BRICS

Basic Research in Computer Science

Characteristic Formulae for Timed Automata

Luca Aceto
Anna Ingólfsdóttir
Mikkel Lykke Pedersen
Jan Poulsen

BRICS Report Series RS-00-23

ISSN 0909-0878 September 2000

See back inner page for a list of recent BRICS Report Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone:	**+45 8942 3360**
Telefax:	**+45 8942 3255**
Internet:	**BRICS@brics.dk**

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/00/23/`

Characteristic Formulae for Timed Automata

Luca Aceto   Anna Ingólfsdóttir
Mikkel Lykke Pedersen   Jan Poulsen
BRICS[*], Department of Computer Science, Aalborg University
Fr. Bajersvej 7E, 9220 Aalborg Ø, Denmark

Abstract

This paper offers characteristic formula constructions in the real-time logic L_ν for several behavioural relations between (states of) timed automata. The behavioural relations studied in this work are timed (bi)similarity, timed ready simulation, faster-than bisimilarity and timed trace inclusion. The characteristic formulae delivered by our constructions have size which is linear in that of the timed automaton they logically describe. This also applies to the characteristic formula for timed bisimulation equivalence, for which an exponential space construction was previously offered by Laroussinie, Larsen and Weise.

1 Introduction

There are two main methodologies for the formal verification of reactive systems, viz. *model checking* and *implementation verification*. In the model checking approach [8], one establishes the correctness of a system with respect to a given specification by checking whether a state-transition graph that models the program satisfies a temporal logic formula expressing the desired specification of the system's behaviour. In implementation verification, both a system and the specification of its desired behaviour are expressed as state-transition graphs. Establishing that a system is correct with respect to its specification then amounts to checking whether the behaviours of the two state-transition graphs are related in some formal sense. In the classic, untimed setting, this correlation between the behaviours of

[*] **B**asic **R**esearch in **C**omputer **S**cience, Centre of the Danish National Research Foundation

two state-transition graphs is usually expressed in terms of a behavioural relation in the linear time-branching time spectrum [12].

One of the bridges between these two approaches to verification is provided by the notion of *characteristic formula* [13, 15, 26]. A characteristic formula is a formula in a temporal logic that completely characterizes the behaviour of a (state in a) state-transition graph modulo a chosen notion of behavioural relation. Using it, checking whether two state-transition graphs A and B are related with respect to a behavioural relation can be reduced to checking whether, say, A is a model of the characteristic formula for B.

The approach to (automated) verification where the problem of checking behavioral relations between finite Labelled Transition Systems (LTSs) [16] is reduced to model checking is advocated by CLEAVELAND AND STEFFEN in [9, 10]. In their approach, the language being model checked is a logic equivalent in expressive power to the alternation-free fragment of the modal μ-calculus [17]. The efficiency of this approach hinges on the following two facts:

1. the characteristic formula associated with a finite labelled transition system has size that is linear in that of the original LTS, and

2. the time complexity of determining whether a process satisfies a formula is proportional to the product of the sizes of the process and the formula.

The resulting algorithm offered in [9] is still considered to be one of the most efficient for checking behavioural preorders.

In the setting of modelling and verification for real time systems, a characteristic formula construction for timed bisimulation equivalence over timed automata [2] has been offered in [19]. In *op. cit.*, LAROUSSINIE, LARSEN AND WEISE have proposed the logic L_ν—a real-time version of Hennessy-Milner Logic [14] with greatest fixed-points—, and have shown that its associated model checking problem is decidable, and that this logic is sufficiently expressive for representing any timed automaton as a single characteristic L_ν formula. Such a formula uniquely characterizes the timed automaton up to timed bisimilarity.

The characteristic formula construction presented in [19], together with a model checking algorithm for the logic L_ν, yields an algorithm for checking whether two timed automata are timed bisimilar, which may be seen as the implementation of the approach advocated in [9] in a real-time setting. Unfortunately, however, the characteristic formula construction for timed automata proposed in [19] produces formulae whose size is exponential in that of the original automaton, and this makes its use in checking timed bisimilarity for timed automata infeasible. The exponential blow-up involved in the characteristic formula construction from *op. cit.* is due to the fact that the formula is essentially constructed by applying the standard, untimed construction developed by INGÓLFSDÓTTIR, GODSKESEN AND ZEEBERG [15] to the region graph associated with the timed automaton [2]. As shown by ALUR AND DILL [2], the size of the region graph is exponential in that of the original timed automaton.

This study offers characteristic formula constructions for timed automata using the logic L_ν that, like those in the untimed setting and unlike that offered in [19], yield formulae whose size is linear with respect to that of the timed automaton they characterize. We present characteristic formula constructions for timed bisimilarity [28], timed versions of the simulation [21] and ready simulation [5, 20] preorders and for the faster-than preorder [23]. In particular, the characteristic formula construction for timed bisimilarity improves upon that offered in [19]. In addition, since, if B is a deterministic timed automaton, checking whether the set of timed traces afforded by a timed automaton A is included in that of B is equivalent to establishing that B simulates A, the characteristic formula construction for timed simulation can also be applied to checking timed trace inclusion [2].

The constructions we propose constitute a first step towards the application of the model checking approach to implementation verification in the timed setting. A prototype tool based on the theory we present in this study is described in [25].

Further Related Work Characteristic formulae were introduced in [13] to relate equational reasoning about processes to reasoning in a modal logic, and therefore to allow proofs about processes to be carried out in a logical framework. The initial research within characteristic formulae concerned terminating processes and bisimulation equivalences, but extensions to this theory have included finite processes and further equivalences. The unpublished master's thesis [15] presents, amongst other things, characteristic formulae for finite LTSs with respect to bisimulation, and is the precursor of most of the papers on the subject that followed, including ours. In [26] INGÓLFSDÓTTIR AND STEFFEN show how to extend these results to cover bisimulation-like preorders which are sensitive to liveness properties. Their work demonstrates the expressive power of intuitionistically interpreted Hennessy Milner Logic with greatest fixed-points, and builds the theoretical basis for a uniform and efficient method to automatically verify bisimulation-like relations between processes by means of model checking. As previously mentioned, this approach to checking behavioural relations has been advocated by Cleaveland and Steffen in a series of papers (see, e.g., [9]).

All the aforementioned papers use some form of Hennessy-Milner Logic with greatest fixed-points as the logical counterpart of automata. This is, however, by no means the only option pursued in the literature. For example, Browne, Clarke and Grümberg [6] have shown how to characterize Kripke structures in the logic CTL [7] up to bisimilarity.

Roadmap of the Paper After a brief review of background material on timed automata and the logic L_ν (Sect. 2), we present the timed behavioural relations for which we offer characteristic formula constructions (Sect. 3). The constructions of the characteristic formulae are the topic of Sect. 4, where their correctness is also proven. The paper concludes with a discussion of the use of characteristic formulae for checking timed trace inclusion between timed automata in a setting in which the specification automaton is deterministic (Sect. 5).

2 Preliminaries

We begin by briefly reviewing the timed automaton model proposed by Alur and Dill [2] and the logic L_ν [19] that will be used in this study.

Timed Labelled Transition Systems Let Act be a finite set of *actions*, ranged over by a, b, and let $\mathbb{N}$ and $\mathbb{R}_{\geq 0}$ denote the sets of natural and non-negative real numbers, respectively. We use $\mathcal{D}$ to denote the set of *delay actions* $\{\epsilon(d) \mid d \in \mathbb{R}_{\geq 0}\}$, and $\mathcal{L}$ to stand for the union of Act and $\mathcal{D}$. The meta-variable α will range over $\mathcal{L}$.

Definition 2.1 *A* timed labelled transition system *(TLTS) is a structure $\mathcal{T} = (\mathcal{S}, \mathcal{L}, s^0, \longrightarrow)$ where $\mathcal{S}$ is a set of* states, *$s^0 \in \mathcal{S}$ is the initial state, and $\longrightarrow \subseteq \mathcal{S} \times \mathcal{L} \times \mathcal{S}$ is a transition relation satisfying the following properties:*

- (Time Determinism) *for every $s, s', s'' \in \mathcal{S}$ and $d \in \mathbb{R}_{\geq 0}$, if $s \overset{\epsilon(d)}{\rightarrow} s'$ and $s \overset{\epsilon(d)}{\rightarrow} s''$, then $s' = s''$;*
- (Time Additivity) *for every $s, s'' \in \mathcal{S}$ and $d_1, d_2 \in \mathbb{R}_{\geq 0}$, $s \overset{\epsilon(d_1+d_2)}{\rightarrow} s''$ iff $s \overset{\epsilon(d_1)}{\rightarrow} s' \overset{\epsilon(d_2)}{\rightarrow} s''$, for some $s' \in \mathcal{S}$;*

- (0-Delay) *for every* $s, s' \in \mathcal{S}$, $s \overset{\epsilon(0)}{\to} s'$ *iff* $s = s'$.

As usual, we write $s \overset{\alpha}{\to}$ *to mean that there is some state* s' *such that* $s \overset{\alpha}{\to} s'$.

The axioms of time determinism, time additivity and 0-delay are standard in the literature on Yi's TCCS (see, e.g., [28]).

Timed Automata Let C be a set of clocks. We use $\mathcal{B}(C)$ to denote the set of boolean expressions over atomic formulae of the form $x \bowtie p$ and $x - y \bowtie p$, with $x, y \in C$, $p \in \mathbb{N}$, and $\bowtie \in \{<, >, =\}$. Expressions in $\mathcal{B}(C)$ are interpreted over the collection of time assignments. A *time assignment*, or *valuation*, v for C is a function from C to $\mathbb{R}_{\geq 0}$. Given an expression $g \in \mathcal{B}(C)$ and a time assignment v, we write $v \models g$ if v satisfies g. Note that $\mathcal{B}(C)$ is closed under negation. For every time assignment v and $d \in \mathbb{R}_{\geq 0}$, we use $v + d$ to denote the time assignment which maps each clock $x \in C$ to the value $v(x) + d$. Two assignments u and v are said to agree on the set of clocks C' iff they assign the same real number to every clock in C'. For every subset C' of clocks, $v[C' \mapsto 0]$ denotes the assignment for C which maps each clock in C' to the value 0 and agrees with v over $C \backslash C'$.

Definition 2.2 *A* timed automaton *is a quintuple* $A = (\mathsf{Act}, N, n_0, C, E)$ *where* N *is a finite set of* nodes*,* n_0 *is the* initial node*,* C *is a finite set of* clocks*, and* $E \subseteq N \times N \times \mathsf{Act} \times 2^C \times \mathcal{B}(C)$ *is a set of* edges*. The quintuple* $e = (n, n_e, a, r_e, g_e) \in E$ *stands for an edge from node* n *to node* n_e *(the* target *of* e*) with action* a*, where* r_e *denotes the set of clocks to be reset to* 0 *and* g_e *is the enabling condition (or* guard*) over the clocks of* A*.*

A *state* of a timed automaton A is a pair (n, v) where n is a node of A and v is a time assignment for C. The initial state of A is $(n_0, [C \mapsto 0])$ where n_0 is the initial node of A, and $[C \mapsto 0]$ is the time assignment mapping all clocks in C to 0.

The operational semantics of a timed automaton A is given by the TLTS $\mathcal{T}_A = (\mathcal{S}_A, \mathcal{L}, s^0_A, \longrightarrow)$, where $\mathcal{S}_A$ is the set of states of A, s^0_A is the initial state of A, and $\longrightarrow$ is the transition relation defined as follows:

$$(n, v) \overset{a}{\to} (n', v') \text{ iff } \exists e = (n, n', a, r_e, g_e) \in E.\ v \models g_e \wedge v' = v[r_e \mapsto 0]$$

$$(n, v) \overset{\epsilon(d)}{\to} (n', v') \text{ iff } n = n' \text{ and } v' = v + d\ ,$$

where $a \in \mathsf{Act}$ and $\epsilon(d) \in \mathcal{D}$.

The Logic L_ν The logic L_ν is a real-time version of Hennessy-Milner Logic with greatest fixed-points that stems from [19]. We now briefly review its syntax and semantics for the sake of completeness.

Definition 2.3 (Syntax of L_ν) *Let K be a finite set of formula clocks,* **Id** *a finite set of identifiers and k a non-negative integer. The set L_ν of formulae over K,* **Id** *and largest constant k is generated by the abstract syntax below, with φ and ψ ranging over L_ν:*

$$\varphi ::= \mathtt{tt} \mid \mathtt{ff} \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \exists\varphi \mid \forall\varphi \mid \langle a\rangle\varphi \mid [a]\varphi \mid$$

$$x \ \underline{\mathtt{in}}\ \varphi \mid x \bowtie p \mid x + p \bowtie y + q \mid Z$$

where $a \in \mathsf{Act}$, $x, y \in K$, $p, q \in \{0, \dots, k\}$, $\bowtie \in \{=, <, \leq, >, \geq\}$ and $Z \in$ **Id**.

The logic L_ν allows for the recursive definition of formulae by including a finite set **Id** of identifiers. The formula associated with each of the identifiers is specified by a declaration $\mathcal{D}$, i.e. $\mathcal{D}$ assigns a formula of L_ν to each identifier. For an identifier Z we let $Z \overset{\text{def}}{=} \varphi$ denote $\mathcal{D}(Z) = \varphi$. Intuitively Z will stand for the largest solution of the equation $Z \overset{\text{def}}{=} \varphi$. We refer the interested reader to [19] for more information on L_ν.

Given a timed automaton A, whose set of clocks C is disjoint from K, we interpret the formulae in L_ν over extended states. An *extended state* of A is a pair (n, vu), where (n, v) is a state of A, u is a time assignment for K, and we use vu for the assignment over $C \cup K$ that agrees with v over C and with u over K.

Definition 2.4 (Semantics of $\mathbf{L}_\nu$) *Let A be a timed automaton and $\mathcal{D}$ a declaration. The satisfaction relation $\models_{\mathcal{D}}$ is the largest relation satisfying the following implications:*

$$\begin{array}{rcl}
(n, vu) \models_{\mathcal{D}} \mathtt{tt} & \Rightarrow & \textit{true} \\
(n, vu) \models_{\mathcal{D}} \mathtt{ff} & \Rightarrow & \textit{false} \\
(n, vu) \models_{\mathcal{D}} \varphi \wedge \psi & \Rightarrow & (n, vu) \models_{\mathcal{D}} \varphi \textit{ and } (n, vu) \models_{\mathcal{D}} \psi \\
(n, vu) \models_{\mathcal{D}} \varphi \vee \psi & \Rightarrow & (n, vu) \models_{\mathcal{D}} \varphi \textit{ or } (n, vu) \models_{\mathcal{D}} \psi \\
(n, vu) \models_{\mathcal{D}} \exists\varphi & \Rightarrow & \exists d \in \mathbb{R}_{\geq 0}.(n, (v + d)(u + d)) \models_{\mathcal{D}} \varphi \\
(n, vu) \models_{\mathcal{D}} \forall\varphi & \Rightarrow & \forall d \in \mathbb{R}_{\geq 0}.(n, (v + d)(u + d)) \models_{\mathcal{D}} \varphi \\
(n, vu) \models_{\mathcal{D}} \langle a\rangle\varphi & \Rightarrow & \exists(n', v').(n, v) \xrightarrow{a} (n', v') \textit{ and } (n', v'u) \models_{\mathcal{D}} \varphi \\
(n, vu) \models_{\mathcal{D}} [a]\varphi & \Rightarrow & \forall(n', v').(n, v) \xrightarrow{a} (n', v') \textit{ implies } (n', v'u) \models_{\mathcal{D}} \varphi \\
(n, vu) \models_{\mathcal{D}} x \bowtie p & \Rightarrow & u(x) \bowtie p \\
(n, vu) \models_{\mathcal{D}} x + p \bowtie y + q & \Rightarrow & u(x) + p \bowtie u(y) + q \\
(n, vu) \models_{\mathcal{D}} x \ \underline{\mathtt{in}}\ \varphi & \Rightarrow & (n, vu') \models_{\mathcal{D}} \varphi \textit{ where } u' = u[\{x\} \mapsto 0] \\
(n, vu) \models_{\mathcal{D}} Z & \Rightarrow & (n, vu) \models_{\mathcal{D}} \mathcal{D}(Z) \ .
\end{array}$$

Any relation satisfying the above implications is referred to as a satisfiability relation. From standard fixed-point theory [27] we have that $\models_{\mathcal{D}}$ is the union of all satisfiability relations.

3 Timed Behavioural Relations

In the untimed setting various behavioral relations over processes have been proposed (see, e.g., [12] for an encyclopaedic treatment and detailed references to the original literature), and some of them (e.g. bisimulation and trace equivalence) have later been adapted to a timed setting. However, the timed setting also provides specific time-dependent behavioral relations. One such relation is the *faster-than bisimulation* from [23], which explicitly requires one process to execute at least as fast as another, while having the same functional behaviour. (See [3] for a similar proposal in the more classic setting of CCS [22].)

We now proceed to review the timed behavioural relations over TLTSs studied in this paper. The notion of timed bisimulation stems from [28]. It is the obvious adaptation to the timed setting of the classic definition due to Park [24].

Definition 3.1 *Let $\mathcal{T} = (\mathcal{S}, \mathcal{L}, s^0, \longrightarrow)$ be a TLTS. A* timed simulation *is a relation $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$ such that whenever $s_1 \mathcal{R} s_2$ and $\alpha \in \mathcal{L}$, then:*

- *If $s_1 \xrightarrow{\alpha} s_1'$ then $s_2 \xrightarrow{\alpha} s_2'$ for some s_2' such that $s_1' \mathcal{R} s_2'$.*

A timed bisimulation *is a symmetric timed simulation.*

For states s_1, s_2, we write $s_1 \underset{\sim}{\sqsubset}_S s_2$ (resp. $s_1 \sim s_2$) iff there exists a timed simulation (resp. a timed bisimulation) $\mathcal{R}$ with $s_1 \mathcal{R} s_2$.

In the untimed setting, the notion of *ready simulation* stems from [5, 20]. In [5], the ready simulation preorder was characterized as the largest congruence with respect to the GSOS format of operational rules included in completed trace inclusion.

Definition 3.2 *Let $\mathcal{T} = (\mathcal{S}, \mathcal{L}, s^0, \longrightarrow)$ be a TLTS. A* timed ready simulation *is a relation $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$ such that whenever $s_1 \mathcal{R} s_2$, $a \in \mathsf{Act}$ and $\alpha \in \mathcal{L}$ then:*

- *If $s_1 \xrightarrow{\alpha} s_1'$ then $s_2 \xrightarrow{\alpha} s_2'$ for some s_2' such that $s_1' \mathcal{R} s_2'$;*
- *If $s_2 \xrightarrow{a}$ then $s_1 \xrightarrow{a}$.*

For states s_1, s_2, we write $s_1 \precsim_{RS} s_2$ iff there exists a timed ready simulation $\mathcal{R}$ with $s_1 \mathcal{R} s_2$.

Moller and Tofts [23] have proposed a preorder on processes that distinguishes functionally behaviourally equivalent processes which operate at different speed. Their original proposal applied to their calculus TCCS, but it is simple enough to adapt it to the setting of TLTSs.

Definition 3.3 *Let $\mathcal{T} = (\mathcal{S}, \mathcal{L}, s^0, \longrightarrow)$ be a TLTS. A* faster-than bisimulation *is a relation $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$ such that whenever $s_1 \mathcal{R} s_2$, $a \in \mathsf{Act}$ and $d \in \mathbb{R}_{\geq 0}$ then:*

1. *if $s_1 \xrightarrow{a} s_1'$ then there are $d \in \mathbb{R}_{\geq 0}, s_1'', s_2'$ and s_2'' such that $s_1' \xrightarrow{\epsilon(d)} s_1''$, $s_2 \xrightarrow{\epsilon(d)} s_2' \xrightarrow{a} s_2''$, and $s_1' \ \mathcal{R} \ s_2'$;*
2. *If $s_2 \xrightarrow{a} s_2'$ then $s_1 \xrightarrow{a} s_1'$ for some s_1' such that $s_1' \ \mathcal{R} \ s_2'$;*
3. *If $s_1 \xrightarrow{\epsilon(d)} s_1'$ then $s_2 \xrightarrow{\epsilon(d)} s_2'$ for some s_2' such that $s_1' \ \mathcal{R} \ s_2'$;*
4. *If $s_2 \xrightarrow{\epsilon(d)} s_2'$ then $s_1 \xrightarrow{\epsilon(d)} s_1'$ for some s_1' such that and $s_1' \ \mathcal{R} \ s_2'$.*

For states s_1, s_2, we write $s_1 \precsim_{FT} s_2$ iff there exists a faster-than bisimulation $\mathcal{R}$ with $s_1 \mathcal{R} s_2$.

It is well-known that $\precsim_*$ $(* \in \{S, RS, FT\})$ is a preorder. Moreover $\precsim_S$ is the largest timed simulation, $\precsim_{RS}$ is the largest timed ready simulation, and $\precsim_{FT}$ is the largest faster-than bisimulation. Similarly, $\sim$ is an equivalence relation, and is the largest timed bisimulation.

All of the previously defined behavioural relations can be lifted to the setting of timed automata thus:

Definition 3.4 *Let A, B be two timed automata. Then, for every relation $\mathcal{R} \in \{\precsim_S, \precsim_{RS}, \precsim_{FT}, \sim\}$, we write $A \ \mathcal{R} \ B$ iff $s_A^0 \ \mathcal{R} \ s_B^0$ in the TLTS that results by taking the disjoint union of $\mathcal{T}_A$ and $\mathcal{T}_B$.*

In what follows, we shall always use the behavioural relations defined above to compare (states of) timed automata.

4 Characteristic Formula Constructions

We now offer general characteristic formula constructions in terms of L_ν for each of the timed behavioral relations introduced in Sect. 3. The constructions associate with each timed automaton a set of propositional equations (one equation per node of the automaton) that characterizes it up to the given timed behavioural relation.

To increase the readability of the characteristic formulae we make use of some derived constructs in the logic L_ν. These we now present for the sake of clarity.

For a reset set $r = \{x_1, \ldots, x_k\}$, we use the abbreviation $r \ \underline{\texttt{in}}\ \varphi$ to stand for the formula inductively defined thus:

$$\begin{aligned} \emptyset \ \underline{\texttt{in}}\ \varphi &\stackrel{\text{def}}{=} \varphi \\ \{x_1, \ldots, x_k\} \ \underline{\texttt{in}}\ \varphi &\stackrel{\text{def}}{=} x_1 \ \underline{\texttt{in}}\ (\{x_2, \ldots, x_k\} \ \underline{\texttt{in}}\ \varphi) \qquad (k \geq 1) \ . \end{aligned}$$

Note that the order of the clocks is arbitrary because $x \ \underline{\texttt{in}}\ (y \ \underline{\texttt{in}}\ \varphi)$ is logically equivalent to $y \ \underline{\texttt{in}}\ (x \ \underline{\texttt{in}}\ \varphi)$.

The expression $g \Rightarrow \varphi$ will stand for $\overline{g} \vee \varphi$, where $\overline{g}$ is the negation of the guard g. This is a formula in L_ν because the collection of guards is closed under negation.

Given a node n in a timed automaton A, and action a, we define:

$$\mathit{enabled}(n, a) \stackrel{\text{def}}{=} \bigvee_{e \in E(n,a)} g_e \ , \tag{1}$$

where $e = (n, n_e, a, r_e, g_e)$ is an edge, and $E(n, a)$ denotes the set of a-labelled edges from node n. Intuitively, the formula $\mathit{enabled}(n, a)$ describes when action a can be performed from a state of the form (n, v). The negation of the expression $\mathit{enabled}(n, a)$ will be used in the characteristic formula construction for timed ready simulation. Note that, since the collection of guards is closed under negation, the negation of $\mathit{enabled}(n, a)$ can also be expressed in L_ν. Finally, we recall that, as usual, an empty disjunction stands for $\mathtt{ff}$ and an empty conjunction is equivalent to $\mathtt{tt}$.

In the remainder of this section, we shall implicitly assume a given timed automaton A, for which all the characteristic formulae will be defined.

Characteristic Formula for Timed Bisimulation Equivalence For this relation we define the characteristic formula describing the properties presented in Defn. 3.1. A formula characterizing a node of a timed automaton up to timed bisimulation should offer a description of:

1. all the actions that are enabled in the node,

2. which node is entered by taking a given transition, together with the resets associated with it, and

3. the maximal delay that is allowed in the node.

The resulting characteristic formula is presented below. It consists of three sets of conjuncts, each associated to one of the above properties, for each node n of a timed automaton A:

$$\begin{array}{lcl}\Phi^{\sim}(n) & \stackrel{\mathrm{def}}{=} & (\bigwedge_{a\in\mathsf{Act}} \bigwedge_{e\in E(n,a)} g_e \Rightarrow (\langle a\rangle\ r_e\ \underline{\mathtt{in}}\ \Phi^{\sim}(n_e))) \wedge \\ & & \bigwedge_{a\in\mathsf{Act}} [a](\bigvee_{e\in E(n,a)} g_e \wedge (r_e\ \underline{\mathtt{in}}\ \Phi^{\sim}(n_e))) \wedge \\ & & \forall\!\!\!\forall\Phi^{\sim}(n)\end{array}$$

where n is a node of A, $e = (n, n_e, a, r_e, g_e)$, and we recall that $E(n,a)$ denotes the set of a-labelled edges from node n. We shall use $\mathcal{D}_A^{\sim}$ to denote the declaration that consists of the equations above, one for each node of A.

Theorem 4.1 *Let A, B be timed automata with disjoint sets of clocks. Let n be a node of A and m be a node of B. Assume that u and v are valuations for the clocks of A and B, respectively. Then*

$$(n,u) \sim (m,v) \text{ iff } (m,vu) \models \Phi^{\sim}(n) \quad ,$$

where $(m,vu) \models \Phi^{\sim}(n)$ holds with respect to the declaration $\mathcal{D}_A^{\sim}$.

Proof: We separately prove that:

1. $(n,u) \sim (m,v)$ only if $(m,vu) \models \Phi^{\sim}(n)$, and
2. if $(m,vu) \models \Phi^{\sim}(n)$ then $(n,u) \sim (m,v)$.

⟨1⟩ To show that the 'only if' implication holds, consider the relation $\vdash$ defined by structural induction on formulae thus: (m ranges over the nodes of B, and n over those of A)

$$\begin{array}{rcl}
(m, vu) \vdash \mathtt{tt} & \Leftrightarrow & \text{true} \\
(m, vu) \vdash \mathtt{ff} & \Leftrightarrow & \text{false} \\
(m, vu) \vdash \varphi \wedge \psi & \Leftrightarrow & (m, vu) \vdash \varphi \text{ and } (m, vu) \vdash \psi \\
(m, vu) \vdash \varphi \vee \psi & \Leftrightarrow & (m, vu) \vdash \varphi \text{ or } (m, vu) \vdash \psi \\
(m, vu) \vdash \exists\varphi & \Leftrightarrow & \exists d \in \mathbb{R}_{\geq 0}.(m, (v+d)(u+d)) \vdash \varphi \\
(m, vu) \vdash \forall\varphi & \Leftrightarrow & \forall d \in \mathbb{R}_{\geq 0}.(m, (v+d)(u+d)) \vdash \varphi \\
(m, vu) \vdash \langle a \rangle \varphi & \Leftrightarrow & \exists (m', v').(m, v) \xrightarrow{a} (m', v') \text{ and} \\
 & & (m', v'u) \vdash \varphi \\
(m, vu) \vdash [a]\varphi & \Leftrightarrow & \forall (m', v').(m, v) \xrightarrow{a} (m', v') \text{ implies} \\
 & & (m', v'u) \vdash \varphi \\
(m, vu) \vdash x \bowtie p & \Leftrightarrow & u(x) \bowtie p \\
(m, vu) \vdash x + p \bowtie y + q & \Leftrightarrow & u(x) + p \bowtie u(y) + q \\
(m, vu) \vdash x \ \underline{\mathtt{in}}\ \varphi & \Leftrightarrow & (m, vu') \vdash \varphi \text{ where } u' = u[\{x\} \mapsto 0] \\
(m, vu) \vdash \Phi^{\sim}(n) & \Leftrightarrow & (m, v) \sim (n, u) \ .
\end{array}$$

We prove that $\vdash$ is a satisfiability relation. The only interesting part of the proof is to show that if $(m, vu) \vdash \Phi^{\sim}(n)$, then $(m, vu) \vdash \mathcal{D}_A^{\sim}(\Phi^{\sim}(n))$. This we now present in detail.

Assume that $(m, vu) \vdash \Phi^{\sim}(n)$ and let ξ be a conjunct of $\mathcal{D}_A^{\sim}(\Phi^{\sim}(n))$. We prove that $(m, vu) \vdash \xi$ holds for each of the three types of conjuncts of the characteristic formula.

$\langle 1 \rangle.1$ Case $\xi \equiv g_e \Rightarrow (\langle a \rangle r_e \ \underline{\mathtt{in}}\ \Phi^{\sim}(n_e))$, where $a \in \mathsf{Act}$ and $e \in E(n, a)$.

The claim is trivial if $u \not\models g_e$. Assume now that $u \models g_e$. We wish to argue that

$$(m, vu) \quad \vdash \quad \langle a \rangle r_e \ \underline{\mathtt{in}}\ \Phi^{\sim}(n_e) \ . \qquad (2)$$

Since $u \models g_e$ and $e \in E(n, a)$, it follows that $(n, u) \xrightarrow{a} (n_e, u[r_e \mapsto 0])$. By the assumption that $(m, vu) \vdash \Phi^{\sim}(n)$, we have that $(n, u) \sim (m, v)$. Thus there is a transition $(m, v) \xrightarrow{a} (m', v')$ with

$$(m', v') \quad \sim \quad (n_e, u[r_e \mapsto 0]) \ .$$

By the definition of $\vdash$, it follows that

$$(m', v'(u[r_e \mapsto 0])) \quad \vdash \quad \Phi^{\sim}(n_e) \ .$$

Thus, again by the definition of $\vdash$, it holds that

$$(m', v'u) \quad \vdash \quad r_e \ \underline{\mathtt{in}}\ \Phi^{\sim}(n_e) \ ,$$

from which (2) follows because $(m, v) \xrightarrow{a} (m', v')$.

$\langle 1 \rangle.2$ Case $\xi \equiv [a](\bigvee_{e \in E(n,a)} g_e \wedge (r_e \ \underline{\mathtt{in}}\ \Phi^{\sim}(n_e)))$, where $a \in \mathsf{Act}$.

Assume that $(m, v) \xrightarrow{a} (m', v')$. We prove that

$$(m', v'u) \quad \vdash \quad \bigvee_{e \in E(n,a)} g_e \wedge (r_e \ \underline{\mathtt{in}}\ \Phi^{\sim}(n_e)) \ . \tag{3}$$

To this end, note that, since $(n, u) \sim (m, v)$ by the assumption that $(m, vu) \vdash \Phi^{\sim}(n)$, there is a transition $(n, u) \xrightarrow{a} (n', u')$ with

$$(n', u') \quad \sim \quad (m', v') \ . \tag{4}$$

Since $(n, u) \xrightarrow{a} (n', u')$ holds, there is an edge $e \in E(n, a)$ such that

- $u \models g_e$,
- $n' = n_e$, and
- $u' = u[r_e \mapsto 0]$.

Thus $(m', v'u) \vdash g_e$ and, by (4), $(m', v'(u[r_e \mapsto 0])) \vdash \Phi^{\sim}(n_e)$. By the definition of $\vdash$, we may now infer that

$$(m', v'u) \vdash r_e \ \underline{\mathtt{in}}\ \Phi^{\sim}(n_e)$$

from which (3) finally follows.

$\langle 1 \rangle .3$ Case $\xi \equiv \forall\!\!\!\forall \Phi^{\sim}(n)$.
Assume that $d \in \mathbb{R}_{\geq 0}$. We prove that

$$(m, (v + d)(u + d)) \quad \vdash \quad \Phi^{\sim}(n) \ . \tag{5}$$

Since $(m, v) \overset{\epsilon(d)}{\to} (m, v + d)$, $(n, u) \overset{\epsilon(d)}{\to} (n, u + d)$, and $(m, v) \sim (n, u)$ hold, it follows by time determinism that $(m, v + d) \sim (n, u + d)$ also holds. The definition of $\vdash$ now yields (5), which was to be shown.

The proof of statement $\langle 1 \rangle$ is now complete.

$\langle 2 \rangle$ We prove that the relation

$$\mathcal{R} = \{((n, u), (m, v)), ((m, v), (n, u)) \mid (m, vu) \models \Phi^{\sim}(n)\}$$

is a timed bisimulation. Note, first of all, that $\mathcal{R}$ is symmetric by definition. We proceed to prove that the relation $\mathcal{R}$ satisfies the clauses in Defn. 3.1. Assume to this end that $(n, u)\mathcal{R}(m, v)$ because $(m, vu) \models \Phi^{\sim}(n)$.

$\langle 2 \rangle .1$ Case $(n, u) \xrightarrow{a} (n', u')$.
Since $(n, u) \xrightarrow{a} (n', u')$ holds, there is an edge $e = (n, n_e, a, g_e, r_e) \in E(n, a)$ such that

(*i*) $u \models g_e$,

(*ii*) $n' = n_e$, and

(iii) $u' = u[r_e \mapsto 0]$.

Since $(m, vu) \models \Phi^{\sim}(n)$ and *(i)* holds, it follows that

$$(m, vu) \models \langle a \rangle r_e \ \underline{\texttt{in}}\ \Phi^{\sim}(n_e) \ .$$

This means that there is a state (m', v') such that $(m, v) \xrightarrow{a} (m', v')$ and $(m', v'(u[r_e \mapsto 0])) \models \Phi^{\sim}(n_e)$. For such an (m', v') we infer that $(n', u')\mathcal{R}(m', v')$ by *(ii)* and *(iii)*.

$\langle 2 \rangle.2$ Case $(n, u) \overset{\epsilon(d)}{\rightarrow} (n, u + d)$.
Since $(m, vu) \models \forall\!\!\!\forall \Phi^{\sim}(n)$, we have that

$$(m, (v + d)(u + d)) \models \Phi^{\sim}(n) \ .$$

By the definition of $\mathcal{R}$, it follows that $(n, u + d)\mathcal{R}(m, v + d)$, and, as $(m, v) \overset{\epsilon(d)}{\rightarrow} (m, v + d)$, we are done.

We now consider the case that $(m, v)\mathcal{R}(n, u)$ because $(m, vu) \models \Phi^{\sim}(n)$.

$\langle 2 \rangle.3$ Case $(m, v) \xrightarrow{a} (m', v')$.
Since $(m, vu) \models [a](\bigvee_{e \in E(n,a)} g_e \wedge (r_e \ \underline{\texttt{in}}\ \Phi^{\sim}(n_e)))$, we have that $(m', v'u) \models \bigvee_{e \in E(n,a)} g_e \wedge (r_e \ \underline{\texttt{in}}\ \Phi^{\sim}(n_e))$. It follows that, for some $e \in E(n, a)$,

(i) $u \models g_e$, and
(ii) $(m', v'(u[r_e \mapsto 0])) \models \Phi^{\sim}(n_e)$.

By *(i)* we have that $(n, u) \xrightarrow{a} (n_e, u[r_e \mapsto 0])$. By *(ii)* and the definition of $\mathcal{R}$, it follows that $(m', v')\mathcal{R}(n_e, u[r_e \mapsto 0])$ and we are done.

$\langle 2 \rangle.4$ Case $(m, v) \overset{\epsilon(d)}{\rightarrow} (m, v + d)$, where $d \in \mathbb{R}_{\geq 0}$. Since $(m, vu) \models \forall\!\!\!\forall \Phi^{\sim}(n)$, it follows that $(m, (v + d)(u + d)) \models \Phi^{\sim}(n)$. Thus $(m, v + d)\mathcal{R}(n, u + d)$ holds. Moreover $(n, u) \overset{\epsilon(d)}{\rightarrow} (n, u + d)$ and we are done.

This completes the proof of the theorem □

Note that the characteristic formula for timed bisimulation has size that is linear in that of the argument automaton. Laroussine, Larsen and Weise [19] have proposed a characteristic formula construction for timed automata up to timed bisimilarity. However, their construction is based on directly mimicking the standard construction from the untimed setting on the region graph, and the size of their characteristic formula is therefore linear in the size of the region graph. Unfortunately, however, as observed by Alur and Dill [2], the region graph has size that is exponential in the length of the clock constraints of the argument automaton.

Remark: LAROUSSINE, LARSEN AND WEISE [19] have shown that the logic L_ν characterizes timed bisimilarity over timed automata. This means that two timed automata are timed bisimilar if, and only if, they satisfy the same formulae in the logic L_ν. As a consequence of Thm. 4.1, we obtain that the existential delay modality $\exists$ is not necessary to obtain this logical characterization of timed bisimilarity.

As a further corollary of Thm. 4.1, and the EXPTIME upper bound on the complexity of model checking for L_ν [1], we have another proof of the following well-known result.

Corollary 4.2 *The problem of deciding whether two timed automata are timed bisimilar is decidable in exponential time.*

Characteristic Formula for Timed Simulation The characteristic formula for $\underset{\sim}{\sqsubset}_S$ is a minor variation on that for $\sim$, and is defined thus:

$$\Phi^{\underset{\sim}{\sqsubset}_S}(n) \quad \stackrel{\mathrm{def}}{=} \quad \Big(\bigwedge_{a\in\mathsf{Act}} \ \bigwedge_{e\in E(n,a)} g_e \Rightarrow (\langle a\rangle\ r_e\ \underline{\mathtt{in}}\ \Phi^{\underset{\sim}{\sqsubset}_S}(n_e))\Big) \ \wedge (\forall \Phi^{\underset{\sim}{\sqsubset}_S}(n)) \quad ,$$

where n is a node of A, $e = (n, n_e, a, r_e, g_e)$ and $E(n, a)$ denotes the set of a labelled edges from node n. We shall use $\mathcal{D}_A^{\underset{\sim}{\sqsubset}_S}$ to denote the declaration that consists of the equations above, one for each node of A.

A minor variation on the proof of Thm. 4.1 now establishes that:

Theorem 4.3 *Let A, B be timed automata with disjoint sets of clocks. Let n be a node of A and m be a node of B. Assume that u and v are valuations for the clocks of A and B, respectively. Then*

$$(n, u) \underset{\sim}{\sqsubset}_S (m, v) \ \textit{iff}\ (m, vu) \models \Phi^{\underset{\sim}{\sqsubset}_S}(n) \quad ,$$

where $(m, vu) \models \Phi^{\underset{\sim}{\sqsubset}_S}(n)$ holds with respect to the declaration $\mathcal{D}_A^{\underset{\sim}{\sqsubset}_S}$.

The full proof of this above theorem may be found in [25].

Corollary 4.4 *The problem of deciding whether $A \underset{\sim}{\sqsubset}_S B$ holds for timed automata A, B is decidable in exponential time.*

Timed ready simulation The characteristic formula for timed ready simulation is presented below:

$$\begin{aligned}\Phi^{\underset{\sim}{\sqsubset}_{RS}}(n) \quad \overset{\text{def}}{=} \quad & (\bigwedge_{a \in \mathsf{Act}} \bigwedge_{e \in E(n,a)} g_e \Rightarrow (\langle a \rangle \; r_e \; \underline{\mathtt{in}} \; \Phi^{\underset{\sim}{\sqsubset}_{RS}}(n_e))) \wedge \\ & (\bigwedge_{a \in \mathsf{Act}} (\overline{enabled(n,a)} \Rightarrow [a]\mathtt{ff}) \wedge \\ & \forall\!\!\!\forall \Phi^{\underset{\sim}{\sqsubset}_{RS}}(n)\end{aligned}$$

where n is a node of A, $e = (n, n_e, a, r_e, g_e)$ and we recall that $E(n,a)$ denotes the set of a labelled edges from node n. The notation $\overline{enabled(n,a)}$ stands for the negation of the formula $enabled(n,a)$ given in (1). We shall use $\mathcal{D}_A^{\underset{\sim}{\sqsubset}_{RS}}$ to denote the declaration that consists of the equations above, one for each node of A.

A minor variation on the proof of Thm. 4.1 now establishes that:

Theorem 4.5 *Let A, B be timed automata with disjoint sets of clocks. Let n be a node of A and m be a node of B. Assume that u and v are valuations for the clocks of A and B, respectively. Then*

$$(n,u) \underset{\sim}{\sqsubset}_{RS} (m,v) \textit{ iff } (m, vu) \models \Phi(n)^{\underset{\sim}{\sqsubset}_{RS}} \quad ,$$

where $(m, vu) \models \Phi^{\underset{\sim}{\sqsubset}_{RS}}(n)$ holds with respect to the declaration $\mathcal{D}_A^{\underset{\sim}{\sqsubset}_{RS}}$.

The full proof of this result may also be found in [25].

Corollary 4.6 *The problem of deciding whether $A \underset{\sim}{\sqsubset}_{RS} B$ holds for timed automata A, B is decidable in exponential time.*

Faster-than preorder In the characteristic formula constructions that we have presented so far no use was made of the existential modality $\exists\!\!\!\exists$ over delay transitions. The use of the $\exists\!\!\!\exists$ modality will instead play a crucial role in the definition of the characteristic property for the faster-than bisimulation preorder. This we now proceed to present.

For every node n in a timed automaton A, we define:

$$\begin{aligned}\Phi^{\underset{\sim}{\sqsubset}_{FT}}(n) \quad \overset{\text{def}}{=} \quad & (\bigwedge_{a \in \mathsf{Act}} \bigwedge_{e \in E(n,a)} g_e \Rightarrow (r_e \; \underline{\mathtt{in}} \; \exists\!\!\!\exists \langle a \rangle \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n_e))) \wedge \\ & (\bigwedge_{a \in \mathsf{Act}} [a](\bigvee_{e \in E(n,a)} g_e \wedge (r_e \; \underline{\mathtt{in}} \; \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n_e))) \wedge \\ & \forall\!\!\!\forall \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n) \quad ,\end{aligned}$$

where $e = (n, n_e, a, r_e, g_e)$ and $E(n, a)$ denotes the set of a labelled edges from node n. We shall use $\mathcal{D}_A^{\underset{\sim}{\sqsubset}_{FT}}$ to denote the declaration that consists of the equations above, one for each node of A.

Theorem 4.7 *Let A, B be timed automata with disjoint sets of clocks. Let n be a node of A and m be a node of B. Assume that u and v are valuations for the clocks of A and B, respectively. Then*

$$(n, u) \underset{\sim}{\sqsubset}_{FT} (m, v) \textit{ iff } (m, vu) \models \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n) \ ,$$

where $(m, vu) \models \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n)$ holds with respect to the declaration $\mathcal{D}_A^{\underset{\sim}{\sqsubset}_{FT}}$.

Proof: We separately prove that:

1. $(n, u) \underset{\sim}{\sqsubset}_{FT} (m, v)$ only if $(m, vu) \models \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n)$, and
2. if $(m, vu) \models \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n)$ then $(n, u) \underset{\sim}{\sqsubset}_{FT} (m, v)$.

$\langle 1 \rangle$ To show that the 'only if' implication holds, consider the relation $\vdash$ defined by structural induction on formulae thus: (m ranges over the nodes of B, and n over those of A)

$$\begin{array}{rcl}
(m, vu) \vdash \mathtt{tt} & \Leftrightarrow & \text{true} \\
(m, vu) \vdash \mathtt{ff} & \Leftrightarrow & \text{false} \\
(m, vu) \vdash \varphi \wedge \psi & \Leftrightarrow & (m, vu) \vdash \varphi \text{ and } (m, vu) \vdash \psi \\
(m, vu) \vdash \varphi \vee \psi & \Leftrightarrow & (m, vu) \vdash \varphi \text{ or } (m, vu) \vdash \psi \\
(m, vu) \vdash \exists\!\!\exists \varphi & \Leftrightarrow & \exists d \in \mathbb{R}_{\geq 0}.(m, (v + d)(u + d)) \vdash \varphi \\
(m, vu) \vdash \forall\!\!\forall \varphi & \Leftrightarrow & \forall d \in \mathbb{R}_{\geq 0}.(m, (v + d)(u + d)) \vdash \varphi \\
(m, vu) \vdash \langle a \rangle \varphi & \Leftrightarrow & \exists (m', v').(m, v) \xrightarrow{a} (m', v') \text{ and} \\
 & & (m', v'u) \vdash \varphi \\
(m, vu) \vdash [a] \varphi & \Leftrightarrow & \forall (m', v').(m, v) \xrightarrow{a} (m', v') \text{ implies} \\
 & & (m', v'u) \vdash \varphi \\
(m, vu) \vdash x \bowtie p & \Leftrightarrow & u(x) \bowtie p \\
(m, vu) \vdash x + p \bowtie y + q & \Leftrightarrow & u(x) + p \bowtie u(y) + q \\
(m, vu) \vdash x \ \underline{\mathtt{in}} \ \varphi & \Leftrightarrow & (m, vu') \vdash \varphi \text{ where } u' = u[\{x\} \mapsto 0] \\
(m, vu) \vdash \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n) & \Leftrightarrow & (n, u) \underset{\sim}{\sqsubset}_{FT} (m, v) \ .
\end{array}$$

We prove that $\vdash$ is a satisfiability relation. The only interesting part of the proof is to show that if $(m, vu) \vdash \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n)$, then $(m, vu) \vdash \mathcal{D}_A^{\underset{\sim}{\sqsubset}_{FT}}(\Phi^{\underset{\sim}{\sqsubset}_{FT}}(n))$. This we now proceed to prove.

Assume that $(m, vu) \vdash \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n)$ and let ξ be a conjunct of $\mathcal{D}_A^{\underset{\sim}{\sqsubset}_{FT}}(\Phi^{\underset{\sim}{\sqsubset}_{FT}}(n))$. We prove that $(m, vu) \vdash \xi$ holds for the first type of conjunct of the characteristic formula. The proof for the other two types of conjuncts is similar to the corresponding cases of the proof of Thm. 4.1.

– Case $\xi \equiv g_e \Rightarrow (r_e \ \underline{\texttt{in}}\ \exists\langle a\rangle \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n_e))$, where $a \in \mathsf{Act}$ and $e \in E(n,a)$. The claim is trivial if $u \not\models g_e$. Assume thus that $u \models g_e$ for some a-labelled edge e emanating from n. We wish to argue that

$$(m, vu) \quad \vdash \quad r_e \ \underline{\texttt{in}}\ \exists\langle a\rangle \ \underline{\texttt{in}}\ \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n_e) \ . \tag{6}$$

To this end, it is sufficient to prove that

$$(m, v(u[r_e \mapsto 0])) \quad \vdash \quad \exists\langle a\rangle \ \underline{\texttt{in}}\ \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n_e) \ . \tag{7}$$

Since $u \models g_e$ and $e \in E(n,a)$, it follows that $(n,u) \xrightarrow{a} (n_e, u[r_e \mapsto 0])$. As $(n,u)\underset{\sim}{\sqsubset}_{FT}(m,v)$ holds, there are a $d \in \mathbb{R}_{\geq 0}$ and a state (m', v') such that

- $(m,v) \xrightarrow{\epsilon(d)} (m, v+d)$,
- $(m, v+d) \xrightarrow{a} (m', v')$, and
- $(n_e, u[r_e \mapsto 0]) \xrightarrow{\epsilon(d)} (n_e, u[r_e \mapsto 0] + d)$, with

$$(n_e, (u[r_e \mapsto 0]) + d) \quad \underset{\sim}{\sqsubset}_{FT} \quad (m', v') \ . \tag{8}$$

Hence it is sufficient to prove that

$$(m, (v+d)(u[r_e \mapsto 0] + d)) \quad \vdash \quad \langle a\rangle \ \underline{\texttt{in}}\ \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n_e) \ . \tag{9}$$

Since $(m, v+d) \xrightarrow{a} (m', v')$, by the definition of $\vdash$ and by (8) it follows that $(m', v'(u[r_e \mapsto 0] + d)) \vdash \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n_e)$, from which we may derive that (9), (7) and finally (6) hold.

$\langle 2\rangle$ We now show that the 'if' implication holds. To this end we prove that the relation

$$\mathcal{R} = \{((n,u),(m,v)) | (m, vu) \models \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n)\}$$

is a faster-than bisimulation.

Assume that $(n,u)\mathcal{R}(m,v)$. We proceed to check that all of the defining properties of a faster-than bisimulation are met.

$\langle 2\rangle.1$ Case $(n,u) \xrightarrow{a} (n', u')$.

Then there is an edge $e \in E(n,a)$ such that

(*i*) $u \models g_e$,

(*ii*) $n' = n_e$, and

(*iii*) $u' = u[r_e \mapsto 0]$.

Since $(m, vu) \models \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n)$ and (i) holds, it follows that

$$(m, vu) \models r_e \ \underline{\mathtt{in}}\ \exists\langle a\rangle\Phi^{\underset{\sim}{\sqsubset}_{FT}}(n_e) \ .$$

Hence $(m, vu') \models \exists\langle a\rangle\Phi^{\underset{\sim}{\sqsubset}_{FT}}(n_e)$. This means that there are a delay $d \in \mathbb{R}_{\geq 0}$ and a state (m', v') such that

$$(m, v) \overset{\epsilon(d)}{\rightarrow} (m, v + d) \overset{a}{\rightarrow} (m', v')$$

and $(m', v'u'') \models \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n_e)$, where $u'' = u' + d$. By the definition of $\mathcal{R}$, we have $(n', u'')\mathcal{R}(m', v')$. Moreover, $(n', u') \overset{\epsilon(d)}{\rightarrow} (n', u'')$, and we are done.

$\langle 2\rangle$.2 Case $(n, u) \overset{\epsilon(d)}{\rightarrow} (n, u + d)$.

Since $(m, vu) \models \forall\Phi^{\underset{\sim}{\sqsubset}_{FT}}(n)$, we have that

$$(m, (v + d)(u + d)) \models \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n) \ .$$

By the definition of $\mathcal{R}$, it follows that $(n, u + d)\mathcal{R}(m, v + d)$ and, as $(m, v) \overset{\epsilon(d)}{\rightarrow} (m, v + d)$, we are done.

$\langle 2\rangle$.3 Case $(m, v) \overset{a}{\rightarrow} (m', v')$.

Since $(m, vu) \models [a](\bigvee_{e\in E(n,a)} g_e \wedge (r_e \ \underline{\mathtt{in}}\ \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n_e)))$, we have that $(m', v'u) \models \bigvee_{e\in E(n,a)} g_e \wedge (r_e \ \underline{\mathtt{in}}\ \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n_e))$. It follows that, for some $e \in E(n, a)$,

(i) $u \models g_e$, and

(ii) $(m', v'(u[r_e \mapsto 0])) \models \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n)$.

By (i) $(n, u) \overset{a}{\rightarrow} (n_e, u[r_e \mapsto 0])$. By (ii) and the definition of $\mathcal{R}$, it follows that $(n_e, u[r_e \mapsto 0])\mathcal{R}(m', v')$ and we are done.

$\langle 2\rangle$.4 Case $(m, v) \overset{\epsilon(d)}{\rightarrow} (m, v + d)$, where $d \in \mathbb{R}_{\geq 0}$.

Since $(m, vu) \models \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n)$, it follows that

$$(m, (v + d)(u + d)) \quad\models\quad \Phi^{\underset{\sim}{\sqsubset}_{FT}}(n) \ .$$

Thus it holds that $(n, u + d)\mathcal{R}(m, v + d)$. Moreover $(n, u) \overset{\epsilon(d)}{\rightarrow} (n, u + d)$ and we are done.

This completes the proof of the theorem. □

As for the previous behavioural relations studied in this section, we have that:

Corollary 4.8 *The problem of deciding whether $A \underset{\sim}{\sqsubset}_{FT} B$ holds for timed automata A, B is decidable in exponential time.*

5 Concluding Remarks

In their seminal paper [2], ALUR AND DILL proved that the problem of checking timed trace inclusion between a timed automaton A and a *deterministic* timed automaton B is PSPACE-complete. Following the classic automata theoretic approach, they achieved this result by reducing this problem to checking for the emptiness of the language accepted by a timed automaton that can be built in polynomial time from A and B. We shall now argue that the use of characteristic formulae offers an alternative approach to checking timed trace inclusion.

For the sake of clarity, we begin with some preliminary definitions.

Definition 5.1 *A* sequence of actions $\overline{\sigma} = a_1 a_2 a_3 \dots$ *is a possibly infinite sequence with* $a_i \in \mathsf{Act}$.

A sequence of time instants $\overline{t} = t_1 t_2 t_3 \dots$ *is a possibly infinite, nondecreasing sequence with* $t_i \in \mathbb{R}_{\geq 0}$.

A timed trace ρ *is a pair* $(\overline{\sigma}, \overline{t})$*, where* $\overline{\sigma}$ *is a sequence of actions and* $\overline{t}$ *is a sequence of time instants. The sequences* $\overline{\sigma}$ *and* $\overline{t}$ *are either both infinite or both finite and of the same length.*

In a timed trace ρ, the real number t_i denotes the absolute time instant at which action a_i occurs. In particular, t_1 always denotes the time instant at which the first action of the timed trace occurs. Assume, for the sake of simplicity, that every timed automaton is supplied with an extra clock x_0 which is never reset. Such a clock will measure the time that has elapsed since a timed automaton started its execution.

Definition 5.2 *Let* $A = (\mathsf{Act}, N, n_0, C, E)$ *be a timed automaton. We say that* $(\overline{\sigma}, \overline{t})$*, with* $\overline{\sigma} = a_1 a_2 \dots a_k$ *and* $\overline{t} = t_1 t_2 \dots t_k$ *(*$k \geq 0$*), is a* timed trace *of* A *iff*

$$(n_0, [C \mapsto 0]) \overset{\epsilon(d_1)}{\rightarrow} \overset{a_1}{\rightarrow} (n_1, v_1) \overset{\epsilon(d_2)}{\rightarrow} \overset{a_2}{\rightarrow} (n_2, v_2) \cdots (n_{k-1}, v_{k-1}) \overset{\epsilon(d_k)}{\rightarrow} \overset{a_k}{\rightarrow} (n_k, v_k)$$

for some delays $d_1, d_2, \dots, d_k \in \mathbb{R}_{\geq 0}$*, valuations* $v_1, v_2, \dots v_k$ *such that* $t_i = v_i(x_0)$ *for every* $i \in \{1, \dots, k\}$*, and nodes* $n_1, \dots, n_k$ *of* A*. The set of timed traces of* A *will be written traces*(A)*.*

Let A *and* B *be timed automata. We write* $A \sqsubseteq_T B$ *iff traces*$(A) \subseteq$ *traces*(B)*.*

As shown by ALUR AND DILL [2], the relation $\sqsubseteq_T$ is undecidable for timed automata. It becomes decidable if the specification automaton B is deterministic.

Definition 5.3 *A timed automaton is* deterministic *iff for every node n, action $a \in \mathsf{Act}$ and distinct edges $e, e' \in E(n, a)$, the guards g_e and $g_{e'}$ are disjoint, i.e., $g_e \wedge g_{e'}$ is unsatisfiable.*

A standard argument, that may be found in [25], now suffices to establish the following result. (See, e.g., [11] for a similar statement in the classic, untimed setting.)

Proposition 5.4 *Let A and B be timed automata. Then the following statements hold:*

1. *$A \underset{\sim}{\sqsubset}_S B$ implies $A \underset{\sim}{\sqsubset}_T B$;*

2. *$A \underset{\sim}{\sqsubset}_T B$ implies $A \underset{\sim}{\sqsubset}_S B$, if B is deterministic.*

The import of the above result is that, if B is a deterministic timed automaton, checking whether the set of timed traces of a timed automaton A is included in that of B can be reduced to checking whether B satisfies the characteristic formula of A with respect to timed simulation.

The feasibility of the approach based on establishing behavioural relations for timed automata via model checking characteristic formulae needs to be established experimentally. The master's thesis [25] describes a prototype implementation of a tool for checking behavioural relations for timed automata based on the theory presented in this study. This tool is rather inefficient, and cannot handle reasonably sized examples. However, we expect that an efficient tool for verifying behavioural equivalences for timed automata can be obtained by implementing a front-end to the L_ν-model checker CMC [18] that generates the different characteristic formula constructions we have presented.

References

[1] L. Aceto and F. Laroussinie, *Is your model checker on time? On the complexity of model checking for timed modal logics*, in Mathematical foundations of computer science 1999 (Szklarska Poreba), Springer-Verlag, Berlin, 1999, pp. 125–136.

[2] R. Alur and D. L. Dill, *A theory of timed automata*, Theoretical Comput. Sci., 126 (1994), pp. 183–235. Fundamental Study.

[3] S. Arun-Kumar and M. Hennessy, *An efficiency preorder for processes*, Acta Informatica, 29 (1992), pp. 737–760.

[4] J. Baeten and J. Klop, eds., *Proceedings CONCUR 90,* Amsterdam, vol. 458 of Lecture Notes in Computer Science, Springer-Verlag, 1990.

[5] B. Bloom, S. Istrail, and A. R. Meyer, *Bisimulation can't be traced*, J. Assoc. Comput. Mach., 42 (1995), pp. 232–268.

[6] M. Browne, E. Clarke, and O. Grümberg, *Characterizing finite Kripke structures in propositional temporal logic*, Theoretical Comput. Sci., 59 (1988), pp. 115–131.

[7] E. Clarke and E. Emerson, *Design and synthesis of synchronization skeletons using branching-time temporal logic*, in Proceedings of the Workshop on Logic of Programs, Yorktown Heights, D. Kozen, ed., vol. 131 of Lecture Notes in Computer Science, Springer-Verlag, 1981, pp. 52–71.

[8] E. Clarke, O. Grümberg, and D. Peled, *Model Checking*, MIT Press, 2000.

[9] R. Cleaveland and B. Steffen, *Computing behavioral relations, logically*, in ICALP '91: Automata, Languages and Programming, J. L. Albert, B. Monien, and M. R. Artalejo, eds., vol. 510 of Lecture Notes in Computer Science, Madrid, July 1991, Springer-Verlag, pp. 127–138.

[10] ——, *A linear-time model-checking algorithm for the alternation-free modal μ-calculus*, Formal Methods in Systems Design, 2 (1993), pp. 121–147.

[11] J. Engelfriet, *Determinacy → (observation equivalence = trace equivalence)*, Theoretical Comput. Sci., 36 (1985), pp. 21–25.

[12] R. J. van Glabbeek, *The linear time – branching time spectrum*, in Baeten and Klop [4], pp. 278–297.

[13] S. Graf and J. Sifakis, *A modal characterization of observational congruence on finite terms of CCS*, Information and Control, 68 (1986), pp. 125–145.

[14] M. Hennessy and R. Milner, *Algebraic laws for nondeterminism and concurrency*, J. Assoc. Comput. Mach., 32 (1985), pp. 137–161.

[15] A. Ingólfsdóttir, J. C. Godskesen, and M. Zeeberg, *Fra Hennessy-Milner logik til CCS-processer*, Master's thesis, Department of Computer Science, Aalborg University, 1987. In Danish.

[16] R. Keller, *Formal verification of parallel programs*, Comm. ACM, 19 (1976), pp. 371–384.

[17] D. Kozen, *Results on the propositional mu-calculus*, Theoretical Comput. Sci., 27 (1983), pp. 333–354.

[18] F. Laroussinie and K. G. Larsen, *CMC: A tool for compositional model-checking of real-time systems*, in Proc. IFIP Joint Int. Conf. Formal Description Techniques & Protocol Specification, Testing, and Verification (FORTE-PS TV'98), Kluwer Academic Publishers, 1998, pp. 439–456.

[19] F. Laroussinie, K. G. Larsen, and C. Weise, *From timed automata to logic - and back*, in Proceedings of the 20^{th} Symposium on Mathematical Foundations of Computer Science, vol. 969 of Lecture Notes in Computer Science, Springer-Verlag, 1995, pp. 529–540.

[20] K. G. Larsen and A. Skou, *Bisimulation through probabilistic testing*, Information and Computation, 94 (1991), pp. 1–28.

[21] R. Milner, *An algebraic definition of simulation between programs*, in Proceedings 2^{nd} Joint Conference on Artificial Intelligence, William Kaufmann, 1971, pp. 481–489. Also available as Report No. CS-205, Computer Science Department, Stanford University.

[22] ——, *Communication and Concurrency*, Prentice Hall, Englewood Cliffs, 1989.

[23] F. Moller and C. Tofts, *Relating processes with respect to speed*, in CONCUR '91: 2^{nd} International Conference on Concurrency Theory, J. C. M. Baeten and J. F. Groote, eds., vol. 527 of Lecture Notes in Computer Science, Amsterdam, The Netherlands, 26–29 Aug. 1991, Springer-Verlag, pp. 424–438.

[24] D. Park, *Concurrency and automata on infinite sequences*, in 5^{th} GI Conference, Karlsruhe, Germany, P. Deussen, ed., vol. 104 of Lecture Notes in Computer Science, Springer-Verlag, 1981, pp. 167–183.

[25] M. L. Pedersen and J. Poulsen, *Model-checking characteristic formulae — a method for proving timed behavioural relations*, Master's thesis, Department of Computer Science, Aalborg University, June 1999.

[26] B. Steffen and A. Ingólfsdóttir, *Characteristic formulae for processes with divergence*, Information and Computation, 110 (1994), pp. 149–163.

[27] A. Tarski, *A lattice-theoretical fixpoint theorem and its applications*, Pacific Journal of Mathematics, 5 (1955).

[28] Y. Wang, *Real-time behaviour of asynchronous agents*, in Baeten and Klop [4], pp. 502–520.

Recent BRICS Report Series Publications

RS-00-23 Luca Aceto, Anna Ingólfsdóttir, Mikkel Lykke Pedersen, and Jan Poulsen. *Characteristic Formulae for Timed Automata*. September 2000. 23 pp.

RS-00-22 Thomas S. Hune and Anders B. Sandholm. *Using Automata in Control Synthesis — A Case Study*. September 2000. 20 pp. Appears in Maibaum, editor, *Fundamental Approaches to Software Engineering: First International Conference*, FASE '00 Proceedings, LNCS 1783, 2000, pages 349–362.

RS-00-21 M. Oliver Möller and Rajeev Alur. *Heuristics for Hierarchical Partitioning with Application to Model Checking*. August 2000. 30 pp.

RS-00-20 Luca Aceto, Willem Jan Fokkink, and Anna Ingólfsdóttir. *2-Nested Simulation is not Finitely Equationally Axiomatizable*. August 2000. 13 pp.

RS-00-19 Vinodchandran N. Variyam. *A Note on* $\mathrm{NP} \cap \mathrm{coNP}/\mathrm{poly}$. August 2000. 7 pp.

RS-00-18 Federico Crazzolara and Glynn Winskel. *Language, Semantics, and Methods for Cryptographic Protocols*. August 2000. ii+42 pp.

RS-00-17 Thomas S. Hune. *Modeling a Language for Embedded Systems in Timed Automata*. August 2000. 26 pp. Earlier version entitled *Modelling a Real-Time Language* appeared in Gnesi and Latella, editors, *Fourth International ERCIM Workshop on Formal Methods for Industrial Critical Systems*, FMICS '99 Proceedings of the FLoC Workshop, 1999, pages 259–282.

RS-00-16 Jiří Srba. *Complexity of Weak Bisimilarity and Regularity for BPA and BPP*. June 2000. 20 pp. To appear in Aceto and Victor, editors, *Expressiveness in Concurrency: Fifth International Workshop EXPRESS '00 Proceedings*, ENTCS, 2000.

RS-00-15 Daniel Damian and Olivier Danvy. *Syntactic Accidents in Program Analysis: On the Impact of the CPS Transformation*. June 2000. Extended version of an article to appear in *Proceedings of the fifth ACM SIGPLAN International Conference on Functional Programming*, 2000.