ICECCS 2020

5 March 2021

Parametric non-interference in timed automata

Étienne André and Aleksander Kryukov

Université de Lorraine, CNRS, Inria, LORIA, Nancy, France

Supported by the ANR-NRF French-Singaporean research program ProMiS (ANR-19-CE25-0015)

Outline

Context: security

Security of computer systems

- Threats coming from an intruder or an unsafe medium (Internet)

[Koc96] Paul C. Kocher. "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems". In: *CRYPTO* (Aug. 18–22, 1996). Vol. 1109. LNCS. Santa Barbara, California, USA: Springer, 1996, pp. 104–113. DOI: 10.1007/3-540-68697-5_9

[Ben+15] Gilles Benattar, Franck Cassez, Didier Lime, and Olivier H. Roux. "Control and synthesis of non-interferent timed systems". In: *International Journal of Control* 88.2 (2015), pp. 217–236. DOI: 10.1080/00207179.2014.944356

[GMR07] Guillaume Gardey, John Mullins, and Olivier H. Roux. "Non-Interference Control Synthesis for Security Timed Automata". In: *Electronic Notes in Theoretical Computer Science* 180.1 (2007), pp. 35–53. DOI: 10.1016/j.entcs.2005.05.046

Context: security

Security of computer systems

- Threats coming from an intruder or an unsafe medium (Internet)

Risk: consequence of external actions onto critical internal behaviors: non-interference

[Koc96] Paul C. Kocher. “Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems”. In: *CRYPTO* (Aug. 18–22, 1996). Vol. 1109. LNCS. Santa Barbara, California, USA: Springer, 1996, pp. 104–113. DOI: 10.1007/3-540-68697-5_9

[Ben+15] Gilles Benattar, Franck Cassez, Didier Lime, and Olivier H. Roux. “Control and synthesis of non-interferent timed systems”. In: *International Journal of Control* 88.2 (2015), pp. 217–236. DOI: 10.1080/00207179.2014.944356

[GMR07] Guillaume Gardey, John Mullins, and Olivier H. Roux. “Non-Interference Control Synthesis for Security Timed Automata”. In: *Electronic Notes in Theoretical Computer Science* 180.1 (2007), pp. 35–53. DOI: 10.1016/j.entcs.2005.05.046

Context: security

Security of computer systems

- Threats coming from an intruder or an unsafe medium (Internet)

Risk: consequence of external actions onto critical internal behaviors: non-interference

Timed systems: challenging

- time is a potential attack vector against secure systems [Koc96][Ben+15]
- a non-interferent system can become interferent when timing information is added [GMR07]

[Koc96] Paul C. Kocher. “Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems”. In: *CRYPTO* (Aug. 18–22, 1996). Vol. 1109. LNCS. Santa Barbara, California, USA: Springer, 1996, pp. 104–113. DOI: 10.1007/3-540-68697-5_9

[Ben+15] Gilles Benattar, Franck Cassez, Didier Lime, and Olivier H. Roux. “Control and synthesis of non-interferent timed systems”. In: *International Journal of Control* 88.2 (2015), pp. 217–236. DOI: 10.1080/00207179.2014.944356

[GMR07] Guillaume Gardey, John Mullins, and Olivier H. Roux. “Non-Interference Control Synthesis for Security Timed Automata”. In: *Electronic Notes in Theoretical Computer Science* 180.1 (2007), pp. 35–53. DOI: 10.1016/j.entcs.2005.05.046

Context: non-interference

Measure the disturbance of a system

A system is non-interferent when some disturbance on some high-level actions does not affect the observable behavior ("low-level" actions)

When adding time information: Question of the frequency

[Bar+02] Roberto Barbuti, Nicoletta De Francesco, Antonella Santone, and Luca Tesei. "A Notion of Non-Interference for Timed Automata". In: *Fundamenta Informaticae* 51.1-2 (2002), pp. 1–11

[BT03] Roberto Barbuti and Luca Tesei. "A Decidable Notion of Timed Non-Interference". In: *Fundamenta Informaticae* 54.2-3 (2003), pp. 137–150

Context: non-interference

Measure the disturbance of a system

A system is non-interferent when some disturbance on some high-level actions does not affect the observable behavior ("low-level" actions)

When adding time information: Question of the frequency

Key point: frequency

Does performing an arbitrary high-level action at a given frequency disturbs the observable behavior?

- [Bar+02]: observable behavior = timed language
- [BT03]: observable behavior = set of discrete states

[Bar+02] Roberto Barbuti, Nicoletta De Francesco, Antonella Santone, and Luca Tesei. "A Notion of Non-Interference for Timed Automata". In: *Fundamenta Informaticae* 51.1-2 (2002), pp. 1–11

[BT03] Roberto Barbuti and Luca Tesei. "A Decidable Notion of Timed Non-Interference". In: *Fundamenta Informaticae* 54.2-3 (2003), pp. 137–150

Context: non-interference

Measure the disturbance of a system

A system is non-interferent when some disturbance on some high-level actions does not affect the observable behavior ("low-level" actions)

When adding time information: Question of the frequency

Key point: frequency

Does performing an arbitrary high-level action at a given frequency disturbs the observable behavior?

- [Bar+02]: observable behavior = timed language
- [BT03]: observable behavior = set of discrete states

Here, we will address a parametric version of the problem, and synthesize this frequency: at which frequency can we perform high-level actions without disturbing the observable behavior?

[Bar+02] Roberto Barbuti, Nicoletta De Francesco, Antonella Santone, and Luca Tesei. "A Notion of Non-Interference for Timed Automata". In: *Fundamenta Informaticae* 51.1-2 (2002), pp. 1–11

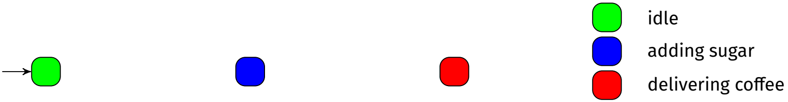
[BT03] Roberto Barbuti and Luca Tesei. "A Decidable Notion of Timed Non-Interference". In: *Fundamenta Informaticae* 54.2-3 (2003), pp. 137–150

Outline

Outline

Timed automaton (TA)

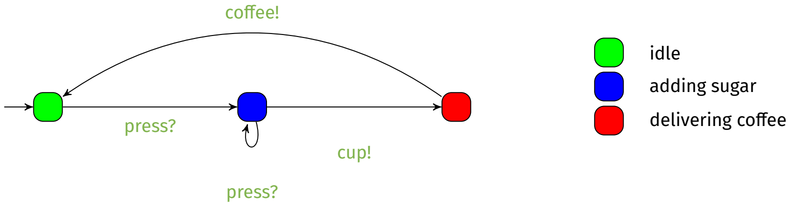
- Finite state automaton (sets of locations)



[AD94] Rajeev Alur and David L. Dill. “A theory of timed automata”. In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Timed automaton (TA)

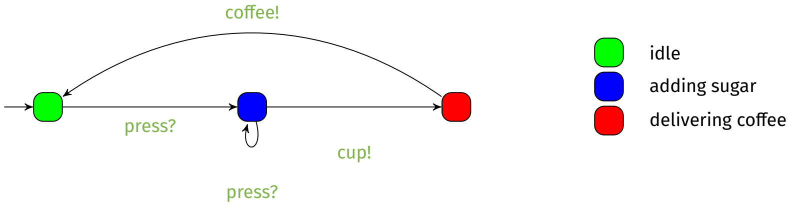
- Finite state automaton (sets of locations and actions)



[AD94] Rajeev Alur and David L. Dill. “A theory of timed automata”. In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Timed automaton (TA)

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate

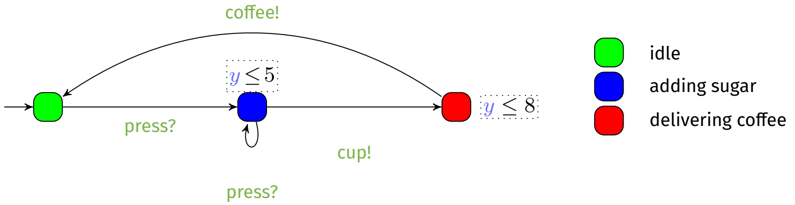


[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Timed automaton (TA)

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate
  - Can be compared to integer constants in invariants

- Features
  - Location invariant: property to be verified to stay at a location

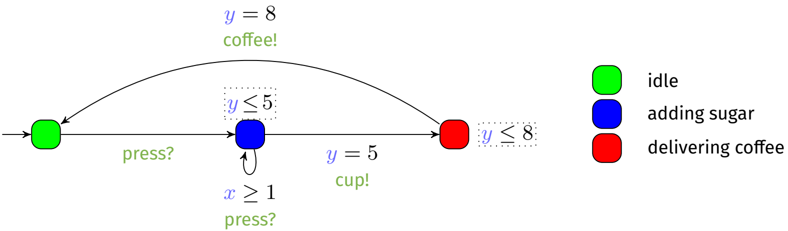


[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Timed automaton (TA)

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate
  - Can be compared to integer constants in invariants and guards

- Features
  - Location invariant: property to be verified to stay at a location
  - Transition guard: property to be verified to enable a transition

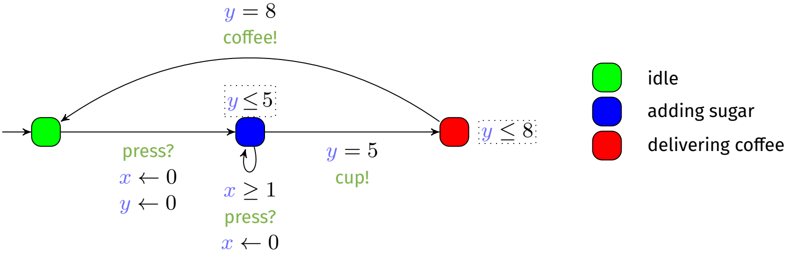


[AD94] Rajeev Alur and David L. Dill. “A theory of timed automata”. In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Timed automaton (TA)

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate
  - Can be compared to integer constants in invariants and guards

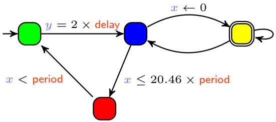
- Features
  - Location invariant: property to be verified to stay at a location
  - Transition guard: property to be verified to enable a transition
  - Clock reset: some of the clocks can be set to 0 along transitions



[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Outline

timed model checking



A model of the system

$$\stackrel{?}{\models}$$

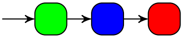
 is unreachable

A property to be satisfied

- Question: does the model of the system satisfy the property?

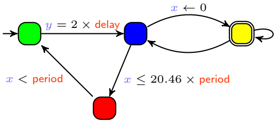
Yes

No



Counterexample

Parametric timed model checking



A model of the system

$$\overset{?}{\models}$$

 is unreachable

A property to be satisfied

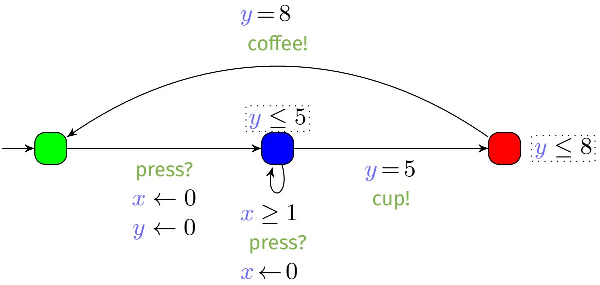
- Question: for what values of the parameters does the model of the system satisfy the property?

Yes if...

$$2 \times \text{delay} > 20.46 \times \text{period}$$

Parametric Timed Automaton (PTA)

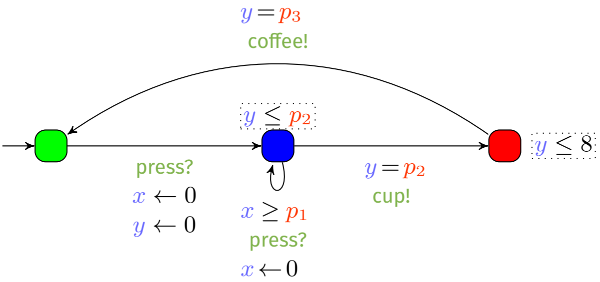
- Timed automaton (sets of locations, actions and clocks)



[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. “Parametric real-time reasoning”. In: *STOC* (May 16–18, 1993). San Diego, California, United States: ACM, 1993, pp. 592–601. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242

Parametric Timed Automaton (PTA)

- Timed automaton (sets of locations, actions and clocks) augmented with a set P of parameters [AHV93]
  - Unknown constants compared to a clock in guards and invariants



[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC* (May 16–18, 1993). San Diego, California, United States: ACM, 1993, pp. 592–601. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242

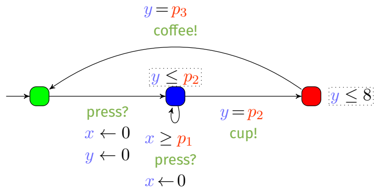
Notation: Valuation of a PTA

- Given a PTA $\mathcal{A}$ and a parameter valuation v, we denote by $v(\mathcal{A})$ the (non-parametric) timed automaton where each parameter p is valuated by $v(p)$

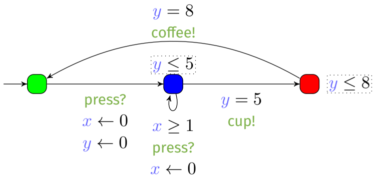
Notation: Valuation of a PTA

- Given a PTA $\mathcal{A}$ and a parameter valuation v, we denote by $v(\mathcal{A})$ the (non-parametric) timed automaton where each parameter p is valuated by $v(p)$

v (



) =



$$\text{with } v : \begin{cases} p_1 & \rightarrow & 1 \\ p_2 & \rightarrow & 5 \\ p_3 & \rightarrow & 8 \end{cases}$$

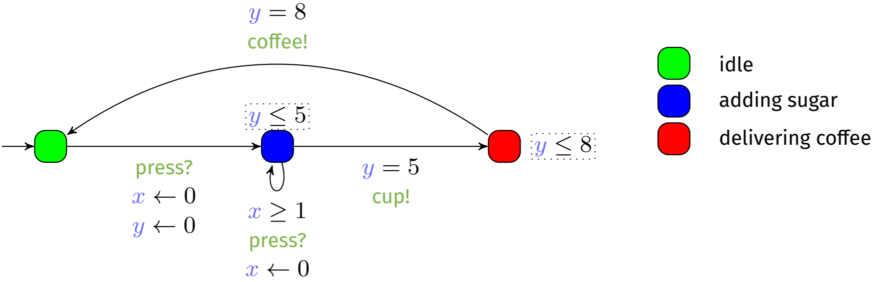
Concrete semantics of timed automata

- Concrete state of a TA: pair (ℓ, w), where
  - ℓ is a location,
  - w is a valuation of each clock

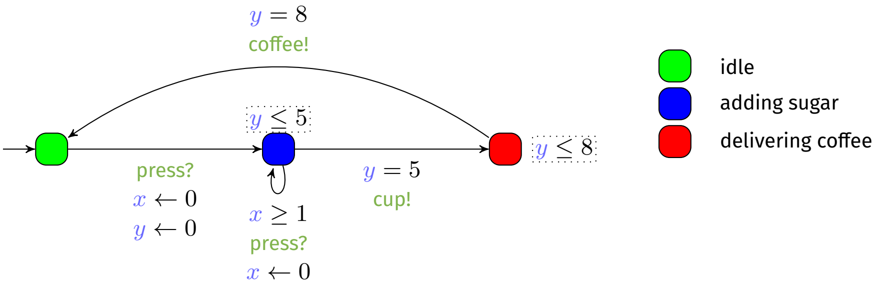
  Example: ( , $\binom{x=1.2}{y=3.7}$)

- Concrete run: alternating sequence of concrete states and actions or time elapse

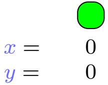
The most critical system: The coffee machine



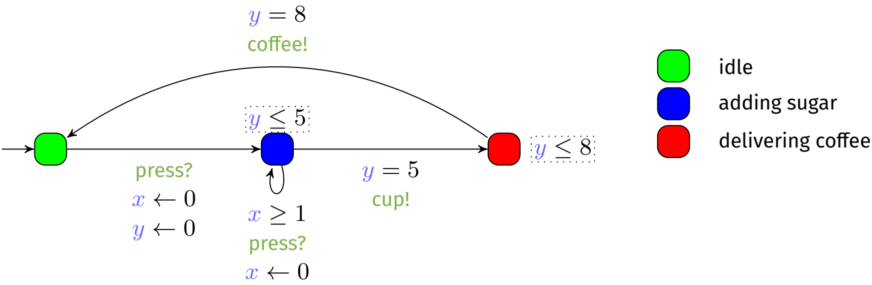
The most critical system: The coffee machine



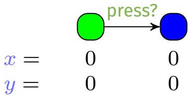
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



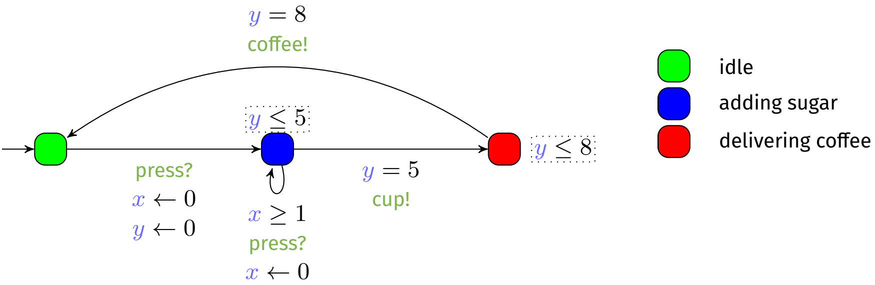
The most critical system: The coffee machine



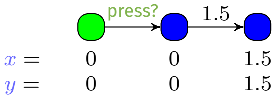
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



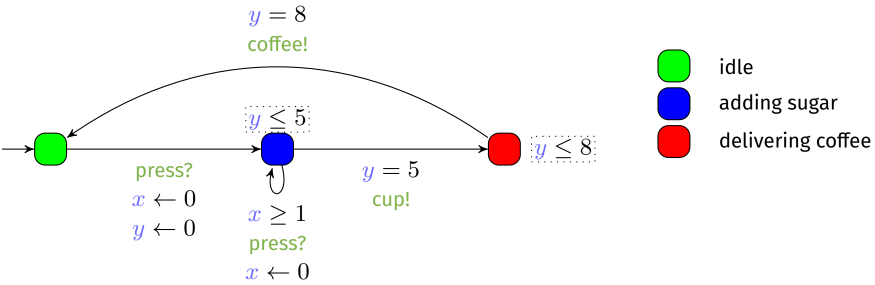
The most critical system: The coffee machine



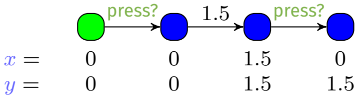
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



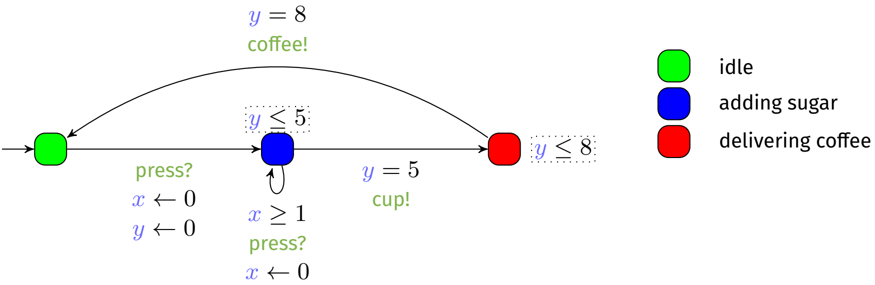
The most critical system: The coffee machine



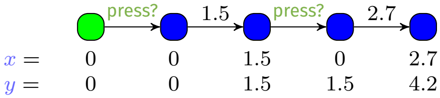
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



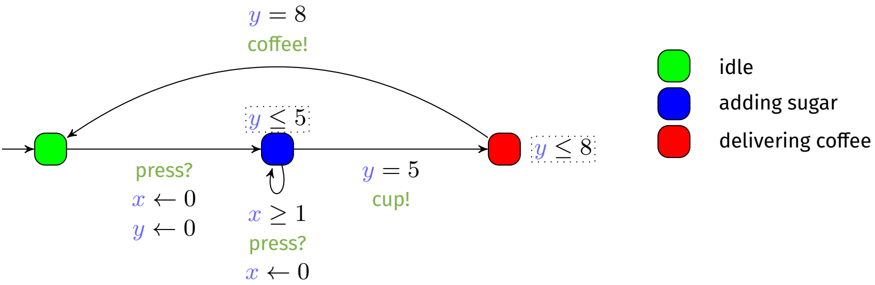
The most critical system: The coffee machine



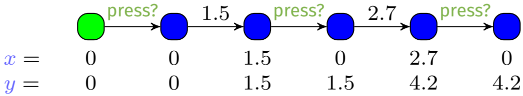
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



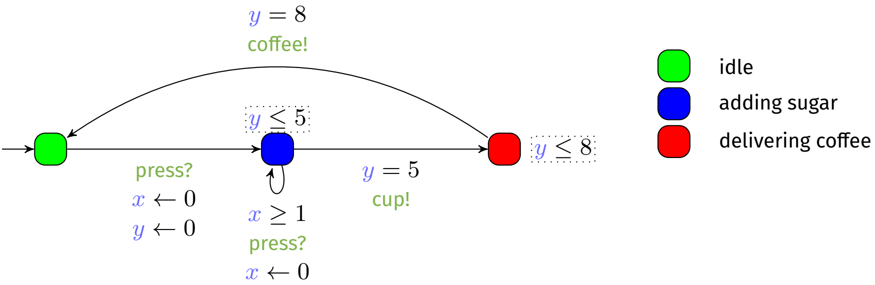
The most critical system: The coffee machine



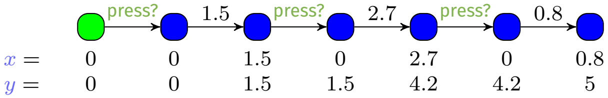
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



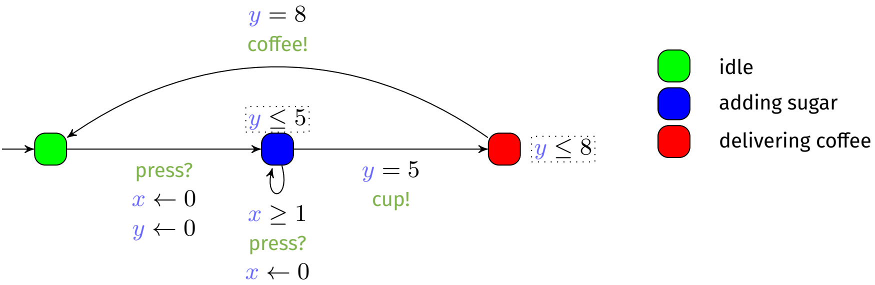
The most critical system: The coffee machine



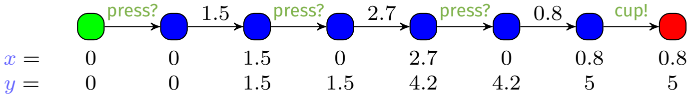
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



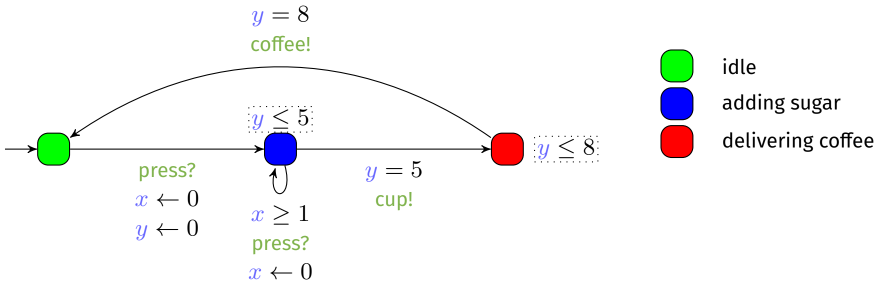
The most critical system: The coffee machine



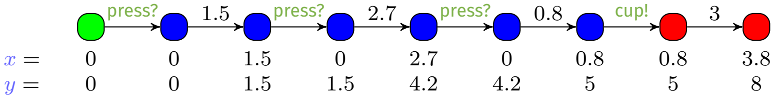
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



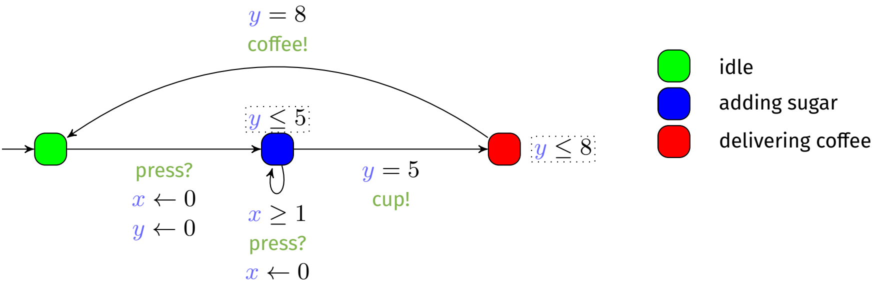
The most critical system: The coffee machine



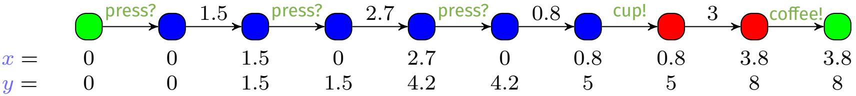
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



The most critical system: The coffee machine



- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



Outline

n-location-non-interference: Definition

Let $\Sigma = L \uplus H$

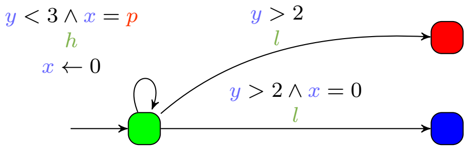
- L: low-level actions
- H: high-level actions

Definition

A TA $\mathcal{A}$ is n-location-non-interferent if the sets of reachable locations are equal in the following TAs:

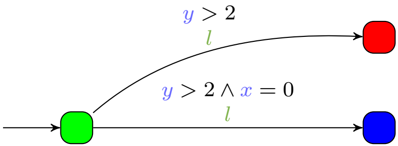
1. $\mathcal{A}$ without any high-level action
2. $\mathcal{A}$ with high-level actions separated by at least n time units

n-location-non-interference: Example



Fix $p = 1.1$

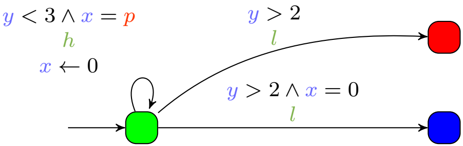
n-location-non-interference: Example



Fix $p = 1.1$

1. Locations reachable in $\mathcal{A}$ without any high-level action: $\{$, $\}$

n-location-non-interference: Example

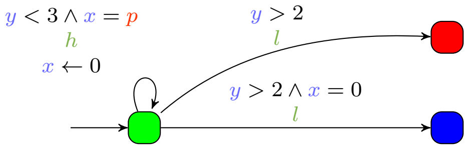


Fix $p = 1.1$

1. Locations reachable in $\mathcal{A}$ without any high-level action: {, }
2. Locations reachable in $\mathcal{A}$ with high-level actions separated by at least 1 time unit: {, , }

$\Rightarrow$ $\mathcal{A}$ is not 1-location-non-interfering

n-location-non-interference: Example



Fix $p = 1.1$

1. Locations reachable in $\mathcal{A}$ without any high-level action: $\{$, $\}$
2. Locations reachable in $\mathcal{A}$ with high-level actions separated by at least 1 time unit: $\{$, , $\}$
3. Locations reachable in $\mathcal{A}$ with high-level actions separated by at least 2 time units: $\{$, $\}$

$\Rightarrow$ $\mathcal{A}$ is not 1-location-non-interfering

$\Rightarrow$ $\mathcal{A}$ is 2-location-non-interfering

Problem

Problem: n-location-non-interference synthesis

Inputs:

- A parametric TA $\mathcal{A}$ with parameters P
- A parameter n

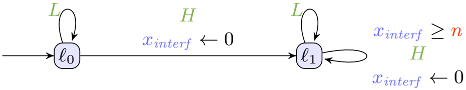
Goal:
"Synthesize valuations v of P and of n such that $v(\mathcal{A})$ is n-location-non-interfering."

Outline

Our approach in a nutshell: Gadget

We take the parallel product of

- $\mathcal{A}$, and
- a special gadget PTA "$\mathcal{I}nterf^{n}_{H}$" constraining any high-level action to be separated by at least n time units



Our approach in a nutshell: Reachability synthesis

Then, we compute

1. a set of locations G to be reached for some desired property in $\mathcal{A}$
2. the set of parameter valuations for which G is reachable in $\mathcal{A} \parallel \mathcal{I}nterf^{n}_{H}$

[JLR15] Aleksandra Jovanović, Didier Lime, and Olivier H. Roux. "Integer Parameter Synthesis for Real-Time Systems". In: *IEEE Transactions on Software Engineering* 41.5 (2015), pp. 445–461. DOI: 10.1109/TSE.2014.2357445

Our approach in a nutshell: Reachability synthesis

Then, we compute

1. a set of locations G to be reached for some desired property in $\mathcal{A}$
2. the set of parameter valuations for which G is reachable in $\mathcal{A} \parallel \mathcal{I}nterf^{n}_{H}$

Toolkit:

- Semi-algorithm: reachability synthesis [JLR15]
  - semi-algorithm: no theoretical guarantee on termination
- implemented in IMITATOR

[JLR15] Aleksandra Jovanović, Didier Lime, and Olivier H. Roux. "Integer Parameter Synthesis for Real-Time Systems". In: *IEEE Transactions on Software Engineering* 41.5 (2015), pp. 445–461. DOI: 10.1109/TSE.2014.2357445

Outline

Outline

Fischer mutual exclusion protocol

Two processes P_1 and P_2 running in parallel compete for the critical section.

Atomic reads and writes are permitted to a shared variable `v`

Every access to the shared memory containing `v` takes acc units of time.

[BT03] Roberto Barbuti and Luca Tesei. "A Decidable Notion of Timed Non-Interference". In: *Fundamenta Informaticae* 54.2-3 (2003), pp. 137–150

Fischer mutual exclusion protocol

Two processes P_1 and P_2 running in parallel compete for the critical section.
Atomic reads and writes are permitted to a shared variable v
Every access to the shared memory containing v takes acc units of time.
Each process P_i executes the following code: [BT03]

```
repeat
    await v = 0
    v := i
    delay b
until v = i
v := 0
(* Critical section* )
```

[BT03] Roberto Barbuti and Luca Tesei. "A Decidable Notion of Timed Non-Interference". In: *Fundamenta Informaticae* 54.2-3 (2003), pp. 137–150

Fischer mutual exclusion protocol

Two processes P_1 and P_2 running in parallel compete for the critical section.

Atomic reads and writes are permitted to a shared variable v

Every access to the shared memory containing v takes acc units of time.

Each process P_i executes the following code: [BT03]

```
repeat
    await v = 0
    v := i
    delay b
until v = i
v := 0
(* Critical section* )
```

An assignment takes (at most) a time units

Maximum time needed to execute the critical section is ucs

[BT03] Roberto Barbuti and Luca Tesei. "A Decidable Notion of Timed Non-Interference". In: *Fundamenta Informaticae* 54.2-3 (2003), pp. 137–150

Fischer mutual exclusion protocol

Two processes P_1 and P_2 running in parallel compete for the critical section.
Atomic reads and writes are permitted to a shared variable $\mathtt{v}$
Every access to the shared memory containing $\mathtt{v}$ takes acc units of time.
Each process P_i executes the following code: [BT03]

```
repeat
    await v = 0
    v := i
    delay b
until v = i
v := 0
(* Critical section* )
```

An assignment takes (at most) a time units
Maximum time needed to execute the critical section is ucs
Crux: P_i is allowed into the critical section only when $\mathtt{v} = i$

[BT03] Roberto Barbuti and Luca Tesei. "A Decidable Notion of Timed Non-Interference". In: *Fundamenta Informaticae* 54.2-3 (2003), pp. 137–150

Fischer: intruder

Attacker scheme

An intruder can take anytime a high-level transition “att”, nondeterministically changing `v` to 0, 1 or 2

Fischer: objective

Objective

Automatically infer conditions over n, a, b, acc and ucs guaranteeing n-location-non-interference.

Fischer: objective

Objective

Automatically infer conditions over n, a, b, acc and ucs guaranteeing n-location-non-interference.

Put it differently: offer guarantees that the Fischer protocol will still be valid even in the situation of an attack on the variable $\mathtt{v}$, with a maximum frequency n

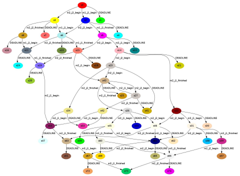
In particular, since the reachable locations do not change, the location where both processes are in the critical section at the same time (safety violation) remains unreachable

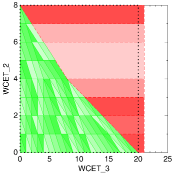
Outline

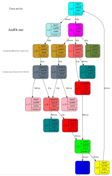
IMITATOR

- A tool for modeling and verifying timed concurrent systems with unknown constants modeled with parametric timed automata
  - Communication through (strong) broadcast synchronization
  - Rational-valued shared discrete variables
  - Stopwatches, to model schedulability problems with preemption
  - Multi-rate clocks

- Synthesis algorithms
  - (non-Zeno) parametric model checking (using a subset of TCTL)
  - Language and trace preservation, and robustness analysis
  - Parametric deadlock-freeness checking







IMITATOR

Under continuous development since 2008 [And+12]

A library of benchmarks [And19]

- Communication protocols
- Schedulability problems
- Asynchronous circuits
- ...and more

Free and open source software: Available under the GNU-GPL license

[And+12] Étienne André, Laurent Fribourg, Ulrich Kühne, and Romain Soulat. “IMITATOR 2.5: A Tool for Analyzing Robustness in Scheduling Problems”. In: *FM* (Aug. 27–31, 2012). Vol. 7436. LNCS. Paris, France: Springer, Aug. 2012, pp. 33–36. DOI: 10.1007/978-3-642-32759-9_6

[And19] Étienne André. “A benchmark library for parametric timed model checking”. In: *FTSCS* (Nov. 16, 2018). Vol. 1008. CCIS. Gold Coast, Australia: Springer, 2019, pp. 75–83. DOI: 10.1007/978-3-030-12988-0_5

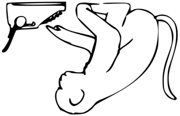
IMITATOR

Under continuous development since 2008 [And+12]

A library of benchmarks [And19]

- Communication protocols
- Schedulability problems
- Asynchronous circuits
- ...and more

Free and open source software: Available under the GNU-GPL license



Try it! www.imitator.fr

[And+12] Étienne André, Laurent Fribourg, Ulrich Kühne, and Romain Soulat. "IMITATOR 2.5: A Tool for Analyzing Robustness in Scheduling Problems". In: *FM* (Aug. 27–31, 2012). Vol. 7436. LNCS. Paris, France: Springer, Aug. 2012, pp. 33–36. DOI: 10.1007/978-3-642-32759-9_6

[And19] Étienne André. "A benchmark library for parametric timed model checking". In: *FTSCS* (Nov. 16, 2018). Vol. 1008. CCIS. Gold Coast, Australia: Springer, 2019, pp. 75–83. DOI: 10.1007/978-3-030-12988-0_5

Some success stories

- Modeled and verified an asynchronous memory circuit by ST-Microelectronics
- Parametric schedulability analysis of a prospective architecture for the flight control system of the next generation of spacecrafts designed at ASTRIUM Space Transportation [Fri+12]
- Verification of software product lines [Lut+17]
- Formal timing analysis of music scores [FJ13]
- Solution to a challenge related to a distributed video processing system by Thales
- Parametric timed pattern matching

[Fri+12] Laurent Fribourg, David Lesens, Pierre Moro, and Romain Soulat. "Robustness Analysis for Scheduling Problems using the Inverse Method". In: *TIME* (Sept. 12–14, 2012). Leicester, UK: IEEE Computer Society Press, Sept. 2012, pp. 73–80. DOI: 10.1109/TIME.2012.10

[Lut+17] Lars Luthmann, Andreas Stephan, Johannes Bürdek, and Malte Lochau. "Modeling and Testing Product Lines with Unbounded Parametric Real-Time Constraints". In: *SPLC, Volume A* (Sept. 25–29, 2017). Sevilla, Spain: ACM, 2017, pp. 104–113. DOI: 10.1145/3106195.3106204

[FJ13] Léa Fanchon and Florent Jacquemard. "Formal Timing Analysis Of Mixed Music Scores". In: *ICMC* (Aug. 12–16, 2013). Perth, Australia: Michigan Publishing, Aug. 2013

Outline

A modified model

Modified the model from [BT03]:

- Corrected some (non-trivial) aspects
  - Entirely rewrote the serializer (responsible for synchronizing the processes and the intruder)
- Added parameters, notably n
- Added the n-non-interference gadget

Target set of locations G:

- All locations except those where the mutual exclusion is violated (both processes in the critical section together)

[BT03] Roberto Barbuti and Luca Tesei. "A Decidable Notion of Timed Non-Interference". In: *Fundamenta Informaticae* 54.2-3 (2003), pp. 137–150

Preliminary results

- ☹ Analysis with IMITATOR does not terminate
- ☺ …but an over-approximation is synthesized

[LPY97] Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. "UPPAAL in a Nutshell". In: *International Journal on Software Tools for Technology Transfer* 1.1-2 (1997), pp. 134–152. DOI: 10.1007/s100090050010

Preliminary results

- ☹ Analysis with IMITATOR does not terminate
- ☺ …but an over-approximation is synthesized

Claim: this result might be exact

- ☺ We “tested” dozens of parameter valuations with UPPAAL [LPY97]
- ☹ No formal guarantee of soundness!

[LPY97] Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. “UPPAAL in a Nutshell”. In: *International Journal on Software Tools for Technology Transfer* 1.1-2 (1997), pp. 134–152. DOI: 10.1007/s100090050010

Preliminary results

- ☹ Analysis with IMITATOR does not terminate
- ☺ …but an over-approximation is synthesized

Claim: this result might be exact

- ☺ We "tested" dozens of parameter valuations with UPPAAL [LPY97]
- ☹ No formal guarantee of soundness!

One disjunct among the synthesized constraint:

$$
\begin{array}{ll}
 & n \geq 0 \\
\wedge & b \geq acc + n \\
\wedge & b \geq 3 \times acc \\
\wedge & a > 0 \\
\wedge & acc > ucs > 0
\end{array}
$$

Sources, binaries, models, results available at `www.imitator.fr/static/ICECCS20`

[LPY97] Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. "UPPAAL in a Nutshell". In: *International Journal on Software Tools for Technology Transfer* 1.1-2 (1997), pp. 134–152. DOI: 10.1007/s100090050010

Outline

Conclusion and perspectives

Conclusion

- A first notion of parametric n-non-interference
- Helps to quantify the admissible frequency of attacks without any effect on the intended behavior
- Dually: quantify the effect of internal actions (by admins) without observable behavior from the outside
- Approximated constraint for Fischer protocol
  - Toolkit: IMITATOR

[Ben+15] Gilles Benattar, Franck Cassez, Didier Lime, and Olivier H. Roux. "Control and synthesis of non-interferent timed systems". In: *International Journal of Control* 88.2 (2015), pp. 217–236. DOI: 10.1080/00207179.2014.944356

Conclusion and perspectives

Conclusion

- A first notion of parametric n-non-interference
- Helps to quantify the admissible frequency of attacks without any effect on the intended behavior
- Dually: quantify the effect of internal actions (by admins) without observable behavior from the outside
- Approximated constraint for Fischer protocol
  - Toolkit: IMITATOR

Perspectives:

- Theoretical issues: decidable subclasses?
- Non-interference w.r.t. the language
- Extend to control [Ben+15]

[Ben+15] Gilles Benattar, Franck Cassez, Didier Lime, and Olivier H. Roux. "Control and synthesis of non-interferent timed systems". In: *International Journal of Control* 88.2 (2015), pp. 217–236. DOI: 10.1080/00207179.2014.944356

We hire!

- Who:
  - PhD students
  - Research fellows (post-doc)
- Project ANR-NRF ProMiS
  - quantitative formal methods + security (2020-2023)
- Where: France (Nancy / Nantes), Singapore

…starting anytime!

Bibliography

References I

Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8.

Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC* (May 16–18, 1993). San Diego, California, United States: ACM, 1993, pp. 592–601. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242.

Étienne André and Aleksander Kryukov. "Parametric non-interference in timed automata". In: *ICECCS* (Mar. 4–6, 2021). Singapore, 2020, pp. 37–42. DOI: 10.1109/ICECCS51672.2020.00012.

Étienne André, Laurent Fribourg, Ulrich Kühne, and Romain Soulat. "IMITATOR 2.5: A Tool for Analyzing Robustness in Scheduling Problems". In: *FM* (Aug. 27–31, 2012). Vol. 7436. LNCS. Paris, France: Springer, Aug. 2012, pp. 33–36. DOI: 10.1007/978-3-642-32759-9_6.

Étienne André. "A benchmark library for parametric timed model checking". In: *FTSCS* (Nov. 16, 2018). Vol. 1008. CCIS. Gold Coast, Australia: Springer, 2019, pp. 75–83. DOI: 10.1007/978-3-030-12988-0_5.

Roberto Barbuti, Nicoletta De Francesco, Antonella Santone, and Luca Tesei. "A Notion of Non-Interference for Timed Automata". In: *Fundamenta Informaticae* 51.1-2 (2002), pp. 1–11.

References II

Gilles Benattar, Franck Cassez, Didier Lime, and Olivier H. Roux. "Control and synthesis of non-interferent timed systems". In: *International Journal of Control* 88.2 (2015), pp. 217–236. DOI: 10.1080/00207179.2014.944356.

Roberto Barbuti and Luca Tesei. "A Decidable Notion of Timed Non-Interference". In: *Fundamenta Informaticae* 54.2-3 (2003), pp. 137–150.

Léa Fanchon and Florent Jacquemard. "Formal Timing Analysis Of Mixed Music Scores". In: *ICMC* (Aug. 12–16, 2013). Perth, Australia: Michigan Publishing, Aug. 2013.

Laurent Fribourg, David Lesens, Pierre Moro, and Romain Soulat. "Robustness Analysis for Scheduling Problems using the Inverse Method". In: *TIME* (Sept. 12–14, 2012). Leicester, UK: IEEE Computer Society Press, Sept. 2012, pp. 73–80. DOI: 10.1109/TIME.2012.10.

Guillaume Gardey, John Mullins, and Olivier H. Roux. "Non-Interference Control Synthesis for Security Timed Automata". In: *Electronic Notes in Theoretical Computer Science* 180.1 (2007), pp. 35–53. DOI: 10.1016/j.entcs.2005.05.046.

Aleksandra Jovanović, Didier Lime, and Olivier H. Roux. "Integer Parameter Synthesis for Real-Time Systems". In: *IEEE Transactions on Software Engineering* 41.5 (2015), pp. 445–461. DOI: 10.1109/TSE.2014.2357445.

References III

Paul C. Kocher. "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems". In: *CRYPTO* (Aug. 18–22, 1996). Vol. 1109. LNCS. Santa Barbara, California, USA: Springer, 1996, pp. 104–113. DOI: 10.1007/3-540-68697-5_9.

Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. "UPPAAL in a Nutshell". In: *International Journal on Software Tools for Technology Transfer* 1.1-2 (1997), pp. 134–152. DOI: 10.1007/s100090050010.

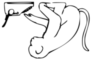
Lars Luthmann, Andreas Stephan, Johannes Bürdek, and Malte Lochau. "Modeling and Testing Product Lines with Unbounded Parametric Real-Time Constraints". In: *SPLC, Volume A* (Sept. 25–29, 2017). Sevilla, Spain: ACM, 2017, pp. 104–113. DOI: 10.1145/3106195.3106204.

Licensing

Source of the graphics used I

Title: Smiley green alien big eyes (aaah)
Author: LadyofHats
Source: `https://commons.wikimedia.org/wiki/File:Smiley_green_alien_big_eyes.svg`
License: public domain

Title: Smiley green alien big eyes (cry)
Author: LadyofHats
Source: `https://commons.wikimedia.org/wiki/File:Smiley_green_alien_big_eyes.svg`
License: public domain



Title: Typing monkey
Author: KaterBegemot
Source: `https://commons.wikimedia.org/wiki/File:Typing_monkey.svg`
License: CC by-sa 3.0

License of this document

Author: **Étienne André**