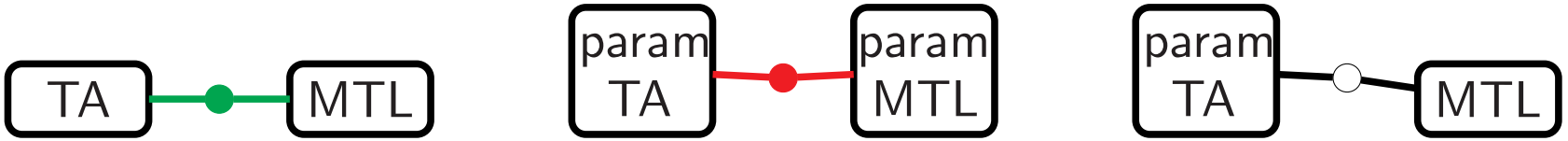
MTL-Model Checking of One-Clock Parametric Timed Automata is Undecidable

SynCop 2014
1st International Workshop on Synthesis of Continuous Parameters

Karin Quaas
University of Leipzig

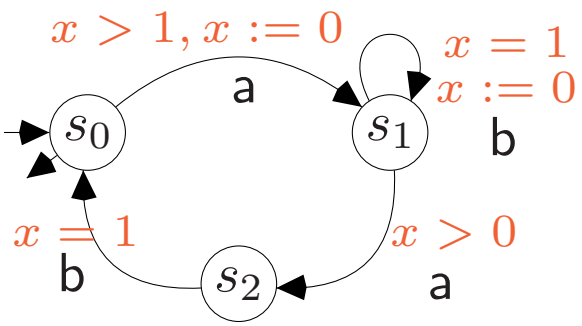
6th April 2014

Outline of the Talk



Timed Automata [AD90]

- Finite automata extended with a finite set of clocks



a clock

- ranges over $\mathbb{R}_{\geq 0}$
- grows monotonically while time elapses in a state
- can be compared with constants in $\mathbb{N}$ at the edges
- can be reset to zero at the edges

Runs

$$(s_0, 0) \xrightarrow{3,a} (s_1, 0) \xrightarrow{0.1,a} (s_2, 0.1) \xrightarrow{0.9,b}$$

Timed words

$$(a, 3)(a, 3.1)(b, 4)$$

[AD90] Alur, Dill: A Theory of Timed Automata, 1990.

Metric Temporal Logic (MTL)

Σ... a finite alphabet.

$$\varphi ::= a \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \mathsf{U}_I \varphi$$

$p \in P$, $I \subseteq \mathbb{R}_{\geq 0}$ is an interval with endpoints in $\mathbb{N} \cup \{\infty\}$.

- Evaluated over timed words $w = (\sigma_1, t_1)(\sigma_2, t_2)\dots(\sigma_n, t_n)$, $i \in \{1, \dots, n\}$

$$(w, i) \models \varphi_1 \mathsf{U}_I \varphi_2 \quad \Leftrightarrow \quad \exists j > i.(w, j) \models \varphi_2, t_j - t_i \in I, \forall i < k < j.(w, k) \models \varphi_1$$

Example:

$\Sigma = \{a, b\}$, $\varphi = a\mathsf{U}_{[0,1]} b$, $w = (a, 3)(a, 3.1)(b, 4)$, $(w, 1) \models \varphi$

The Model Checking Problem

The MTL-Model Checking Problem:

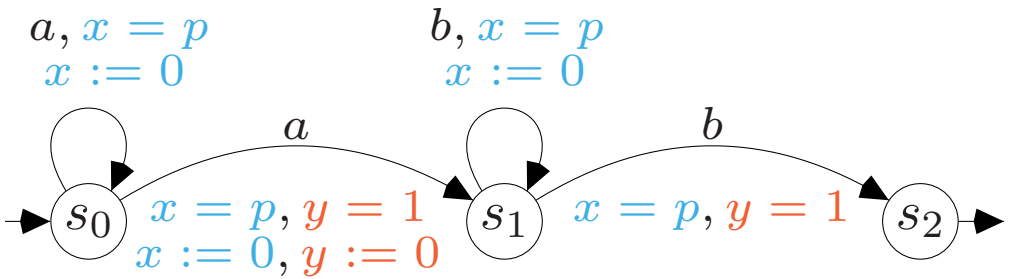
Instance: Timed automaton $\mathcal{A}$, MTL formula φ

Question: Does $w \models \varphi$ hold for all timed words accepted by $\mathcal{A}$?

- MTL-model checking for timed automata is decidable with non-primitive recursive complexity [OW05]

[OW05] Ouaknine, Worrell: On the decidability of Metric Temporal Logic, 2005.

Parametric Timed Automata [AHV93]



a parametric clock
- is a special clock
- can be compared with parameters
- a parameter valuation π determines the behaviour of the automaton

π-Runs

$$\pi(p) = \tfrac{1}{3}, \quad (s_0, 0, 0) \xrightarrow{\frac{1}{3}, a} (s_0, 0, \tfrac{1}{3}) \xrightarrow{\frac{1}{3}, a} (s_0, 0, \tfrac{2}{3}) \xrightarrow{\frac{1}{3}, a} (s_1, 0, 0) \xrightarrow{\frac{1}{3}, b} \dots$$

[AHV93] Alur, Henzinger, Vardi: Parametric real-time reasoning, 1993.

A problem that's been open for a long time...

The Emptiness Problem:

Instance: Parametric timed automaton $\mathcal{A}$.

Question: Is there some parameter valuation such that the set of timed words accepted by $\mathcal{A}$ is non-empty?

- The emptiness problem is undecidable if $\mathcal{A}$ uses more than two parametric clocks. The emptiness problem is decidable if $\mathcal{A}$ uses at most one parametric clock. [AHV93]
- So what about two parametric clocks?
- The emptiness problem is decidable if $\mathcal{A}$ uses at most two parametric clocks and at most one parameter. [BO14]

[AHV93] Alur, Henzinger, Vardi: Parametric real-time reasoning, 1993.

[BO14] Bundala, Ouaknine: Advances in Parametric Real-Time Reasoning, 2014

Extending MTL with Parameters...

- ...is not a good idea:

- Extend LTL with parametric Until modalities of the form $\varphi_1 \mathsf{U}_{=p} \varphi_2$
- Let $w = \sigma_1 \sigma_2 \dots \sigma_k$, $i \in \{1, \dots, k\}$, π a parameter valuation

$$(w, i, \pi) \models \varphi_1 \mathsf{U}_{=p} \varphi_2 \quad \Leftrightarrow \quad (w, i+\pi(p)) \models \varphi_2, \forall i < k < i+\pi(p).(w, k) \models \varphi_1$$

- Model checking finite automata with this logic is undecidable [AETP01]

[AETP01] Alur, Etessami, La Torre, Peled: Parametric Temporal Logic for "Model Measuring", 2001

The Model Checking Problem

The MTL-Model Checking Problem:

Instance: Parametric Timed automaton $\mathcal{A}$, MTL formula φ

Question: Is there some parameter valuation π such that $w \models \varphi$ holds for all timed words accepted by $\mathcal{A}$ under π?

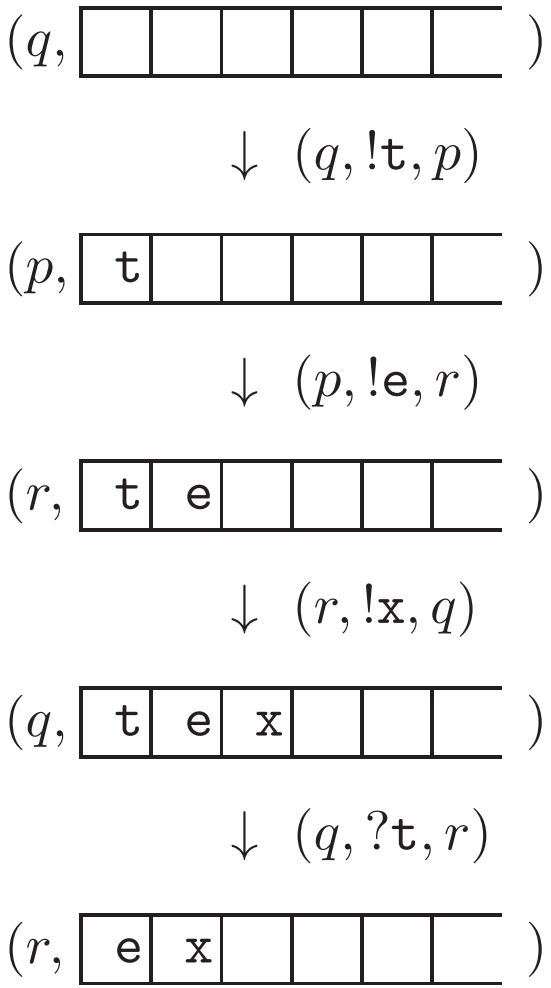
Main Theorem

The MTL-model checking problem for parametric timed automata is undecidable, even if $\mathcal{A}$ uses at most one parametric clock, one parameter, and $\mathcal{A}$ is deterministic.

Proof

- Reduction of the (undecidable) reachability problem for **channel machines**

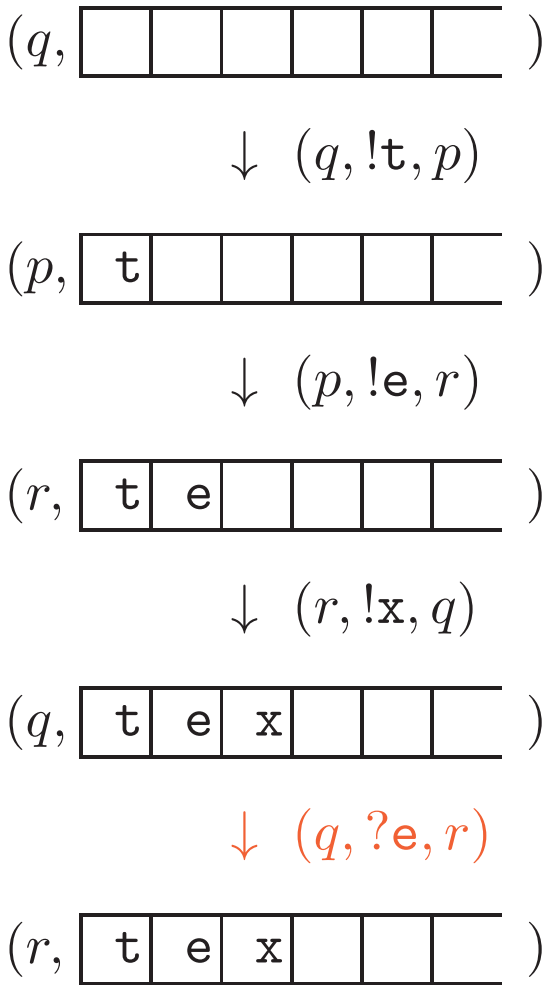
Channel Machines



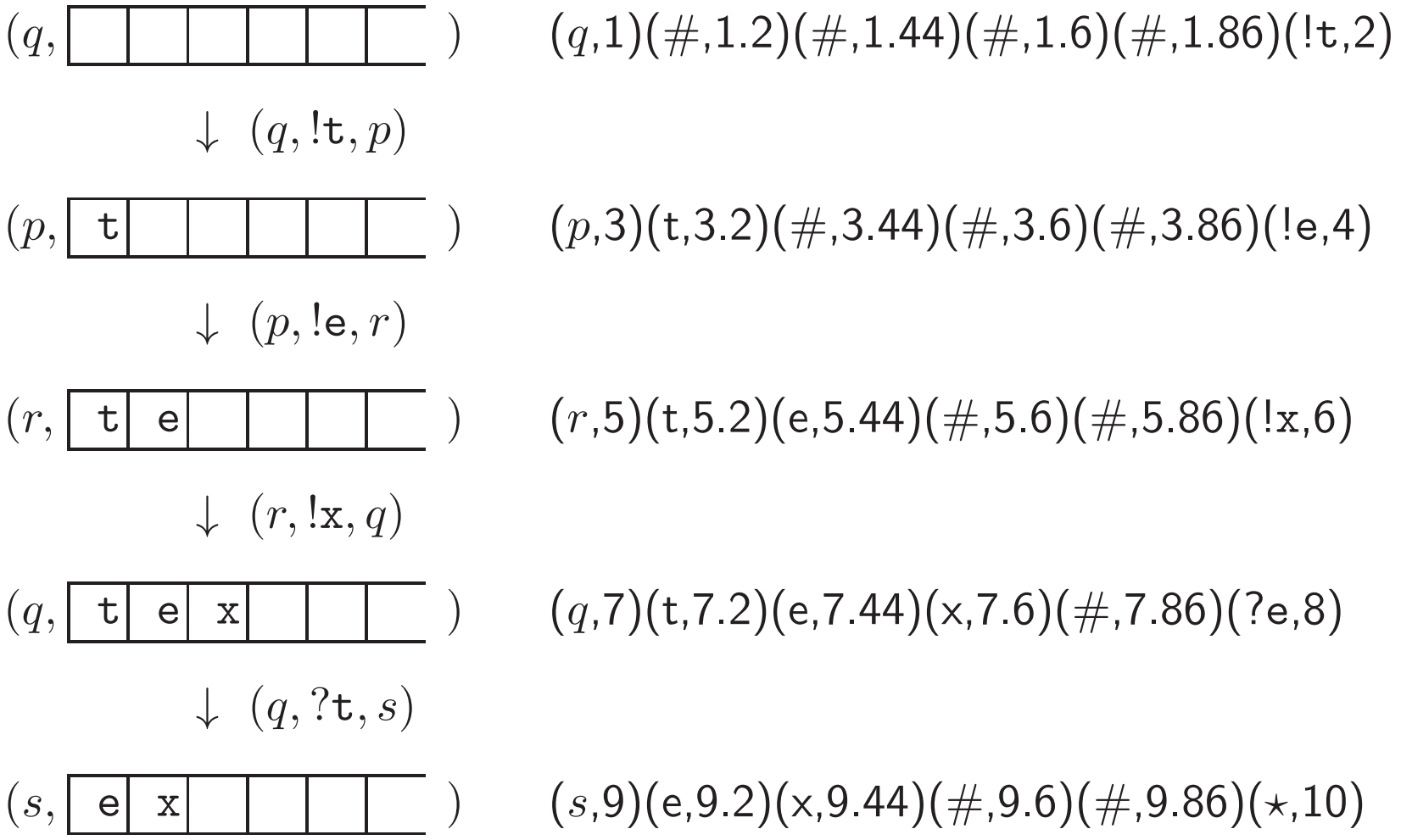
Proof

- Reduction of the (undecidable) reachability problem for **channel machines**
- Given a channel machine $\mathcal{C}$ and a state q, we define a timed language $L(\mathcal{C}, q)$ that encodes computations of $\mathcal{C}$ with insertion errors

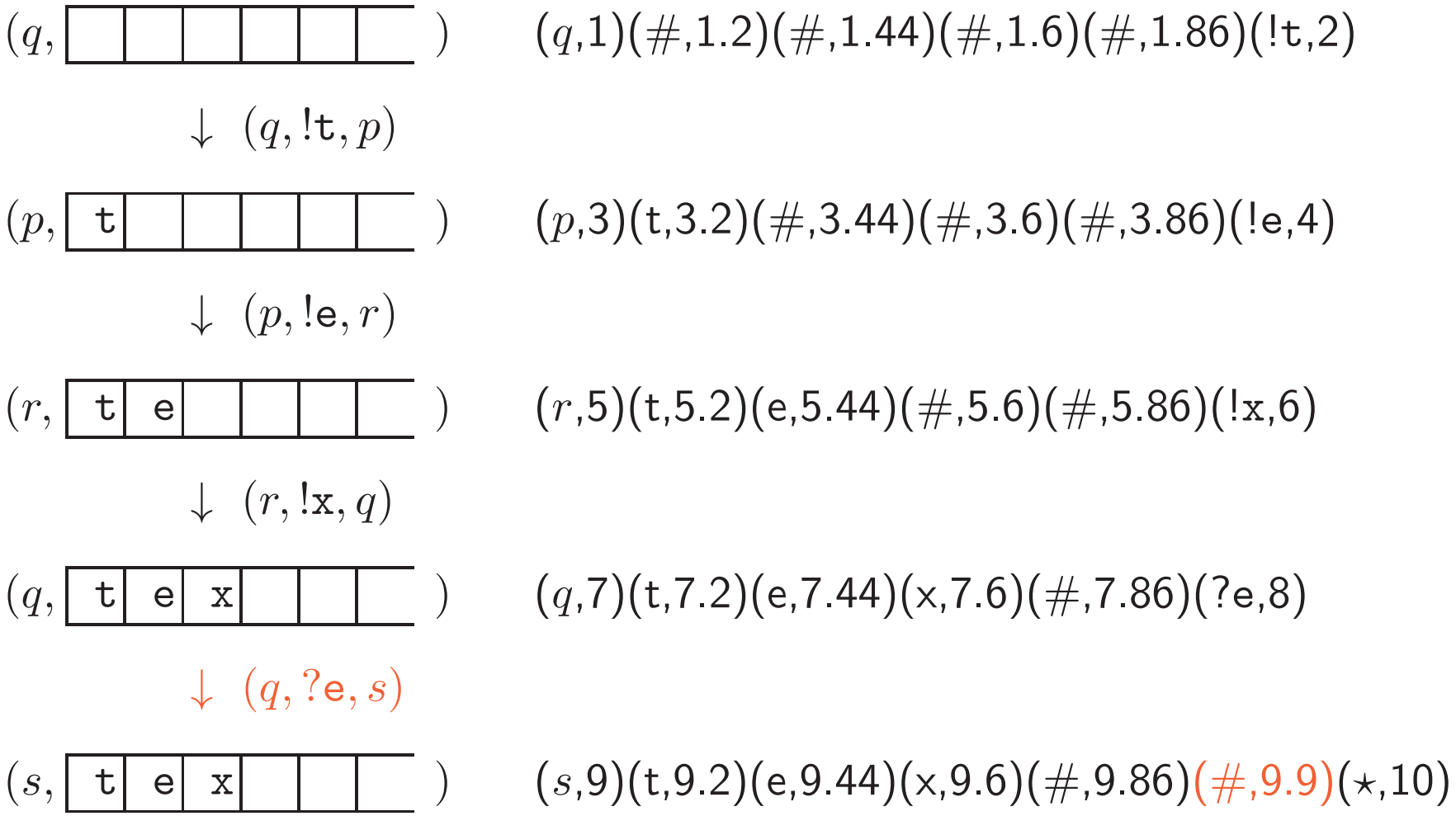
Channel Machines with Insertion Errors



Channel Machines with Insertion Errors: Encoding



Channel Machines with Insertion Errors: Encoding

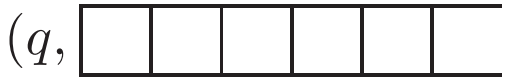


Proof

- Reduction of the (undecidable) reachability problem for **channel machines**
- Given a channel machine $\mathcal{C}$ and a state q, we define a timed language $L(\mathcal{C}, q)$ that encodes computations of $\mathcal{C}$ with insertion errors
- One can define MTL-formula φ such that $L(\varphi) = L(\mathcal{C}, q)$ [OW05]

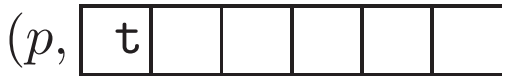
[OW05] Ouaknine, Worrell: On the decidability of Metric Temporal Logic, 2005.

MTL Formula defining $L(\mathcal{C}, q)$



$(q,1)(\#,1.2)(\#,1.44)(\#,1.6)(\#,1.86)(!\mathtt{t},2)$

$\downarrow\ (q, !\mathtt{t}, p)$



$(p,3)(\mathsf{t},3.2)(\#,3.44)(\#,3.6)(\#,3.86)(!\mathsf{e},4)$

$$\varphi_{\text{trans}} := \mathsf{G}\left\langle \bigwedge_{s \in S} s \rightarrow \left(\bigvee_{(s,op,s') \in \Delta} (\mathsf{F}_{[1,1]} op \wedge \mathsf{F}_{[2,2]} s') \right) \right\rangle$$

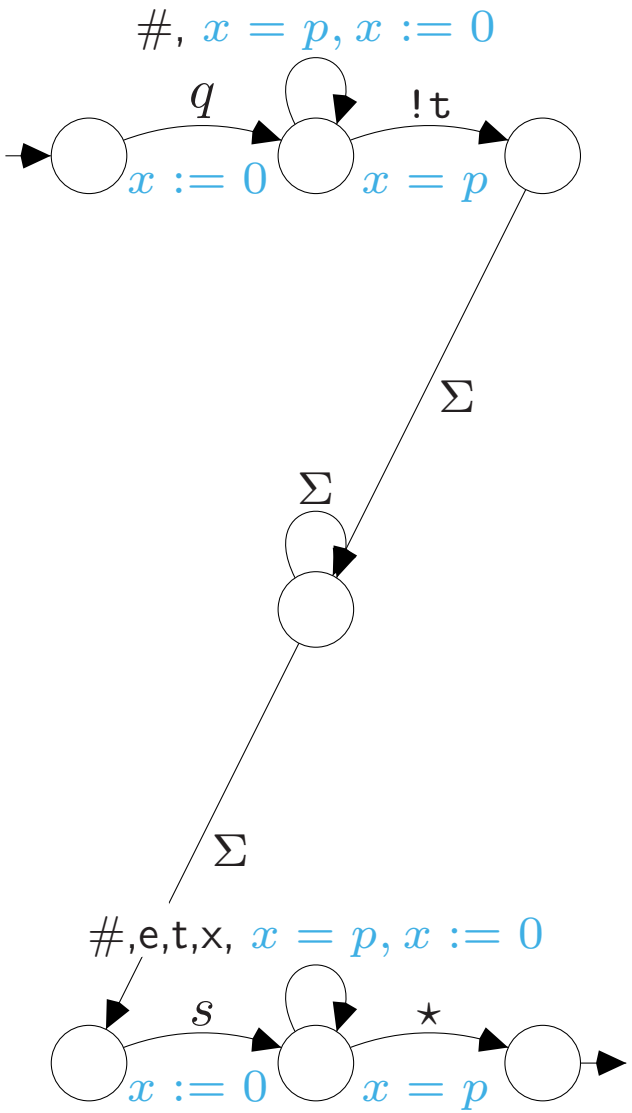
Proof

- Reduction of the (undecidable) reachability problem for **channel machines**
- Given a channel machine $\mathcal{C}$ and a state q, we define a timed language $L(\mathcal{C}, q)$ that encodes computations of $\mathcal{C}$ with insertion errors
- One can define MTL-formula φ such that $L(\varphi) = L(\mathcal{C}, q)$ [OW05]
- We construct a parametric timed automaton $\mathcal{A}$ to exclude insertion errors:

$$L(\varphi) \cap L(\mathcal{A}) = L(\mathcal{C}, q) \backslash \{\text{encoding of computation with insertion errors}\}$$

[OW05] Ouaknine, Worrell: On the decidability of Metric Temporal Logic, 2005.

Parametric Timed Automaton to Exclude Insertion Errors



$(q,1)(\#,1.2)(\#,1.44)(\#,1.6)(\#,1.86)(!t,2)$

$(p,3)(t,3.2)(\#,3.44)(\#,3.6)(\#,3.86)(!e,4)$

$(r,5)(t,5.2)(e,5.44)(\#,5.6)(\#,5.86)(!x,6)$

$(q,7)(t,7.2)(e,7.44)(x,7.6)(\#,7.86)(?e,8)$

$(s,9)(t,9.2)(e,9.44)(x,9.6)(\#,9.86)(\#,9.9)(\star,10)$

Proof

- Reduction of the (undecidable) reachability problem for **channel machines**
- Given a channel machine $\mathcal{C}$ and a state q, we define a timed language $L(\mathcal{C}, q)$ that encodes computations of $\mathcal{C}$ with insertion errors
- One can define MTL-formula φ such that $L(\varphi) = L(\mathcal{C}, q)$ [OW05]
- We construct a parametric timed automaton $\mathcal{A}$ to exclude insertion errors:

$$L(\varphi) \cap L(\mathcal{A}) = L(\mathcal{C}, q) \backslash \{\text{encoding of computation with insertion errors}\}$$

- We obtain

$$\mathcal{C} \text{ does not reach } q \Leftrightarrow L(\mathcal{A}) \cap L(\varphi) = \emptyset \Leftrightarrow L(\mathcal{A}) \subseteq L(\neg\varphi)$$

[OW05] Ouaknine, Worrell: On the decidability of Metric Temporal Logic, 2005.

The Model Checking Problem

The MTL-Model Checking Problem:

Instance: Parametric Timed automaton $\mathcal{A}$, MTL formula φ

Question: Is there some parameter valuation π such that $w \models \varphi$ holds for all timed words accepted by $\mathcal{A}$ under π?

Main Theorem

The MTL-model checking problem for parametric timed automata is undecidable, even if $\mathcal{A}$ uses at most one parametric clock, one parameter, and $\mathcal{A}$ is deterministic.

Open Problems

- In our proof, we use parametric *equality* constraints
- What about MTL-model checking for parametric timed automata with syntactic restriction on parametric constraints, eg., LU-automata [HRSV01]?

- The proof works only for parameter valuations mapping parameters to non-negative rationals
- What about parameter valuations mapping to non-negative integers?

[HRSV01] Hune, Romijn, Stoelinga, Vaandrager: Linear Parametric Model Checking of Timed Automata, 2001

Thank you for your attention!