Liveness of Parameterized Timed Networks

Florian Zuleger
Technische Universität Wien

Joint work with Benjamin Aminof,
Sasha Rubin, Francesco Spegni
Labeled transition system:

- finite set of **states**
  (one **initial state**)
- finite set of **clocks**
- transitions labeled by **guards** and ** resets**
- guard = **comparison** of a **clock** to a **constant**

Time is either **continuous** or **discrete**.

\[
\begin{align*}
  x &= 0 \\
y &\geq 1 \\
x &:= 0; y := 0
\end{align*}
\]
Timed Automata - Semantics

- transitions = time passes

Alternative Representation:
- Explicit passage of time
- Clock values in states
- Finite number of clock values are sufficient
Timed Automata – Alternative Representation

Forget about clocks!

For the rest of the talk, we use this representation.

transitions = time passes
Timed Networks

Timed Network = finite number of copies of the same timed automaton + communication via rendezvous transitions
**Timed Networks**

Communication alphabet $\Sigma = \{a!,a?\} \cup \{\square\}$

Example run:

|   | p |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | p |   |   |   |   |   |   |
| 2 | p |   |   |   |   |   |   |
| 3 | p |   |   |   |   |   |   |

![Diagram](image-url)
Timed Networks

Communication alphabet $\Sigma = \{a!, a?\} \cup \{\square\}$

Example run:

1. $p \quad a! \quad p$
2. $p \quad a? \quad q$
3. $p \quad p$

Rendezvous transition
Timed Networks

Communication alphabet $\Sigma = \{a!, a?\} \cup \{\square\}$

Example run:

|   | p | a! | p | a! | p |
|---|---|----|---|----|---|
| ① | p | a? | q | q  | q |
| ② | p | a? | q | q  | q |
| ③ | p | a? | q | q  | q |
Timed Networks

Communication alphabet $\Sigma = \{a!, a?\} \cup \{\Box\}$

Example run:

1. $p \quad a! \quad p \quad a! \quad p \quad \Box \quad p$
2. $p \quad a? \quad q \quad q \quad \Box \quad p$
3. $p \quad p \quad a? \quad q \quad \Box \quad p$

Time passing transition
**Timed Networks**

Communication alphabet $\Sigma = \{a!,a?\} \cup \{\Box\}$

Example run:

|   | p | a! | p | a! | p | □ | p | a? | q |
|---|----|----|----|----|----|----|----|----|----|
| 1 | p  | a! | p  | a! | p  | □  | p  | a? | q |
| 2 | p  | a? | q  | q  | □  | p  | □  | p  | p |
| 3 | p  | □  | p  | a? | q  | □  | p  | a! | p |

\[\begin{array}{cccccccc}
1 & p & a! & p & a! & p & □ & p & a? & q \\
2 & p & a? & q & q & □ & p & □ & p & p \\
3 & p & □ & p & a? & q & □ & p & a! & p \\
\end{array}\]
Timed Networks

Communication alphabet $\Sigma = \{a!, a?\} \cup \{\square\}$

Example run:

|   | p | a! | p | a! | p | □ | p | a? | q | ... |
|---|---|----|---|----|---|---|---|----|---|-----|
|① | p | a! | p | a! | p | □ | p | a? | q | ... |
|② | p | a? | q | q | □ | p | p | p | ... |
|③ | p | p | a? | q | □ | p | a! | p | p | ... |
Timed Networks

Communication alphabet $\Sigma = \{a!, a?\} \cup \{\Box\}$

Example run:

|   | p | a! | p | a! | p | □ | p | a? | q | ... |
|---|---|----|---|----|---|---|---|----|---|-----|
| ① | p | a? | q | q | □ | p | p | ... |
| ② | p | a? | q | q | □ | p | p | ... |
| ③ | p | p | a? | q | □ | p | a! | p | ... |

Execution of ③ in the run:

| a? | □ | a! | ... |

execution = a sequence in $\Sigma^\omega$
Parameterized Model Checking

Timed automaton $A$

Communication alphabet $\Sigma$

$\text{Exec}(A_n) =$ all executions of a timed network with $n$ copies of automaton $A$

$\text{Exec}(A) = \bigcup_{n \geq 0} \text{Exec}(A_n)$

Parameterized Model Checking Problem (PMCP):

Given a language $L \subseteq \Sigma^\omega$, Liveness Property
decide $\text{Exec}(A) \subseteq L$?
Timed Networkds = RB-Systems

RB Systems = finite automata communicating via
- rendezvous transitions
- symmetric broadcast transitions

|   | p | a! | p | a! | p | □ | p | a? | q | ...
|---|---|----|---|----|---|----|---|----|---|---|
| 1 | p |    | p | a! | p | □ | p | a? | q | ...
| 2 | p | a? | q | q | □ | p | p | p |   |   |
| 3 | p |    | p | a? | q | □ | p | a! | p | ...

TU Wien
Florian Zuleger
(I) Why RB-Systems?

PMCP of liveness properties for finite automata communicating via (asymmetric) broadcast is undecidable (Esparza, Finkel, Mayr, LICS 1999)

Asymmetric broadcast is very powerful:
- allows to establish a controller process
- allows to simulate rendezvous transitions
(II) Why RB-Systems?

PMCP of liveness properties is undecidable (Abdulla, Jonsson, TCS 2003) for timed networks with
- continuous-time
- a distinguished controller process
- rendezvous transitions

Proof heavily relies on
- time being dense
- controller for coordination
Theorem
Given a timed automaton $A$, we can compute a $B$-automaton $B$ such that $\text{Exec}(A) = L(B)$.

Corollary
PMCP is decidable for specifications given by a BS-automaton*.
Theorem
Given a timed automaton $A$, we can compute a B-automaton $B$ such that $\text{Exec}(A) = L(B)$.

Florian Zuleger
TU Wien

Corollary
PMCP is decidable for specifications given by a BS-automaton*.

BS-automata (Bojanczyk, Colcombet LICS 2006):
- decidable emptiness
- closed under union, intersection
- not closed under complement
- subclasses B- and S-automata that are closed under complement
- strictly generalize $\omega$-regular languages
Why BS-automata?

\[ a!, a? \text{ may only boundedly often be taken between two } \square! \]
Why BS-automata?

a!, a? may only boundedly often be taken between two □!

„boundedly often“ =

| a? | □ | a! | a? | □ | a! | a? | □ | ... |
|----|----|----|----|----|----|----|----|-----|

there is a k ∈ N with

≤ k

≤ k

≤ k
Why BS-automata?

$a!, a?$ may only boundedly often be taken between two $\square$!

"boundedly often" =

there is a $k \in \mathbb{N}$ with $\leq k$
BS-automata

BS-automata have finite number of counters.

Counters can be
1) reset,
2) incremented,
3) assigned to other counters.

Acceptance condition = positive boolean combination of Büchi condition + “counter is bounded” + “counter goes to ∞”
4 Types of Automata Edges

**Red:** appears at most finitely often on any execution

**Blue:** appears infinitely times on some execution, but only finitely often on every execution with infinitely many broadcasts

**Orange:** appears infinitely times on some execution with infinitely many broadcasts, but only boundedly many times between two broadcasts

**Green:** otherwise
4 Types of Automata Edges

**Red:** appears at most finitely many times on any execution

Proof strategy:
1) establish edge types
2) add a suitable counter update to each transition
3) equip the automaton with a suitable B-acceptance condition

**Green:** otherwise
Lasso Shaped Reachability Graph

- Initial states
- States after a broadcast
- States reachable via rendezvous
Deciding Edge Types

Essential question:
Is there a cyclic run of the lasso that uses edge $e$?
Linear Program by Example

variables $x_1, x_2, y_1, y_2 \in \mathbb{Q}$ for the number of automata in state $p$ resp. $q$ at $I_1$ resp. $P_1$

executing rendezvous transitions (with $c \in \mathbb{Q}$):

rendezvous transition is taken at least once:

executing broadcast:

$x_1 = y_1 + y_2$

$c \geq 1$

$y_1 = x_1 - c$

$y_2 = x_2 + c$

$p = I_1 = \{p\}$

$q = P_1 = \{p, q\}$

$p$
variables $x_1, x_2, y_1, y_2 \in \mathbb{Q}$ for the number of automata in state $p$ resp. $q$ at $I_1$ resp. $P_1$.

- $x_1, x_2, y_1, y_2 \geq 0$
- $y_1 = x_1 - c$
- $y_2 = x_2 + c$
- $c \geq 1$
- $x_1 = y_1 + y_2$
- $x_2 = 0$

We can use LPs over $\mathbb{Q}$ because solutions are always scalable thanks to symmetry.
Linear Programs: A Complication

An assignment

\[ y = x + c_1 \cdot t_1 + \ldots + c_n \cdot t_n \]

does not guarantee that there is a path from \( x \) to \( y \), e.g.,

\[
\begin{bmatrix}
3 \\
0 \\
0
\end{bmatrix} = \begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix} + \begin{bmatrix}
1 \\
-1 \\
1
\end{bmatrix} + \begin{bmatrix}
1 \\
1 \\
-1
\end{bmatrix},
\]

because coordinates can become negative.

Key Lemma:
If there is a path from \( x \in \mathbb{Q}^d \) to \( y \in \mathbb{Q}^d \), then there also is a path such that on \( q \) the vector components with a 0 do not change and \( p_1, p_2 \) are of form \( t_1^* \ldots t_d^* \) for some transitions \( t_1, \ldots, t_d \).
Linear Programs: A Complication

An assignment

\[ y = x + c_1 \cdot t_1 + ... + c_n \cdot t_n \]

does not guarantee that there is a path from \( x \) to \( y \), e.g.,

\[
\begin{pmatrix}
3 \\
0 \\
0
\end{pmatrix}
= \begin{pmatrix}
1 \\
0 \\
0
\end{pmatrix}
+ \begin{pmatrix}
1 \\
-1 \\
1
\end{pmatrix}
+ \begin{pmatrix}
1 \\
1 \\
-1
\end{pmatrix},
\]

because coordinates can become negative.

Key Lemma:

If there is a path from \( x \in \mathbb{Q}^d \) to \( y \in \mathbb{Q}^d \),

\[ x - y \] is a path

and \( p_1, p_2 \) are of form \( t_1 \times ... \times t_d \) for some transitions \( t_1, ..., t_d \).

\[ \rightarrow \] Finite number of linear programs
Summary

- Decidability for liveness properties of timed networks
- New communication primitive „symmetric broadcast“
- New proof techniques: hopefully are useful in similar settings