

LUCA ACETO

ANNA INGÓLFSDÓTTIR

MIKKEL LYKKE PEDERSEN

JAN POULSEN

Characteristic formulae for timed automata

Informatique théorique et applications, tome 34, n° 6 (2000),
p. 565-584

http://www.numdam.org/item?id=ITA_2000__34_6_565_0

© AFCET, 2000, tous droits réservés.

L'accès aux archives de la revue « Informatique théorique et applications » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

CHARACTERISTIC FORMULAE FOR TIMED AUTOMATA

LUCA ACETO¹, ANNA INGÓLFSÐÓTTIR¹,
MIKKEL LYKKE PEDERSEN¹ AND JAN POULSEN¹

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.

Mathematics Subject Classification. 68Q60, 68Q10.

INTRODUCTION

There are two main methodologies for the formal verification of reactive systems, viz. *model checking* and *refinement 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 refinement 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 two state-transition graphs is usually expressed in terms of a behavioural relation in the linear time-branching time spectrum [12].

¹ BRICS (Basic Research in Computer Science, Centre of the Danish National Research Foundation), Department of Computer Science, Aalborg University, Fr. Bajersvej 7E, 9220 Aalborg Ø, Denmark.

One of the bridges between these two approaches to verification is provided by the notion of *characteristic formula* [13, 15, 27]. 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 *et al.* 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 Ingólfssdóttir *et al.* [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 [29], timed versions of the simulation [22] and ready simulation [5, 21] preorders and for the faster-than preorder [24]. 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 refinement verification in the timed setting. A prototype tool based on the theory we present in this study is described in [26].

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 on 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 [27] Ingólfssdóttir and Steffen showed 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 *et al.* [6] have shown how to characterize Kripke structures in the logic CTL [7] up to bisimilarity.

Using the characteristic formula constructions presented in this paper, it is not hard to show that checking any of the relations of timed bisimilarity, timed (ready) simulation and the faster-than preorder can be done in exponential time over timed automata. These results are mentioned here just as an application of the general characteristic formulae constructions, and are not really novel. Indeed, Theorem 4.8 in [20] shows that any behavioural relation lying between the (timed) simulation preorder and bisimilarity is EXPTIME-hard over timed automata. Moreover, Remark 4.9 in *op. cit.* states that, in light of results previously published in [1], deciding timed bisimulation and simulation over timed automata are, in fact, EXPTIME-complete problems.

ROADMAP OF THE PAPER. After a brief review of background material on timed automata and the logic L_ν (Sect. 1), we present the timed behavioural relations for

which we offer characteristic formula constructions (Sect. 2). The constructions of the characteristic formulae are the topic of Section 3, 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. 4).

1. 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 1.1. 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 $s \xrightarrow{\epsilon(d)} s'$ and $s \xrightarrow{\epsilon(d)} s''$, then $s' = s''$;
- (TIME ADDITIVITY) for every $s, s'' \in \mathcal{S}$ and $d_1, d_2 \in \mathbb{R}_{\geq 0}$, $s \xrightarrow{\epsilon(d_1+d_2)} s''$ iff $s \xrightarrow{\epsilon(d_1)} s' \xrightarrow{\epsilon(d_2)} s''$, for some $s' \in \mathcal{S}$;
- (0-DELAY) for every $s, s' \in \mathcal{S}$, $s \xrightarrow{\epsilon(0)} s'$ iff $s = s'$.

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

The axioms of time determinism, time additivity and 0-delay are standard in the literature on Yi's TCCS (see, *e.g.* [29]).

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 $C \setminus C'$.

Definition 1.2. A *timed automaton* is a quintuple $A = (\text{Act}, N, n_0, C, E)$ where N is a finite set of *nodes*, n_0 is the *initial node*, C is a finite set of *clocks*, and $E \subseteq N \times N \times \text{Act} \times 2^C \times \mathcal{B}(C)$ 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 $\mathcal{T}_A = (\mathcal{S}_A, \mathcal{L}, s_A^0, \longrightarrow)$, where \mathcal{S}_A is the set of states of A , s_A^0 is the initial state of A , and \longrightarrow is the transition relation defined as follows:

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

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

where $a \in \text{Act}$ 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 1.3 (Syntax of L_ν). Let K be a finite set of formula clocks, \mathbf{Id} a finite set of identifiers and k a non-negative integer. The set L_ν of formulae over K , \mathbf{Id} and largest constant k is generated by the abstract syntax below, with φ and ψ ranging over L_ν :

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

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

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

The logic L_ν allows for the recursive definition of formulae by including a finite set \mathbf{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 $Z \stackrel{\text{def}}{=} \varphi$ denote $\mathcal{D}(Z) = \varphi$. Intuitively Z will stand for the largest solution of the equation $Z \stackrel{\text{def}}{=} \varphi$. 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 1.4 (Semantics of L_ν). 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:

$$\begin{aligned}
(n, vu) \models_{\mathcal{D}} \mathbf{t} &\Rightarrow \text{true} \\
(n, vu) \models_{\mathcal{D}} \mathbf{ff} &\Rightarrow \text{false} \\
(n, vu) \models_{\mathcal{D}} \varphi \wedge \psi &\Rightarrow (n, vu) \models_{\mathcal{D}} \varphi \text{ and } (n, vu) \models_{\mathcal{D}} \psi \\
(n, vu) \models_{\mathcal{D}} \varphi \vee \psi &\Rightarrow (n, vu) \models_{\mathcal{D}} \varphi \text{ or } (n, vu) \models_{\mathcal{D}} \psi \\
(n, vu) \models_{\mathcal{D}} \exists \varphi &\Rightarrow \exists d \in \mathbb{R}_{\geq 0}. (n, (v+d)(u+d)) \models_{\mathcal{D}} \varphi \\
(n, vu) \models_{\mathcal{D}} \forall \varphi &\Rightarrow \forall d \in \mathbb{R}_{\geq 0}. (n, (v+d)(u+d)) \models_{\mathcal{D}} \varphi \\
(n, vu) \models_{\mathcal{D}} \langle a \rangle \varphi &\Rightarrow \exists (n', v'). (n, v) \xrightarrow{a} (n', v') \text{ and } (n', v'u) \models_{\mathcal{D}} \varphi \\
(n, vu) \models_{\mathcal{D}} [a] \varphi &\Rightarrow \forall (n', v'). (n, v) \xrightarrow{a} (n', v') \text{ implies } (n', v'u) \models_{\mathcal{D}} \varphi \\
(n, vu) \models_{\mathcal{D}} x \bowtie p &\Rightarrow u(x) \bowtie p \\
(n, vu) \models_{\mathcal{D}} x + p \bowtie y + q &\Rightarrow u(x) + p \bowtie u(y) + q \\
(n, vu) \models_{\mathcal{D}} x \underline{\text{in}} \varphi &\Rightarrow (n, vu') \models_{\mathcal{D}} \varphi \text{ where } u' = u[\{x\} \mapsto 0] \\
(n, vu) \models_{\mathcal{D}} Z &\Rightarrow (n, vu) \models_{\mathcal{D}} \mathcal{D}(Z).
\end{aligned}$$

Any relation satisfying the above implications is referred to as a satisfiability relation. From standard fixed-point theory [28] we have that $\models_{\mathcal{D}}$ is the union of all satisfiability relations.

Remark 1.5. Although the syntax of the atomic formulae of $\mathcal{B}(C)$ differs from that of the atomic formulae in L_{ν} , it is easy to see that every guard in $\mathcal{B}(C)$ can be expressed in L_{ν} .

2. TIMED BEHAVIOURAL RELATIONS

In the untimed setting various behavioural relations over processes have been proposed (see, *e.g.* [12] for an encyclopedic 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 behavioural relations. One such relation is the *faster-than bisimulation* from [24], 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 [23].)

We now proceed to review the timed behavioural relations over TLTSs studied in this paper. The notion of timed bisimulation stems from [29]. It is the obvious adaptation to the timed setting of the classic definition due to Park [25].

Definition 2.1. 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:

$$- \text{ If } s_1 \xrightarrow{\alpha} s'_1 \text{ then } s_2 \xrightarrow{\alpha} s'_2 \text{ for some } s'_2 \text{ such that } s'_1 \mathcal{R} s'_2.$$

A *timed bisimulation* is a symmetric timed simulation.

For states s_1, s_2 , we write $s_1 \sqsubseteq_{\mathcal{S}} s_2$ (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 [5,21]. 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 2.2. 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 $s_1 \mathcal{R} s_2$, $a \in \text{Act}$ 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 $s_2 \xrightarrow{\alpha}$ then $s_1 \xrightarrow{\alpha}$.

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

Moller and Tofts [24] 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 2.3. 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 $s_1 \mathcal{R} s_2$, $a \in \text{Act}$ 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 $s_1 \sqsubseteq_{FT} s_2$ iff there exists a faster-than bisimulation \mathcal{R} with $s_1 \mathcal{R} s_2$.

It is well-known that \sqsubseteq_* ($*$ $\in \{S, RS, FT\}$) is a preorder. Moreover \sqsubseteq_S is the largest timed simulation, \sqsubseteq_{RS} is the largest timed ready simulation, and \sqsubseteq_{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 2.4. Let A, B be two timed automata. Then, for every relation $\mathcal{R} \in \{\sqsubseteq_S, \sqsubseteq_{RS}, \sqsubseteq_{FT}, \sim\}$, 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.

3. CHARACTERISTIC FORMULA CONSTRUCTIONS

We now offer general characteristic formula constructions in terms of L_ν for each of the timed behavioural relations introduced in Section 2. 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 $r = \{x_1, \dots, x_k\}$, we use the abbreviation $r \text{ \underline{in} } \varphi$ to stand for the formula inductively defined thus:

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

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

The expression $g \Rightarrow \varphi$ will stand for $\bar{g} \vee \varphi$, where \bar{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:

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

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 $\text{enabled}(n, a)$ describes when action a can be performed from a state of the form (n, v) . The negation of the expression $\text{enabled}(n, a)$ 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 $\text{enabled}(n, a)$ can also be expressed in L_ν . Finally, we recall that, as usual, an empty disjunction stands for \mathbf{ff} and an empty conjunction is equivalent to \mathbf{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 Definition 2.1. 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 fact that arbitrary delays are allowed in the node.

The resulting characteristic formula is presented below, where we consider each $\Phi(n)^\sim$ to be an identifier. The formula consists of three sets of conjuncts, each associated to one of the above properties, for each node n of a timed automaton A :

$$\begin{aligned} \Phi^{\sim}(n) \stackrel{\text{def}}{=} & \left(\bigwedge_{a \in \text{Act}} \bigwedge_{e \in E(n,a)} g_e \Rightarrow (\langle a \rangle r_e \text{ in } \Phi^{\sim}(n_e)) \right) \wedge \\ & \bigwedge_{a \in \text{Act}} [a] \left(\bigvee_{e \in E(n,a)} g_e \wedge (r_e \text{ in } \Phi^{\sim}(n_e)) \right) \wedge \\ & \forall \Phi^{\sim}(n) \end{aligned}$$

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 . We shall use \mathcal{D}_A^{\sim} to denote the declaration that consists of the equations above, one for each node of A .

Theorem 3.1. *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*

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

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)

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

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.

(1).1 Case $\xi \equiv g_e \Rightarrow (\langle a \rangle r_e \text{ in } \Phi^\sim(n_e))$, where $a \in \text{Act}$ 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

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

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

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

By the definition of \vdash , it follows that

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

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

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

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

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

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

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

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

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

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

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

from which (3) finally follows.

(1).3 Case $\xi \equiv \forall \Phi^\sim(n)$.

Assume that $d \in \mathbb{R}_{\geq 0}$. We prove that

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

Since $(m, v) \xrightarrow{\epsilon(d)} (m, v + d)$, $(n, u) \xrightarrow{\epsilon(d)} (n, u + d)$, 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 (1) is now complete.

(2) 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 Definition 2.1.

Assume to this end that $(n, u)\mathcal{R}(m, v)$ because $(m, vu) \models \Phi^\sim(n)$.

(2).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

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

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).

(2).2 Case $(n, u) \xrightarrow{\epsilon(d)} (n, u + d)$.

Since $(m, vu) \models \forall \Phi^\sim(n)$, 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 $(m, v) \xrightarrow{\epsilon(d)} (m, v + d)$, we are done.

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

(2).3 Case $(m, v) \xrightarrow{a} (m', v')$.

Since $(m, vu) \models [a](\bigvee_{e \in E(n, a)} g_e \wedge (r_e \text{ in } \Phi^\sim(n_e)))$, we have that $(m', v'u) \models$

$\bigvee_{e \in E(n, a)} g_e \wedge (r_e \text{ in } \Phi^\sim(n_e))$. 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.

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

This completes the proof of the theorem. □

Note that, since we assume that the set of actions Act is fixed, the characteristic formula for timed bisimulation has size that is linear in that of the argument automaton. Laroussinie *et al.* [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 3.2. In defining our characteristic formula construction for timed bisimilarity over timed automata, we have used the logic L_ν as presented in [19]. As already remarked, this allows us to construct characteristic formulae for timed bisimulation whose size is linear in that of the argument automaton, if the set of actions Act is fixed. This result can be extended to the case in which the set of actions is not fixed by adding atomic propositions to L_ν , following the example of [27]. These new atomic propositions take the form P_A , where A is a subset of Act , and hold in states whose initially enabled actions are contained in A . The easy modification of $\Phi^\sim(n)$ using these atomic propositions is left as an exercise for the reader.

Remark 3.3. Laroussinie *et al.* [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 Theorem 3.1, we obtain that the existential delay modality \exists is not necessary to obtain this logical characterization of timed bisimilarity.

As a further corollary of Theorem 3.1, 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 3.4. *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 \approx_S is a minor variation on that for \sim , and is defined thus:

$$\Phi^{\approx_S}(n) \stackrel{\text{def}}{=} \left(\bigwedge_{a \in \text{Act}} \bigwedge_{e \in E(n,a)} g_e \Rightarrow \left(\langle a \rangle r_e \text{ in } \Phi^{\approx_S}(n_e) \right) \right) \wedge \left(\mathbb{W}\Phi^{\approx_S}(n) \right),$$

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 $\mathcal{D}_A^{\sqsubseteq S}$ to denote the declaration that consists of the equations above, one for each node of A .

A minor variation on the proof of Theorem 3.1 now establishes that:

Theorem 3.5. *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*

$$(n, u) \sqsubseteq_S (m, v) \text{ iff } (m, vu) \models \Phi^{\sqsubseteq S}(n),$$

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

The full proof of this above theorem may be found in [26].

Corollary 3.6. *The problem of deciding whether $A \sqsubseteq_S B$ holds for timed automata A, B is decidable in exponential time.*

CHARACTERISTIC FORMULA FOR TIMED READY SIMULATION. The characteristic formula for timed ready simulation is presented below:

$$\begin{aligned} \Phi^{\sqsubseteq RS}(n) \stackrel{\text{def}}{=} & \left(\bigwedge_{a \in \text{Act}} \bigwedge_{e \in E(n, a)} g_e \Rightarrow \left((a) r_e \text{ in } \Phi^{\sqsubseteq RS}(n_e) \right) \right) \wedge \\ & \left(\bigwedge_{a \in \text{Act}} \overline{\text{enabled}(n, a)} \Rightarrow [a] \mathbf{f} \right) \wedge \\ & \forall \Phi^{\sqsubseteq RS}(n) \end{aligned}$$

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 $\overline{\text{enabled}(n, a)}$ stands for the negation of the formula $\text{enabled}(n, a)$ given in (1). We shall use $\mathcal{D}_A^{\sqsubseteq RS}$ to denote the declaration that consists of the equations above, one for each node of A .

A minor variation on the proof of Theorem 3.1 now establishes that:

Theorem 3.7. *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*

$$(n, u) \sqsubseteq_{RS} (m, v) \text{ iff } (m, vu) \models \Phi(n)^{\sqsubseteq RS},$$

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

The full proof of this result may also be found in [26].

Corollary 3.8. *The problem of deciding whether $A \sqsubseteq_{RS} B$ holds for timed automata A, B is decidable in exponential time.*

CHARACTERISTIC FORMULA FOR THE FASTER-THAN PREORDER. In the characteristic formula constructions that we have presented so far no use was made of the existential modality \exists over delay transitions. The use of the \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:

$$\begin{aligned} \Phi^{\sqsubset FT}(n) \stackrel{\text{def}}{=} & \left(\bigwedge_{a \in \text{Act}} \bigwedge_{e \in E(n,a)} g_e \Rightarrow \left(r_e \text{ in } \exists(a) \Phi^{\sqsubset FT}(n_e) \right) \right) \wedge \\ & \left(\bigwedge_{a \in \text{Act}} [a] \left(\bigvee_{e \in E(n,a)} g_e \wedge \left(r_e \text{ in } \Phi^{\sqsubset FT}(n_e) \right) \right) \right) \wedge \\ & \forall \Phi^{\sqsubset FT}(n), \end{aligned}$$

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 $\mathcal{D}_A^{\sqsubset FT}$ to denote the declaration that consists of the equations above, one for each node of A .

Theorem 3.9. *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*

$$(n, u) \sqsubset_{FT} (m, v) \text{ iff } (m, vu) \models \Phi^{\sqsubset FT}(n),$$

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

Proof. We separately prove that:

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

- (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)

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

We prove that \vdash is a satisfiability relation. The only interesting part of the proof is to show that if $(m, vu) \vdash \Phi^{\square}_{FT}(n)$, then it holds that

$$(m, vu) \vdash \mathcal{D}_A^{\square}_{FT} \left(\Phi^{\square}_{FT}(n) \right).$$

This we now proceed to prove. Assume that $(m, vu) \vdash \Phi^{\square}_{FT}(n)$ and let ξ be a conjunct of $\mathcal{D}_A^{\square}_{FT}(\Phi^{\square}_{FT}(n))$. 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 Theorem 3.1.

- Case $\xi \equiv g_e \Rightarrow (r_e \mathbf{in} \exists \langle a \rangle \Phi^{\square}_{FT}(n_e))$, where $a \in \text{Act}$ 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

$$(m, vu) \vdash r_e \mathbf{in} \exists \langle a \rangle \mathbf{in} \Phi^{\square}_{FT}(n_e). \quad (6)$$

To this end, it is sufficient to prove that

$$(m, v(u[r_e \mapsto 0])) \vdash \exists \langle a \rangle \mathbf{in} \Phi^{\square}_{FT}(n_e). \quad (7)$$

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 $(n, u) \sqsubseteq_{FT} (m, v)$ 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

$$(n_e, (u[r_e \mapsto 0] + d)) \sqsubseteq_{FT} (m', v'). \quad (8)$$

Hence it is sufficient to prove that

$$(m, (v + d)(u[r_e \mapsto 0] + d)) \vdash (a) \text{ in } \Phi^{\square}_{FT}(n_e). \quad (9)$$

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

(2) We now show that the ‘if’ implication holds. To this end we prove that the relation

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

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.

(2).1 Case $(n, u) \xrightarrow{\alpha} (n', u')$.

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

$$(m, vu) \models r_e \text{ in } \exists \langle a \rangle \Phi^{\square}_{FT}(n_e).$$

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

$$(m, v) \xrightarrow{\epsilon(d)} (m, v + d) \xrightarrow{\alpha} (m', v')$$

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

(2).2 Case $(n, u) \xrightarrow{\epsilon(d)} (n, u + d)$.

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

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

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

(2).3 Case $(m, v) \xrightarrow{\alpha} (m', v')$.

Since $(m, vu) \models [a](\bigvee_{e \in E(n,a)} g_e \wedge (r_e \text{ in } \Phi^{\sqsubseteq_{FT}}(n_e)))$, we have that

$(m', v'u) \models \bigvee_{e \in E(n,a)} g_e \wedge (r_e \text{ in } \Phi^{\sqsupset_{FT}}(n_e))$. It follows that, for

some $e \in E(n, a)$,

(i) $u \models g_e$, and

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

By (i) $(n, u) \xrightarrow{a} (n_e, u[r_e \mapsto 0])$. 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.

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

Since $(m, vu) \models \Phi^{\sqsubseteq_{FT}}(n)$, it follows that

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

Thus it holds that $(n, u + d) \mathcal{R}(m, v + d)$. Moreover $(n, u) \xrightarrow{\epsilon(d)} (n, u + d)$ 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 3.10. *The problem of deciding whether $A \sqsubseteq_{FT} B$ holds for timed automata A, B is decidable in exponential time.*

4. FURTHER 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 4.1. A *sequence of actions* $\bar{\sigma} = a_1 a_2 a_3 \dots$ is a possibly infinite sequence with $a_i \in \text{Act}$.

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

A *timed trace* ρ is a pair $(\bar{\sigma}, \bar{t})$, where $\bar{\sigma}$ is a sequence of actions and \bar{t} is a sequence of time instants. The sequences $\bar{\sigma}$ and \bar{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 4.2. Let $A = (\text{Act}, N, n_0, C, E)$ be a timed automaton. We say that $(\bar{\sigma}, \bar{t})$, with $\bar{\sigma} = a_1 a_2 \dots a_k$ and $\bar{t} = t_1 t_2 \dots t_k$ ($k \geq 0$), is a *timed trace* of A iff

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

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 $\text{traces}(A)$.

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

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 4.3. A timed automaton is *deterministic* iff for every node n , action $a \in \text{Act}$ 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 [26], now suffices to establish the following result. (See, *e.g.* [11] for a similar statement in the classic, untimed setting.)

Proposition 4.4. *Let A and B be timed automata. Then the following statements hold:*

1. $A \sqsubseteq_S B$ implies $A \sqsubseteq_T B$;
2. $A \sqsubseteq_T B$ implies $A \sqsubseteq_S B$, 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 [26] 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. Since the tool CMC already

supports timed bisimilarity checking using a different approach, it would be interesting to compare the performance of the two techniques on benchmark examples.

Acknowledgements. We thank the anonymous referees for their constructive comments that helped us improve the paper.

REFERENCES

- [1] L. Aceto and F. Laroussinie, *Is your model checker on time? On the complexity of model checking for timed modal logics*. Springer-Verlag, Berlin, *Math. Foundations of Comput. Sci.* **1999** (Szklarska Poreba) (1999) 125-136.
- [2] R. Alur and D.L. Dill, A theory of timed automata. *Theoret. Comput. Sci.* **126** (1994) 183-235. (Fundamental Study).
- [3] S. Arun-Kumar and M. Hennessy, An efficiency preorder for processes. *Acta Inform.* **29** (1992) 737-760.
- [4] J. Baeten and J. Klop, in *Proc. CONCUR 90*, Amsterdam. Springer-Verlag, *Lecture Notes in Comput. Sci.* **458** (1990).
- [5] B. Bloom, S. Istrail and A.R. Meyer, Bisimulation can't be traced. *J. Assoc. Comput. Mach.* **42** (1995) 232-268.
- [6] M. Browne, E. Clarke and O. Grümberg, Characterizing finite Kripke structures in propositional temporal logic. *Theoret. Comput. Sci.* **59** (1988) 115-131.
- [7] E. Clarke and E. Emerson, Design and synthesis of synchronization skeletons using branching-time temporal logic, in *Proc. of the Workshop on Logic of Programs*, Yorktown Heights, edited by D. Kozen. Springer-Verlag, *Lecture Notes in Comput. Sci.* **131** (1981) 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*, edited by J.L. Albert, B. Monien and M.R. Artaleso. Springer-Verlag, *Lecture Notes in Comput. Sci.* **510** (1991) 127-138.
- [10] ———, A linear-time model-checking algorithm for the alternation-free modal μ -calculus. *Formal Methods in Systems Design* **2** (1993) 121-147.
- [11] J. Engelfriet, Determinacy \rightarrow (observation equivalence = trace equivalence). *Theoret. Comput. Sci.* **36** (1985) 21-25.
- [12] R.J.V. 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. *Inform. and Control* **68** (1986) 125-145.
- [14] M. Hennessy and R. Milner, Algebraic laws for nondeterminism and concurrency. *J. Assoc. Comput. Mach.* **32** (1985) 137-161.
- [15] A. Ingólfssdóttir, J.C. Godskesen and M. Zeeberg, *Fra Hennessy-Milner logik til CCS-processor*. Master's thesis, Department of Computer Science, Aalborg University (1987).
- [16] R. Keller, Formal verification of parallel programs. *Comm. ACM* **19** (1976) 371-384.
- [17] D. Kozen, Results on the propositional mu-calculus. *Theoret. Comput. Sci.* **27** (1983) 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) 439-456.

- [19] F. Laroussinie, K.G. Larsen and C. Weise, From timed automata to logic - and back, in *Proc. of the 20th Symposium on Mathematical Foundations of Computer Science*. Springer-Verlag, *Lecture Notes in Comput. Sci.* **969** (1995) 529-540.
- [20] F. Laroussinie and P. Schnoebelen, The state explosion problem from trace to bisimulation equivalence, in *Proc. of the 3rd International Conference on Foundations of Software Science and Computation Structures (FOSSACS'2000)*. Springer-Verlag, *Lecture Notes in Comput. Sci.* **1784** (2000) 192-207.
- [21] K.G. Larsen and A. Skou, Bisimulation through probabilistic testing. *Inform. and Comput.* **94** (1991) 1-28.
- [22] R. Milner, An algebraic definition of simulation between programs, in *Proc. 2nd Joint Conference on Artificial Intelligence*. William Kaufmann (1971) 481-489. Also available as Report No. CS-205, Computer Science Department, Stanford University.
- [23] ———, *Communication and Concurrency*. Prentice Hall, Englewood Cliffs (1989).
- [24] F. Moller and C. Tofts, Relating processes with respect to speed, in *CONCUR '91: 2nd International Conference on Concurrency Theory*, edited by J.C.M. Baeten and J.F. Groote. Springer-Verlag, Amsterdam, The Netherlands, *Lecture Notes in Comput. Sci.* **527** (1991) 424-438.
- [25] D. Park, *Concurrency and automata on infinite sequences*, in *5th GI Conference*, Karlsruhe, Germany, edited by P. Deussen. Springer-Verlag, *Lecture Notes in Comput. Sci.* **104** (1981) 167-183.
- [26] 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 (1999).
- [27] B. Steffen and A. Ingólfssdóttir, Characteristic formulae for processes with divergence. *Inform. and Comput.* **110** (1994) 149-163.
- [28] A. Tarski, A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.* **5** (1955) 285-309.
- [29] Y. Wang, *Real-time behaviour of asynchronous agents*, in Baeten and Klop [4], pp. 502-520.

Communicated by Z. Ésik.

Received August 15, 2000. Accepted March 14, 2001.