The Timestamp of Timed Automata

Amnon Rosenmann

Graz University of Technology

rosenmann@math.tugraz.at

Introduction

- Timed automata (TA) are finite automata extended with clocks that measure the time that elapsed since past events in order to control the triggering of future events
- Defined [Alur and Dill, 1994] as an abstract model of real-time systems
- A fundamental problem is the reachability problem: is a given location of a TA reachable from the initial location?
- The reachability problem was shown to be decidable (of complexity PSPACE-complete) [Alur and Dill, 1994] through the construction of a region automaton
- We generalize the reachability problem: we show that the problem of computing the set of all time values on which any transition occurs (and thus, a location is reached) is solvable

Main results

- Given a non-deterministic timed automaton with silent transitions A, we effectively compute its timestamp: the set of all pairs (time value, action) of all observable timed traces of A
- The timestamp is in the form of a union of action-labeled intervals with integral end-points and is eventually periodic
- One can compute a simple deterministic timed automaton with the same timestamp as that of A
- Partial method, not bounded by time or number of steps, for the general language non-inclusion problem for timed automata
- The language of A is periodic with respect to suffixes

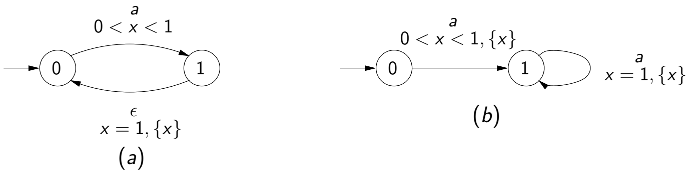
Example (A non-determinizable TA and its timestamp)

- The TA in figure (a) is non-determinizable and its language is

$$\mathfrak{L}(A) = \{(0+\delta_0, a), \cdots, (k+\delta_k, a) : \ k \in \mathbb{N}_0, 0 < \delta_i < 1\}$$

- The TA in figure (b) is deterministic and has the same timestamp:

$$\mathbb{R}_{\geq 0} \setminus \mathbb{N}_0$$



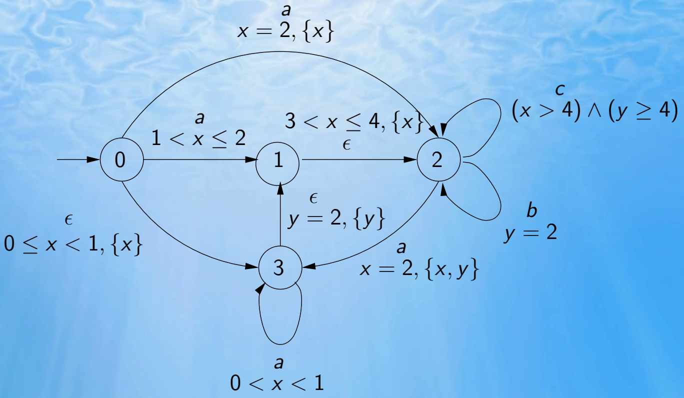
Non-deterministic timed automaton - definition

Definition (Timed automaton)

A non-deterministic timed automaton with silent transitions is a tuple $(\mathcal{Q}, q_0, \Sigma_\epsilon, \mathcal{C}, \mathcal{T})$:

- $\mathcal{Q}$ - a finite set of locations, q_0 - the initial location
- $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$ - a finite set of transition labels, or actions, Σ - observable, ϵ - silent
- $\mathcal{C}$ - a finite set of clocks
- $\mathcal{T} \subseteq \mathcal{Q} \times \Sigma_\epsilon \times \mathcal{G} \times \mathcal{P}(\mathcal{C}) \times \mathcal{Q}$ - 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

Example (Fishy)



The semantics of a TA

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

Definition (Semantics of a TA)

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

- $S = \{(q, v) \in \mathcal{Q} \times \mathcal{V}\}$ - the set of states, $s_0 = (q_0, \mathbf{0})$ - the initial state
- $T \subseteq S \times (\Sigma_\epsilon \cup \mathbb{R}_{\geq 0}) \times S$ - the transition relation:
  - Timed transitions (delays): $(q, v) \xrightarrow{d} (q, v + d)$, $d \in \mathbb{R}_{\geq 0}$
  - Discrete transitions (jumps): $(q, v) \xrightarrow{a} (q', v')$, $a \in \Sigma_\epsilon$ 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 $v' = v[\mathcal{C}_{rst}]$

Run, timed trace, language

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

Definition (Language)

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

The trail of a path

- In order to track the timestamp of an event along a path in the TA A with clocks $x_1, \cdots, x_s$ we first add a global clock t that displays absolute time
- A run along a path in A induces a trajectory in the non-negative part of the $tx_1 \cdots x_s$-space in direction $\mathbf{1}$, except for the projections during events with clocks reset
- The set of all runs along a given path forms a trail
- The trail is triangulated into symplices called regions
- Each region sits on the integral grid within a unit hyper-cube and defines a fixed ordering among the partial parts of the clocks and it has its immediate time-successor

The timestamp of an event

Definition (Timestamp of an event in a path)

The timestamp of an event in a path is the union of the timestamps (time, action) of that event of all runs along the path

Proposition

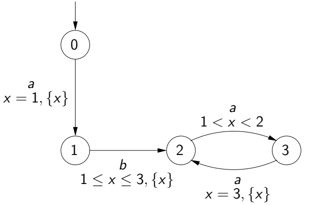
The timestamp of each event is a labeled interval between points m and n, $m \leq n$, $m \in \mathbb{N}_0$ and $n \in \mathbb{N} \cup \infty$

Proof.

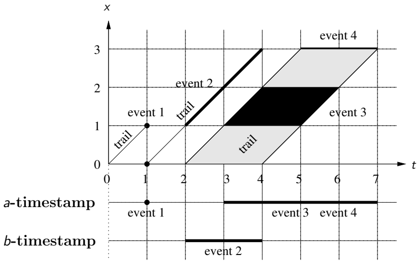
It suffices to show that the timestamp of a single simplex is of the required form.
Another proof is by representing events i by variables t_i and showing that max/min solutions of a corresponding linear programming problem has integer solutions. $\square$

Example (Trail, timestamp and regions of a path)

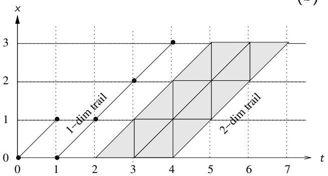
We look at the path: $(0) \xrightarrow{a} (1) \xrightarrow{b} (2) \xrightarrow{a} (3) \xrightarrow{a} (2)$



(a)



(b)



(c)

Infinite augmented region automaton - definition

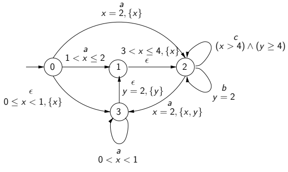
- We augment A with the clock t that measures absolute time and never resets

Definition (Infinite augmented region automaton)

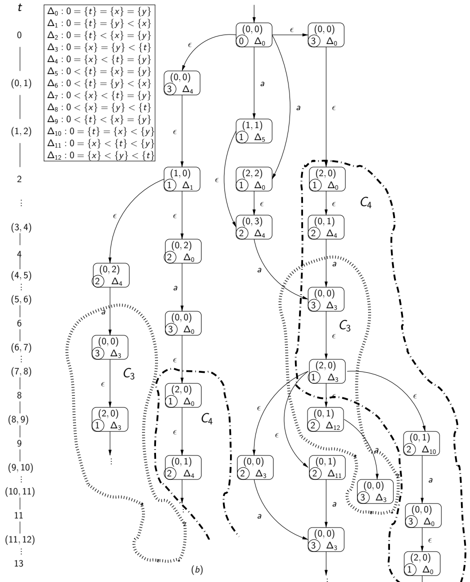
The infinite augmented region automaton $\mathfrak{R}^t_\infty(A)$ is a tuple $(V, v_0, E, \Sigma_\epsilon)$:

- V - the infinite (in general) set of vertices $(q, \mathbf{n}, \Delta)$, where q - a location of A, $(\mathbf{n}, \Delta)$ - a region:
  - $\mathbf{n} = (n_0, n_1, \ldots, n_s) \in \mathbb{N}_0 \times \{0, 1, \ldots, M, \top\}^s$ - the integral parts of the clocks $t, x_1, \ldots, x_s$
  - Δ - the simplex defined by the order of the fractional parts of the clocks
- $v_0 = (q_0, \mathbf{0}, \mathbf{0})$ - 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'
- $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$ - the set of actions

Example: Infinite augmented region automaton



(a)



(b)

Augmented region automaton

- We now fold $\mathfrak{R}^t_\infty(A)$ by ignoring the integral part of t
- The result is a finite augmented region automaton $\mathfrak{R}^t(A)$ obtained by identifying vertices that contain the same data except for the integral part of t
- As a compensation, we assign weights to the edges of $\mathfrak{R}^t(A)$ which equal the integral time difference between the target and source locations
- $\mathfrak{R}^t(A)$ and $\mathfrak{R}^t_\infty(A)$ are equally informative and more informative than the regular region automaton: we can construct from $\mathfrak{R}^t(A)$ a deterministic automaton which approximates A with a maximal error of $1/2$ time units at each observed transition

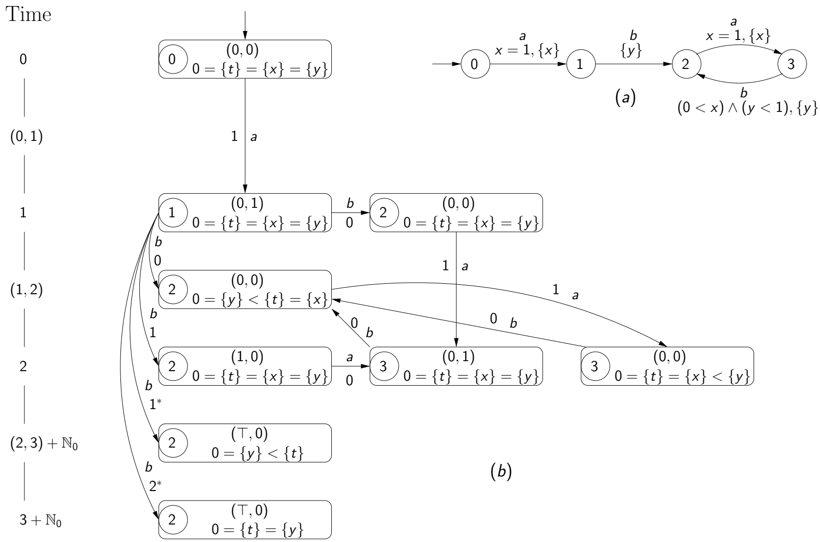
Augmented region automaton - definition

Definition (Augmented region automaton)

The augmented region automaton $\mathfrak{R}^t(A)$ is a tuple $(V, v_0, E, \Sigma_\epsilon, W^*)$:

- V - the set of vertices $(q, \mathbf{n}, \Delta)$ without the integral part of t,
- 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
- $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$ - the set of actions
- W^* - the set of weights on the edges: $m = \lfloor t_1 \rfloor - \lfloor t_0 \rfloor \in [0..M]$, where $\lfloor t_1 \rfloor$ is the integral part of t in the target location and $\lfloor t_0 \rfloor$ - in the source location in the corresponding run of A
  $m^* := m, m+1, m+2, \ldots$

Example: Augmented region automaton



Definition (Duration of a path)

Given a path γ in $\mathfrak{R}^t(A)$, its minimal (integral) duration $d(\gamma) \in \mathbb{N}_0$ is the sum of the weights on its edges, where a weight m^* is counted as m

Lemma

There exists a minimal positive integer t_{nz}, the non-Zeno threshold time, such that every path γ of $\mathfrak{R}^t(A)$ that is of (minimal) duration t_{nz} or more contains a vertex belonging to some non-Zeno cycle (a cycle of duration greater than 0)

A Period of $\mathfrak{R}^t(A)$

Definition (Covering set of non-Zeno cycles)

A set C of non-Zeno cycles of $\mathfrak{R}^t(A)$ is called a covering set of non-Zeno cycles if every path γ of $\mathfrak{R}^t(A)$ whose duration $d(\gamma)$ is at least t_{nz} intersects a cycle in C in a common vertex.

Definition (Period of $\mathfrak{R}^t(A)$)

A (time) period L of $\mathfrak{R}^t(A)$ is a common multiple of the set of durations $d(\pi)$, $\pi \in C$, for some fixed (minimal) covering set of non-Zeno cycles C

Eventual Periodicity of $\mathfrak{R}^t_\infty(A)$

Let t_{nz}, C, L be as above, with C fixed. We denote by $\mathfrak{R}^t_\infty(A)|_{t \geq n}$ the subgraph of $\mathfrak{R}^t_\infty(A)$ that starts at time-level n, that is, the set of vertices of $\mathfrak{R}^t_\infty(A)$ with absolute time $t \geq n$ and their out-going edges.

Definition (L-shift in time)

Given a subgraph G of $\mathfrak{R}^t_\infty(A)$, an *L-shift in time* of G, denoted $G + L$, is the graph obtained by adding the value L to each value of the integral part of the clock t in G and leaving the rest of the data unaltered

Lemma

If $\mathfrak{R}^t_\infty(A)$ is not bounded in time then

$$\mathfrak{R}^t_\infty(A)|_{t \geq t_{nz}} + L \subseteq \mathfrak{R}^t_\infty(A)|_{t \geq t_{nz}+L}$$

Eventual Periodicity of $\mathfrak{R}^t_\infty(A)$

- Let V_k, $k = 0, 1, 2, \ldots$, be the set of vertices

$$V_k = V(\mathfrak{R}^t_\infty(A)|_{t \geq t_{nz} + kL}) \setminus V(\mathfrak{R}^t_\infty(A)|_{t \geq t_{nz} + (k+1)L})$$

Theorem

The infinite augmented region automaton $\mathfrak{R}^t_\infty(A)$ is eventually periodic: there exists an integral time $t_{per} > 0$ such that

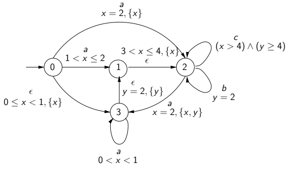
$$\mathfrak{R}^t_\infty(A)|_{t \geq t_{per}} + L = \mathfrak{R}^t_\infty(A)|_{t \geq t_{per} + L}$$

A possible value for t_{per} can be effectively computed by the following:

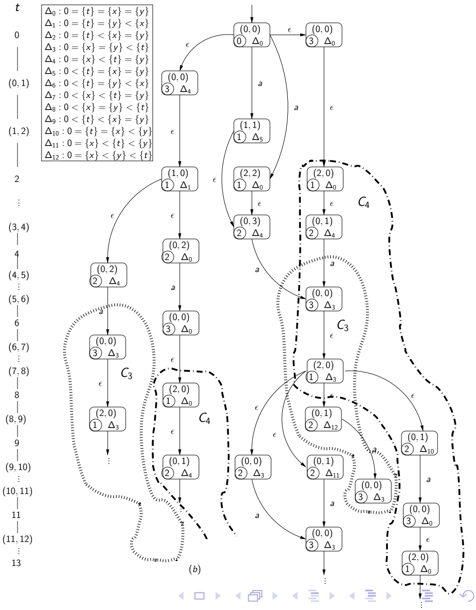
Proposition

If $|V_k| = |V_{k+1}| = |V_{k+2}|$ for some k then we can set $t_{per} = t_{nz} + kL$

Example: periodic structure



(a)



(b)

Suffix-periodicity of the language of TA

- As is known, a TA may be totally non-periodic in the sense that no single timed trace of it is eventually periodic
- However, a special kind of periodicity, which we call suffix-periodicity, holds between different timed traces, as shown in the following theorem

Theorem

The language of A, $\mathfrak{L}(A)$, is suffix-periodic: if $t_r > t_{per}$ and

$$\lambda = (t_1, a_1), \ldots, (t_{r-1}, a_{r-1}), (t_r, a_r), (t_{r+1}, a_{r+1}), \ldots, (t_{r+m}, a_{r+m})$$

is an observable timed trace of $\mathfrak{L}(A)$ then, for each $k \in L\mathbb{Z}$, if $t_r + k > t_{per}$ then there exists an observable timed trace $\lambda' \in \mathfrak{L}(A)$ such that

$$\lambda' = (t'_1, a'_1), \ldots, (t'_s, a'_s), (t_r + k, a_r), (t_{r+1} + k, a_{r+1}), \ldots, (t_{r+m} + k, a_{r+m})$$

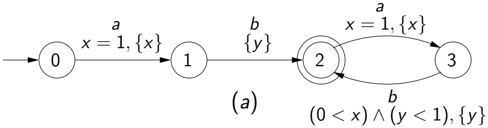
Periodic augmented region automaton

- After revealing the periodic structure of $\mathfrak{R}^t_\infty(A)$, it is natural to fold it into a finite graph according to this period, which we call periodic augmented region automaton, denote by $\mathfrak{R}^t_{per}(A)$
- The construction of $\mathfrak{R}^t_{per}(A)$ is done by first taking the subgraph of $\mathfrak{R}^t_\infty(A)$ of time $t < t_{per} + L$ and then folding the infinite subgraph of $\mathfrak{R}^t_\infty(A)$ of time $t \geq t_{per} + L$ onto the subgraph of time $t_{per} \leq t < t_{per} + L$, which becomes the periodic subgraph

Example (Periodic augmented region automaton)

- The following known example from Alur and Dill (1994) shows a totally non-periodic TA: every word accepted by this automaton has the property that the sequence of time differences between a and the following b is strictly decreasing
- The language accepted by it is

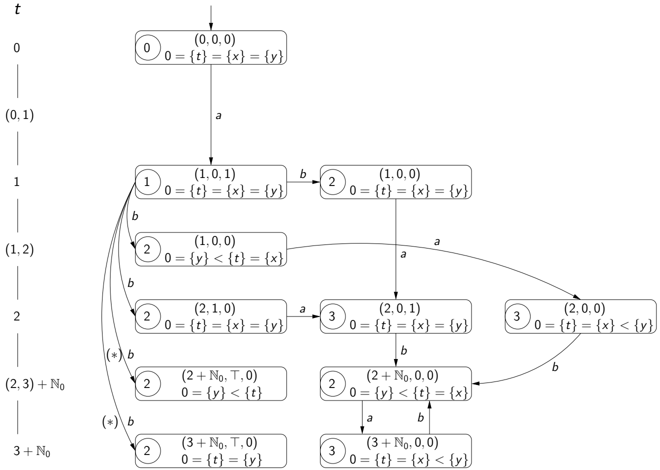
$$\begin{aligned}\mathfrak{L}(A) = &\{(1,a),(t,b)\} \cup \\ &\{(1,a),(1+\delta_1,b),(2,a),(2+\delta_2,b),\cdots,(k,a),(k+\delta_k,b) \\ &: k \in \mathbb{N}, 1 \geq \delta_1 > \delta_2 > \cdots > \delta_k\}\end{aligned}$$



(a)

Example (Periodic augmented region automaton (cont.))

- This non-periodicity is irrelevant when considering the periodic augmented region automaton:



(b)

Timestamp

Theorem

- *The timestamp of a TA A is a union of action-labeled integral points and open unit intervals with integral end-points*
- *It is either finite or forms an eventually periodic subset of $\mathbb{R}_{\geq 0} \times \Sigma$ and is effectively computable*

- The timestamp is easily extracted from $\mathfrak{R}^t_{per}$ or from the subgraph of $\mathfrak{R}^t_{\infty}$ up to level $t_{per} + L$

Corollary (Language non-inclusion)

Given two timed automata A, B, the question of non-inclusion of their timestamps is decidable, thus providing a sufficient condition for $\mathfrak{L}(A) \nsubseteq \mathfrak{L}(B)$

Timestamp automata

Definition (Timestamp automaton)

- Given a TA A, a timestamp automaton $\tilde{A}$ is a deterministic (finite) timed automaton with a single clock and with timestamp identical to that of A
- $\tilde{A}$ is the union of the timestamp automata $\tilde{A}_a$, $a \in \Sigma$, having a common initial vertex, where each $\tilde{A}_a$ is in the shape of a linear graph and possibly ending in a simple loop

Theorem

$\tilde{A}$ can be effectively constructed

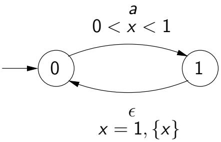
Example (A non-determinizable TA and its timestamp automaton)

- The TA in figure (a) is non-determinizable and its language is

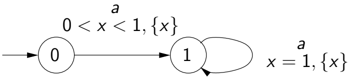
$$\mathfrak{L}(A) = \{(0+\delta_1, a), \cdots, (k+\delta_k, a) : \; k \in \mathbb{N}_0, 0 < \delta_i < 1\}$$

- The TA in figure (b) is deterministic and has the same timestamp:

$$\mathbb{R}_{\geq 0} \setminus \mathbb{N}_0$$



(a)

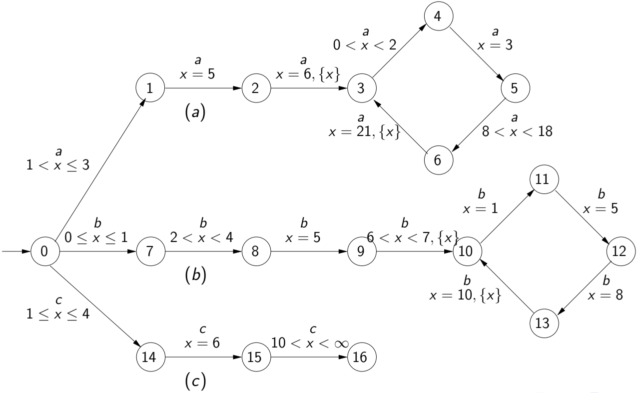


(b)

Example (Timestamp automaton)

Let A be a TA with timestamp

$$\begin{aligned}
\mathbf{TS}(A_a) &= (1,3] \cup \{5\} \cup (6 + ([0,2) \cup \{3\} \cup (8,18)) + 21\mathbb{N}_0) \times \{a\}, \\
\mathbf{TS}(A_b) &= [0,1] \cup (2,4) \cup \{5\} \cup (6 + ((0,1) \cup (1,2) \cup (5,6) \cup (8,9)) + 10\mathbb{N}_0) \\
&\quad \times \{b\}, \\
\mathbf{TS}(A_c) &= [1,4] \cup \{6\} \cup (10,\infty) \times \{c\}.
\end{aligned}$$



Conclusion

- The timestamp consists of the set of all action-labeled times at which locations can be reached by observable transitions
- The problem of computing the timestamp is a generalization of the fundamental problem of reachability
- The timestamp can be effectively computed, also when the TA is non-deterministic and includes silent transitions
- A sufficient condition for language non-inclusion in TA
- By a suitable unfolding of the augmented region automaton one can compute the timestamp of the k-th time a specific location is reached
- Future research: extend the computation of the timestamp to more complicated (extensions of) timed automata, e.g., more general clocks' behavior and transition guards, hybrid automata