Parameter Synthesis for Timed Automata with Clock-Aware LTL Properties

Nikola Beneš

joint work with
Peter Bezděk, Ivana Černá, Vojtěch Havel, Jiří Barnat

Masaryk University
Brno, Czech Republic

April 7, 2019
Based on


Parametric Timed Automata

Parameter Synthesis Problem: given a PTA $A$ and a specification $\phi$, compute the set of all parameter valuations $v$ such that $A^v$ satisfies $\phi$.
Parameter Synthesis Problem
given a PTA $A$ and a specification $\phi$
compute the set of all parameter valuations $v$ such that $A_v$ satisfies $\phi$. 

$\begin{align*}
\text{start} & \rightarrow \text{green } x \leq 5 \\
& \xrightarrow{x \geq 5, x \leftarrow 0} \text{yellow } x \leq 1 \\
& \xrightarrow{x \geq 1, x \leftarrow 0} \text{red} \\
& \xleftarrow{x \geq 6, x \leftarrow 0}
\end{align*}$
Parametric Timed Automata

Parameter Synthesis Problem
given a PTA $A$ and a specification $\varphi$ compute the set of all parameter valuations $v$ such that $A_v$ satisfies $\varphi$.
Parametric Timed Automata

Initial value:
\[
\begin{align*}
x &\geq 6, x \leftarrow 0 \\
x &\geq 5, x \leftarrow 0 \\
x &\geq 1, x \leftarrow 0
\end{align*}
\]

Parameter Synthesis Problem: Given a PTA \( A \) and a specification \( \varphi \), compute the set of all parameter valuations \( v \) such that \( A^v \) satisfies \( \varphi \).
Parametric Timed Automata

Parameter Synthesis Problem:
Given a PTA $A$ and a specification $\phi$, compute the set of all parameter valuations $v$ such that $A^v$ satisfies $\phi$.

\[ \begin{align*}
  &x \leq 5, x \leftarrow 0 \\
  &x \geq 6, x \leftarrow 0
\end{align*} \]
Parametric Timed Automata

Parameter Synthesis Problem
given a PTA $A$ and a specification $\phi$ compute the set of all parameter valuations $v$ such that $A_v$ satisfies $\phi$.
Parametric Timed Automata

Parameter Synthesis Problem
given a PTA $A$ and a specification $\phi$ compute the set of all parameter valuations $v$ such that $A_v$ satisfies $\phi$. 

SynCoP 2019, Prague, 7. 4. 2019
Parametric Timed Automata

Parameter Synthesis Problem

Given a PTA $A$ and a specification $\phi$, compute the set of all parameter valuations $v$ such that $A^v$ satisfies $\phi$.
**Parametric Timed Automata**

Parameter Synthesis Problem

- given a PTA $A$ and a specification $\varphi$
- compute the set of all parameter valuations $v$ such that $A_v$ satisfies $\varphi$
<table>
<thead>
<tr>
<th></th>
<th>discrete time integer parameters</th>
<th>continuous time integer parameters</th>
<th>continuous time real parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>L/U-automata</td>
<td>decidable</td>
<td>decidable</td>
<td>decidable</td>
</tr>
<tr>
<td>1c-PTA</td>
<td>decidable</td>
<td>decidable</td>
<td>decidable</td>
</tr>
<tr>
<td>1pc-PTA</td>
<td>decidable</td>
<td>decidable</td>
<td>decidable</td>
</tr>
<tr>
<td>2c-PTA</td>
<td>open</td>
<td>open</td>
<td>open</td>
</tr>
<tr>
<td>1p-PTA (3c)</td>
<td><strong>undecidable</strong></td>
<td><strong>undecidable</strong></td>
<td>undecidable</td>
</tr>
<tr>
<td>PTA (3c)</td>
<td>undecidable</td>
<td>undecidable</td>
<td>undecidable</td>
</tr>
</tbody>
</table>

Our Focus

Bounded Integer Parameter Synthesis Problem

- given a PTA $A$ and a specification $\varphi$
- given integer bounds for each parameter
- compute the set of all integer parameter valuations $v$ within the given bounds such that $A_v$ satisfies $\varphi$
Our Focus

**Bounded Integer Parameter Synthesis Problem**

- given a PTA $A$ and a specification $\varphi$
- given integer bounds for each parameter
- compute the set of all integer parameter valuations $\nu$ within the given bounds such that $A_\nu$ satisfies $\varphi$

**Solutions**

- explicit (on parameters)
  - enumeration of all (finitely many) admissible parameter valuations
- symbolic (on parameters)
Linear Temporal Logic

**LTL**
- evaluated over runs
- atomic propositions (labels of locations)
- Boolean operators
- temporal operators
  - Future
  - Globally
  - Until
  - F G
  - G F

**Automata-Based Model Checking**
- Büchi automaton for the (negation of) the formula
- combine with the model of a system (timed automaton)
- check emptiness of the product (timed Büchi automaton)
Clock-Aware Linear Temporal Logic

CA-LTL

- evaluated over runs
- atomic propositions (labels of locations)
  - simple comparisons over clocks
- Boolean operators
- temporal operators

Examples:

- $\text{FG } x < 10$
- $x < 5 \text{ U } ready$

Model Checking – ???
Clock-Aware Linear Temporal Logic

CA-LTL

- evaluated over runs
- atomic propositions (labels of locations)
  + simple comparisons over clocks
- Boolean operators
- temporal operators

Examples:

- \( \text{FG} x < 10 \)
- \( x < 5 \text{ U ready} \)

Model Checking – ???

- \( x < 42 \text{ U x} \geq 42 \) cannot be represented as a TBA
  - TA transitions are instantaneous
LTL Parameter Synthesis

Parametric timed automaton $A$ (system under investigation)

LTL formula $\varphi$ (specification)

Büchi automaton $A_{\neg \varphi}$

Parametric timed Büchi automaton $A \otimes A_{\neg \varphi}$

Büchi automaton $B$ with monotonic annotation $f$ (finite abstraction of $A \otimes A_{\neg \varphi}$)

Cumulative NDFS
LTL Parameter Synthesis

Parametric timed automaton $A$ (system under investigation)

LTL formula $\varphi$ (specification)

Büchi automaton $A_{\neg \varphi}$

Parametric timed Büchi automaton $A \otimes A_{\neg \varphi}$

Büchi automaton $B$ with monotonic annotation $f$ (finite abstraction of $A \otimes A_{\neg \varphi}$)

Cumulative NDFS
Symbolic State Space of Timed Automata

**Zone**
- convex set of clock valuations given by conjunction of guards
- represents all possible clock valuations in one particular state

**Data structure**
- difference bound matrix (DBM)
- efficient operations, canonical form

\[
\begin{array}{c|ccc}
& 0 & x & y \\
\hline
0 & - & (\leq, 0) & (\leq, 0) \\
x & (\leq, 2) & - & (\leq, 2) \\
y & (\leq, 1) & (\leq, 0) & - \\
\end{array}
\]
**Symbolic State Space of Timed Automata**

**Zone**
- convex set of clock valuations given by conjunction of guards
- represents all possible clock valuations in one particular state

**Data structure**
- difference bound matrix (DBM)
- efficient operations, canonical form

![Diagram](image)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>−</td>
<td>(≤, 0)</td>
<td>(≤, 0)</td>
</tr>
<tr>
<td>x</td>
<td>(≤, 2)</td>
<td>−</td>
<td>(≤, 2)</td>
</tr>
<tr>
<td>y</td>
<td>(≤, 1)</td>
<td>(≤, 0)</td>
<td>−</td>
</tr>
</tbody>
</table>

\[ x \leq 2 \]
Symbolic State Space of Timed Automata

**Zone**
- convex set of clock valuations given by conjunction of guards
- represents all possible clock valuations in one particular state

**Data structure**
- difference bound matrix (DBM)
- efficient operations, canonical form

```
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>¬</td>
<td>(≤,0)</td>
<td>(≤,0)</td>
</tr>
<tr>
<td>x</td>
<td>(≤,2)</td>
<td>¬</td>
<td>(≤,2)</td>
</tr>
<tr>
<td>y</td>
<td>(≤,1)</td>
<td>(≤,0)</td>
<td>¬</td>
</tr>
</tbody>
</table>
```

\[ y \leq 1 \]
Symbolic State Space of Timed Automata

Zone
- convex set of clock valuations given by conjunction of guards
- represents all possible clock valuations in one particular state

Data structure
- difference bound matrix (DBM)
- efficient operations, canonical form

\[
\begin{array}{c|ccc}
 & 0 & x & y \\
\hline
0 & - & (\leq, 0) & (\leq, 0) \\
x & (\leq, 2) & - & (\leq, 2) \\
y & (\leq, 1) & (\leq, 0) & - \\
\end{array}
\]

\[y - x \leq 0\]
Symbolic State Space of Timed Automata

**Zone**
- convex set of clock valuations given by conjunction of guards
- represents all possible clock valuations in one particular state

**Data structure**
- difference bound matrix (DBM)
- efficient operations, canonical form

\[
\begin{array}{c|ccc}
   & 0 & x & y \\
\hline
0 & - & (\leq, 0) & (\leq, 0) \\
x & (\leq, 2) & - & (\leq, 2) \\
y & (\leq, 1) & (\leq, 0) & - \\
\end{array}
\]

\[y \geq 0\]
Symbolic State Space of PTA

Parametric zone
- given by conjunction of parametric guards, and
- constraints on parameter values (context)

Data structure
- Constrained parametric difference bound matrix (CPDBM)

CPDBM example
- Context = \{3 < p, p \leq 10\}

\[\begin{array}{c|ccc}
 & 0 & x & y \\
\hline
0 & - & (\leq, 0) & (\leq, 0) \\
x & (\leq, p) & - & (\leq, p) \\
y & (\leq, 1) & (\leq, 0) & - \\
\end{array}\]

\(^{1}\)Hune, T., Romijn, J., Stoelinga, M., Vaandrager, F.: Linear parametric model checking of timed automata. JLAP 52 (2002)
Symbolic State Space of PTA

Parametric zone

- given by conjunction of parametric guards, and
- constraints on parameter values (context)

Data structure

- Constrained parametric difference bound matrix (CPDBM) \(^1\)

CPDBM example

- Context = \(\{3 < p, p \leq 10\}\)

\[\begin{array}{c|ccc}
  & 0 & x & y \\
\hline
  0 & - & (\leq, 0) & (\leq, 0) \\
  x & (\leq, p) & - & (\leq, p) \\
  y & (\leq, 1) & (\leq, 0) & - \\
\end{array}\]

\(^1\)Hune, T., Romijn, J., Stoelinga, M., Vaandrager, F.: Linear parametric model checking of timed automata. JLAP 52 (2002)
result of CPDBM operations can be ambiguous

the application of a guard leads to a **split** of the parametric context

Example: $x \leq q$
- result of CPDBM operations can be ambiguous
- the application of a guard leads to a split of the parametric context

Example: $x \leq q$
result of CPDBM operations can be ambiguous

the application of a guard leads to a split of the parametric context

Example: $x \leq q$
result of CPDBM operations can be ambiguous

- the application of a guard leads to a split of the parametric context

Example: $x \leq q$
result of CPDBM operations can be ambiguous
the application of a guard leads to a **split** of the parametric context

Example: $x \leq q$

$$C \gets C \cup \{p \leq q\}$$

$$C \gets C \cup \{p > q\}$$
The number of (non-parametric) zones can be unbounded

**k-extrapolation**

- zones that differ only in bounds exceeding the *maximal bound* on clock valuations cannot be distinguished
- replace the bounds with $\infty$
**pk-extrapolation**

- based on k-extrapolation
- parametric zone bounds may exceed the maximal bound for only a subset of the allowed parameter valuations
- leads to a **split** of the parametric context
pk-extrapolation

- based on k-extrapolation
- parametric zone bounds may exceed the maximal bound for only a subset of the allowed parameter valuations
- leads to a split of the parametric context
**pk-extrapolation**

- based on k-extrapolation
- parametric zone bounds may exceed the maximal bound for only a subset of the allowed parameter valuations
- leads to a **split** of the parametric context
State Space Exploration

- state space storage needs unique representation of states
- one state represented with syntactically different CPDBMs
  ⇒ semantic equivalence checks

Heuristics

- representative: CPDBM of the state’s first occurrence
- integer hull\(^2\) of the state; hashtable
- caching

---

LTL Parameter Synthesis

Parametric timed automaton $A$ (system under investigation)

LTL formula $\varphi$ (specification)

Büchi automaton $A_{\neg \varphi}$

Parametric timed Büchi automaton $A \otimes A_{\neg \varphi}$

Büchi automaton $B$ with monotonic annotation $f$
(finite abstraction of $A \otimes A_{\neg \varphi}$)

Cumulative NDFS
Input

- Büchi automaton
- each state is associated a set of parameter valuations

Monotonicity property

- the set of associated parameter valuations does not grow along a run
- \( \Rightarrow \) does not change on a cycle
Cumulative Nested Depth First Search

Goal
- find the set of all parameter valuations associated with an accepting cycle

Algorithm
- based on Nested Depth First Search
- detects multiple accepting cycles on-the-fly
- the parameter valuations from the accepting cycles are accumulated during the computation
- backtracks when all parameter valuations associated with current state are already in the accumulated set
Cumulative Nested Depth First Search

Found: $v_1, v_2, v_3$
Cumulative Nested Depth First Search

Found:

$v_2, v_1, v_2, v_3$
Found: $v_2$
Cumulative Nested Depth First Search

Found: $v_2$
Found: $v_2$
Found: $v_2$
Cumulative Nested Depth First Search

Found: $v_2$
Found: $v_2$
Cumulative Nested Depth First Search

Found: $v_2, v_3$
Cumulative Nested Depth First Search

Found: $v_2, v_3$
Found: $v_2, v_3$
Found: $v_2$, $v_3$
Found: $v_2$, $v_3$
Found: $v_2$, $v_3$
Clock-Aware LTL Model Checking

Timed automaton $A$ (system under investigation)

CA-LTL formula $\varphi$ (specification)

$A_{ZURA}$ as a Kripke structure (finite abstraction of $A$)

Büchi automaton $A_{\neg \varphi}$

Büchi automaton $B = A_{ZURA} \otimes A_{\neg \varphi}$

Emptiness check
Formula guard satisfaction may not be consistent
Naive Partitioning

Consider the set of all constraints appearing in the formula

\[ G = \{ y \leq 2, x \leq 2 \} \]
consider the set of all constraints appearing in the formula
partition the zones w.r.t. the set

\[
G = \{y \leq 2, x \leq 2\}
\]
consider the set of all constraints appearing in the formula
partition the zones w.r.t. the set
add delay transitions

\[ G = \{ y \leq 2, x \leq 2 \} \]
Naive Partitionining

- consider the set of all constraints appearing in the formula
- partition the zones w.r.t. the set
- add delay transitions

Incorrect: consider the formula: \((y \leq 2 \land x \leq 2) \cup (y > 2 \land x > 2)\)
Correct Partitioning

Ultraregions

- partition with respect to $G$
- **add diagonals** to the partitioning

$$G = \{y \leq 2, x \leq 2\}$$
Ultraregions

- partition with respect to $G$
- add diagonals to the partitioning

$$G = \{ y \leq 2, x \leq 2, x < 3 \}$$
Zone-Ultraregion Abstraction

$A_{ZURA}$

- combine zone-based abstraction with ultraregions
- symbolic states $(I, Z, U)$
- action + delay transitions
- branching reset operation
- preserves all runs (w.r.t. CA-LTL)
- atomic propositions + satisfaction of the formula clock guards
Zone-Ultraregion Abstraction

\( A_{ZURA} \)
- combine zone-based abstraction with ultraregions
- symbolic states (\( I, Z, U \))
- action + delay transitions
- branching reset operation
- preserves all runs (w.r.t. CA-LTL)
- atomic propositions + satisfaction of the formula clock guards

**Parametric version**
- use parametric zones + pk-extrapolation
- finite-state symbolic Kripke structure \( A_{sym} \) with annotations
  (sets of parameter valuations)
- every \( \nu \)-run of \( A_{sym} \) is equivalent to a run in \( A_{\nu} \) and vice versa
CA-LTL Parameter Synthesis

- Parametric timed automaton $A$ (system under evaluation)
- CA-LTL formula $\varphi$ (specification)
- $A_{\text{sym}}$ with monotonic annotation $f$ (finite abstraction of $A$)
- Büchi automaton $A_{\neg \varphi}$

$B = A_{\text{sym}} \otimes A_{\neg \varphi}$ with monotonic annotation $f$

Cumulative NDFS
Experimental Evaluation

Implementation
- prototype tool: https://paradise.fi.muni.cz/parameterSynthesis/
- symbolic manipulation using Parma Polyhedra Library

Experimental Model
- parametric timed network of three sensors + a controller
- controller gathers data from sensors and provides a final value
- seven parameters
- properties to check:
  - $\varphi_1 = \mathbf{G}((l_1 \lor l_6) \Rightarrow (y \leq 500 U l_8))$
  - $\varphi_2 = \mathbf{G}((l_1 \lor l_6) \Rightarrow (y \leq 150 U l_8))$
  - $\varphi_3 = \mathbf{G}((l_1 \lor l_6) \Rightarrow \mathbf{F} l_8)$
### Table: Impact of model parameter count

<table>
<thead>
<tr>
<th></th>
<th>2 params</th>
<th>3 params</th>
<th>4 params</th>
<th>5 params</th>
<th>6 params</th>
<th>7 params</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi_1$ explicit</td>
<td>3.5 s</td>
<td>351 s</td>
<td>TO (17%)</td>
<td>TO (0%)</td>
<td>TO (0%)</td>
<td>TO (0%)</td>
</tr>
<tr>
<td>$\varphi_1$ CNDFS</td>
<td>0.4 s</td>
<td>2.2 s</td>
<td>3.3 s</td>
<td>5.7 s</td>
<td>8.6 s</td>
<td>36 s</td>
</tr>
<tr>
<td>$\varphi_2$ explicit</td>
<td>2.5 s</td>
<td>302 s</td>
<td>TO (20%)</td>
<td>TO (0%)</td>
<td>TO (0%)</td>
<td>TO (0%)</td>
</tr>
<tr>
<td>$\varphi_2$ CNDFS</td>
<td>2 s</td>
<td>25 s</td>
<td>151 s</td>
<td>1188 s</td>
<td>4924 s</td>
<td>TO</td>
</tr>
<tr>
<td>$\varphi^*_2$ CNDFS</td>
<td>2.5 s</td>
<td>29 s</td>
<td>193 s</td>
<td>866 s</td>
<td>3120 s</td>
<td>TO</td>
</tr>
<tr>
<td>$\varphi_3$ explicit</td>
<td>1.7 s</td>
<td>213 s</td>
<td>TO (22%)</td>
<td>TO (0%)</td>
<td>TO (0%)</td>
<td>TO (0%)</td>
</tr>
<tr>
<td>$\varphi_3$ CNDFS</td>
<td>0.5 s</td>
<td>3.9 s</td>
<td>52 s</td>
<td>124 s</td>
<td>189 s</td>
<td>1383 s</td>
</tr>
<tr>
<td>$\varphi^*_3$ CNDFS</td>
<td>0.3 s</td>
<td>1.5 s</td>
<td>2 s</td>
<td>3.8 s</td>
<td>5.6 s</td>
<td>24 s</td>
</tr>
</tbody>
</table>

* run with larger maximum constant (500) for pk-extrapolation

timeout 2 hours
**Experimental Evaluation**

**Table:** Impact of parameter range size

<table>
<thead>
<tr>
<th></th>
<th>[1, 10]</th>
<th>[1, 50]</th>
<th>[51, 100]</th>
<th>[1, 100]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi_1$ explicit</td>
<td>427 s</td>
<td>TO (0%)</td>
<td>TO (0%)</td>
<td>TO (0%)</td>
</tr>
<tr>
<td>$\varphi_1$ CNDFS</td>
<td>8.4 s</td>
<td>8.4 s</td>
<td>8.5 s</td>
<td>8.6 s</td>
</tr>
<tr>
<td>$\varphi_2$ explicit</td>
<td>426 s</td>
<td>TO (0%)</td>
<td>TO (0%)</td>
<td>TO (0%)</td>
</tr>
<tr>
<td>$\varphi_2$ CNDFS</td>
<td>8.4 s</td>
<td>33 s</td>
<td>1231 s</td>
<td>4924 s</td>
</tr>
<tr>
<td>$\varphi_2^*$ CNDFS</td>
<td>8.4 s</td>
<td>35 s</td>
<td>864 s</td>
<td>3120 s</td>
</tr>
<tr>
<td>$\varphi_3$ explicit</td>
<td>357 s</td>
<td>TO (0%)</td>
<td>TO (0%)</td>
<td>TO (0%)</td>
</tr>
<tr>
<td>$\varphi_3$ CNDFS</td>
<td>189 s</td>
<td>190 s</td>
<td>6.6 s</td>
<td>189 s</td>
</tr>
<tr>
<td>$\varphi_3^*$ CNDFS</td>
<td>6.2 s</td>
<td>6.2 s</td>
<td>6.2 s</td>
<td>6.3 s</td>
</tr>
</tbody>
</table>

* run with larger maximum constant (500) for pk-extrapolation

timeout 2 hours
Conclusion

Summary
- CA-LTL extends LTL with simple clock constraints
- symbolic method for CA-LTL parameter synthesis
  - new parametric abstraction (pk-extrapolation)
  - Cumulative NDFS
  - ultraregion technique for CA-LTL properties
- experimental evaluation
  - better than parameter scan
  - performance hard to predict
  - larger max for pk-extrapolation may help

Future Work
- try different abstractions
- parallel version of CNDFS
- extension of CA-LTL (action-based, difference constraints)
- parameters in CA-LTL properties