On the Distance between Timed Automata

Amnon Rosenmann

Graz University of Technology

rosenmann@math.tugraz.at

Introduction

- Timed automata (TA) were introduced by Alur and Dill (1994) as an abstract model for real-time systems by extending finite automaton with continuous clocks
- Language inclusion: The fundamental problem of inclusion of the language accepted by a timed automaton A (e.g., the implementation) in the language accepted by TA B (e.g., the specification) is undecidable when A, B are non-deterministic TA
- Decidability was shown to hold for various restricted and modificated TA
- Other investigations were of robustness in the language or perturbations in the clocks. However, by allowing a fixed imprecision, undecidability problems due to working over continuous time do not vanish

- Digitization of timed systems were studied e.g. by Henzinger et al., Ouaknine and Worrell to show when language inclusion is decidable
- Suman et al. showed that $\mathfrak{L}(A) \subseteq \mathfrak{L}(B)$ is decidable when B is reset only on integral time
- We go further with this approach. We work over discretized time, but without restricting or modifying the definition of a TA.
- The key ingredients:
  1. Add a clock that measures absolute time
  2. discretize over intervals that are fractions of a time unit
- We show how to effectively construct deterministic TA A_d, a discretization of A, s.t. A and and A_d differ from each other by at most $\frac{1}{6}$ time units on each occurrence of an event; similarly for B and B_d

- Language inclusion $\mathfrak{L}(A_d) \subseteq \mathfrak{L}(B_d)$ - decidable
- $\mathfrak{L}(A) \subseteq \overline{\mathfrak{L}(B)}$ - decidable, where $\overline{\mathfrak{L}(B)}$ is the closure of $\mathfrak{L}(B)$ in the Euclidean topology

Specifically:

- $\mathfrak{L}(A_d) \nsubseteq \mathfrak{L}(B_d) \Rightarrow \mathfrak{L}(A) \nsubseteq \mathfrak{L}(B)$
- $\mathfrak{L}(A_d) \subseteq \mathfrak{L}(B_d) \Rightarrow \mathfrak{L}(A) \subseteq \overline{\mathfrak{L}(B)}$

- The next natural question, in case $\mathfrak{L}(A) \not\subseteq \mathfrak{L}(B)$, is how far away are the timed traces of $\mathfrak{L}(A)$ from those of $\mathfrak{L}(B)$: e.g. the implementation may not meet the ideal specification model, but we need to check that the time imprecision is bounded, or we may check that the imprecision is within a safety time zone
- For that matter we define the distance between the languages of timed automata as the limit on how far away a timed trace of one timed automaton can be from the closest timed trace of the other timed automaton
- We show how one can decide whether the distance between two TA is finite or infinite for a (perhaps) restricted version of the problem

Definition (Non-deterministic timed automaton)

A non-deterministic timed automaton is a tuple $(\mathcal{Q}, q_0, \Sigma, \mathcal{F}, \mathcal{C}, \mathcal{T})$:

- $\mathcal{Q}$ - a finite set of locations, q_0 - the initial location
- $\mathcal{F} \subseteq \mathcal{Q}$ - the set of accepting locations
- Σ - a finite set of transition labels, or actions
- $\mathcal{C}$ - a finite set of clocks
- $\mathcal{T} \subseteq \mathcal{Q} \times \Sigma \times \mathcal{G} \times \mathcal{P}(\mathcal{C}) \times \mathcal{Q}$ - a finite set of transitions $(q, a, g, \mathcal{C}_{rst}, q')$:
    - $q, q' \in \mathcal{Q}$ - the source and the target locations, respectively
    - $a \in \Sigma_\epsilon$ - the transition action
    - $g \in \mathcal{G}$ - the transition guard
    - $\mathcal{C}_{rst} \subseteq \mathcal{C}$ - the clocks to be reset

Let

- $v : \mathcal{C} \to \mathbb{R}_{\geq 0}$ - a clock valuation
- $\mathcal{V}$ - the set of all clock valuations

Definition (The semantics of a TA)

The semantics of a TA A is the timed transition system $[\![A]\!] = (S, s_0, \mathbb{R}_{\geq 0}, \Sigma, T)$:

- $S = \{(q, v) \in \mathcal{Q} \times \mathcal{V}\}$ - the set of states, $s_0 = (q_0, \mathbf{0})$ - the initial state
- $T \subseteq S \times (\Sigma \cup \mathbb{R}_{\geq 0}) \times S$ - the transition relation:
  - Timed transitions (delays): $(q, v) \xrightarrow{d} (q, v + d)$, $d \in \mathbb{R}_{\geq 0}$
  - Discrete transitions (jumps): $(q, v) \xrightarrow{a} (q', v')$, $a \in \Sigma$ where there exists a transition $(q, a, g, \mathcal{C}_{rst}, q')$ in $\mathcal{T}$, such that the valuation v satisfies the guard g and $v' = v[\mathcal{C}_{rst}]$

Definition (Run)

A (finite) run ϱ of a TA A - a sequence of alternating timed and discrete transitions:

$$(q_0, \mathbf{0}) \xrightarrow{d_1} (q_0, \mathbf{d}_1) \xrightarrow{a_1} (q_1, v_1) \xrightarrow{d_2} \cdots \xrightarrow{d_k} (q_{k-1}, v_{k-1} + d_k) \xrightarrow{a_k} (q_k, v_k)$$

Definition (Timed trace)

A timed trace (timed word) - a sequence of pairs:

$$\lambda = (t_1, a_1), (t_2, a_2), \ldots, (t_k, a_k),$$

with $a_i \in \Sigma$ and $t_i = \Sigma_{j=1}^{i} d_i$

Definition (Language)

The language $\mathfrak{L}(A)$ - the set of accepted timed traces of A

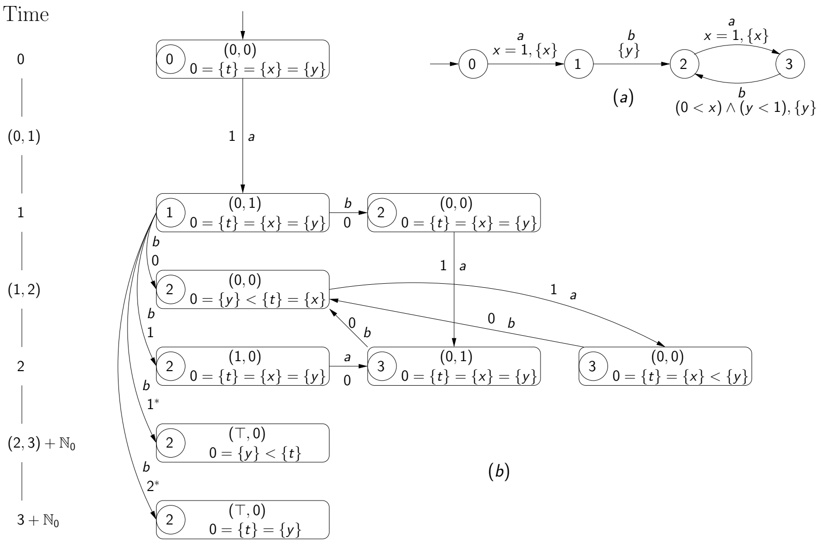
Augmented region automaton

- The region automaton $\mathfrak{R}(A)$ is a finite discretized version of A, such that time is abstracted and both automata define the same untimed language
- Instead of looking at the clocks-space as a continuous space it is partitioned into regions, each region is characterized by the integral values of the clocks and the ordering of their fractional parts
- In order to be able to compare the language of two TA we augment $\mathfrak{R}(A)$ with the clock t that is never reset, obtaining $\mathfrak{R}^t(A)$
- We can do without the integral part of t, instead - each edge of $\mathfrak{R}^t(A)$ is assigned a 'weight', the time difference in the integral part of t between the target and the source regions

Definition (Augmented region automaton)

- V - the set of vertices $(q, \mathbf{n}, \Delta)$ without the integral part of t, q - a location of A , $r = (\mathbf{n}, \Delta)$ - a region, with $\mathbf{n}$ the integral parts of the clocks and Δ - a simplex
- v_0 - the initial vertex
- E - the set of labeled edges: $(q, r) \xrightarrow{a} (q', r') \in E$ iff $\exists$ a run of A containing $(q, v) \xrightarrow{d} (q, v + d) \xrightarrow{a} (q', v')$, where v - clock valuation belonging to region r and similarly with v', r', when ignoring the integral part of the time measured by t
- Σ - the set of actions
- W^* - the set of weights on the edges: $m = \lfloor t_1 \rfloor - \lfloor t_0 \rfloor \in [0..M]$, $m^* := m, m+1, m+2, \ldots$

Example: Augmented region automaton



Definition (Discretized timed automaton)

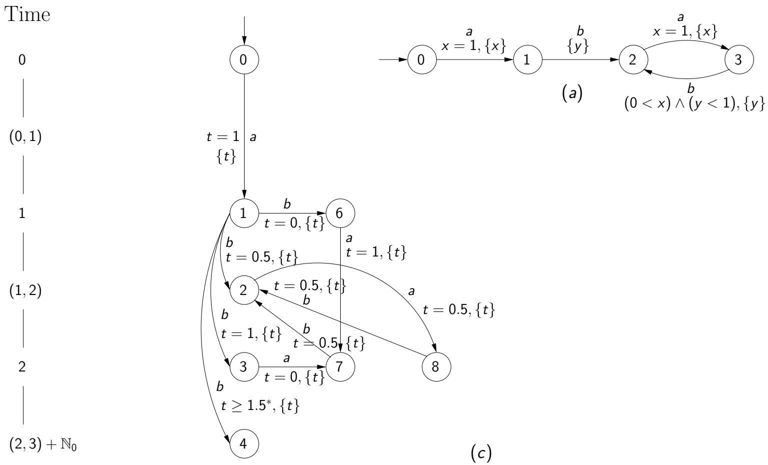
A discretized timed automaton A_d is a deterministic TA constructed from the augmented region automaton $\mathfrak{R}^t(A)$ in the following way.

- Locations, edges and actions - as in $\mathfrak{R}^t(A)$
- Clock t which is reset on each transition
- Transition guards: let $e = v_0 \to v_1$ be an edge of $\mathfrak{R}^t(A)$ with weight $w(e)$ and let $\{t_0\}, \{t_1\} \in [0, 1)$ be any fractional parts of t in the source and target regions. Let

$$\delta = \frac{1}{2}(\lceil\{t_1\}\rceil - \lceil\{t_0\}\rceil) \in \{-\frac{1}{2}, 0, \frac{1}{2}\},$$

  where $\lceil\{t_i\}\rceil \in \{0, 1\}$. Then the transition guard of the corresponding edge of A_d is $t = w(e) + \delta$
  - If $w(e) = m^*$ then the transition guard is $t \geq m + \delta$

Example: Discretized timed automaton



Definition (Distance between timed traces)

We define the ∞-metric or max-metric d on a set T of timed traces: given timed traces

$$\tau_1 = (t_1^{\tau_1}, a_1^{\tau_1}), (t_2^{\tau_1}, a_2^{\tau_1}), \ldots, (t_m^{\tau_1}, a_m^{\tau_1}),$$
$$\tau_2 = (t_1^{\tau_2}, a_1^{\tau_2}), (t_2^{\tau_2}, a_2^{\tau_2}), \ldots, (t_n^{\tau_2}, a_n^{\tau_2}),$$

the distance between τ_1 and τ_2 is

$$d(\tau_1, \tau_2) = \begin{cases} \infty, & \text{if } m \neq n \text{ or } a_i^{\tau_1} \neq a_i^{\tau_2} \text{ for some } i, \\ \max_i |t_i^{\tau_1} - t_i^{\tau_2}|, & \text{otherwise.} \end{cases}$$

Definition (Conformal distance between timed languages)

- Given two timed languages $\mathfrak{L}_1$ and $\mathfrak{L}_2$, $\mathfrak{L}_1$ is ε-included in $\mathfrak{L}_2$, denoted $\mathfrak{L}_1 \subseteq_\varepsilon \mathfrak{L}_2$, if for every timed trace $\tau_1 \in \mathfrak{L}_1$ there exists a timed trace $\tau_2 \in \mathfrak{L}_2$ such that $d(\tau_1, \tau_2) \leq \varepsilon$
- The conformance distance $c(\mathfrak{L}_1, \mathfrak{L}_2)$ between $\mathfrak{L}_1$ and $\mathfrak{L}_2$ is

$$c(\mathfrak{L}_1, \mathfrak{L}_2) = \inf\{\varepsilon : \mathfrak{L}_1 \subseteq_\varepsilon \mathfrak{L}_2\},$$

  that is,

$$c(\mathfrak{L}_1, \mathfrak{L}_2) = \sup_{\tau_1 \in \mathfrak{L}_1} \inf_{\tau_2 \in \mathfrak{L}_2} d(\tau_1, \tau_2) = \sup_{\tau_1 \in \mathfrak{L}_1} d(\tau_1, \mathfrak{L}_2)$$

- The distance $d(\mathfrak{L}_1, \mathfrak{L}_2)$ between $\mathfrak{L}_1$ and $\mathfrak{L}_2$ is

$$d(\mathfrak{L}_1, \mathfrak{L}_2) = \max\{c(\mathfrak{L}_1, \mathfrak{L}_2), c(\mathfrak{L}_2, \mathfrak{L}_1)\}$$

- Subadditivity: $c(\mathfrak{L}_1, \mathfrak{L}_3) \leq c(\mathfrak{L}_1, \mathfrak{L}_2) + c(L_2, L_3)$

- When an event in a run on A occurs at time $t_0 \in \mathbb{N}_0$ then the corresponding event on A_d occurs also at t_0
- When $t_0 = n + \varepsilon$, $n \in \mathbb{N}_0$, $0 < \varepsilon < 1$ then on A_d it occurs at $n + \frac{1}{2}$
- Since the clock t of A_d is synchronized with that added to A the cumulative error does not increase and A_d is a $\frac{1}{2}$-time-unit approximation of A: there exits a surjective mapping

$$\pi : \mathfrak{L}(A) \twoheadrightarrow \mathfrak{L}(A_d),$$

such that if $\pi(\tau) = \tilde{\tau}$ then $d(\tau, \tilde{\tau}) < \frac{1}{2}$

Theorem

$d(\mathfrak{L}(A), \mathfrak{L}(A_d)) \leq \frac{1}{2}$

- Since t resets in $\frac{1}{2}\mathbb{N}_0$ then A_d is determinizable
- Since t resets at each transition, we can remove it altogether to obtain an action-labeled, weighted directed graph

The discretization interval

Let us look now at $\mathfrak{L}(A)$ and $\mathfrak{L}(B)$

Theorem

If A, B are TA then $c(\mathfrak{L}(A), \mathfrak{L}(B)) \in \frac{1}{2}\mathbb{N}_0 \cup \{\infty\}$.

- In order to compute distances between TA we need to make the basic discretization interval Δ smaller
- When $\Delta = \frac{1}{n}$ we get: $d(\mathfrak{L}(A), \mathfrak{L}(A_d)) \leq \frac{1}{n}$ (in the expense of complexity)
- However, it turns out that it suffices to choose $\Delta = \frac{1}{6}$ in order to get the maximal precision for $d(\mathfrak{L}(A), \mathfrak{L}(B))$

The conformance distance

Theorem

With $\Delta = \frac{1}{6}$ *we have*

- $|\, c(\mathfrak{L}(A), \mathfrak{L}(B)) - c(\mathfrak{L}(A_d), \mathfrak{L}(B_d))\, | \leq \frac{1}{6}$
- *If* $c(\mathfrak{L}(A_d), \mathfrak{L}(B_d))$ *is known then* $c(\mathfrak{L}(A), \mathfrak{L}(B))$ *is known*

In particular:

- $\mathfrak{L}(A_d) \nsubseteq \mathfrak{L}(B_d) \;\Rightarrow\; \mathfrak{L}(A) \nsubseteq \mathfrak{L}(B)$
- $\mathfrak{L}(A_d) \subseteq \mathfrak{L}(B_d) \;\Rightarrow\; \mathfrak{L}(A) \subseteq \overline{\mathfrak{L}(B)}$
- *The language inclusion problem between* $\mathfrak{L}(A)$ *and the topological closure of* $\mathfrak{L}(B)$ *is decidable*

Computing the conformance distance - a game approach

- we need to compute $c(\mathfrak{L}(A_d), \mathfrak{L}(B_d))$
- The general goal in computing $c(\mathfrak{L}(A_d), \mathfrak{L}(B_d))$ is to find the timed trace of $\mathfrak{L}(A_d)$ that is the farthest from $\mathfrak{L}(B_d)$ (or a sequence of such timed traces if the distance is ∞)
- First, we determinize A_d
- A heuristic approach is to play a timed game in which the player in white moves along A_d and tries to maximize her wins, while the player in black moves along B_d and tries to minimize his losses. One strategy to cope with the complexity of the game is a greedy max-min algorithm

Infinite distance

Let consider a seemingly easier question: is $c(\mathfrak{L}(A_d), \mathfrak{L}(B_d)) = \infty$? An infinite conformal distance occurs the following situations:

S1. The untimed language of A_d is not included in that of B_d: there exists a path $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n$ in A_d, with q_n an accepting location, which either cannot be realized in B_d with the same sequence of actions, or all such paths in B_d do not terminate in an accepting location

S2. There exists a path in A_d of the form $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n$, where the transition $q_{n-1} \xrightarrow{a_n} q_n$ has guard $t \geq m$, whereas for any path in B_d of the form $q'_0 \xrightarrow{a_1} q'_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q'_n$ the guard of the last transition $q'_{n-1} \xrightarrow{a_n} q'_n$ bounds t from above

S3. For each $N \in \mathbb{N}$ there exists a timed trace $\tau \in \mathfrak{L}(A_d)$, such that for each $\sigma \in \mathfrak{L}(B_d)$, $d(\tau, \sigma) > N$ and not because of S2

Decidability of S1 and S2

We want to check whether $c(\mathfrak{L}(A_d), \mathfrak{L}(B_d)) = \infty$ because of **S1** or **S2**:

- Extend A_d and (and, similarly, B_d) by adding $\bar{\Sigma} = \{\bar{a} : a \in \Sigma\}$, a copy of Σ
- For each transition $q \xrightarrow{a} q'$ of A_d with time constraint $t \geq m$, add a transition $q \xrightarrow{\bar{a}} q'$ with guard $t = \infty$
- Complete B_d by adding a 'sink'
- Construct the untimed automaton $U(A_d)$, a determinization of A_d with respect to actions (ignoring time) and similarly $U(B_d)$
- Construct $U(A_d) \times U(B_d)$ with accepting locations (Q, Q'), where $Q \in U(A_d)$ - accepting and $Q' \in U(B_d)$ - not accepting

Theorem

$c(\mathfrak{L}(A_d), \mathfrak{L}(B_d)) = \infty$ because of **S1** *or* **S2** *iff $U(A_d) \times U(B_d)$ contains an accepting location*

Checking S3

- Assume that options **S1** or **S2** were ruled out and we want to check **S3**
- The goal is to find a sequence of traces in A_d which 'run-away' from B_d, and now we are interested in the exact delays between consecutive transitions
- This problem may be of very high complexity and even it is not clear whether it is decidable
- It can be shown that the following is decidable:

- Extend A_d and B_d with actions $\bar{\Sigma}$ as before, referring to transitions that are unbounded by time
- In order to compare each timed trace of A_d simultaneously with all (untimed) equivalent time traces of B_d, determinize B_d to $D(B_d)$, so that the transitions of $D(B_d)$ retain the set of transitions of B_d including source and target locations
- Construct the product automaton $A_d \times D(B_d)$ with at most $L = |\mathcal{Q}^{A_d}| \cdot 2^{|\mathcal{Q}^{B_d}|}$ locations, where each location is of the form

$$Q^{A_d \times D(B_d)} = (q^{A_d}, \{q_1^{B_d}, \dots, q_m^{B_d}\})$$

- At each transition of a run on $A_d \times D(B_d)$ we subtract the delay of the edge of A_d from each of the delays of the corresponding edges of B_d and record at each location $q_i^{B_d}$ of $Q^{A_d \times D(B_d)}$ the set of accumulated time differences (ATDs)

- Every run ρ on $A_d \times D(B_d)$ can be uniquely written in the form

$$\rho = \rho_0 \sigma_1^{i_1} \rho_1 \sigma_2^{i_2} \rho_2 \cdots \sigma_r^{i_r} \rho_r,$$

  for some $r \in \mathbb{N}_0$, $i_j \in \mathbb{N}$ and where each σ_j is a simple cycle of positive length and each ρ_j is without cycles and of length $0 \leq l < L$
- We say that the number of power cycles of ρ is r, written $pc(\rho) = r$

Theorem

It is decidable whether there exists a fixed $K \in \mathbb{N}$, such that for every $N \in \mathbb{N}$ there exists a timed trace $\tau \in \mathfrak{L}(A_d)$, such that $d(\tau, \mathfrak{L}(B_d)) > N$ and the corresponding run ρ on $A_d \times D(B_d)$ satisfies $pc(\rho) \leq K$.

- We emphasize that K is not given in advance
- The proof of the theorem is by constructing an algorithm that contains an iterated process and guarantees to terminate

Conclusion

- We introduced a natural definition of the distance between the languages of TA based on the accumulated time difference between TA that are supposed to be (almost) the same or that one TA confors to the other
- We can effectively construct discretized deterministic TA and obtain the distance between the original TA from the distance between the discretized versions
- The problem of language inclusion $\mathfrak{L}(A) \subseteq \mathfrak{L}(B)$, undecidable in general, is decidable for $\mathfrak{L}(A) \subseteq \overline{\mathfrak{L}(B)}$
- It is decidable whether the distance between TA is finite or infinite for a restricted(?) version of the problem

Suggested future work

- Is finiteness of the distance decidable? What other problems are decidable/undecidable?
- Special cases on which the accumulated distance can be computed

Other possible definitions of the distance between TA:

- A maximal time difference on a single transition - easy
- Time difference mean on simple cycles - easy
- If the TA are equipped with probabilities on transitions then compute the accumulated distance or expected value of absolute differences with respect to these probabilities