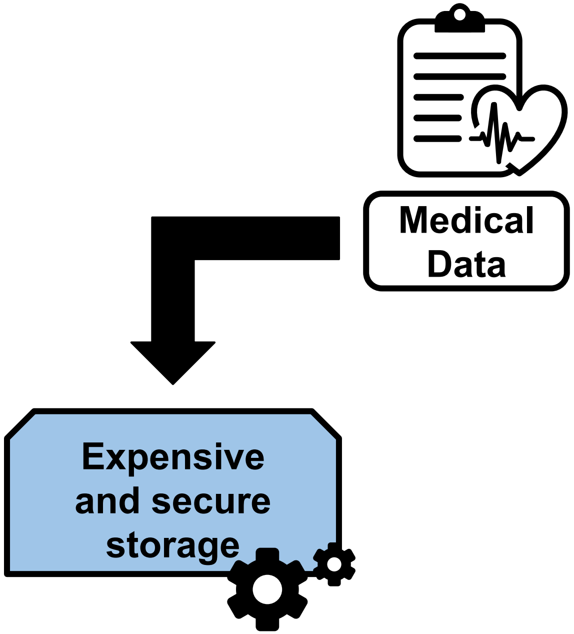
Verifying Opacity of Discrete Timed Automata

International Conference on Formal Methods in Software Engineering
April 14, 2024

Julian Klein, Paul Kogel, Sabine Glesner

SOFTWARE AND EMBEDDED SYSTEMS ENGINEERING

Motivation

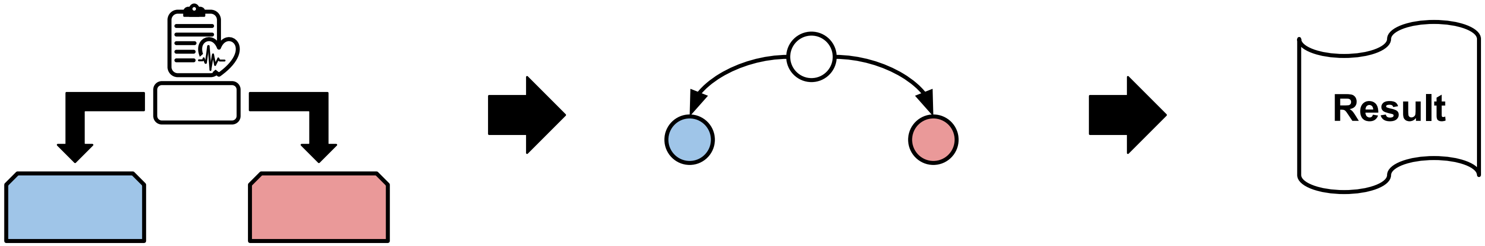

Medical Data
Expensive and secure storage
Cheap and vulnerable storage

Motivation

- Opacity to guarantee confidentiality
- Formal verification requires accurate model (TA)

Problem: Opacity verification of TA is **undecidable**

➔ **Restriction required**



Real System | **Formal (Timed) Model** | **Verification**

Goal

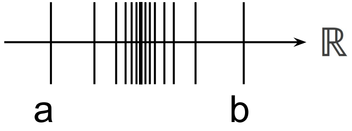
Goal: Verify opacity of TA with **minimal** restrictions

Criteria:

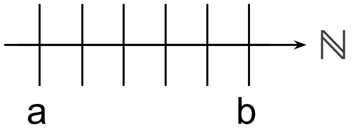
- **Safe**: no restrictions on the notion of opacity
- **Applicable**: no restrictions on the class of TA
- **Scalable**: large TA can be verified in reasonable time

Key Idea: Discrete-time setting to make opacity verification problem decidable.

Dense Time

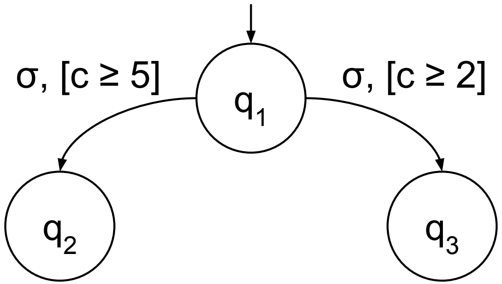


Discrete Time



Background: Opacity

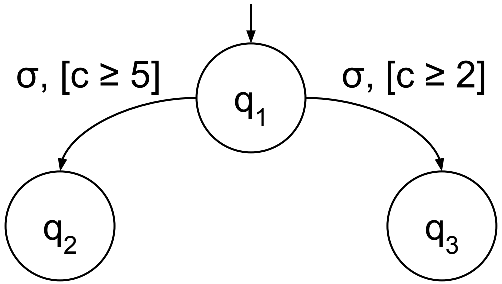
S:



Assumption:
Attacker knows structure of **S**

Background: Opacity

S:



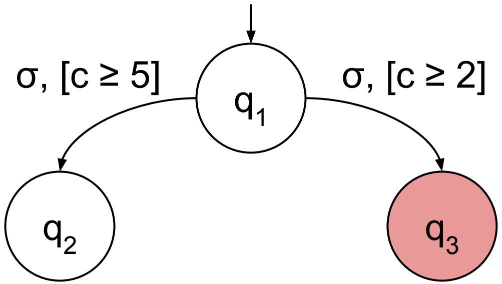
Assumption:
Attacker knows structure of **S**

- Multiple notions of opacity in the literature

➔ Equivalent in our setting

Background: Opacity

S:



Assumption:
Attacker knows structure of **S**

- Multiple notions of opacity in the literature

➔ Equivalent in our setting

Example: Current-State Opacity (CSO)

Question: Is the current state a secret state?

$(\sigma, 7) \rightarrow q_2$ and q_3 can be active

$(\sigma, 2) \rightarrow$ only q_3 can be active!

Related Work

- Literature considers only **dense time** setting
- Two restrictions to overcome undecidability:
  - Weaken notion of opacity
  - Restrict class of TA

Authors	Approach	Applicable	Safe
André et al, 2023 **[3]**	Measure only total runtime	✔	✘
Ammar et al, 2021 **[4]**	Time is bounded	✔	✘
Zhang, 2024 **[5]**	Real Time Automata	✘	✔
Wang and Zhan, 2018 **[6]**	Real Time Automata	✘	✔
Marques et al, 2023 **[7]**	Real Time Automata	✘	✔
Li et al, 2021 **[8]**	Weighted Automata (weight = time instance)	✘	✔

Our Method

Two-Step Verification Method

TA

↓

Step 1: Transformation

Transform timed automaton (TA) to equivalent finite tick automaton (τ-FA)

↓

τ-FA

↓

Step 2: Verification

Employ known opacity verification approach for FA

↓

Result

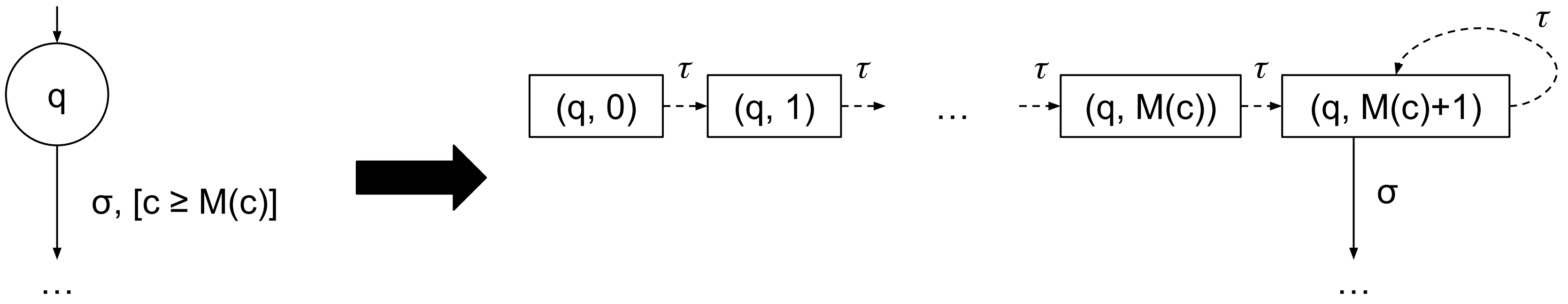
Transformation

- Transformation from single clock TA to τ-FA, Gruber et al. (2005) [1]
- Extension to TA with **arbitrary many** clocks

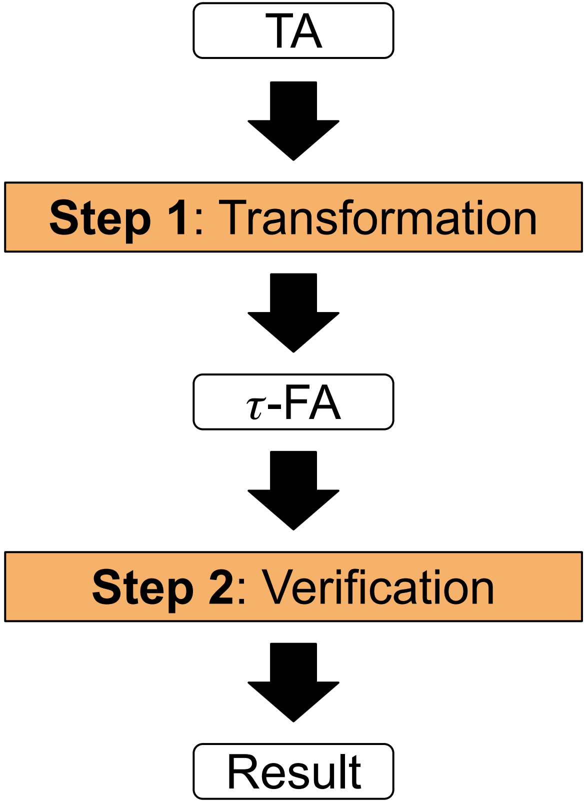
Idea: use standard region abstraction $\boldsymbol{\alpha_R}$:

- M(**c**) = largest constant that can be compared to clock **c**

➔ all k > M(**c**) cannot be distinguished



State Explosion



Problem:
Opacity verification scales **exponential**
➔ Small τ-FA required!

- $\boldsymbol{\alpha}_R$ produces more states than necessary

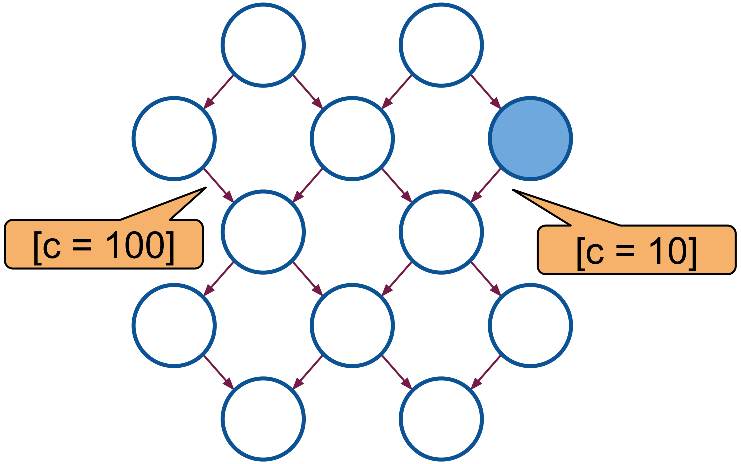
➔ Tighter time abstraction to improve scalability

Local Time Abstraction

Standard region abstraction α_R:

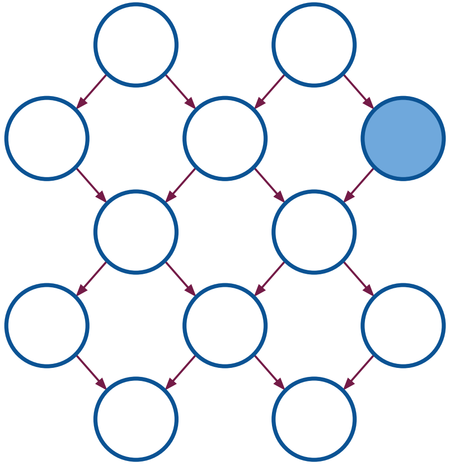
For a clock **c**

- Collect **all** locations in set **Q**
- Largest constant determines number of states



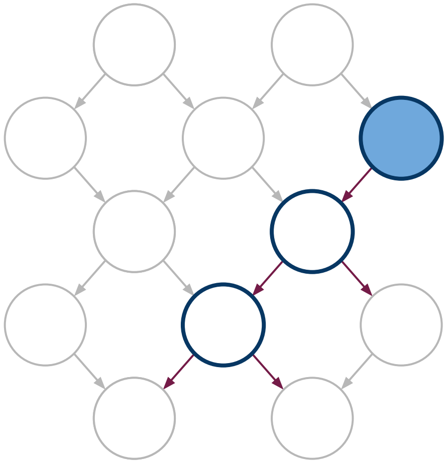
Local Time Abstraction

Standard region abstraction α_R:
For a clock **c**
- Collect **all** locations in set **Q**
- Largest constant determines number of states



Local time abstraction α_L:
For a clock **c**
- Collect **specific** locations in set **Q'**

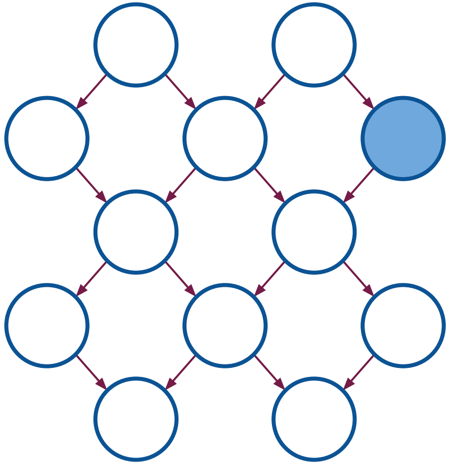
➔ Largest constant determines number of states



Local Time Abstraction

Standard region abstraction α_R:
For a clock **c**

- Collect **all** locations in set **Q**
- Largest constant determines number of states

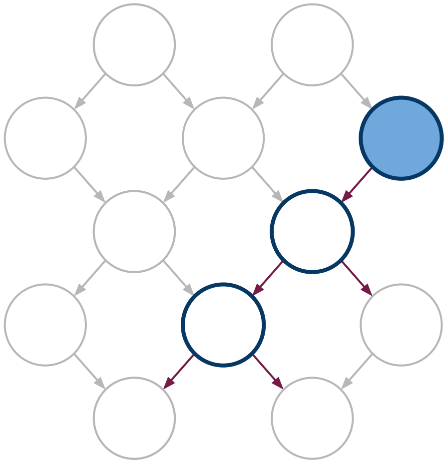


Collect **only** locations that can distinguish clock values of **c** (**Clock Reach**)

Local time abstraction α_L:
For a clock **c**

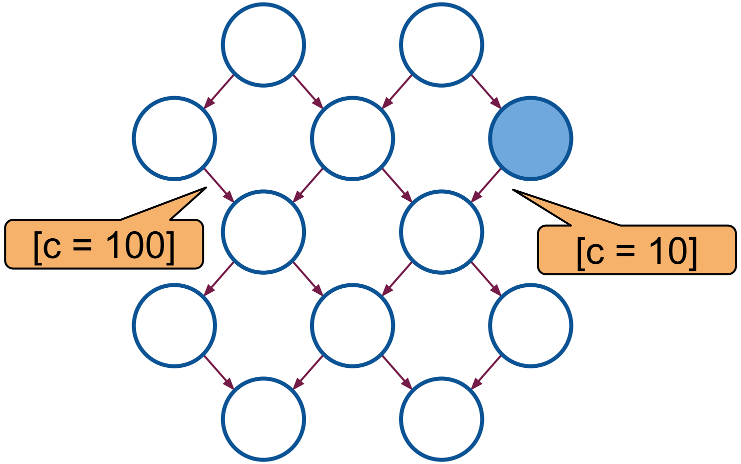
- Collect **specific** locations in set **Q'**

➔ Largest constant determines number of states



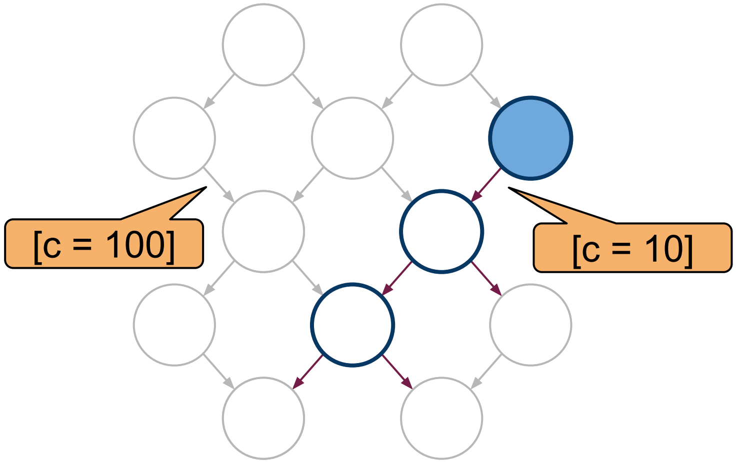
Local Time Abstraction

Standard region abstraction α_R:
For a clock **c**
- Collect **all** locations in set **Q**
- Largest constant determines number of states



Local time abstraction α_L:
For a clock **c**
- Collect **specific** locations in set **Q’**

➔ Largest constant determines number of states



Evaluation

Observation: Computation of α_L is more expensive than computation of α_R

Question 1: How significant is the **cost** of computing α_L?
Question 2: How significant are the **overall gains** due to the **state reduction** of α_L?

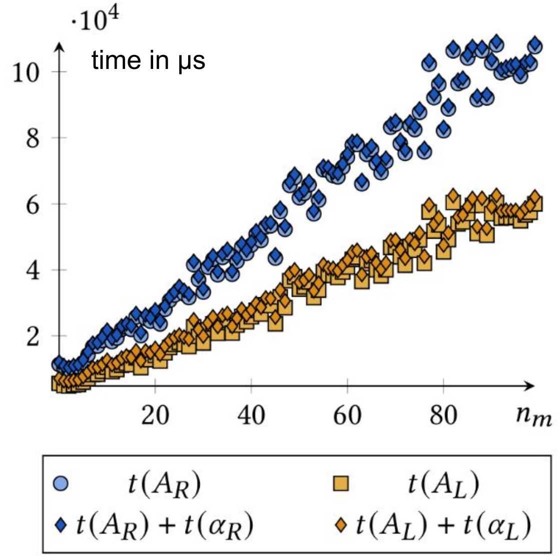
Randomized Systems

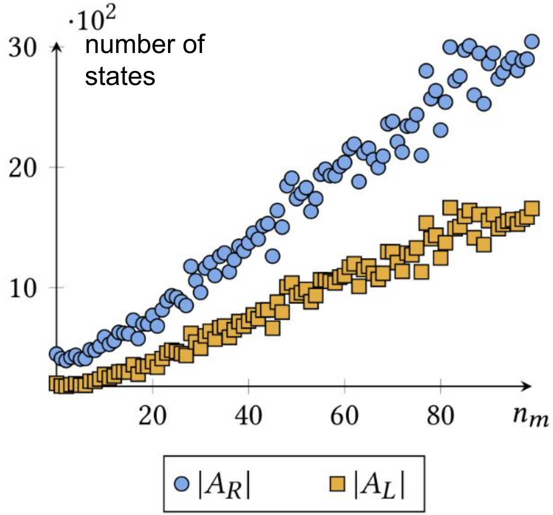
- Scaled by parameter
- No assumptions on system structure
- Average over all possible systems

Case Studies

- Realistic systems found in literature
- Logical structure
- Allow comparison in future works

Randomized Systems





Abbreviations:

- **n_m** = largest constant in any guard of TA
- **α** = time abstraction
- **A** = tick automaton

Standard region abstraction α_R

Local time abstraction α_L

Case Studies

Four realistic case studies:

- ➔ A_C : **Cloud service** to purchase products online **[3]**
- ➔ A_M : **Medical cloud** service to process patient data **[8]**
- ➔ A_S : **Sensor network** to locate agent in area **[9]**
- ➔ A_A : **ATM** with password authentication **[10]**

Case Study	Properties of case studies				Computation time of abstraction in µs		Number of states of the τ-FA		Time to verify Opacity in µs		Saved
	|L|	|Δ|	M	opaque	α_R	α_L	α_R	α_L	α_R	α_L	
A_C	18	20	21	yes	0.34	3.43	1301	175	216.38	29.82	**-86.22%**
A_M	8	10	10	no	0.29	0.55	96	49	22.06	9.83	**-55.42%**
A_S	25	64	20	yes	0.85	1.71	800	299	443.53	112.63	**-74.61%**
A_A	16	23	100	no	0.91	17.48	22909	12121	98782.03	37693.67	**-61.84**

Conclusion

Contributions:

- Novel algorithmic approach to verify opacity of timed automata

➔ Opacity verification is decidable in discrete time

- Local time abstraction to improve scalability of verification method by 55%-86%

Future Work:

- Explore more compact time models that avoid time step enumeration
- Investigate opacity enforcement techniques for our model

References (1/2)

- **[1]** Hermann Gruber, Markus Holzer, Astrid Kiehn, and Barbara König. 2005. On Timed Automata With Discrete Time - Structural And Language Theoretical Characterization. In International Conference On Developments In Language Theory. Springer, 272–283. https://doi.org/10.1007/11505877_24
- **[2]** Franck Cassez. 2009. The Dark Side Of Timed Opacity. In International Conference On Information Security And Assurance. Springer, 21–30. https://doi.org/10.1007/978-3-642-02617-1_3
- **[3]** Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. 2022. Guaranteeing Timed Opacity Using Parametric Timed Model Checking. ACM Transactions On Software Engineering And Methodology 31 (2022), 64–100. https://doi.org/10.1145/3502851
- **[4]** Ikhlass Ammar, Yamen El Touati, Moez Yeddes, and John Mullins. 2021. Bounded Opacity For Timed Systems. Journal Of Information Security And Applications 61 (2021), 2214–2126. https://doi.org/10.1016/j.jisa.2021.102926
- **[5]** Zhang, Kuize. State-based opacity of labeled real-time automata. *Theoretical Computer Science* 987 (2024), https://doi.org/10.1016/j.tcs.2023.114373
- **[6]** Lingtai Wang and Naijun Zhan. 2018. Decidability Of The Initial-State Opacity Of Real-Time Automata. In Symposium On Real-Time And Hybrid Systems. Springer, 44–60. https://doi.org/10.1007/978-3-030-01461-2_3

References (2/2)

- **[7]** Marques, Mariana Guimarães, Raphael Julio Barcelos, and João Carlos Basilio. The use of time-interval automata in the modeling of timed discrete event systems and its application to opacity. *IFAC-PapersOnLine* 56.2 (2023), https://doi.org/10.1016/j.ifacol.2023.10.042
- **[8]** Jun Li, Dimitri Lefebvre, Christoforos N Hadjicostis, and Zhiwu Li. 2021. Observers For A Class Of Timed Automata Based On Elapsed Time Graphs. IEEE Trans. Automat. Control 67, 2 (2021), 767–779. https://doi.org/10.1109/TAC.2021.3064542
- **[9]** Wen Zeng, Maciej Koutny, and Paul Watson. 2015. Opacity In Internet Of Things With Cloud Computing. In 2015 IEEE 8th International Conference On Service-Oriented Computing And Applications (SOCA). IEEE, 201–207. https://doi.org/10.1109/SOCA.2015.33
- **[10]** Anooshiravan Saboori. 2010. Verification And Enforcement Of State-Based Notions Of Opacity In Discrete Event Systems. dissertation. University Of Illinois At Urbana-Champaign. https://hdl.handle.net/2142/18226
- **[11]** Étienne André, Shapagat Bolat, Engel Lefaucheux, and Dylan Marinho. 2022. StrategFTO: Untimed Control For Timed Opacity. In Proceedings Of The 8th ACM SIGPLAN International Workshop On Formal Techniques For Safety-Critical Systems. 27–33. https://doi.org/10.1145/3563822.3568013
- **[12]** Kuize Zhang. 2023. A Unified Concurrent-Composition Method To State/Event Inference And Concealment In Labeled Finite-State Automata As Discrete-Event Systems. Annual Reviews in Control 56 (2023), 100902. https://doi.org/10.1016/j.arcontrol.2023.100902