Revisiting Digitization, Robustness, and Decidability for Timed Automata

Joël Ouaknine
(joint work with James Worrell, Tulane University)

SVC Presentation
February 4, 2003
Timed Automata

- Untimed automata with clocks.
- Timed trace semantics: sequences of events with non-decreasing real-valued timestamps. E.g., \( u = \langle (0.3, a), (2, b), (2, c), (3.1, a) \rangle \).

\[ [A] \triangleq \text{set of timed traces accepted by } A. \]

- Standard real-time modelling formalism.
Shortcomings

- PSPACE-complete emptiness problem ($\mathbb{[}A\mathbb{]} = \emptyset$) (Alur-Dill 94).
- Undecidable universality problem ($\mathbb{[}A\mathbb{]} = \mathbb{T}\mathbb{T}$) (idem).
- Excessive ‘precision’.

Various restrictions on timed automata proposed to remedy these points...
Digitization Techniques

Introduced by Henzinger-Manna-Pnueli 92.

- **Under appropriate conditions**, reduce dense-time language inclusion problems to discrete time:

  \[ [A] \subseteq [B] \iff \mathbb{Z}[A] \subseteq \mathbb{Z}[B]. \]

- Very successful and widespread. Useful in practice.
Digitization: An Example

```
error
```

```
2 < x < 3? b
```

```
y ≥ 4? a
```

```
Impl. ⊆ Spec.
```

```
error
```

```
2 ≤ x ≤ 3? b
```

```
y ≥ 4? a
```

```
a
```

```
a, b
```
Digitization: Prerequisites

- **Prerequisites**: Implementation must be *closed under digitization*,
  Specification must be *closed under inverse digitization*.

- Closure under digitization is **decidable**.

- Closure under inverse digitization is **undecidable**.
Are Timed Automata Too Expressive?

Example: Nuclear meltdown if in ‘hot’ state for strictly longer than 3s. Is the following system safe?

- ‘Infinite precision’ of timed automata also originally blamed for undecidability of universality problem.
- Require ‘safety margins’: make timed automata robust.
- Robustness also vital for ensuring the soundness and convergence of numerical approximation tools.
Robust Timed Automata

- What is robustness? If \( u \in []A[] \), then all timed traces ‘sufficiently close’ to \( u \) should also be in \( []A[] \).
  (If a behaviour is ‘safe’, small perturbations of it should also be safe.)

- Robustness corresponds to the removal of equality testing:
  - ‘Syntactic robustness’ \( \sim \) open timed automata.
  - ‘Semantic robustness’ \( \sim \) robust timed automata
    (Gupta-Henzinger-Jagadeesan 97).
The \(d\)-Topology

\[ u = \langle (t_1, a_1), \ldots, (t_m, a_m) \rangle, \quad u' = \langle (t'_1, a'_1), \ldots, (t'_n, a'_n) \rangle. \]

\[ d(u, u') = \infty, \text{ if } \langle a_1, \ldots, a_m \rangle \neq \langle a'_1, \ldots, a'_n \rangle, \]

\[ d(u, u') = \max\{|t_i - t'_i| : 1 \leq i \leq m\}, \text{ if } \text{untime}(u) = \text{untime}(u'). \]

Two traces are ‘close’ if they have the same sequence of events, occurring at neighbouring times.

(GHJ 97: All ‘reasonable’ metrics actually yield the same topology.)
The Robust Semantics for Timed Automata

A tube is a $d$-open set of timed traces.

The robust semantics assigns sets of tubes to timed automata, rather than sets of timed traces.

A tube $u$ is accepted if $\llbracket A \rrbracket$ is dense in $u$.

- Tube-emptiness problem is decidable (Gupta-Henzinger-Jagadeesan 97).
- It was believed that tube-universality might be decidable. Eventually disproved (Henzinger-Raskin 00).
- Current understanding is that robust semantics yields roughly same theory as standard semantics (idem for hybrid automata). Not so!
Convert Robust Semantics to Timed Traces

Can equivalently capture the robust semantics by considering only the largest accepted tube:

\[
\widehat{\lbrack A \rbrack} \equiv (\overline{\lbrack A \rbrack})^{\text{int}}.
\]

In this way, both \( \lbrack A \rbrack \) and \( \widehat{\lbrack A \rbrack} \) are sets of timed traces, and can directly be compared.
Robust vs. Open Timed Automata

Open timed automata have only strict inequalities (e.g., $x < 3$ rather than $x \leq 3$) as clock constraints.

- Open timed automata: **Syntactic** removal of equality.
- Robust timed automata: **Semantic** removal of equality.

Both types of automata are ‘acceptance-robust’: whenever they accept a trace, they also accept all sufficiently close neighbouring traces.

- Are their respective expressive powers comparable?
Robust vs. Standard: Incomparable Expressive Powers

- There exists a timed automaton $A$ such that, for every timed automaton $B$, $\lceil A \rceil \neq \lceil B \rceil$.

- (Also: There exists an open timed automaton $B$ such that, for every timed automaton $A$, $\lceil A \rceil \neq \lceil B \rceil$.)
Universality

- Undecidability of robust universality problem established by Henzinger-Raskin 00 (over strongly monotonic time).
  Universality of open timed automata left there as open question.

- Universality of open timed automata recently settled (OW 03):
  - Undecidable over strongly monotonic time.
  - Decidable over weakly monotonic time.

Strongly monotonic: time strictly increasing — no two events have same timestamp.

Weakly monotonic: time merely non-decreasing. Events can occur simultaneously.
Universality over Weakly Monotonic Time

Fact: open timed automata are closed under inverse digitization.

Universality: $\mathbb{T} = [\mathbb{A}] \iff \mathbb{T} \subseteq [\mathbb{A}] \iff \mathbb{Z}\mathbb{T} \subseteq \mathbb{Z}[\mathbb{A}]$?

But $\mathbb{Z}[\mathbb{A}]$ is regular! Thus decidable.

Robust timed automata are also closed under inverse digitization. Thus

$\mathbb{T} = \widehat{[\mathbb{A}]} \iff \mathbb{T} \subseteq \widehat{[\mathbb{A}]} \iff \mathbb{Z}\mathbb{T} \subseteq \mathbb{Z}[\widehat{\mathbb{A}}]$?

Yet robust universality (over weakly monotonic time) turns out to be . . . 

**undecidable**! What is going on?
Discrete Robust Languages Are Non-Regular!

It turns out that $\mathbb{Z}[\hat{A}]$ is (in general) not regular.

In particular, robust integral universality ($\mathbb{Z}[\hat{A}] = \mathbb{Z}^T$?) undecidable.

– Open question: is robust integral emptiness ($\mathbb{Z}[\hat{A}] = \emptyset$?) decidable?

(Recall: robust emptiness ($\hat{A} = \emptyset$?) is decidable.)
Summary

Digitization and robustness are important and well-studied topics.

- Closure under digitization **decidable**.
- Closure under inverse digitization **undecidable**.
- These two results **reversed** under the robust semantics.
- Expressive powers of robust and standard semantics **incomparable**.
- **Robust semantics much less tractable**: Undecidable (non-regular) discrete-time theory, contrary to standard semantics.
- Consequence: impossible to combine digitization techniques with robust semantics.
- Better introduce robustness explicitly — **syntactically**.
- Positive side: robust semantics is still **recursive**.
Future Work

- What about hybrid automata?
- Is robust integral emptiness decidable?