IMITATOR
Formal Verification of Real-time Systems Under Uncertainty

Étienne André[1] Giuseppe Lipari[2] and Youcheng Sun[3]

[1]LIPN, CNRS UMR 7030, Université Paris 13, France
[2]CRIStAL – UMR 9189, Université de Lille, USR 3380 CNRS, France
[3]Scuola Superiore Sant'Anna, Pisa, Italy

Context: Formal Verification of Real-Time Systems

Critical systems involve timing constants and concurrency

- Bugs can be dramatic (risk of loss of lives or huge financial loss)


⇒ Need for formal verification

Problem: what if the system constants are uncertain or are not yet known?

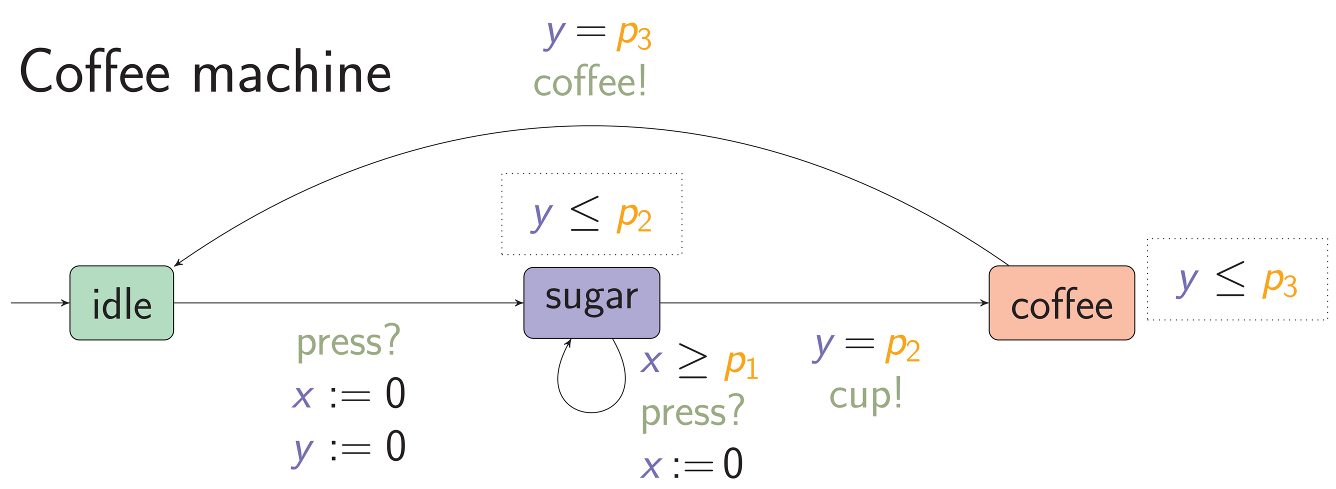
Solution: parametric verification

- Timing constants become parameters

Objective: derive values for these parameters ensuring the absence of bug (usually under the form of a set of constraints)

Parametric Timed Automata (PTA) [Alur et al., 1993]

- Finite automata (sets of locations and actions) extended with:
  - Clocks: real-valued variables evolving linearly
  - Parameters: unknown constants
- Example: Coffee machine



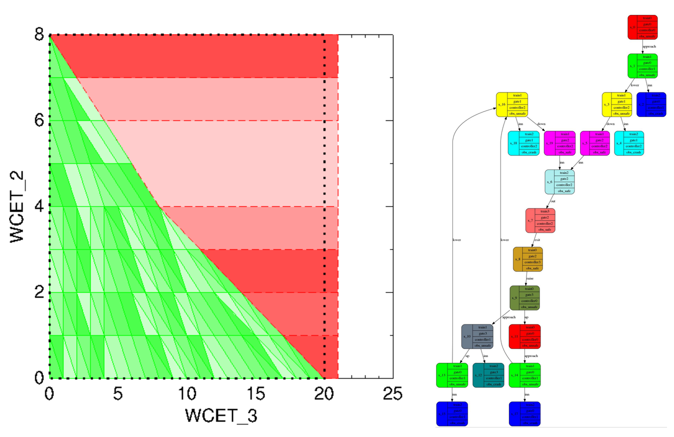
IMITATOR: Parameter Synthesis for Critical Systems

Input: a real-time system modeled by a network of PTA

Output: a constraint over the parameters guaranteeing the system correctness (e.g., non-reachability of some unsafe state)

Several algorithms:

- Non-reachability synthesis
- Parametric language preservation
- Behavioral cartography



Try IMITATOR! [André et al., 2012]

- Entirely written in OCaml
- Graphical outputs (behaviors, parameter constraints, etc.)
- Large repository of benchmarks
  - Asynchronous hardware circuits, scheduling problems, communication protocols, train controllers... and more!
- Available for free under the GNU-GPL license

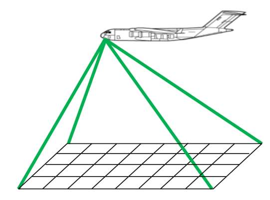
`www.imitator.fr`

What's next?

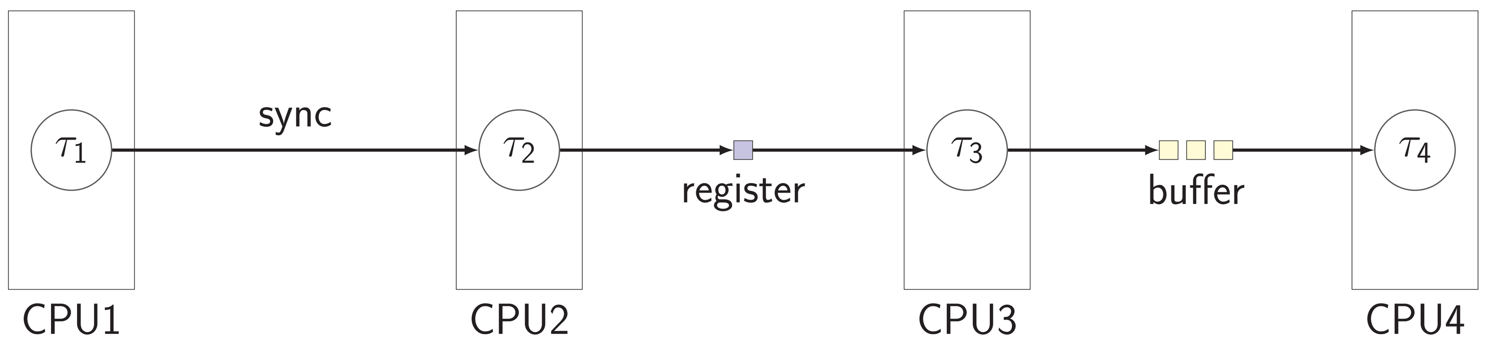
- Improved optimizations to address scalability
- Distributed and multi-core algorithms
- An input language for IMITATOR dedicated to real-time systems
  - Followed by a translation to PTA

A Case Study: The FMTV Challenge

- A problem proposed by Thales Research & Technology for the video capture in an aerial video system (2014)



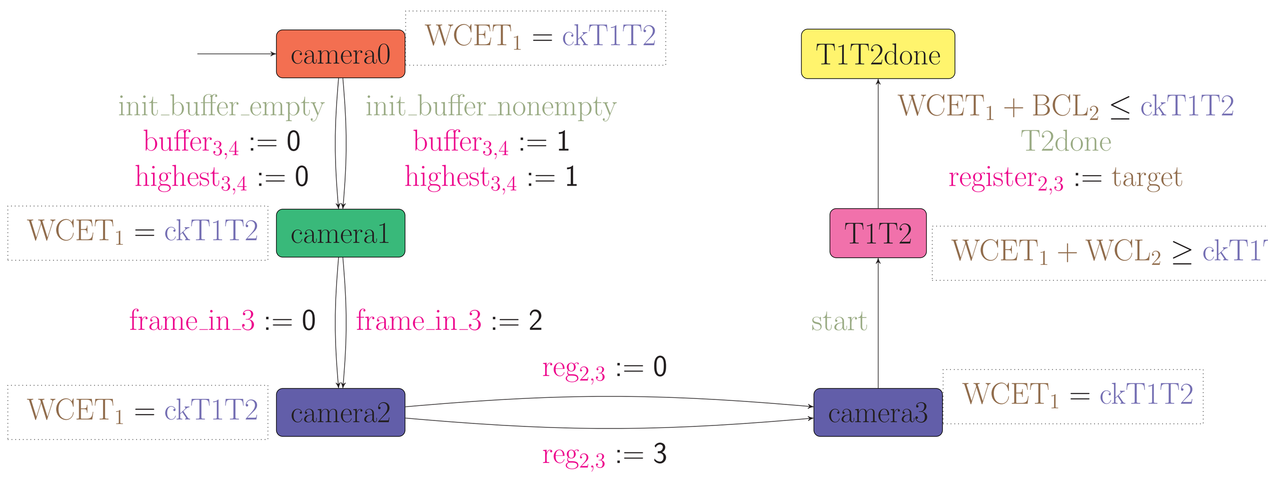
- A distributed video processing system (abstract view)



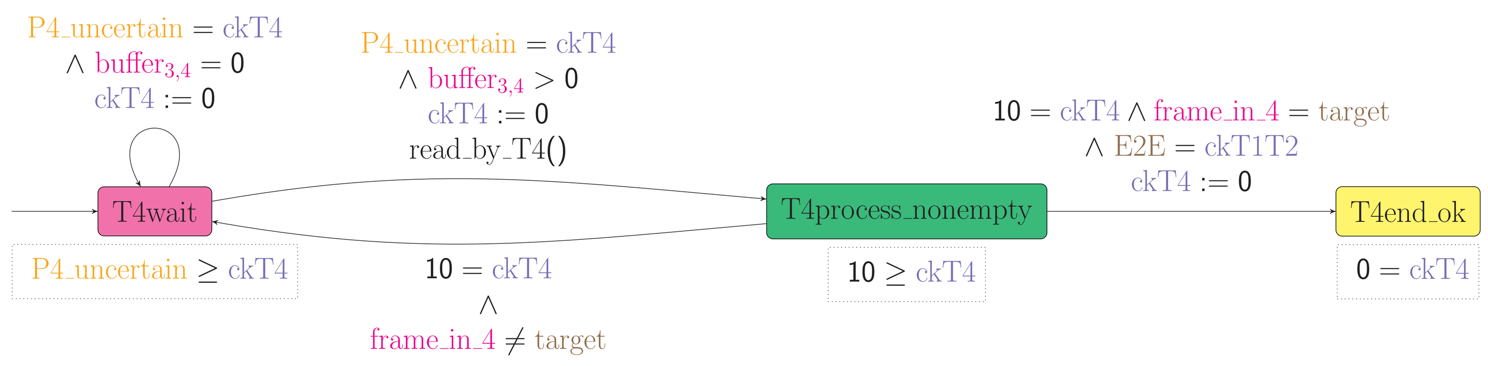
- τ_1, τ_3 and τ_4 are periodic tasks
  - The exact value for each task's period is constant but unknown
    - $P1 \in [40 - 0.004\,\text{ms}, 40 + 0.004\,\text{ms}]$
    - $P3 \in [\frac{40}{3} - \frac{1}{150}\,\text{ms}, \frac{40}{3} + \frac{1}{150}\,\text{ms}]$
    - $P4 \in [40 - 0.004\,\text{ms}, 40 + 0.004\,\text{ms}]$
- τ_2 is triggered by the completion of τ_1
- The FIFO buffer between τ_3 and τ_4 has a size $n = 1$ or $n = 3$
- **Challenge**: find the min/max end-to-end latency that a frame may experience in this system

Our Solution: Parametric Analysis [André et al., 2015]

- Task periods are modeled as parameters
  - E.g., $P4\_uncertain \in [40 - 0.004\,\text{ms}, 40 + 0.004\,\text{ms}]$
- Another parameter: the end-to-end latency E2E
  - To focus on the E2E of an arbitrary frame (denoted as target)
- Some of the PTA modeling the system (for $n = 1$)
  - The system status is initialized to be arbitrary so that the worst-case and best-case scenarios for E2E will be included



  - PTA model for task τ_4



- The end-to-end latency results returned by IMITATOR
  - $E2E \in [63\,\text{ms}, 145.008\,\text{ms}]$ (for $n = 1$)
  - $E2E \in [63\,\text{ms}, 225.016\,\text{ms}]$ (for $n = 3$)
- Runtime costs: 7.908 s with $n = 1$ and 115.247 s with $n = 3$

Conclusion

- Solved a problem with uncertain timing constants using parametric analysis, which turned out to be an efficient option

References

Alur, R., Henzinger, T. A., and Vardi, M. Y. (1993).
Parametric real-time reasoning.
In *STOC*, pages 592–601. ACM.

André, É., Fribourg, L., Kühne, U., and Soulat, R. (2012).
IMITATOR 2.5: A tool for analyzing robustness in scheduling problems.
In *FM*, volume 7436 of *Lecture Notes in Computer Science*, pages 33–36. Springer.

André, É., Lipari, G., and Sun, Y. (2015).
Verification of two real-time systems using parametric timed automata.
In *WATERS*.