalent, but $(\ell_0, \binom{0.7}{0.2}) \not\models \varphi$

# Model Checking Timed Automata & Parametric TCTL

## Timed Automata (TA)




$$\binom{0.5}{0.2} \xrightarrow{0.4} \binom{0}{0.6} \xrightarrow{0.4} \binom{0.4}{1.0} \xrightarrow{0.7} \binom{1.1}{1.7} \dots$$

## Parametric TCTL

$$\varphi = \exists\theta\, \exists\Diamond_{=\theta} (p_1 \wedge \exists\Diamond_{=\theta} p_2)$$

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?

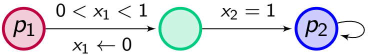
## Model Checking

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

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

# Model Checking Timed Automata & Parametric TCTL

## Timed Automata (TA)




$\begin{pmatrix} 0.5 \\ 0.2 \end{pmatrix} \xrightarrow{0.4} \begin{pmatrix} 0 \\ 0.6 \end{pmatrix} \xrightarrow{0.4} \begin{pmatrix} 0.4 \\ 1.0 \end{pmatrix} \xrightarrow{0.7} \begin{pmatrix} 1.1 \\ 1.7 \end{pmatrix} \dots$

## Parametric TCTL

$$\varphi = \exists\theta\, \exists\Diamond_{=\theta}\, (p_1 \wedge \exists\Diamond_{=\theta}\, p_2)$$

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 $(\ell_0, \begin{pmatrix} 0.7 \\ 0.2 \end{pmatrix}) \not\models \varphi$: $\begin{pmatrix} 0.7 \\ 0.2 \end{pmatrix} \xrightarrow{0.4} \lightning$

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 $\varphi_{\ell,\ell'}(\mathbf{y}, \mathbf{y}')$ such that $(\mathbf{v}, \mathbf{v}') \models \varphi_{\ell,\ell'}$ iff $(\ell, \mathbf{v}) \to^* (\ell', \mathbf{v}')$.

# Goal

### Computing the Reachability Relation

**Instance:** A timed automaton and locations $\ell, \ell'$
**Output:** First-order formula $\varphi_{\ell,\ell'}(\mathbf{y}, \mathbf{y}')$ such that $(\mathbf{v}, \mathbf{v}') \models \varphi_{\ell,\ell'}$ iff $(\ell, \mathbf{v}) \to^* (\ell', \mathbf{v}')$.

### 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 $\varphi_{\ell,\ell'}(\mathbf{y}, \mathbf{y}')$ such that
$(\mathbf{v}, \mathbf{v}') \models \varphi_{\ell,\ell'}$ iff $(\ell, \mathbf{v}) \rightarrow^* (\ell', \mathbf{v}')$.

### Historical Remarks

Alur and Dill, 1990: control-state reachability without clock values
Comon & Jurski CONCUR 1999, Dang TCS 2003, Dima LICS 2002

# Goal

### Computing the Reachability Relation

**Instance:** A timed automaton and locations $\ell, \ell'$
**Output:** First-order formula $\varphi_{\ell,\ell'}(\mathbf{y}, \mathbf{y}')$ such that
$(\mathbf{v}, \mathbf{v}') \models \varphi_{\ell,\ell'}$ iff $(\ell, \mathbf{v}) \to^* (\ell', \mathbf{v}')$.

### Historical Remarks

Alur and Dill, 1990: control-state reachability without clock values
Comon & Jurski CONCUR 1999, Dang TCS 2003, Dima LICS 2002

### Our Logic

Existential fragment of mix of real-arithmetic and Presburger arithmetic

# Notation

Given a timed automaton with $n$ clocks:

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

# Notation

Given a timed automaton with $n$ clocks:

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

# 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 $\mathrm{fr}(\mathbf{v}) = (\mathrm{fr}(v_1), \ldots, \mathrm{fr}(v_n))$

# 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 $\mathrm{fr}(\mathbf{v}) = (\mathrm{fr}(v_1), \ldots, \mathrm{fr}(v_n))$

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

# 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 $\mathrm{fr}(\mathbf{v}) = (\mathrm{fr}(v_1), \ldots, \mathrm{fr}(v_n))$

  Note: $\mathbf{v} = \lfloor \mathbf{v} \rfloor + \mathrm{fr}(\mathbf{v})$.
- Integer-valued variables $\mathbf{z} = (z_1, \ldots, z_n)$

# 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 $\mathrm{fr}(\mathbf{v}) = (\mathrm{fr}(v_1), \ldots, \mathrm{fr}(v_n))$

  Note: $\mathbf{v} = \lfloor \mathbf{v} \rfloor + \mathrm{fr}(\mathbf{v})$.
- 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 $\mathbf{v}$
**Output:** $\varphi_{\ell,\ell',\mathbf{v}}(\mathbf{z}, \mathbf{r}, \mathbf{z}', \mathbf{r}')$ such that
$(\lfloor \mathbf{v} \rfloor, \mathrm{fr}(\mathbf{v}), \lfloor \mathbf{v}' \rfloor, \mathrm{fr}(\mathbf{v}')) \models \varphi_{\ell,\ell',\mathbf{v}}$ iff $(\ell, \mathbf{v}) \to^* (\ell', \mathbf{v}')$

# Solution: Step I

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

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

# Solution: Step I

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

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

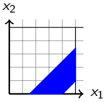
- sound and complete with respect to reachability

# Solution: Step I

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

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

- 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



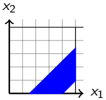

$$\begin{array}{cc} & \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array} & \begin{bmatrix} (\leq, 0) & (<, -0.3) & (\leq, 0) \\ (\leq, 1) & (\leq, 0) & (\leq, 0.8) \\ (<, 0.7) & (<, -0.3) & (\leq, 0) \end{bmatrix} \end{array}$$

# Solution: Step I

**Instance:** Locations $\ell, \ell'$, concrete valuation $\mathbf{v}$
**Output:** $\varphi_{\ell,\ell',\mathbf{v}}(\mathbf{z}, \mathbf{r}, \mathbf{z}', \mathbf{r}')$ such that
$(\lfloor \mathbf{v} \rfloor, \mathrm{fr}(\mathbf{v}), \lfloor \mathbf{v}' \rfloor, \mathrm{fr}(\mathbf{v}')) \models \varphi_{\ell,\ell',\mathbf{v}}$ iff $(\ell, \mathbf{v}) \to^* (\ell', \mathbf{v}')$

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

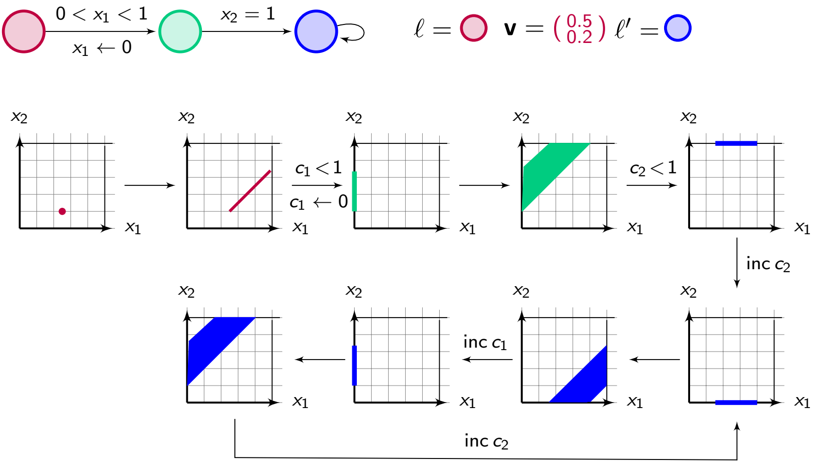
- sound and complete with respect to reachability
- control states $q = (\ell, D)$, where $D$ is a DBM representing a *zone* over $[0,1]^n$ representing the fractional parts of the clock values




$$\begin{array}{c c} & \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array} & \begin{bmatrix} (\leq, 0) & (<, -0.3) & (\leq, 0) \\ (\leq, 1) & (\leq, 0) & (\leq, 0.8) \\ (<, 0.7) & (<, -0.3) & (\leq, 0) \end{bmatrix} \end{array}$$

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

# Solution: Step I, Example



# Solution: Step I

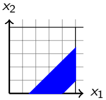
**Instance:** Locations $\ell, \ell'$, concrete valuation $\mathbf{v}$

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

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

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

- 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




$$\begin{array}{cc} & \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array} & \begin{bmatrix} (\leq, 0) & (<, -0.3) & (\leq, 0) \\ (\leq, 1) & (\leq, 0) & (\leq, 0.8) \\ (<, 0.7) & (<, -0.3) & (\leq, 0) \end{bmatrix} \end{array}$$

$$(0.3 < r'_1 \leq 1) \wedge (0 \leq r'_2 < 0.7) \wedge (0.3 < r_1 - r_2 \leq 0.8)$$

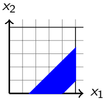
- counter $c_i$ stores the integer part of clock $x_i$

# Solution: Step I

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

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

- 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



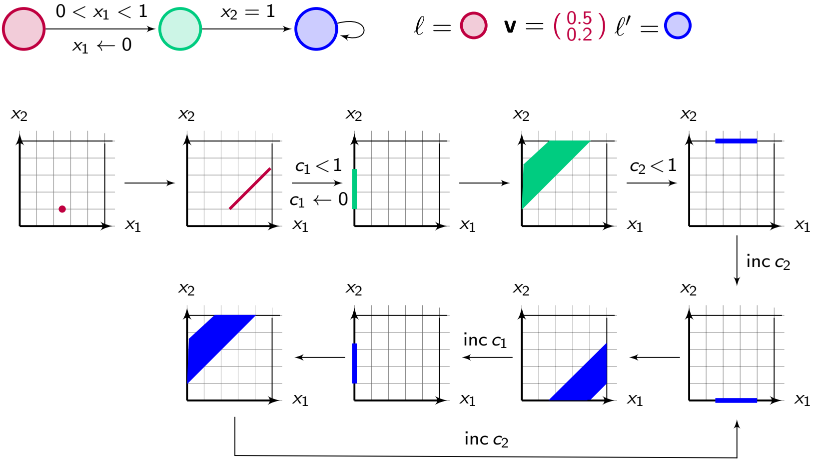

$$\begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \end{array} \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \left[ \begin{array}{ccc} (\leq, 0) & (<, -0.3) & (\leq, 0) \\ (\leq, 1) & (\leq, 0) & (\leq, 0.8) \\ (<, 0.7) & (<, -0.3) & (\leq, 0) \end{array} \right] \end{array}$$

$$(0.3 < r'_1 \leq 1) \wedge (0 \leq r'_2 < 0.7) \wedge (0.3 < r_1 - r_2 \leq 0.8)$$

- 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 I, Example



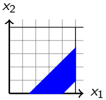

$$\varphi_{\ell,\ell',\mathbf{v}} = \big((z_2' - z_1' = 1) \wedge (0.3 < r_1' \leq 1) \wedge (0 \leq r_2' < 0.7) \wedge (0.3 < r_1' - r_2' \leq 0.8)\big)$$
$$\vee \big((z_2' - z_1' = 0) \wedge (0 \leq r_1' \leq 0.8) \wedge (0.3 < r_2' \leq 1) \wedge (0.3 < 1 + r_1' - r_2' \leq 0.8)\big)$$

# Solution: Step I

**Instance:** Locations $\ell, \ell'$, concrete valuation $\mathbf{v}$
**Output:** $\varphi_{\ell,\ell',\mathbf{v}}(\mathbf{z}, \mathbf{r}, \mathbf{z}', \mathbf{r}')$ such that
$(\lfloor \mathbf{v} \rfloor, \mathrm{fr}(\mathbf{v}), \lfloor \mathbf{v}' \rfloor, \mathrm{fr}(\mathbf{v}')) \models \varphi_{\ell,\ell',\mathbf{v}}$ iff $(\ell, \mathbf{v}) \to^* (\ell', \mathbf{v}')$

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

- 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




$$\begin{array}{cc} & \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array} & \left[\begin{array}{ccc} (\leq, 0) & (<, -0.3) & (\leq, 0) \\ (\leq, 1) & (\leq, 0) & (\leq, 0.8) \\ (<, 0.7) & (<, -0.3) & (\leq, 0) \end{array}\right] \end{array}$$

$$(0.3 < r_1' \leq 1) \wedge (0 \leq r_2' < 0.7) \wedge (0.3 < r_1 - r_2 \leq 0.8)$$

- 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 $\mathbf{v}$
**Output:** $\varphi_{\ell,\ell',\mathsf{type}(\mathbf{u})}(\mathbf{z}, \mathbf{r}, \mathbf{z}', \mathbf{r}')$ such that

$(\lfloor \mathbf{u} \rfloor, \mathsf{fr}(\mathbf{u}), \lfloor \mathbf{v}' \rfloor, \mathsf{fr}(\mathbf{v}')) \models \varphi_{\ell,\ell',\mathsf{type}(\mathbf{v})}$ iff $(\ell, \mathbf{u}) \rightarrow^* (\ell', \mathbf{v}')$

for all $\mathbf{u}$ with $\mathsf{type}(\mathbf{u}) = \mathsf{type}(\mathbf{v})$.

# Solution: Step II

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

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

$(\lfloor \mathbf{u} \rfloor, \mathsf{fr}(\mathbf{u}), \lfloor \mathbf{v}' \rfloor, \mathsf{fr}(\mathbf{v}')) \models \varphi_{\ell,\ell',\mathsf{type}(\mathbf{v})}$ iff $(\ell, \mathbf{u}) \to^* (\ell', \mathbf{v}')$

for all $\mathbf{u}$ with $\mathsf{type}(\mathbf{u}) = \mathsf{type}(\mathbf{v})$.

### Difference Types

$\mathsf{type}(\mathbf{v})$ is collection of formulas of the form

$$c + r_i - r_j \;\leq\; c' + r_{i'} - r_{j'}$$

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

# Solution: Step II

**Instance:** Locations $\ell, \ell'$, concrete valuation $\mathbf{v}$
**Output:** $\varphi_{\ell,\ell',\mathsf{type}(\mathbf{u})}(\mathbf{z}, \mathbf{r}, \mathbf{z}', \mathbf{r}')$ such that

$(\lfloor \mathbf{u} \rfloor, \mathsf{fr}(\mathbf{u}), \lfloor \mathbf{v}' \rfloor, \mathsf{fr}(\mathbf{v}')) \models \varphi_{\ell,\ell',\mathsf{type}(\mathbf{v})}$ iff $(\ell, \mathbf{u}) \rightarrow^* (\ell', \mathbf{v}')$

for all $\mathbf{u}$ with $\mathsf{type}(\mathbf{u}) = \mathsf{type}(\mathbf{v})$.

### Difference Types

$\mathsf{type}(\mathbf{v})$ is collection of formulas of the form

$$c + r_i - r_j \ \leq \ c' + r_{i'} - r_{j'}$$

that are satisfied by $\mathbf{v}$, 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 $\mathsf{type}\left(\begin{smallmatrix} 0.5 \\ 0.2 \end{smallmatrix}\right)$, but it is not in $\mathsf{type}\left(\begin{smallmatrix} 0.7 \\ 0.2 \end{smallmatrix}\right)$

# Solution: Step II

**Instance:** Locations $\ell, \ell'$, concrete valuation $\mathbf{v}$
**Output:** $\varphi_{\ell,\ell',\mathrm{type}(\mathbf{u})}(\mathbf{z}, \mathbf{r}, \mathbf{z}', \mathbf{r}')$ such that

$(\lfloor \mathbf{u} \rfloor, \mathrm{fr}(\mathbf{u}), \lfloor \mathbf{v}' \rfloor, \mathrm{fr}(\mathbf{v}')) \models \varphi_{\ell,\ell',\mathrm{type}(\mathbf{v})}$ iff $(\ell, \mathbf{u}) \rightarrow^* (\ell', \mathbf{v}')$

for all $\mathbf{u}$ with $\mathrm{type}(\mathbf{u}) = \mathrm{type}(\mathbf{v})$.

## Difference Types

$\mathrm{type}(\mathbf{v})$ is collection of formulas of the form

$$c + r_i - r_j \;\leq\; c' + r_{i'} - r_{j'}$$

that are satisfied by $\mathbf{v}$, 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 $\mathrm{type}\binom{0.5}{0.2}$, but it is not in $\mathrm{type}\binom{0.7}{0.2}$

Note: there are only finitely many distinct different types.

# Solution: Step II ctd.

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

- 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)

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

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

# Solution: Step II ctd.

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

- 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)

$$\begin{array}{c|ccc} & x_0 & x_1 & x_2 \\ \hline x_0 & (\leq,[0]) & (\leq,[0-r_1]) & (\leq,[0]) \\ x_1 & (\leq,[1]) & (\leq,[0]) & (\leq,[1]) \\ x_2 & (<,[1-r_1]) & (<,[0-r_1]) & (\leq,[0]) \end{array}$$

$$(r_1 < r_1' \leq 1) \wedge (0 \leq r_2' \leq 1 - r_1) \wedge (-r_1 < r_1 - r_2 \leq 1)$$

- 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:** $\varphi_{\ell,\ell'}(\mathbf{z}, \mathbf{r}, \mathbf{z}', \mathbf{r}')$ such that
$(\lfloor \mathbf{v} \rfloor, \mathrm{fr}(\mathbf{v}), \lfloor \mathbf{v}' \rfloor, \mathrm{fr}(\mathbf{v}')) \models \varphi_{\ell,\ell'}$ iff $(\ell, \mathbf{v}) \to^* (\ell', \mathbf{v}')$

# Solution: Final Step

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

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:** $\varphi_{\ell,\ell'}(\mathbf{z}, \mathbf{r}, \mathbf{z}', \mathbf{r}')$ such that
$(\lfloor \mathbf{v} \rfloor, \mathrm{fr}(\mathbf{v}), \lfloor \mathbf{v}' \rfloor, \mathrm{fr}(\mathbf{v}')) \models \varphi_{\ell,\ell'}$ iff $(\ell, \mathbf{v}) \to^* (\ell', \mathbf{v}')$

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

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

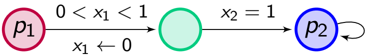
- construct for each of the finitely many difference types $T$ the corresponding formula $\varphi_{\ell,\ell',T}$, and
- define $\varphi_{\ell,\ell'}$ as the disjunction of all these formulas.

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

# Back to Model Checking

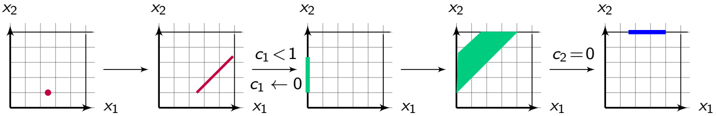
- reachability fragment of parametric TCTL (without until modality)
- we reduce model checking to deciding the truth of formula in a fragment of first-order logic
- the formula contains reachability formulas
- yields a EXPSPACE-decision procedure for model checking

# Model Checking Example

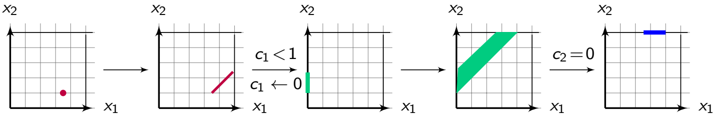



$$\varphi = \exists\theta\, \exists\Diamond_{=\theta} \left(p_1 \wedge \exists\Diamond_{=\theta} p_2\right)$$

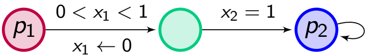
$\binom{0.5}{0.2} \models \varphi$




$\binom{0.7}{0.2} \not\models \varphi$

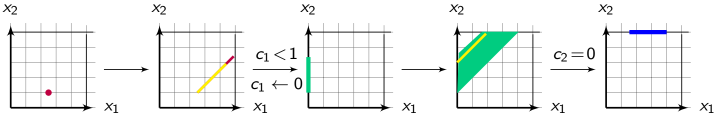


# Model Checking Example

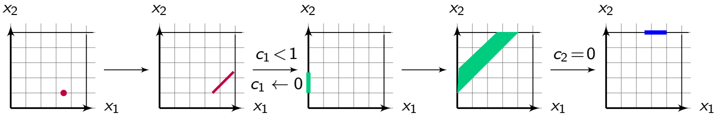



$$\varphi = \exists\theta\, \exists\Diamond_{=\theta} \left(p_1 \wedge \exists\Diamond_{=\theta} p_2\right)$$

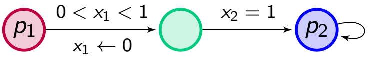
$\binom{0.5}{0.2} \models \varphi$




$\binom{0.7}{0.2} \not\models \varphi$

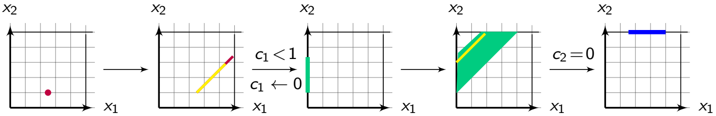


# Model Checking Example

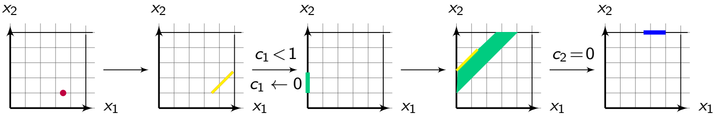



$$\varphi = \exists\theta\, \exists\Diamond_{=\theta}\, (p_1 \wedge \exists\Diamond_{=\theta}\, p_2)$$

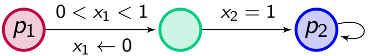
$\binom{0.5}{0.2} \models \varphi$



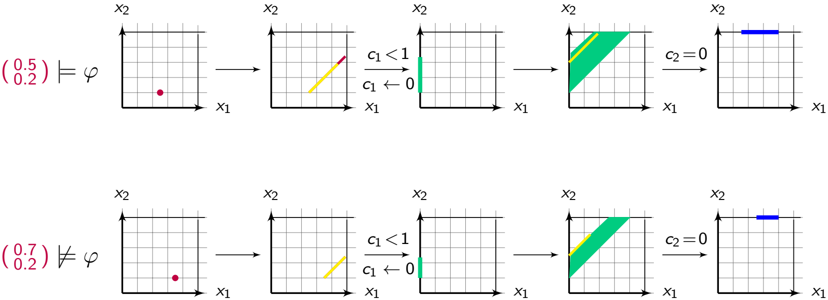

$\binom{0.7}{0.2} \not\models \varphi$



# Model Checking Example




$$\varphi = \exists\theta\, \exists\Diamond_{=\theta}\, (p_1 \wedge \exists\Diamond_{=\theta}\, p_2)$$




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