Checking Timed Bisimulation with Bounded Zone-History Graphs – Technical Report

Lars Luthmann∗
Real-Time Systems Lab
TU Darmstadt, Germany
lars.luthmann@es.tu-darmstadt.de

Hendrik Göttmann
Real-Time Systems Lab
TU Darmstadt, Germany
h.goettmann@stud.tu-darmstadt.de

Malte Lochau∗
Real-Time Systems Lab
TU Darmstadt, Germany
malte.lochau@es.tu-darmstadt.de

Timed automata (TA) are a well-established formalism for discrete-state/continuous-time behaviors of time-critical reactive systems. Concerning the fundamental analysis problem of comparing a candidate implementation against a specification, both given as TA, it has been shown that timed trace equivalence is undecidable, whereas timed bisimulation equivalence is decidable. The corresponding proof utilizes region graphs, a finite, but very space-consuming characterization of TA semantics. In practice, most TA tools use zone graphs instead, a symbolic and generally more efficient representation of TA semantics, to automate analysis tasks. However, zone graphs only produce sound results for analysis tasks being reducible to plain reachability problems thus being too imprecise for checking timed bisimilarity. To the best of our knowledge, no practical tool is currently available for automated timed bisimilarity-checking. In this paper, we propose bounded zone-history graphs, a novel characterization of TA semantics facilitating an adjustable trade-off between precision and scalability of timed-bisimilarity checking. Our approach supports non-deterministic timed automata with silent moves. We further present experimental results gained from applying our tool TimBRCheck to a collection of community benchmarks, providing insights into trade-offs between precision and efficiency.

1 Introduction

Timed automata (TA) are frequently used to specify discrete-state/continuous-time behaviors of time-critical reactive (software) systems [2, 6]. TA extend labeled state-transition graphs of classical automata models by a set $C$ of clocks constituting constantly and synchronously increasing, yet independently resettable numerical read-only variables. Clock values are referenced by clock constraints in order to specify boundaries for time intervals to be satisfied by occurrences of actions in valid TA runs.

A fundamental analysis problem arises from the comparison of a candidate implementation against a specification, both given as TA. It has been shown that timed trace inclusion is undecidable, whereas timed (bi-)simulation is decidable thus making timed bisimilarity a particularly useful equivalence notion for verifying time-critical behaviors [8, 26]. The original proof is based on region graphs, a finite, but generally very space-consuming representation of TA semantics (i.e., having $O(|C|! \cdot k^{|C|})$ many regions, where $k$ is the maximal constant occurring in a clock constraint). Instead, most recent TA analysis tools use zone graphs, constituting a symbolic and, on average, more efficient representation of TA semantics

∗This work was funded by the Hessian LOEWE initiative within the Software-Factory 4.0 project.
than region graphs. However, zone graphs only produce sound results for analysis tasks being reducible to plain reachability problems thus being too imprecise for checking timed bisimilarity [27].

**Conceptual Contributions.** In this paper, we propose a novel characterization of TA semantics, called *bounded zone-history graphs*, by enriching zones with additional history information required for sound timed bisimilarity-checking. The approach further incorporates a bound parameter to restrict the length of histories thus facilitating an adjustable trade-off between precision and scalability of timed-bisimilarity checking. Our approach handles non-deterministic TA and supports weak and strong bisimilarity of timed (safety) automata with silent τ-moves. All proofs are available in Appendix A.

**Tool Support and Reproducibility.** Our tool TimBRCheck supports the UPPAAL file format for input models and is available at [https://www.es.tu-darmstadt.de/timbrcheck/](https://www.es.tu-darmstadt.de/timbrcheck/) This web page also contains all experimental data and further information for reproducing the evaluation results.

**Experimental Evaluation.** Our experimental results gained from applying TimBRCheck to a collection of community benchmarks [3, 19, 17, 14] provide insights into trade-offs between precision and efficiency of checking timed similarity using bounded zone-history graphs.

**Related Work.** The notion of timed bisimulation goes back to Moller and Toft's [21] as well as Yi [28], both defined on real-time extensions of the process algebra CCS. Similarly, Nicollin and Sifakis [22] define timed bisimulation on ATP (Algebra of Timed Processes). However, none of these works initially incorporated a technique for effectively checking bisimilarity. The pioneering work of Čeráns [8] includes the first decidability proof of timed bisimulation on TA, by providing a finite characterization of bisimilarity-checking on region graphs. The improved (i.e., less space-consuming) approach of Weise and Lenzkes [27] employs a variation of zone graphs, called FBS graphs, which also builds the basis for our zone-history graphs. Guha et al. [13, 11] also follow a zone-based approach for bisimilarity-checking on TA as well as the weaker notion of timed prebisimilarity, by employing so-called zone-valuation graphs and the notion of spans as also used in our approach. Finally, Tanimoto et al. [25] employ timed bisimulation to check if a given behavioral abstraction preserves time-critical system behaviors.

Hence, to the best of our knowledge, no practical tool is currently available for effectively checking timed bisimilarity.

## 2 Preliminaries

We first introduce basic notions of timed automata and timed bisimulation.

### 2.1 Timed Automata

**Syntax.** A timed automaton (TA) consists of finite state-transition graph, whose states are called *locations* (including a distinguished *initial location*) and whose edges, denoting transitions between locations, are called *switches* [2]. Switches are either labeled with names from a finite alphabet $\Sigma$ of *visible actions*, or by a distinguished symbol $\tau \not\in \Sigma$, denoting *internal actions* (silent moves). We range over $\Sigma$ by $\sigma$ and over $\Sigma_\tau = \Sigma \cup \{\tau\}$ by $\mu$. A TA further consists of a finite set $C$ of *clocks*, defined over a numerical *clock domain* $\mathbb{T}_C$ (e.g., $\mathbb{T}_C = \mathbb{N}_0$ for modeling *discrete-time* and $\mathbb{T}_C = \mathbb{R}_+$ for modeling *dense-time*).
Clocks may be considered as constantly and synchronously increasing yet independently resettable variables over $\mathbb{T}_C$. Clocks allow for measuring and restricting time intervals corresponding to durations—or delays between occurrences—of actions in runs of a TA. Those restrictions are expressed by clock constraints $\varphi$ to denote guards for switches and invariants for locations. Guards restrict time intervals in which particular switches are enabled, whereas invariants restrict time intervals in which TA runs are permitted to reside in particular locations. In addition, each switch is labeled with a subset of clocks $R \subseteq C$ to be reset.

**Definition 1 (Timed Automaton).** A TA is a tuple $(L, \ell_0, \Sigma, C, I, E)$, where

- $L$ is a finite set of locations with initial location $\ell_0 \in L$,
- $\Sigma$ is a finite set of actions such that $\tau \notin \Sigma$,
- $C$ is a finite set of clocks such that $C \cap \Sigma = \emptyset$,
- $I : L \to \mathcal{B}(C)$ is a function assigning invariants to locations, and
- $E \subseteq L \times \mathcal{B}(C) \times \Sigma \times 2^C \times L$ is a relation defining switches.

The set $\mathcal{B}(C)$ of clock constraints $\varphi$ over $C$ is inductively defined as

$$\varphi := \text{true} | c \sim n | c - c' \sim n | \varphi \land \varphi, \quad \text{where } \sim \in \{<,\leq,\geq,>,\}, c, c' \in C, n \in \mathbb{T}_C.$$

We denote TA defined over sets $C$ and $\Sigma$ by $\mathcal{A}$ where we may omit an explicit mentioning of $C$ and/or $\Sigma$ if clear from the context. We denote switches $(\ell, g, \mu, R, \ell') \in E$ by $\ell \xrightarrow{g, \mu, R} \ell'$ for convenience.

Clock constraints neither contain operators for equality nor disjunction as both are expressible in the given formalism (e.g., $x = 2$ may be expressed by $x \leq 2 \land x \geq 2$, and $x < 2 \lor x > 2$ may be expressed by two different switches labeled $x < 2$ or $x > 2$, respectively). We further consider diagonal-free TA with clock constraints only containing atomic constraints of the form $c \sim n$ as for every TA, language-equivalent diagonal-free TA can be constructed [7]. Hence, we include difference constraints $c - c' \sim n$ in $\mathcal{B}(C)$ only for the sake of a concise representation of our subsequent constructions. Similarly, we assume location invariants being unequal to true to be downward-closed (i.e., only having clauses $c \leq n$ or $c < n$). However, as two actual restrictions, we limit our considerations to (1) constants $n \in \mathbb{Q}_0$ as real-valued bounds obstruct fundamental decidability properties of TA, as well as to (2) so-called timed safety automata not including distinguished acceptance locations employing Büchi acceptance semantics for infinite TA runs [16, 2].

**Semantics.** The operational semantics of TA defining all valid runs may be defined in terms of Timed Labeled Transition Systems (TLTS) [15]. A TLTS state is a pair $(\ell, u)$ of active location $\ell \in L$ and clock valuation $u \subseteq C$ assigning to each clock $c \in C$ the amount of time $u(c)$ elapsed since the last reset of $c$. Thereupon, TLTS comprise two kinds of transitions: (1) passage of time of duration $d \in \mathbb{T}_C$ while (inactively) residing in location $\ell$, leading to an updated clock valuation $u'$, and (2) instantaneous execution of switches $\ell \xrightarrow{g, \mu, R} \ell'$, leading from location $\ell$ to $\ell'$ by an occurrence of action $\sigma \in \Sigma$. Given clock valuation $u$, by $u + d$, with $d \in \mathbb{T}_C$, we denote the updated clock valuation mapping each clock $c \in C$ to the new value $u(c) + d$. By $[R \mapsto 0]u$, with $R \subseteq C$, we further denote the updated clock valuation mapping each clock $c \in R$ to value 0 (clock reset) while preserving the values $u(c')$ of all other clocks $c' \in C \setminus R$. Finally, by $u \models \varphi$, we denote that clock valuation $u$ satisfies clock constraint $\varphi \in \mathcal{B}(C)$. Concerning $\tau$-labeled transitions, we distinguish between strong and weak TLTS semantics, where $\tau$-transitions are invisible in the latter.
Definition 2 (Timed Labeled Transition System). The TLTS of TA $\mathcal{A}$ over $\Sigma$ is a tuple $(S,s_0,\hat{\Sigma},\rightarrow)$, where

- $S = L \times (\mathcal{E} \rightarrow T_C)$ is a set of states with initial state $s_0 = (\ell_0,[C \mapsto 0]) \in S$.
- $\hat{\Sigma} = \Sigma \cup \Delta$ is a set of transition labels, where $\Delta = T_C$ is bijective such that $(\Sigma \cup \{\tau\}) \cap \Delta = \emptyset$.
- $\rightarrow \subseteq S \times (\hat{\Sigma} \cup \{\tau\}) \times S$ is a set of strong transitions being the least relation satisfying the rules:
  - $\langle \ell,u \rangle \xrightarrow{\rho} \langle \ell,u+d \rangle$ if $(u+d) \in I(\ell)$ for $d \in T_C$, and
  - $\langle \ell,u \rangle \xrightarrow{\rho,\mu} \langle \ell',u' \rangle$ if $\ell \xrightarrow{\rho,\mu} \ell'$, $u \in g, u' \in [R \mapsto 0]u$, $u' \in I(\ell')$ and $\mu \in (\Sigma \cup \{\tau\})$.

By $\Rightarrow \subseteq S \times \hat{\Sigma} \times S$, we denote a set of weak transitions being the least relation satisfying the rules:

- $s \xrightarrow{\rho} s'$ if $s \xrightarrow{\rho} s_1 \xrightarrow{\tau_1} \ldots \xrightarrow{\tau_n} s'$ with $n,m \in \mathbb{N}_0$.
- $s \xrightarrow{\rho} s'$ if $s \xrightarrow{\rho} s'$,
- $s \xrightarrow{\rho,\mu} s'$ if $s \xrightarrow{\rho} s'$ with $n \in \mathbb{N}_0$, and
- $s \xrightarrow{d+d'} s'$ if $s \xrightarrow{d} s''$ and $s'' \xrightarrow{d'} s'$.

We consider TA with strongly convergent TLTS (i.e., without infinite $\tau$-sequences) and refer to the TLTS semantics of TA $\mathcal{A}$ as $\mathcal{J}_d$ or simply $\mathcal{J}$ if clear from the context. In addition, if not further stated, we apply strong TLTS semantics, where the weak version can by obtained by replacing $\rightarrow$ by $\Rightarrow$.

Example 1. Figure 1 shows two sample TA specifying (simplified) coffee machines with corresponding TLTS extracts shown in Figures 1c and 1d. In state (Warm Up,$\times = 0$), we can only let further time pass whereas in (Warm Up,$\times = 1$), we have to choose coffee due to the invariant. In contrast, as neither location Idle nor Fill Cup has an invariant, we may wait for an unlimited amount of time thus resulting in infinitely many consecutive TLTS states. Further note that the TLTS in Fig. 1d contains a $\tau$-transition which is only visible in the strong case.
2.2 Timed Bisimulation

We next revisit the notion of timed bisimulation to semantically compare different TA defined over the same alphabet. A timed (bi-)simulation relation may be defined by directly adapting the classical notion of (bi-)simulation on LTS to TLTS. State \(s'\) of TLTS \(\mathcal{J}_{\mathcal{A}}\) timed simulates state \(s\) of TLTS \(\mathcal{J}_{\mathcal{A}}\) if every transition enabled in \(s\), either labeled with action \(\mu \in \Sigma_r\) or delay \(d \in \Delta\), is also enabled in \(s'\) and the target state in \(\mathcal{J}_{\mathcal{A}}\), again, timed simulates the respective target state in \(\mathcal{J}_{\mathcal{A}}\). Hence, TA \(\mathcal{A}'\) timed simulates \(\mathcal{A}\) if initial state \(s'_0\) timed simulates initial state \(s_0\) and \(\mathcal{A}'\) and \(\mathcal{A}\) are timed bisimilar if the timed simulation relation is symmetric.

Definition 3 (Timed Bisimulation \[27\]). Let \(\mathcal{A}, \mathcal{A}'\) be TA over \(\Sigma\) with \(C \cap C' = \emptyset\) and \(\mathcal{R} \subseteq S \times S'\) such that for all \((s_1, s'_1) \in \mathcal{R}\) it holds that

- if \(s_1 \xrightarrow{\mu} s_2\) with \(\mu \in \Sigma_r\), then \(s'_1 \xrightarrow{\mu} s'_2\) and \((s_2, s'_2) \in \mathcal{R}\) and
- if \(s_1 \xrightarrow{d} s_2\) with \(d \in \Delta\) then \(s'_1 \xrightarrow{d} s'_2\) with \((s_2, s'_2) \in \mathcal{R}\).

\(\mathcal{A}'\) (strongly) timed simulates \(\mathcal{A}\), denoted \(\mathcal{A} \sqsubseteq \mathcal{A}'\), iff \((s_0, s'_0) \in \mathcal{R}\). In addition, \(\mathcal{A}'\) and \(\mathcal{A}\) are (strongly) timed bisimilar, denoted \(\mathcal{A} \simeq \mathcal{A}'\), iff \(\mathcal{R}\) is symmetric.

Weak timed (bi-)simulation can, again, be obtained by replacing \(\xrightarrow{}\) with \(\xRightarrow{}\) in all definitions (which we will omit if not relevant).

Lemma 1. If \(\mathcal{A}'\) strongly timed simulates \(\mathcal{A}\), then \(\mathcal{A}'\) weakly timed simulates \(\mathcal{A}\).

Example 2. Consider TA \(\mathcal{A}\) and \(\mathcal{A}'\) in Figs. 1a and 1b. Strong timed (bi-)simulation does not hold due to the \(\tau\)-step in \(\mathcal{A}'\). In contrast, for the weak case, we have \(\mathcal{A} \sqsubseteq \mathcal{A}'\) as every action and delay of \(\mathcal{A}\) is also permitted by \(\mathcal{A}'\) (cf. TLTS in Figs. 1c and 1d and Example 1). Similarly, \(\mathcal{A}' \sqsubseteq \mathcal{A}\) also holds such that \(\mathcal{A}\) and \(\mathcal{A}'\) are weakly timed bisimilar.

3 Checking Timed Bisimulation with Bounded Zone-History Graphs

As TLTS are, in general, infinite-state and infinitely-branching LTS, they are only of theoretical interest, but do not facilitate effective timed (bi-)similarity checking. In \[8\], a finite, yet often unnecessarily space-consuming characterization of timed bisimilarity is given using region graphs instead of TLTS. In contrast, Weise and Lenzkes \[27\] use so-called full backward stable (FBS) graphs, an adaption of the symbolic zone-graph representation \[10\] of TA semantics enriched by transition labels. Zone graphs are, in most cases, less space-consuming than region graphs. We will also build upon FBS graphs in the following, but propose a novel definition, called (bounded) zone-history graphs, to permit a more concise characterization and scalable checking of timed (bi-)simulation.

3.1 Zone Graphs

A symbolic state of TA \(\mathcal{A}\) is a pair \((\ell, \varphi)\) consisting of a location \(\ell \in L\) and a zone \(\varphi \in \mathcal{B}(C)\), where \(\varphi\) represents the maximum set \(D = \{u : C \rightarrow \mathbb{T}_C | u \in \varphi\}\) of clock valuations \(u\) satisfying clock constraint \(\varphi\). Hence, symbolic state \((\ell, \varphi)\) comprises all TLTS states \((\ell, u) \in \mathcal{J}_{\mathcal{A}}\) with \(u \in D\), where we may use \(\varphi\) and \(D\) interchangeably in the following. The construction of a zone graph for a timed automaton is based on two operations:

- \(D^\dagger = \{u + d | u \in D, d \in \mathbb{T}_C\}\) denotes the future of zone \(D\), and
- \(R(D) = \{[R \mapsto 0]u | u \in D\}\) denotes the application of a set of clock resets \(R \subseteq C\) on zone \(D\).
By $D_0$, we denote the initial zone in which all clock values are mapped to 0. For each switch $\ell \xrightarrow{g, u, R} \ell'$, a corresponding transition $\langle \ell, D \rangle \xrightarrow{u} \langle \ell', D' \rangle$ is added with target zone $D'$ derived from source zone $D$ by considering future $D^\uparrow$ of $D$, restricted by switch guard $g$, location invariants of $\ell$ and $\ell'$ and clock resets $R$.

**Definition 4 (Zone Graph).** The zone graph of $TA$ $\mathcal{A}$ over $\Sigma$ is a tuple $\langle \mathcal{Z}, z_0, \Sigma, \sim \rangle$, where

- $\mathcal{Z} = L \times \mathcal{B}(C)$ is a set of symbolic states with initial state $z_0 = \langle \ell_0, D_0 \rangle$,
- $\Sigma$ is a set of actions, and
- $\sim \subseteq \mathcal{Z} \times \Sigma \times \mathcal{Z}$ is the least relation satisfying the rule:
  \[ \langle \ell, D \rangle \xrightarrow{u} \langle \ell', D' \rangle \text{ if } \ell \xrightarrow{g, u, R} \ell' \text{ and } D' = R(D^\uparrow \wedge g \wedge I(\ell)) \wedge I(\ell'). \]

Although zone graphs according to Def. 4 are, again, not necessarily finite, an equivalent, finite zone-graph representation for any given TA can be obtained (1) by constructing an equivalent diagonal-free TA only containing atomic clock constraints of the form $x \sim r$ [7], and (2) by constructing for this TA a $k$-bounded zone graph with all zones being bound by a maximum global clock ceiling $k$ using $k$-normalization [24, 23].

The comparison of zones from different TA while checking timed bisimilarity is based on the notion of spans [12]. The span of clock $c \in C$ in zone $D$ is the interval $(lo, up)$ between the minimum valuation $lo$ and maximum valuation $up$ of $c$ in $D$. The span of zone $D$ is the least interval covering the spans of all clocks in $D$. By $\infty$, we denote upward-open intervals (i.e., $d < \infty$ for all $d \in \mathbb{T}_C$), where $\infty$ is handled in calculations as usual. Furthermore, we apply two operators for comparing spans $sp_1$ and $sp_2$: $sp_1 \preceq sp_2$ denotes that $sp_1$ is contained in $sp_2$, and $sp_1 \preceq sp_2$ compares the length of spans.

**Definition 5 (Span).** Given zone $D$ and $c \in C$, we use the following notations.

- $\text{span}(c, D) = (lo, up) \in \mathbb{T}_C \times (\mathbb{T}_C \cup \{\infty\})$ is the smallest interval such that $\forall u \in D : u(c) \geq lo \land u(c) \leq up$.
- $(lo, up) \preceq (lo', up') \iff lo \geq lo' \land up \leq up'$.
- $(lo, up) \preceq (lo', up') \iff up - lo \leq up' - lo'$.
- $\text{span}(D) = (lo, up) \iff \forall c \in C : (lo, up) \preceq \text{span}(c, D) \wedge \exists c', c'' \in C : \text{span}(c', D) = (lo, up') \land \text{span}(c'', D) = (lo'', up)$.

Based on spans, we are able compare timing constraints of action occurrences of two different TA independent of the names of locations and clocks. However, due to non-observability of clock resets, it is not sufficient for timed (bi-)simulation checking to just compare spans of pairs of potentially similar symbolic states one-by-one as will be illustrated in the following.

**Example 3.** Considering TA $\mathcal{A}$ and $\mathcal{A}'$ in Figs. 2a and 2b, the span of action $a$ is $(0, 2)$ in both TA due to the switch guards. Additionally, the span for action $b$ is $(0, 5)$ in both TA. However, in $\mathcal{A}$, we may only wait for 5 time units before performing $b$ if we have instantaneously (i.e., with 0 delay) performed $a$ before, whereas in $\mathcal{A}'$, the delay for performing $b$ is independent of previous delays due to the reset of $z$. Hence, $\mathcal{A} \simeq \mathcal{A}'$ does not hold which is, however, not concluded by a pairwise comparison of spans of (presumably bisimilar) symbolic states.

In [27], this issue is tackled by further considering so-called good sequences of FBS graphs in a separate post-check. In contrast, we propose an alternative solution being more aligned with the concepts of (bi-)simulation equivalence relations on state-transition graphs (i.e., by enriching symbolic states with additionally discriminating information).
3.2 Zone-History Graphs

Similar to the notion of causal history as, for instance, proposed for history-preserving event-structure semantics \cite{hda}, we extend symbolic states \( (\ell, D) \) to triples \( (\ell, D, \mathcal{H}) \) further comprising a zone history \( \mathcal{H} \in \mathcal{B}(C)^* \) to memorize sequences of clock constraints corresponding to the zones of predecessor states. When stepping from zone \( D \) to a subsequent zone \( D' \), history \( \mathcal{H} \) is updated to \( \mathcal{H}' \) according to the updates applied to \( D \) leading to \( D' \). By introducing a fresh clock \( \chi \notin C \) which is never explicitly reset, we measure the respective spans of histories \( \mathcal{H} \) in order to compare the sequences of intervals through which the current states are reachable from their predecessors. By \( H \cdot \mathcal{H} \) and \( \mathcal{H}' \cdot H \), respectively, we denote the concatenation of further elements \( H \) in front of, or after, history sequences \( \mathcal{H}' \), where \( \varepsilon \) denotes the empty sequence with \( \mathcal{H} \cdot \varepsilon = \varepsilon \cdot \mathcal{H} = \mathcal{H} \). By \( |\mathcal{H}| \), we further denote the length of sequence \( \mathcal{H} \) and by \( \mathcal{H} \downarrow_k \), \( k > 0 \), we denote the postfix of \( \mathcal{H} \) of length \( k \) (or whole \( \mathcal{H} \) if \( k \geq |\mathcal{H}| \)). In this way, we are able to compare sequences of spans of histories of differing lengths by only considering a respective postfix of the longer one.

**Definition 6 (Zone History).** Let \( \mathcal{H} \in \mathcal{B}(C \cup \{\chi\})^* \) with \( \chi \notin C \) be a zone history. The update of history \( \mathcal{H} \) for a switch \( \ell \xrightarrow{g,H,R} \ell' \) leading from zone \( D \) to \( D' = R(D^g \land g \land I(\ell)) \land I(\ell') \) is recursively defined by

- update\( (\mathcal{H},D,D') = R(D^g \land g \land I(\ell)) \land I(\ell') \cdot \text{update}(\mathcal{H}',D,D') \) if \( \mathcal{H} = H \cdot \mathcal{H}' \),
- update\( (\mathcal{H},D,D') = R((D \land \chi = 0)^g \land g \land I(\ell)) \land I(\ell') \) if \( \mathcal{H} = \varepsilon \).

The comparison of the spans of histories \( \mathcal{H} \) and \( \mathcal{H}' \) is recursively defined by

- \( \mathcal{H} \preceq \mathcal{H}' \) if \( |\mathcal{H}| = |\mathcal{H}'| \land \mathcal{H} = H \cdot \mathcal{H}' \land \mathcal{H}' = H' \cdot \mathcal{H}'' \),
- \( \mathcal{H} \preceq \mathcal{H}' \iff \text{span}(\mathcal{H},H) \preceq \text{span}(\mathcal{H}',H') \land \mathcal{H}'' \preceq \mathcal{H}''' \) if \( |\mathcal{H}| = |\mathcal{H}'| \land \mathcal{H} = H \cdot \mathcal{H}' \land \mathcal{H}' = H' \cdot \mathcal{H}'' \),
- \( \mathcal{H} \preceq \mathcal{H}' \iff \mathcal{H} \downarrow_k \preceq \mathcal{H}' \downarrow_k \) if \( |\mathcal{H}| \neq |\mathcal{H}'| \) and \( k = \min(|\mathcal{H}|,|\mathcal{H}'|) \), and
- \( \mathcal{H} \succeq \mathcal{H}' \iff \mathcal{H} \preceq \mathcal{H} \) and \( \mathcal{H}' \preceq \mathcal{H} \).

We define the zone-history graph of TA \( \mathcal{A} \) by extending plain zone graphs (see Def.\( \ref{def:zone-graph} \)) with zone histories. Initial state \( z_0 = (\ell_0, D_0, \varepsilon) \) comprises initial location \( \ell_0 \), initial zone \( D_0 \) and the empty history. The target state \( (\ell', D', \mathcal{H}') \) of a transition \( (\ell, D, \mathcal{H}) \xrightarrow{g,H,R} (\ell', D', \mathcal{H}') \) corresponding to a switch \( \ell \xrightarrow{g,H,R} \ell' \) is either newly constructed by updating zone \( D \) to \( D' \) as described before and by updating history \( \mathcal{H} \) to \( \mathcal{H}' = \text{update}(\mathcal{H}, D, D') \), or a compatible target state \( (\ell', D', \mathcal{H}'') \) has been already constructed in a previous step. In the latter case, we require \( \mathcal{H}''' \preceq \text{update}(\mathcal{H}, D, D') \) (i.e., either the history of the already
existing state is a *postfix* of the history resulting from the (updated) history of the currently reached state \( \mathcal{H}' \), or vice versa. In this way, zone histories are cut during state-space construction whenever states with similar location-zone pairs and compatible zone-history postfixes have already been reached before.

**Definition 7 (Zone-History Graph).** The zone-history graph of a TA \( \mathcal{A} \) over \( \Sigma \) is a tuple \( (\mathcal{Z}, z_0, \Sigma, \rightsquigarrow) \), where

- \( \mathcal{Z} = L \times \mathcal{B}(C) \times \mathcal{B}(C)^* \) is a set of symbolic states with \( z_0 = (\ell_0, D_0, \epsilon) \),
- \( \Sigma \) is a set of actions, and
- \( \rightsquigarrow \subseteq \mathcal{Z} \times \Sigma \times \mathcal{Z} \) is the least relation satisfying the rule:
  \[ \langle \ell, D, \mathcal{H} \rangle \overset{\mu}{\Rightarrow} \langle \ell', D', \mathcal{H}' \rangle \text{ if } \ell \overset{\mu, R}{\Rightarrow} \ell', D' = R(D' \land g \land I(\ell)) \land I(\ell'), \text{ and update}(\mathcal{H}, D, D') \xRightarrow{\epsilon} \mathcal{H}'. \]

By \( \mathcal{Z}_\mathcal{H} \), we denote the zone-history graph of TA \( \mathcal{A} \) and may omit subscript \( \mathcal{A} \).

**Example 4.** Figures 2c and 2d show extracts from zone-history graphs of TA \( \mathcal{A} \) and \( \mathcal{A}' \), respectively (cf. Figs. 2a and 2b), where \( \mathcal{A}' \) has two clocks, \( y \) and \( z \). The initial state of \( \mathcal{Z}_\mathcal{H} \) starts in location \( \ell_0 \) and zone \( y = 0 \land y = z \). Considering the switch labeled with \( a \), \( y \leq 2 \) and reset of \( z \), we track clock differences in zone-history graphs (cf. \( y = z \) in the initial state) as usual, and update difference constraints in case of clock resets \( [10, 27] \). Due to \( y \leq 2 \), the difference between \( y \) and \( z \) may increase, thus resulting in \( y \leq z + 2 \). The updated zone history yields \( \chi \leq 2 \) with span \( (0, 2) \). Next, we update the existing entry of the zone history and append a new entry for the current step. Here, the next transition labeled \( c \) does not meet the cut criterion, but rather imposes further unrolling.

We require one further concept to compare TA with non-deterministic behaviors (including \( \tau \)-steps) as illustrated by the following example.

**Example 5.** Considering Fig. 3, we have \( \mathcal{A} \simeq \mathcal{A}' \) as both TA permit action \( a \) within span \( (0, 3) \) and both switches of \( \mathcal{A}' \) labeled \( a \) can be simulated by \( \mathcal{A} \). However, the one switch of \( \mathcal{A}' \) cannot be simulated by either of the two switches of \( \mathcal{A} \). Hence, generating comparable zone-history graphs for timed-bisimilarity checking may require a splitting of states in case of non-determinism, as shown in Figs. 3c and 3d for \( \mathcal{A} \) and \( \mathcal{A}' \). Here, states are split due to overlapping spans of guards. We call this construction a composite zone-history graph.

The (non-symmetric) construction of a composite zone-history graph \( \mathcal{Z}_\mathcal{H} \otimes \otimes \mathcal{A} \) for TA \( \mathcal{A} \) with respect to \( \mathcal{A}' \) is based on the zone-history graph \( \mathcal{Z}_\mathcal{H} \otimes \otimes \mathcal{A} \) for the (synchronous) parallel product \( \mathcal{A} \times \mathcal{A}' \), comprising only behaviors shared by \( \mathcal{A} \) and \( \mathcal{A}' \). Additionally, \( \mathcal{Z}_\mathcal{H} \otimes \otimes \mathcal{A} \) also comprises all further behaviors of \( \mathcal{Z}_\mathcal{H} \otimes \mathcal{A} \) potentially not enabled by \( \mathcal{Z}_\mathcal{H} \otimes \mathcal{A} \) such that the result is (1) bisimilar to \( \mathcal{Z}_\mathcal{H} \otimes \mathcal{A} \) and (2) facilitates a (bi-)simulation check with \( \mathcal{Z}_\mathcal{H} \otimes \mathcal{A} \), even in the presence of non-determinism.

In order to construct the composite zone-history graph \( \mathcal{Z}_\mathcal{H} \otimes \otimes \mathcal{A} \), we first define the parallel product \( \mathcal{A} \times \mathcal{A}' \). Next, we employ two auxiliary transition relations for constructing the transition relation \( \otimes \otimes \) of \( \mathcal{Z}_\mathcal{H} \otimes \otimes \mathcal{A} \). Here, \( \otimes \otimes \) defines the transition relation of \( \mathcal{Z}_\mathcal{H} \otimes \otimes \mathcal{A} \) (i.e., the zone-history graph...
of $\mathcal{A} \times \mathcal{A}'$, whereas $\leadsto_1$ refers to the transition relation of $\mathcal{L} \mathcal{H}_{\mathcal{A}}$. First, we require $\sim_x \subseteq \sim_\otimes$. In addition, transition $\langle (\ell_1, \ell_1'), D_1, \mathcal{H}_1 \rangle \stackrel{\mu}{\leadsto_1} \langle (\ell_2, \ell_2'), D_2, \mathcal{H}_2 \rangle$ is also part of $\sim_\otimes$ if $D_2$ is not a subset of $D$ where $D$ is the union of all zones reachable via $\sim_x$ from $\langle (\ell_1, \ell_1'), D_1, \mathcal{H}_1 \rangle$ by the same action $\mu$.

**Definition 8** (Composite Zone-History Graph). Let $\mathcal{A}$, $\mathcal{A}'$ be TA over $\Sigma$ with $\mathcal{C} \cap \mathcal{C}' = \emptyset$. The parallel product $\mathcal{A} \times \mathcal{A}' = (L \times L', (\ell_0, \ell_0'), \Sigma, \mathcal{C} \cup \mathcal{C}', I_x, E_x)$ is a TA with $I_x(\ell, \ell') = I(\ell) \land I(\ell')$ and $E_x$ being the least relation satisfying:

$$
(\ell_1, \ell_1') \xrightarrow{g \land g', \mu, R \cup R'} (\ell_2, \ell_2') \in E_x \text{ if } \begin{cases} 
\ell_1 \xrightarrow{g, \mu, R} \ell_2 \in E \land \ell_1 \xrightarrow{g', \mu, R'} \ell_2' \in E'. 
\end{cases}
$$

The composite zone-history graph $\mathcal{L} \mathcal{H}_{\mathcal{A} \times \mathcal{A}'} = (\mathcal{L}, z_0, \Sigma, \sim_\otimes)$ of $\mathcal{A}$ with respect to $\mathcal{A}'$ is a zone-history graph, where

- $\mathcal{L} = (L \times L') \times \mathcal{B}(\mathcal{C} \cup \mathcal{C}') \times \mathcal{B}(\mathcal{C} \cup \mathcal{C}')^*$ is a set of symbolic states with initial state $z_0 = \langle (\ell_0, \ell_0'), D_0, e \rangle \in \mathcal{L}$,
- $\Sigma$ is a set of actions and $\sim_\otimes \subseteq \mathcal{L} \times \Sigma_t \times \mathcal{L}$ is the least relation satisfying
  - $\sim_x \subseteq \sim_\otimes$ and
  - $\langle (\ell_1, \ell_1'), D_1, \mathcal{H}_1 \rangle \xrightarrow{\mu} \langle (\ell_2, \ell_2'), D_2, \mathcal{H}_2 \rangle \in \sim_\otimes$ if $D \subseteq D_2$ where $D$ is the least set s.t. $\langle (\ell_1, \ell_1'), D_1, \mathcal{H}_1 \rangle \xrightarrow{\mu} \langle (\ell_3, \ell_3'), D_3, \mathcal{H}_3 \rangle \in \sim_x \Rightarrow D_3 \subseteq D$.

The auxiliary transition relations $\sim_x$ and $\sim_1$ are defined as

- $\sim_x \subseteq \mathcal{L} \times \Sigma_t \times \mathcal{L}$ being the least relation satisfying
  $$
  \langle (\ell_1, \ell_1'), D_1, \mathcal{H}_1 \rangle \xrightarrow{} \langle (\ell_2, \ell_2'), D_2, \mathcal{H}_2 \rangle \text{ if } \begin{cases} 
  (\ell_1, \ell_1') \xrightarrow{g, R} (\ell_2, \ell_2'), \\
  D_2 = (R \cup R')(D_1 \land g \land g' \land I_x(\ell_1', \ell_1')) \land I_x(\ell_2', \ell_2)), \text{ and update}(\mathcal{H}_1, D_1, D_2) \times \mathcal{H}_2, \text{ and}
  \end{cases}
  $$

- $\sim_1 \subseteq \mathcal{L} \times \Sigma_t \times \mathcal{L}$ being the least relation satisfying
  $$
  \langle (\ell_1, \ell_1'), D_1, \mathcal{H}_1 \rangle \xrightarrow{} \langle (\ell_2, \ell_2'), D_2, \mathcal{H}_2 \rangle \text{ if } \begin{cases} 
  \ell_1 \xrightarrow{g, R} \ell_2, \\
  D_2 = R(D_1 \land g \land I(\ell_1)) \land I(\ell_2), \text{ and update}(\mathcal{H}_1, D_1, D_2) \times \mathcal{H}_2.
  \end{cases}
  $$

This construction allows us to establish a symbolic version of (strong) timed (bi-)simulation on zone-history graphs such that state $\ell_1'$ simulates state $\ell_1$ if (1) $\ell_1'$ enables the same actions $\mu \in \Sigma_t$ as $\ell_1$, and (2) the spans of zone $D_1'$ as well as of history $\mathcal{H}_1'$ include those of $D_1$ and $\mathcal{H}_1$, respectively.

**Definition 9** (Symbolic Timed Bisimulation). Let $\mathcal{A}$ and $\mathcal{A}'$ be TA over $\Sigma$ with $\mathcal{C} \cap \mathcal{C}' = \emptyset$ and $\mathcal{R} \subseteq \mathcal{L} \times \mathcal{L}'$ such that for all $(\ell_1, \ell_1') \in \mathcal{R}$

- if $\ell_1 \xrightarrow{\mu} \ell_2$ with $\mu \in \Sigma_t$, then $\ell_1' \xrightarrow{\mu} \ell_2'$ and $(\ell_2, \ell_2') \in \mathcal{R}$ and
- span$(D_1 \land I(\ell_1)) \leq$ span$(D_1' \land I'(\ell_1'))$ and $\mathcal{H}_1 \leq \mathcal{H}_1'$.

$\mathcal{A}'$ (strongly) timed simulates $\mathcal{A}$ iff $(z_0, z_0') \in \mathcal{R}$. $\mathcal{A}'$ and $\mathcal{A}$ are (strongly) timed bisimilar, denoted $\mathcal{A} \simeq \mathcal{A}'$, iff $\mathcal{R}$ is symmetric.

We overload $\subseteq$ and $\simeq$ on zone-history graphs, accordingly, and we, again, obtain weak versions of those definition as before. Concerning correctness and decidability of symbolic timed bisimulation on zone-history graphs, we first prove that the composite zone-history graph is semantic-preserving and finite.
**Proposition 1.** Let $\mathcal{A}$, $\mathcal{A}'$ be TA over $\Sigma$. Then it holds that (1) $\mathcal{L}\mathcal{H}_{\mathcal{A}} \simeq \mathcal{L}\mathcal{H}_{\mathcal{A} \cup \mathcal{A}'}$, and (2) $\mathcal{L}\mathcal{H}_{\mathcal{A}}$ and $\mathcal{L}\mathcal{H}_{\mathcal{A} \cup \mathcal{A}'}$ are finite.

Thereupon, we are now able to show correctness of symbolic timed (bi-)simulation.

**Theorem 1.** Let $\mathcal{A}$, $\mathcal{A}'$ be TA over $\Sigma$. Then it holds that (1) $\mathcal{A} \subseteq \mathcal{A}'$ $\iff$ $\mathcal{L}\mathcal{H}_{\mathcal{A} \cup \mathcal{A}'} \subseteq \mathcal{L}\mathcal{H}_{\mathcal{A}'}$, and (2) $\mathcal{L}\mathcal{H}_{\mathcal{A} \cup \mathcal{A}'} \subseteq \mathcal{L}\mathcal{H}_{\mathcal{A}'}$ is decidable.

**Example 6.** The extract from the zone-history graphs in Fig. 4 correspond to the TA in Fig. 1. Starting from the initial state of coffee machine (cf. Fig. 4a) with zone $x = 0$, the zone of the subsequent state is $x = 0$ due to the reset, whereas the following state has zone $x = 1$ due to the invariant of location Warm Up and the guard of switch coffee. Additionally, $\mathcal{H}_1 = (x = 0 \land x \geq y)$ as $x$ is reset, and $\mathcal{H}_2 = (x = 1 \land x \geq y)$ due to the guard and invariant. All elements of $\mathcal{H}_1$ and $\mathcal{H}_2$ equal $(y = 0 \land x \geq y)$ while $\mathcal{H}_3 = (y \geq 0 \land x \geq y) \cdot (y \geq 0 \land x \geq y) \cdot (y \geq 0 \land x = y)$. Hence, both TA are not strongly-, but weakly-bisimilar as, e.g., $\mathcal{H}_1 \leq \mathcal{H}_4$ and $\mathcal{H}_1 \leq \mathcal{H}_1$ (as span $(x, x = 0 \land x \geq y) = \text{span}(y, y = 0 \land x \geq y) = (0, \infty)$). Furthermore, TA in Fig. 4b may immediately produce sugar after action coffee due to silent steps.

As shown in Proposition 1, zone-history graphs are finite and allow for precise timed bisimilarity-checking. However, in case of larger TA models with many locations and clocks, complex clock constraints and frequent clock resets, zone-history graphs may become very large thus obstructing effective timed bisimilarity-checking by practical tools. To also handle realistic models, we next define bounded zone-history graphs to enable potentially imprecise, yet arbitrarily scalable timed bisimilarity-checking.

### 3.3 Bounded Zone-History Graphs

In order to control the size of zone-history graphs, we introduce a bound parameter $b \in \mathbb{N}_0$ restricting each history sequence $\mathcal{H}$ produced by the update-operator (Def. 6) during zone-history graph construction to $\mathcal{H} \downarrow b$ (i.e., memorizing a maximum number of $b$ previous spans). By $\mathcal{A}_b \simeq \mathcal{A}'_b$, we denote that the $b$-bounded zone-history graphs of TA $\mathcal{A}_b$ and $\mathcal{A}'_b$ are timed similar. Hence, $\mathcal{A}_b \simeq \mathcal{A}'_b$ denotes the unbounded case being equivalent to $\mathcal{A}_0 \simeq \mathcal{A}'_0$, whereas $\mathcal{A}_b \simeq_0 \mathcal{A}'_0$ denotes timed bisimilarity-checking on plain zone graphs according to Def. 4.

**Theorem 2.** Let $\mathcal{A}_b$, $\mathcal{A}'_b$ be TA over $\Sigma$.

1. There exists $b < \infty$ such that $\mathcal{A}_b \simeq_0 \mathcal{A}_b'$ iff $\mathcal{A}_b \simeq \mathcal{A}_b'$.
2. If $\mathcal{A}_b \simeq_0 \mathcal{A}_b'$ then $\mathcal{A}_b \simeq_0 \mathcal{A}_b'$ iff $b \geq b'$.

However, identifying a minimal, yet sufficiently large $b$ meeting the first property a-priori is not obvious. In contrast, if $\mathcal{A}_b \nLeftarrow \mathcal{A}_b'$ holds for some $b$, then $\mathcal{A}_b \simeq \mathcal{A}_b'$ does also not hold, whereas $\mathcal{A}_b \simeq b \mathcal{A}_b'$ may be a false positive only if history exceeds bound $b$ at least once during zone-history-graph construction.

**Example 7.** Consider the example in Fig. 2 with $b = 1$. Here, history $\chi \leq 2$ results for the states of $\mathcal{L}\mathcal{H}$ and $\mathcal{L}\mathcal{H}'$ containing $\ell_1$ and $\ell'_1$, respectively. In the states containing $\ell_2$ and $\ell'_2$, respectively, we obtain...
\( \chi \leq 5 \) on both sides as we only consider the last history elements due to \( b = 1 \). Hence, \( \mathcal{A} \approx_1 \mathcal{A}' \). In contrast, \( b \geq 2 \) yields the correct result \( \mathcal{A} \not\approx_b \mathcal{A}' \) as we also consider the different first history elements \( \chi \leq 5 \) and \( \chi \leq 7 \) of the states containing \( \ell_2 \) and \( \ell'_2 \), where we

4 Tool Support and Experimental Evaluation

We now present experimental results gained from applying our tool implementation of the previously presented technique to a collection of TA models.

Tool Support. Our tool for timed-bisimilarity checking, called TIMBRCHECK (timed bisimilarity checker), uses UP\textsc{paal} [18], a tool environment for TA modeling and analysis, as a front end. Our tool therefore supports the UP\textsc{paal} file format as input, generates (bounded) zone-history graphs for a predefined bound value \( b \) and performs a timed-bisimilarity check on the resulting representation. This allows us to apply TIMBRCHECK to already existing community-benchmarks and case studies, where the current version is limited to deterministic TA models with \( \tau \)-moves. Internally, our tool is based on difference bound matrices (DBMs) [5, 10, 6] to construct and manipulate zones and zone histories by utilizing a matrix-based characterization of operations on zones.

Research Questions. Our tool TIMBRCHECK allows us to investigate the impact of parameter \( b \) (see Sect. 3) on efficiency and precision of timed-bisimilarity checking for TA models of different sizes and complexity. Intuitively, we expect that increasing the value of \( b \) has a negative impact on performance, but a positive impact on precision. In addition, we claim that there exists a value for \( b \) yielding the best trade-off between both criteria on average. Hence, we consider the following research questions.

- **RQ1 (Efficiency).** How does parameter \( b \) impact the computational effort of timed-bisimilarity checking?
- **RQ2 (Precision).** How does parameter \( b \) impact precision of timed-bisimilarity checking?
- **RQ3 (Trade-off).** Which parameter value for \( b \) constitutes the best efficiency/precision trade-off on average for timed-bisimilarity checking?

Methods and Experimental Design. In order to systematically investigate and compare the impact of different values of parameter \( b \) on efficiency and precision of timed-bisimilarity checking, we instantiate \( b \) with 10 different values, namely 0, 1, 2, 3, 4, 5, 10, 20, 25, 30 as well as \( \infty \). As described above, \( b = 0 \) means the absence of any history information in zone-history graphs constructed for timed bisimilarity checking, whereas \( b = \infty \) permits unbounded histories. In this regard, \( b = 0 \) is supposed to constitute the most efficient, yet less precise instantiation, whereas \( b = \infty \) guarantees precise results, but presumably causes the highest computational effort. Hence, these two parameterizations serve as our baselines.

We limit our considerations to deterministic input models without \( \tau \)-moves and therefore to strong bisimilarity-checking scenarios to keep the overall number of experiments and respective results comprehensible.

Subject Systems. We consider 5 different TA models, mostly well-known from recent community benchmarks and frequently considered for experimental evaluation of TA analysis techniques and tools [3, 19, 17, 14, 9]. Table 1 provides an overview of key properties of the models, including the number of locations, switches and clocks and the number of (syntactic) occurrences of clock resets within switch
## Table 1: Subject Systems

| Subject System | # Locations | # Switches | # Clocks | # Resets | # Mutants | # Bisimilar Mutants | Description |
|----------------|-------------|------------|----------|----------|-----------|--------------------|-------------|
| TGC            | 14          | 18         | 1        | 6        | 26        | 11                 | A simple level crossing [3]. |
| GC             | 23          | 28         | 1        | 12       | 34        | 15                 | Component of the control system operating in a modern vehicle [10]. |
| CA             | 6           | 13         | 1        | 1        | 15        | 9                  | Models communication among users using an Ethernet-like medium [17]. |
| AVC            | 18          | 30         | 1        | 18       | 33        | 9                  | A messaging protocol for communication between AV components [14]. |
| RCP            | 10          | 26         | 2        | 9        | 26        | 2                  | Root contention protocol of the FireWire Bus [9]. |

guards. Based on these original models, we consider two experimental settings for investigating timed-bisimilarity checking.

1. We simply copy the original models and perform timed-bisimilarity checks between the original model and its copy (which will thus definitely succeed).

2. We copy and mutate the original models and perform bisimilarity checks between original and mutated model (which will either succeed or fail).

For 2., we employ the framework of Aichernig et al. [1] providing canonical mutation operators for TA. In contrast to classical mutation testing for evaluating testing effectiveness, equivalent mutants are not problematic in our setting, but even desirable to investigate efficiency and precision for both negative as well as positive cases. We therefore selected two operators presumably having the highest probability to produce slightly different, yet bisimilar mutants, namely

- **invert resets** (i.e., flipping the reset set $R$ of a switch) and
- **change guards** (i.e., changing a comparison operator in a guard of a switch)

and exhaustively applied both of them to all 5 subject systems. From the resulting overall number of 134 mutants, 46 are equivalent (with respect to timed bisimilarity) to the original model (see Table 1). Hence, our evaluation comprises an overall number of

$$(\text{number of mutants} + 5) \cdot (\text{number of b values}) = 1390$$

runs of TIMBRCHECK of which 510 are should be (true) positives (including the 5 identical copies) and 880 should be (true) negatives.

**Data Collection.** To answer RQ1, we measure (1) CPU time and (2) memory consumption, aggregated over all mutants of each subject system. Concerning (1), we distinguish between CPU time for generating zone-history graphs from CPU time for subsequent bisimilarity checks. According to Theorem 2, the result of bounded timed-bisimilarity checking for a bound value $b < \infty$ may yield false positives, but no false negatives. Hence, to answer RQ2, we only have to count the number of false positives. We executed all experiments on an Intel Core i7-6700k machine with 4x4GHz, 16GB RAM and Linux Mint 19.2. Our tool is implemented in Java using Azul OpenJDK 13.0.1.
Results and Discussion. The measurements for RQ1 (efficiency) are shown in Fig. 5 for the 5 subject systems. As a first observation, the CPU time required for the timed-bisimilarity check (having a peak value of 207ms, but mostly performing much faster) is negligible compared to the CPU time required for the bounded zone-history graph construction (ranging from 246ms to 45s). For all subject systems except for AVC, the average CPU times mostly range from 250ms to 500ms and do not further increase for values of $b$ being greater than 10. In contrast, the computational effort for AVC heavily increases for values of $b$ beyond 20 (where we omitted the measurement of 355s for $b=30$ in Fig. 5). This outlier may be explained through the high number of resets in AVC as compared to the other subject systems. We observed very similar tendencies for the memory consumption (ranging from 24MB to 145MB) which we also omitted in Fig. 5. We further performed an additional experiment for subject system CA with $b=500$, requiring 13s CPU time and 300MB RAM. To summarize, especially the results for AVC indicate a worst-case exponential growth of the overall computational effort for respectively large values of $b$, whereas in the average case, TIMBRCHECK performs quite well (even for RCP having two clocks).

The measurements for RQ2 (precision) are shown in Fig. 5. Here, precision 0 denotes the most imprecise case (i.e., all negatives were falsely reported as positives), whereas a value of 1 denotes the most precise case (i.e., no false positives occur). Interestingly, in case of $b=0$, all negatives were falsely reported as positives thus showing the essential necessity of zone histories during timed-bisimilarity checking even in case of smaller models with only one clock. Conversely, we observe that from $b=3$ on, the probability of false positives drastically decreases and for $b=5$ no more false positives were observed for any subject system (thus being equivalent to $b=\infty$).

Finally, based on these results, we can conclude for RQ3 (trade-off) that $b=3$ appears to be a reasonable bound value for efficient, yet sufficiently precise timed-bisimilarity checking with regard to our subject systems.

Threats to Validity. We first discuss internal threats. The scope of our experimental setting is limited (a) to deterministic TA without $\tau$-moves (and thus strong bisimilarity) as well as (b) to the class of basic safety TA. Concerning (a), the version of TIMBRCHECK used for the experiments supports deterministic TA with $\tau$-moves as well as weak and strong bisimilarity checking, whereas non-determinism is left open for future work (also due to the lack of proper benchmarks and mutations). However, as the construction of composite zone-history graphs, again, yields a proper zone-history graph, we expect comparable results for those cases (i.e., increasing computational effort due to the combinatorial-explosion prob-
lem). Concerning (b), any non-trivial TA extension obstructs essential properties of the underlying zone graphs, obviously making our approach more imprecise or even inapplicable [26]. Concerning the usage of mutation operators to synthetically generate variations of our subject systems, we rely on small and locally restricted changes as usual. Nevertheless, our experiments show that those mutations may produce both TA which are equivalent to the original TA as well as TA which are not, thus indicating mutation to be an appropriate tool for our experiments. Finally, to ensure correctness of (a) our theory and (b) our tool implementation, we (a) provide correctness proves in the accompanying technical report [20] and (b) exhaustively tested our tool by a rich collection of test cases in terms of particularly critical TA fragments.

We identified as external threats (a) a lack of comparison to other tools and (b) the relatively small set of subject systems. Concerning (a), there currently exists, to the best of our knowledge, no competitive tool that provides a functionality being comparable to TIMBRCHECK. Concerning (b), we selected our set of subject systems from well-established community benchmarks of reasonable size and complexity which are frequently used in experiments involving analysis techniques for TA. However, we definitely plan in a future work to consider further case studies, especially including real-world systems.

5 Conclusion

We presented a novel formalism, called bounded zone-history graphs, for precise, yet scalable timed-bisimilarity checking of non-deterministic TA with silent moves. Our tool TIMBRCHECK currently supports strong bisimilarity checking of deterministic TA provided in the UPPAAL file format. Our experimental evaluation shows promising potentials in scaling bisimilarity checking also to larger-scaled models without seriously harming precision. As a future work, we plan to extend our tool and our accompanying experiments to non-deterministic input models as well as more advanced classes of TA [26]. In addition, we are interested in adapting our technique to incorporate further crucial notions of behavioral equivalences beyond timed bisimulation.
References

[1] Bernhard K. Aichernig, Klaus Hörmaier & Florian Lorber (2014): *Debugging with Timed Automata Mutations*. In: SAFECOMP’14, LNCS 8666, Springer, pp. 49–64, doi:10.1007/978-3-319-10506-2_4

[2] Rajeev Alur & David Dill (1990): *Automata for Modeling Real-Time Systems*. In: ICALP’90, LNCS 443, Springer, pp. 322–335, doi:10.1007/BFb0032042

[3] Rajeev Alur, Thomas A. Henzinger & Moshe Y. Vardi (1993): *Parametric Real-time Reasoning*. In: STOC’93, ACM, pp. 592–601, doi:10.1145/167088.167242

[4] Paolo Baldan, Andrea Corradini & Ugo Montanari (1999): *History Preserving Bisimulation for Contextual Nets*. In: WADT’99, LNCS 1827, Springer Berlin Heidelberg, pp. 291–310, doi:10.1007/3-540-44616-3_17

[5] Richard Bellman (1957): *Dynamic Programming*. Princeton University Press.

[6] Johan Bengtsson & Wang Yi (2003): *Timed Automata: Semantics, Algorithms and Tools*. In: ACPN’03, LNCS 3098, Springer, pp. 87–124, doi:10.1007/3-540-30989-2_3

[7] Béatrice Bérard, Antoine Petit, Volker Diekert & Paul Gastin (1998): *Characterization of the Expressive Power of Silent Transitions in Timed Automata*. Fundamenta Informaticae 36(2, 3), pp. 145–182, doi:10.3233/FI-1998-36233

[8] Kārlis Ķerāns (1992): *Decidability of Bisimulation Equivalences for Parallel Timer Processes*. In: CAV’92, LNCS 663, Springer, pp. 302–315, doi:10.1007/3-540-56496-9_24

[9] Aurore Collomb-Annichini & Mihaela Sighireanu (2001): *Parameterized Reachability Analysis of the IEEE 1394 Root Contention Protocol using TReX*

[10] David L Dill (1989): *Timing Assumptions and Verification of Finite-State Concurrent Systems*. In: CAV’89, LNCS 407, Springer, pp. 197–212, doi:10.1007/3-540-52148-8_17

[11] Shibashis Guha, Shankara Narayan Krishna, Chinmay Narayan & S Arun-Kumar (2013): *A Unifying Approach to Decide Relations for Timed Automata and their Game Characterization*. In: EXPRESS/SOS’13, EPTCS 120, arXiv, doi:10.4204/EPTCS.120.5

[12] Shibashis Guha, Chinmay Narayan & S. Arun-Kumar (2012): *Deciding Timed Bisimulation for Timed Automata Using Zone Valuation Graph*

[13] Shibashis Guha, Chinmay Narayan & S. Arun-Kumar (2012): *On Decidability of Prebisimulation for Timed Automata*. In: CAV’12, LNCS 7358, Springer, pp. 444–461, doi:10.1007/978-3-642-31424-7_33

[14] Klaus Havelund, Arne Skou, Kim G. Larsen & Kristian Lund (1997): *Formal Modeling and Analysis of an Audio/Video Protocol: An Industrial Case Study Using UPPAAL*. In: RTSS’97, pp. 2–13, doi:REAL.1997.641264

[15] Thomas A. Henzinger, Zohar Manna & Amir Pnueli (1991): *Timed Transition Systems*. In: REX’91, LNCS 600, Springer, pp. 226–251, doi:10.1007/BFb0031995

[16] Thomas A. Henzinger, Xavier Nicollin, Joseph Sifakis & Sergio Yovine (1994): *Symbolic Model Checking for Real-Time Systems. Information and Computation 111(2)*, pp. 193–244, doi:10.1006/inco.1994.1045

[17] Henrik E. Jensen, Kim G. Larsen & Arne Skou (1996): *Modelling and analysis of a collision avoidance protocol using Spin and Uppaal*. In: DIMACS’96.

[18] Kim G. Larsen, Paul Pettersson & Wang Yi (1997): *UPPAAL in a nutshell*. STTT 1(1), pp. 134–152, doi:10.1007/s100090050010

[19] Magnus Lindahl, Paul Pettersson & Wang Yi (2001): *Formal design and analysis of a gear controller*. STTT 3(3), pp. 353–368, doi:10.1007/BFb0054178

[20] Lars Luthmann, Hendrik Göttmann & Malte Lochau (2019): *Checking Timed Bisimulation with Bounded Zone-History Graphs – Technical Report*. arXiv. Available at http://arxiv.org/abs/1910.08992

[21] Faron Moller & Chris Toft (1990): *A Temporal Calculus of Communicating Systems*. In: CONCUR’90, LNCS 458, Springer, pp. 401–415, doi:10.1007/BFb0039073
[22] Xavier Nicollin & Joseph Sifakis (1994): *The Algebra of Timed Processes, ATP: Theory and Application*. Information and Computation 114(1), pp. 131–178, doi:10.1006/inco.1994.1083

[23] Paul Pettersson (1999): *Modelling and Verification of Real-Time Systems Using Timed Automata: Theory and Practice*. Ph.D. thesis.

[24] Tomas G. Rokicki (1994): *Representing and Modeling Digital Circuits*. Ph.D. thesis.

[25] Tadaaki Tanimoto, Suguru Sasaki, Akio Nakata & Teruo Higashino (2004): *A Global Timed Bisimulation Preserving Abstraction for Parametric Time-Interval Automata*. In: ATVA’04, LNCS 3299, Springer, pp. 179–195, doi:10.1007/978-3-540-30476-0_18

[26] Md Tawhid Bin Waez, Juergen Dingel & Karen Rudie (2013): *A survey of timed automata for the development of real-time systems*. Computer Science Review 9, pp. 1–26, doi:10.1016/j.cosrev.2013.05.001

[27] Carsten Weise & Dirk Lenzkes (1997): *Efficient Scaling-Invariant Checking of Timed Bisimulation*. In: STACS’97, LNCS 1200, Springer, pp. 177–188, doi:10.1007/BFb0023458

[28] Wang Yi (1990): *Real-Time Behaviour of Asynchronous Agents*. In J. C. M. Baeten & J. W. Klop, editors: *CONCUR’90*, LNCS 458, Springer, pp. 502–520, doi:10.1007/BFb0039080
A Proofs

A.1 Proof of Lemma 1

Proof. We prove Lemma 1 by contradiction. Assume TA $\mathcal{A}$ and $\mathcal{A}'$ with $\mathcal{A}'$ strongly timed simulating $\mathcal{A}$ and $\mathcal{A}$ not weakly timed simulating $\mathcal{A}$. In this case, we require TLTS states $\langle \ell_1, u_1 \rangle \in S$ and $\langle \ell'_1, u'_1 \rangle \in S'$ being reachable by a $\tau$-step such that for each $\langle \ell_1, u_1 \rangle \xrightarrow{\eta} \langle \ell'_2, u'_2 \rangle \in \tau^* \mathcal{A}$ with $\eta \in \Sigma$ there exists a $\langle \ell_1, u_1 \rangle \xrightarrow{\eta} \langle \ell_2, u_2 \rangle \in \tau^* \mathcal{A}$. Due to the definition of weak transitions (cf. Def. 2), we also require a transition $\langle \ell_1, u_1 \rangle \xrightarrow{\eta} \langle \ell_2, u_2 \rangle \notin \tau^* \mathcal{A}$ being not enabled in $\langle \ell_1, u_1 \rangle$ to prove that $\mathcal{A}'$ strongly timed simulates $\mathcal{A}$ and $\mathcal{A}'$ weakly timed simulates $\mathcal{A}$. However, as these two assumptions are contradicting, it holds that $\mathcal{A}'$ weakly timed simulates $\mathcal{A}$ if $\mathcal{A}'$ strongly timed simulates $\mathcal{A}$. $\square$

A.2 Proof of Proposition 1

Proof. Let $\mathcal{A}$ and $\mathcal{A}'$ be TA over $\Sigma$. We prove the two parts of Proposition 1 separately.

(1) By definition, $\mathcal{X} \mathcal{H}_{\mathcal{A}_0 \times \mathcal{A}_f}$ contains exactly the shared behaviors of $\mathcal{X} \mathcal{H}_{\mathcal{A}_0}$ and $\mathcal{X} \mathcal{H}_{\mathcal{A}_f}$ (cf. relation $\sim_\times$ of Def. 8). In addition, the remaining behaviors being exclusive to $\mathcal{X} \mathcal{H}_{\mathcal{A}_0}$ are added by relation $\sim_1$ as $\sim_1$ contains exactly the behaviors of $\mathcal{A}_0$. Furthermore, the requirement $D \subset D_2$ ensures that transitions of $\sim_1$ are added to $\sim_\odot$ if and only if the respective behaviors are not already contained in $\sim_\odot$ through $\sim_\otimes$ (again, cf. Def. 8). Hence, it directly follows that $\mathcal{X} \mathcal{H}_{\mathcal{A}_0} \simeq \mathcal{X} \mathcal{H}_{\mathcal{A}_0 \times \mathcal{A}_f}$ due to $\mathcal{A}_0 \simeq \mathcal{A}_f$ as shown with bisimilarity of the corresponding TLTS.

(2) Zone graphs $(\mathcal{Z}, z_0, \Sigma, \sim_\odot)$ (without histories) are not necessarily finite but it has been shown that an equivalent finite zone graph $(\mathcal{Z}, z_0, \Sigma, \sim)$ can be obtained by constructing a $k$-bounded zone graph with all zones being bound by a maximum global clock ceiling $k$ using $k$-normalization [24, 23]. Hence, it remains to be shown that when adding histories, $k$-normalized zone-history graphs $(\mathcal{Z}, z_0, \Sigma, \sim_\odot)$ remain finite. Here, histories $\mathcal{H}$ are constructed in a way such that $\mathcal{H}$ is eventually cut. In particular, whenever there already exists a target state with the same location $\ell$, an equivalent zone $D$, and the existing history $\mathcal{H}'$ has a compatible postfix of length $\min(\mathcal{H}'_\ell, |\mathcal{H}'|)$ to $\mathcal{H}$ (cf. Defs. 7 and 8). As TA are finite state-transition graphs (cf. Def. 11), histories always eventually become regular for the same reasons as $k$-normalization ensures finite zone graphs. In the worst case, zone histories become maximally unrolled after all switches and respective guards have been applied to the zone-history graph. $\square$

A.3 Proof of Theorem 1

Proof. Let $\mathcal{A}$, $\mathcal{A}'$ be TA over $\Sigma$. We prove the two parts of Theorem 1 separately.

(1) It holds, by construction of composite zone-history graphs, that $\sim_\otimes = \sim_\otimes'$ up to renaming of locations and clocks (cf. Def. 8). Hence, w.l.o.g., we have to show that behaviors in $\sim_1$ (i.e., being exclusive to $\mathcal{X} \mathcal{H}_{\mathcal{A}_0 \times \mathcal{A}_f}$) cannot be simulated by $\mathcal{X} \mathcal{H}_{\mathcal{A}_0 \times \mathcal{A}_f}$. This follows directly from the first condition of Def. 9 and the fact that transitions are added to $\sim_1$ iff the corresponding behaviors are exclusive (cf. second rule for $\sim_1$ in Def. 8). Furthermore, exclusive behaviors of $\mathcal{A}$ cannot be simulated by $\mathcal{A}'$ when considering timed bisimulation on TLTS (cf. Defs. 12 and 3). Finally, we have to consider that clock resets hide clock constraints in the sense that a clock constraint $x \sim n$ is no more visible in a zone after $x$ is reset. However, by comparing zone-histories $\mathcal{H}$ and $\mathcal{H}'$, we ensure that the impact of previous clock constraints remain observable by using the fresh clock $\chi$ for tracking respective changes to clock differences including those potentially being hidden by subsequent clock resets. Therefore, it holds that $\mathcal{A} \subseteq \mathcal{A}' \iff \mathcal{X} \mathcal{H}_{\mathcal{A}_0 \times \mathcal{A}_f} \subseteq \mathcal{X} \mathcal{H}_{\mathcal{A}_0 \times \mathcal{A}_f}$. 

L. Luthmann, H. Göttmann, and M. Lochau

17
As composite zone-history graphs are finite (cf. Proposition 1), there are finitely many transitions and spans to check (cf. Def. 9). Hence, $\mathcal{H}_{\mathcal{A} \times \mathcal{A}'} \subseteq \mathcal{H}_{\mathcal{A}' \times \mathcal{A}}$ is decidable.

A.4 Proof of Theorem 2

Proof. We prove (1) and (2) separately.

1. As $\mathcal{A} \approx \mathcal{A}'$ is decidable (cf. Theorem 1) and zone-history graphs have a finite length (cf. Proposition 1), the length of the respective zone history is finite. Hence, there exists $b < \infty$ where $b$ may have the length of the longest zone history when computing $\mathcal{A} \approx \mathcal{A}'$.

2. If it holds that $\mathcal{A} \approx_b \mathcal{A}'$, then it also holds that $\mathcal{A} \approx_{b'} \mathcal{A}'$ if $b \geq b'$ as $\mathcal{A} \approx_{b'} \mathcal{A}'$ considers a shorter history (where the leading elements of the history are equal to considering $b$). Here, recognizing a $\mathcal{A}'$ as not bisimilar would require an element in the zone history to be unequal.