BRICS

Basic Research in Computer Science

BRICS RS-95-2 Laroussinie et al.: From Timed Automata to Logic - and Back

From Timed Automata to Logic - and Back

François Laroussinie
Kim G. Larsen
Carsten Weise

BRICS Report Series RS-95-2

ISSN 0909-0878 January 1995

See back inner page for a list of recent publications in the BRICS Report Series. 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 WWW and anonymous FTP:

`http://www.brics.dk/`
`ftp ftp.brics.dk (cd pub/BRICS)`

From Timed Automata to Logic — and Back [*]

François Laroussinie
BRICS[†], Aalborg Univ., Denmark

Kim G. Larsen
BRICS, Aalborg Univ., Denmark

Carsten Weise
Aachen Univ., Germany

Abstract

One of the most successful techniques for automatic verification is that of *model checking*. For finite automata there exist since long extremely efficient model–checking algorithms, and in the last few years these algorithms have been made applicable to the verification of real–time automata using the region–techniques of Alur and Dill.

In this paper, we continue this transfer of existing techniques from the setting of finite (untimed) automata to that of timed automata. In particular, a timed logic L_ν is put forward, which is sufficiently expressive that we for any timed automaton may construct a single *characteristic* L_ν formula uniquely characterizing the automaton up to timed bisimilarity. Also, we prove decidability of the *satisfiability* problem for L_ν with respect to given bounds on the number of clocks and constants of the timed automata to be constructed. None of these results have as yet been succesfully accounted for in the presence of time [1].

[*] This work has been supported by the European Communities under CONCUR2, BRA 7166

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

[1] An exception occurs in Alur's thesis [Alu91] in which a decidability result is presented for a *linear* timed logic called MITL.

Contents

1 Introduction

One of the most successful techniques for *automatic verification* is that of *model–checking*; i.e. a property is given as a formula of a propositional temporal logic and automatically compared with an automata [2] representing the actual behaviour of the system. Extremely efficient model–checking algorithms have been obtained for *finite* automata with respect to the branching–time temporal logics CTL [CE81, QS82, CES86] and (various fragments of) the modal μ–calculus [Koz82, AC88, EL86, CS91, And92, Xin92].

In the last few years, model–checking has been extended to real–time systems, with time considered to be a dense linear order. A timed extension of finite automata through addition of a finite set of real–valued clocks has been put forward [AD94], and the corresponding model–checking problem has been proven decidable for a number of timed logics including timed extensions of CTL (TCTL) [ACD90] and a timed μ–calculus (T_μ) [HNSY92].

However, in the untimed setting automata and logics enjoy a number of other important relationships which at present are either absent or at best unaccounted for in the setting of real–time automata and the corresponding real–time logics:

— **G**iven a finite automaton, both CTL and the modal μ–calculus are sufficiently expressive that corresponding *characteristic formulas* may be expressed with respect to a number of behavioural preorders and equivalences (e.g. bisimilarity) [BCG88, GS86, IS94]: i.e. an automaton is related to another in the preorder if and only if the first automaton satisfies the characteristic formula of the second. As characteristic formulas can be automatically constructed in time linear in the size of the argument automaton, this yields a preorder checking method that outperforms other known algorithms [CS91]. No such relationship has so far been established between timed automata and any of the proposed real–timed logics;

— **T**he *satisfiability* problems for CTL and the modal μ–calculus have been proven decidable [EC82, EH85, Wol85, KP83]; thus given a logical property it is possible to automatically synthesize a satisfying finite automata (provided any such exists). In contrast, the satisfiability problems for both TCTL and T_μ are undecidable [ACD90, HNSY92]

In this paper we present results establishing both of the two above desired relationships in the presence of real–time (timed automaton). In particular we put forward a timed logic L_ν for which we establish the following:

— **F**irst, we present an effective characteristic formula construction for timed bisimilarity, transforming any timed automaton into a formula of L_ν characterizing precisely the equivalence class of the automaton. Thus, timed bisimilarity between automata reduces to a model–checking problem, which — when combined with the model–checking algorithm for L_ν — yields an alternative algorithm for timed bisimulation compared with [Cer92]. In addition, characteristic formula constructions may be given for time–abstracted equivalence [LW93] and the "faster–than" relation in [FT91], immediately yielding

[2] or a kripke structure

decision procedures for these relationships as well;

— **S**econd, we prove decidability of *bounded satisfiable* for L_ν. That is, we present a model–construction algorithm, which given a formula of L_ν and bounds k and M will synthesize a timed automata with no more than k clocks and no clock being compared with constants greater than M (provided any such exits).

Combining the characteristic formula construction with the bounded model–construction algorithm enables us to decide whether an automaton can be simplified in terms of number of clocks and constants used for comparison.

The remainder of this paper is organized as follows: In the next section we give a short presentation of the notion of timed automata used in this paper; in section 3, the logic L_ν is presented, and in section 4 we review the region technique by Alur and Dill [AD94] and present a decidability result for the model–checking problem of L_ν. Section 5 presents the characteristic formula construction, whereas section 6 presents the bounded model–construction algorithm.

2 Timed Automata

Let $\mathcal{A}$ be a fixed set of actions ranged over by $a, b, c, \ldots$. We denote by $\mathbf{N}$ the set of natural numbers and by $\mathbf{R}$ the set of non–negative real numbers. $\mathcal{D}$ denotes the set of delay actions $\{\epsilon(d) \,|\, d \in \mathbf{R}\}$, and $\mathcal{L}$ denotes the union $\mathcal{A} \cup \mathcal{D}$. If C is a set of clocks, $\mathcal{B}(C)$ denotes the set of formulas built using boolean connectives over atomic formulas of the form $x \leq m$, $m \leq x$, $x \leq y + m$ and $y + m \leq x$ with $x, y \in C$ and $m \in \mathbf{N}$. Moreover $\mathcal{B}_M(C)$ denotes the subset of $\mathcal{B}(C)$ with no constant greater than M.

Definition 1 *A timed automaton A is a tuple $\langle \mathcal{A}, N, \eta_0, C, E \rangle$ where $\mathcal{A}$ is a finite set of actions, N is a finite set of nodes, η_0 is the initial node, C is a finite set of clocks, and $E \subseteq N \times N \times \mathcal{A} \times 2^C \times \mathcal{B}(C)$ corresponds to the set of edges. $e = \langle \eta, \eta', a, r, b \rangle \in E$ represents an edge from the node η to the node η' with action a, r denoting the set of clocks to be reset and b is the enabling condition over the clocks of A.*

Example 1 Consider the 2-clock automaton A described in the left part of figure 1. The automaton has four nodes, η_0, η_1, η_2 and η_3, two clocks, x and y, and three edges. The edge between η_0 and η_1 has a as action, $\{x\}$ as reset set and the enabling condition for the edge is $0 < x < 1$. □

Informally, the system starts at node η_0 with all its clocks initialized to 0. The values of the clocks increase synchronously with time. At any time, the automaton whose current node is η can change node by following an edge $\langle \eta, \eta', a, r, b \rangle \in E$ provided the current values of the clocks satisfy b. With this transition the clocks in r get reset to 0.

A time assignment v for C is a function from C to $\mathbf{R}$. We denote by $\mathbf{R}^C$ the set of time assignments for C. For $v \in \mathbf{R}^C$, $x \in C$ and $d \in \mathbf{R}$, $v + d$ denotes the time assignment which maps each clock x in C to the value $v(x) + d$. For

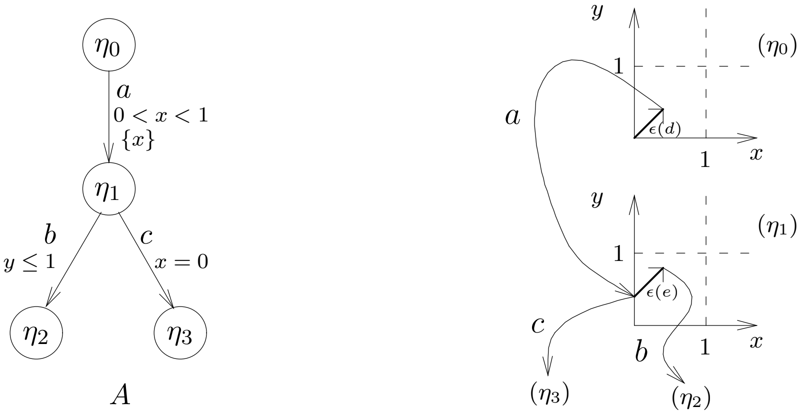


Figure 1: An automaton and its behaviour

$C' \subseteq C$, $[C' \mapsto 0]v$ denotes the assignment for C which maps each clock in C' to the value 0 and agrees with v over $C \backslash C'$. Given a condition $b \in \mathcal{B}(C)$ and a time assignment $v \in \mathbf{R}^C$, $b(v)$ is a boolean value describing whether b is satisfied by v or not. Finally a k–clock automata is a timed automata $\langle \mathcal{A}, S, \eta_0, C, E\rangle$ such that $|C| = k$.

A *state* of an automaton A is a pair $\langle \eta, v\rangle_A$ where η is a node of A and v a time assignment for C. The initial state of A is $\langle \eta_0, v_0\rangle_A$ where v_0 is the time assignment mapping all clocks in C to 0.

The semantics of A is given by a labelled transition system $\mathcal{M}_A = \langle \Sigma_A, \mathcal{L}, \sigma_0, \longrightarrow_A\rangle$, where Σ_A is the set of states of A, σ_0 is the initial state $\langle \eta_0, v_0\rangle_A$, and $\longrightarrow_A$ is the transition relation defined as follows:

$$\langle \eta, v\rangle \xrightarrow{a}_A \langle \eta', v'\rangle \quad \text{iff} \quad \exists r, b.\ \langle \eta, \eta', a, r, b\rangle \in E \ \wedge\ b(v) \ \wedge\ v' = [r \rightarrow 0]v$$
$$\langle \eta, v\rangle \xrightarrow{\epsilon(d)}_A \langle \eta', v'\rangle \quad \text{iff} \quad \eta = \eta' \ \text{ and } \ v' = v + d$$

We may now apply the standard notion of bisimulation [Mil89, Par81] to the labelled transition systems determined by two automata A and B. Letting s_A and s_B range over states of respectively A and B, *stong timed bisimulation* $\sim$ is defined as the largest symmetric relation over $\Sigma_A \times \Sigma_B$ such that whenever $s_A \sim s_B$ and $\ell \in \mathcal{A} \cup \mathcal{D}$ then

- Whenever $s_A \xrightarrow{\ell}_A s'_A$ then there exists s'_B such that $s_B \xrightarrow{\ell}_B s'_B$ and $s'_A \sim s'_B$.

We say that A and B are strong timed bisimular if their initial states are strong bisimilar.

Example 2 Reconsider the automaton A of Figure 1. The two coordinate systems in the right part of the Figure indicates (some of) the states of A. Each point of the coordinate systems represents a unique time assignment, with the top (resp. bottom) coordinate system representing states involving the node η_0 (resp. η_1). In the Figure we have indicated the following transition sequence (where $d < 1$ and $e + d \leq 1$):

$$\langle \eta_0, (0,0) \rangle \xrightarrow{\epsilon(d)} \langle \eta_0, (d,d) \rangle \xrightarrow{a} \langle \eta_1, (0,d) \rangle \xrightarrow{\epsilon(e)} \langle \eta_1, (e, d+e) \rangle \xrightarrow{b}$$

In addition, it is indicated that A can perform a c–transition in the state $\langle \eta_1, (0,d) \rangle$. □

3 Timed Modal Logic L_ν

We consider a dense–time logic L_ν with clocks and recursion. This logic may be seen as a certain fragment [3] of the μ–calculus T_μ presented in [HNSY92].

Definition 2 *Let K a finite set of clocks, Id a set of identifiers and k an integer. The set L_ν of formulae over K, Id and k is generated by the abstract syntax with φ and ψ ranging over L_ν:*

$$\begin{array}{rcl} \varphi & ::= & \mathsf{tt} \mid \mathsf{ff} \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \exists\, \varphi \mid \forall\, \varphi \mid \langle a \rangle\, \varphi \mid [a]\, \varphi \\ & & \mid x \,\mathsf{in}\, \varphi \mid x + n \sim y + m \mid Z \end{array}$$

where $a \in \mathcal{A}$; $x, y \in K$; $n, m \in \{0, 1, \ldots, k\}$; $\sim \in \{=, <, \leq, >, \geq\}$ and $Z \in \mathsf{Id}$.

The meaning of the identifiers is specified by a declaration $\mathcal{D}$ assigning a formula of L_ν to each identifier. When $\mathcal{D}$ is understood we write $Z \overset{\mathrm{def}}{=} \varphi$ for $\mathcal{D}(Z) = \varphi$. The K clocks are called *formula clocks* and a formula φ is said to be *closed* if every formula clock x occurring in φ is in the scope of an "$x \,\mathsf{in} \ldots$" operator.

Given a timed automata $A = \langle \mathcal{A}, N, \eta_0, C, E \rangle$, we interpret the L_ν formulas over an *extended state* $\langle \eta, vu \rangle_{A^+}$ where $\langle \eta, v \rangle_A$ is a state of A and u a time assignment for K. Transitions between extended states are defined by: $\langle \eta, vu \rangle_{A^+} \xrightarrow{\epsilon(d)} \langle \eta', v + d\, u + d \rangle_{A^+}$ and $\langle \eta, vu \rangle_{A^+} \xrightarrow{a} \langle \eta', v'u' \rangle_{A^+}$ iff $\langle \eta, v \rangle_A \xrightarrow{a} \langle \eta', v' \rangle_A$ and $u = u'$.

Informally, $\exists\varphi$ holds in an extended state if there is a delay transition leading to an extended state satisfying φ. Thus $\exists$ denotes existential quantification over (arbitrary) delay transitions. Similarly, $\forall$ denotes universal quantification over delay transitions, and $\langle a \rangle$ (resp. $[a]$) denotes existential (resp. universal) quantification over a–transitions. The formula $(x \,\mathsf{in}\, \varphi)$ introduces a formula clock x and initializes it to 0; i.e. an extended state satisfies the formula in case the modified state with x being reset to 0 satisfies φ. Introduced formula clocks are used by formulas of the type $(x + n \sim y + m)$, which is satisfied by an extended state provided the values of the named formula clocks satisfies the required relationship. Finally, an extended state satisfies an identifier Z if it satisfies the corresponding declaration (or definition) $\mathcal{D}(Z)$. Formally, the satisfaction relation between extended states and formulas is defined as follows:

[3] allowing only maximal recursion and using a slightly different notion of model

Definition 3 *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}
\langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} \mathtt{tt} & \Rightarrow & \mathit{true} \\
\langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} \mathtt{ff} & \Rightarrow & \mathit{false} \\
\langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} \varphi \wedge \psi & \Rightarrow & \langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} \varphi \;\; \mathit{and} \;\; \langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} \psi \\
\langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} \varphi \vee \psi & \Rightarrow & \langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} \varphi \;\; \mathit{or} \;\; \langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} \psi \\
\langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} \exists\, \varphi & \Rightarrow & \exists d \in \mathbf{R}. \;\; \langle \eta, v{+}d \;\, u{+}d\rangle_{A^+} \models_{\mathcal{D}} \varphi \\
\langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} \forall\, \varphi & \Rightarrow & \forall d \in \mathbf{R}. \;\; \langle \eta, v{+}d\, u{+}d\rangle_{A^+} \models_{\mathcal{D}} \varphi \\
\langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} \langle a\rangle\, \varphi & \Rightarrow & \exists\, \langle \eta', v'\rangle_A. \;\; \langle \eta, v\rangle_A \xrightarrow{a} \langle \eta', v'\rangle_A \;\; \mathit{and} \\
 & & \qquad\qquad\quad \langle \eta', v'\, u\rangle_{A^+} \models_{\mathcal{D}} \varphi \\
\langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} [a]\, \varphi & \Rightarrow & \forall\, \langle \eta', v'\rangle_A. \;\; \langle \eta, v\rangle_A \xrightarrow{a} \langle \eta', v'\rangle_A \; \mathit{implies} \\
 & & \qquad\qquad\quad \langle \eta', v'\, u\rangle_{A^+} \models_{\mathcal{D}} \varphi \\
\langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} x{+}m{\sim}y{+}n & \Rightarrow & u(x) + m \sim u(y) + n \\
\langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} x \;\mathsf{in}\; \varphi & \Rightarrow & \langle \eta, v\, u'\rangle_{A^+} \models_{\mathcal{D}} \varphi \;\; \mathit{where} \;\; u' = [\{x\} \rightarrow 0]u \\
\langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} Z & \Rightarrow & \langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} \mathcal{D}(Z)
\end{array}$$

Any relation satisfying the above implications is called a *satisfiability* relation. It follows from standard fixpoint theory [Tar55] that $\models_{\mathcal{D}}$ is the union of all satisfiability relations and that the above implications in fact are biimplications for $\models_{\mathcal{D}}$. We say that A satisfies a closed L_ν formula φ and write $A \models \varphi$ when $\langle \eta_0, v_0\, u\rangle_{A^+} \models_{\mathcal{D}} \varphi$ for any u. Note that if φ is closed, then $\langle \eta, vu\rangle_{A^+} \models_{\mathcal{D}} \varphi$ iff $\langle \eta, vu'\rangle_{A^+} \models_{\mathcal{D}} \varphi$ for any $u, u' \in \mathbf{R}^K$.

The following real–time interval modalities present in the Extended Timed Modal Logic introduced in [HLY92] are obtainable as derived operators of L_ν, e.g.:

$$\begin{array}{rcl}
\exists\,]0;\infty[\; \varphi & \overset{\mathrm{def}}{=} & x \;\mathsf{in}\; \Big(\exists\, (x > 0 \wedge \varphi)\Big) \\
\exists\,]m;n[\; \varphi & \overset{\mathrm{def}}{=} & x \;\mathsf{in}\; \Big(\exists\, (x > m \wedge x < n \wedge \; \varphi)\Big)
\end{array}$$

Thus, $\exists]m;n[\varphi$ is satisfied by and extended state if an extended state satisfying φ can be reached with a delay between m and n. A formula is called a *q-clocks formula* if it contains no more than q formula clocks. Thus formulas using only the derived $\exists\,]m;n[$ or $\exists\,]0;\infty[$ modalities are clearly 1-clock formulas (as each use of an interval modality can be defined using the same formula clock x).

Example 3 Consider the timed automaton described in Figure 1. It may be argued that the initial state $\langle \eta_0, v_0\; u_0\rangle$ satisfies the following L_ν formula φ:

$$\varphi = \exists\,]0;1[\; \langle a\rangle \left[\Big(\langle c\rangle\, \mathtt{tt} \Big) \wedge \Big(\forall\,]0;1[\; [c]\, \mathtt{ff} \Big) \wedge \Big(\exists\,]0;1[\; \langle b\rangle\, \mathtt{tt} \Big) \wedge \Big(\exists\,]0;1[\; [b]\, \mathtt{ff} \Big) \right] \quad (1)$$

Intuitively this formula means that "the action a can be performed after a delay (strictly) between 0 and 1, after which (1) the action c can be performed immediately but not after any positive delay, (2) the action b can be performed after some delay in the interval $]0;1[$, and (3) the action b cannot be performed after some delay in $]0;1[$". □

4 Model Checking

The model-checking problem for L_ν consists in deciding if a given timed automata A satisfies a given specification φ in L_ν. This problem is decidable using the region technique of Alur and Dill [AD94, ACD90] which provides an abstract semantics of timed automata in the form of finite labelled transition systems with the truth value of L_ν formulas being maintained.

The basic idea is that, given a timed automaton A, two states $\langle \eta, v_1 \rangle_A$ and $\langle \eta, v_2 \rangle_A$ which are close enough with respect to their clocks values (we will say that v_1 and v_2 are in the same *region*) can perform the same actions, and two extended states $\langle \eta, v_1\, u_1 \rangle_{A^+}$ and $\langle \eta, v_2\, u_2 \rangle_{A^+}$ where $v_1\, u_1$ and $v_2\, u_2$ are in the same region, satisfy the same L_ν formulas. The notion of region is defined as an equivalence class of a relation over time assignments [HNSY92] [4] . First, for $t \in \mathbf{R}$, let $\lfloor t \rfloor \overset{\text{def}}{=} max\{n \in \mathbf{N} \mid n \leq t\}$ denote the integral part of t, and let $\{t\} \overset{\text{def}}{=} t - \lfloor t \rfloor$ denote its fractional part. Moreover we have: $\lceil t \rceil \overset{\text{def}}{=} min\{n \in \mathbf{N} \mid t \leq n\}$.

Definition 4 *Let $k \in \mathbf{N}$ and let C be a set of clocks. Then $u, u' \in \mathbf{R}^C$ are equivalent with respect to k, denoted by $u \doteq u'$ iff:*

i) $\forall x \in C.\ \ u(x) > k$ *iff* $u'(x) > k$

ii) $\forall x \in C$ *s.t.* $u(x) \leq k.\ \ \lfloor u(x) \rfloor = \lfloor u'(x) \rfloor$ *and* $\{u(x)\} = 0 \Leftrightarrow \{u'(x)\} = 0$

iii) $\forall x, y \in C.\ \ u(x) - u(y) > k$ *iff* $u'(x) - u'(y) > k$

iv) $\forall x, y \in C$ *s.t.* $0 \leq u(x) - u(y) \leq k.\ \ \lfloor u(x) - u(y) \rfloor = \lfloor u'(x) - u'(y) \rfloor$
and $\{u(x) - u(y)\} = 0 \Leftrightarrow \{u'(x) - u'(y)\} = 0$

The equivalence classes under $\doteq$ are called *regions*, and $[u]$ denotes the region which contains the time assignment u. $\mathcal{R}_k^C$ denotes the set of all regions for a set C of clocks and the maximal constant k. From a decision point of view it is important to note that $\mathcal{R}_k^C$ is finite.

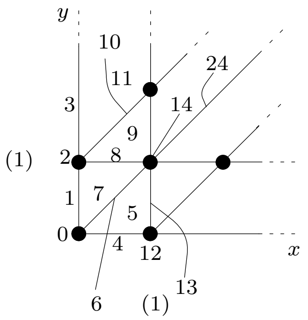
Note that for any condition b in $\mathcal{B}(C)$ with no constant greater than k, we have $b(u) \Leftrightarrow b(u')$, whenever u and u' belong to the same region in $\mathcal{R}_k^C$. Thus for a region $\gamma \in \mathcal{R}_k^C$, we can define $b(\gamma)$ as the truth value of $b(u)$ for any u in γ. Conversely given a region γ, we can easily build a formula of $\mathcal{B}(C)$, called $\beta(\gamma)$, such that $\beta(\gamma)(u) = \mathtt{tt}$ iff $u \in \gamma$ [5]. Thus, given a region γ', $\beta(\gamma)(\gamma')$ is mapped to the value $\mathtt{tt}$ precisely when $\gamma = \gamma'$. Finally, note that $\beta(\gamma)$ itself can be viewed as a L_ν formula.

Given a region $[u]$ in $\mathcal{R}_k^C$ and $C' \subseteq C$ we define the following reset operator: $[C' \to 0][u] = [[C' \to 0]u]$. Moreover, for a region $[u]$, we define the succssor region (denoted by $succ([u])$) as the region $[u']$, where:

$$u'(x) = \begin{cases} u(x) + f & \forall x \in C.\ u(x) > k \vee \{u(x)\} \neq 0 \\ u(x) + f/2 & \exists x \in C.\ u(x) \leq k \wedge \{u(x)\} = 0 \end{cases}$$

[4] The notion of region used in the present paper is slightly more refined.

[5] An obvious way of building $\beta(\gamma)$ is to consider the conjunction of all $\mathcal{B}(C,k)$ formulas satisfied by γ, where $\mathcal{B}(C,k)$ denotes the finite set (modulo boolean reductions) of $\mathcal{B}(C)$ formulas with no constant greater than k.



$$succ(\gamma_0) = \gamma_6$$
$$succ(\gamma_5) = \gamma_{13}$$
$$\dots$$

$$\beta(\gamma_6) = \begin{array}{l} 0 < x < 1 \wedge \\ 0 < y < 1 \wedge \\ y = x \end{array}$$

$$\beta(\gamma_0) = (x = 0 \wedge y = 0)$$
$$\beta(\gamma_8) = (0 < x < 1 \wedge y = 1)$$

$$\beta(\gamma_7) = \begin{array}{l} 0 < x < 1 \wedge \\ 0 < y < 1 \wedge \\ y > x \end{array}$$

Figure 2: $\mathcal{R}_k^C$ with $C = \{x, y\}$ and $k = 1$

where $f = min\{1 - \{u(x)\} \mid u(x) \leq k\}$ [6]. Informally the change from γ to $succ(\gamma)$ correspond to the minimal elapse of time which can modify the enabled actions of the current state.

We denote by γ^l the l^{th} successor region of γ (i.e. $\gamma^l = succ^l(\gamma)$). From each region γ, it is possible to reach a region γ' s.t. $succ(\gamma') = \gamma'$, and we denote by l_γ the required number of step s.t. $\gamma^{l_\gamma} = succ(\gamma^{l_\gamma})$.

Example 4 The Figure 2 gives an overview of the set of regions defined by two clocks x and y, and the maximal constant 1. In this case there are 31 different regions, of which only 14 are numbered in the figure. Corresponding $\mathcal{B}(C)$–formulas as well as successor regions are indicated for some of the regions. In general successor regions are determined by following 45^o lines upwards to the right. □

Given a timed automata $A = \langle \mathcal{A}, N, \eta_0, C, E \rangle$, let k_A be the maximal constant occurring in the enabling condition of the edges E. Then for any $k \geq k_A$ we can define a symbolic semantics of A over symbolic states $[\eta, \gamma]_A$ where $\eta \in S$ and $\gamma \in \mathcal{R}_k^C$ as follows:

$$\begin{array}{lll} [\eta, \gamma]_A \xrightarrow{a} [\eta', \gamma']_A & \text{iff} & \exists\, u \in \gamma,\ \langle \eta, u \rangle_A \xrightarrow{a} \langle \eta', u' \rangle_A \text{ and } u' \in \gamma' \\ [\eta, \gamma]_A \xrightarrow{\chi} [\eta, succ(\gamma)]_A & \text{iff} & \text{true} \end{array}$$

Consider now L_ν with respect to formula clock set K and maximal constant k_L. Also consider a given timed automata $A = \langle \mathcal{A}, N, \eta_0, C, E \rangle$ (s.t. K and C are disjoint). Then an *extended symbolic state* is a pair $[\eta, \gamma]_{A^+}$ where $\eta \in N$ and $\gamma \in \mathcal{R}_k^{C^+}$ with $C^+ = C \cup K$ and $k = max(k_A, k_L)$. Whenever γ is a region over $C \cup K$ we denote by $\gamma_{|C}$ the set of time assignments in γ restricted to the (automata) clock set C. Similarly, $\gamma_{|K}$ denotes the projection of all time assignments in γ to the (formula) clock set K. Now we define a *symbolic semantics* for L_ν as follows:

[6] if this set is empty, then $f = 0$

Definition 5 *$\vdash_{\mathcal{D}}$ is the largest relation satisfying the following implications:*

$$\begin{array}{rl}
i) & [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} \mathsf{tt} \;\Rightarrow\; \textit{true} \\
ii) & [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} \mathsf{ff} \;\Rightarrow\; \textit{false} \\
iii) & [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} \varphi \wedge \psi \;\Rightarrow\; [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} \varphi \;\textit{ and }\; [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} \psi \\
iv) & [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} \varphi \vee \psi \;\Rightarrow\; [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} \varphi \;\textit{ or }\; [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} \psi \\
v) & [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} \exists\, \varphi \;\Rightarrow\; \exists l \in \mathbf{N}.\; [\eta, succ^l(\gamma)]_{A^+} \vdash_{\mathcal{D}} \varphi \\
vi) & [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} \forall\, \varphi \;\Rightarrow\; \forall l \in \mathbf{N}.\; [\eta, succ^l(\gamma)]_{A^+} \vdash_{\mathcal{D}} \varphi \\
vii) & [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} \langle a \rangle\, \varphi \;\Rightarrow\; \exists\, [\eta',\gamma']_{A^+}.\; [\eta,\gamma_{|C}]_A \xrightarrow{a} [\eta',\gamma'_{|C}]_A \;\textit{ and} \\
 & \qquad\qquad\qquad\qquad \gamma'_{|K} = \gamma_{|K} \;\textit{ and }\; [\eta',\gamma']_{A^+} \vdash_{\mathcal{D}} \varphi \\
viii) & [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} [a]\, \varphi \;\Rightarrow\; \forall\, [\eta',\gamma']_{A^+}.\; [\eta,\gamma_{|C}]_A \xrightarrow{a} [\eta',\gamma'_{|C}]_A \;\textit{ and} \\
 & \qquad\qquad\qquad\qquad \gamma'_{|K} = \gamma_{|K} \;\textit{implies}\; [\eta',\gamma']_{A^+} \vdash_{\mathcal{D}} \varphi \\
ix) & [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} x{+}c{\sim}y{+}d \;\Rightarrow\; (x + c \sim y + d)(\gamma) \\
x) & [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} x \;\mathsf{in}\; \varphi \;\Rightarrow\; [\eta, [\{x\} \to 0]\gamma]_{A^+} \vdash_{\mathcal{D}} \varphi \\
xi) & [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} Z \;\Rightarrow\; [\eta,\gamma]_{A^+} \vdash_{\mathcal{D}} \mathcal{D}(Z)
\end{array}$$

Any relation satisfying the above implications is called a *symbolic satisfiability relation*. In the following we write $[\eta,\gamma]$ instead of $[\eta,\gamma]_{A^+}$ when no confusion is possible. The above symbolic interpretation of L_ν is closely related to the standard interpretation from Definition 3 as stated in the following theorem:

Theorem 1 *Let φ be a formula of L_ν, and let $\langle \eta, v\, u\rangle_{A^+}$ be an extended state over some timed automaton A . Then we have* [7]*:*

$$\langle \eta, v\, u\rangle_{A^+} \models_{\mathcal{D}} \varphi \quad \textit{if and only if} \quad [\eta, [v \cdot u]]_{A^+} \vdash_{\mathcal{D}} \varphi$$

It follows that the model checking problem for L_ν is decidable since, given $\varphi \in \mathrm{L}_\nu$, it suffices to to check the truth value of any given L_ν formula φ with respect to the finite transition system $\langle N \times \mathcal{R}_k^{C^+}, \mathcal{A} \cup \{\chi\}, \sigma_0, \rightarrow\rangle$ corresponding to the extended symbolic semantics of A.

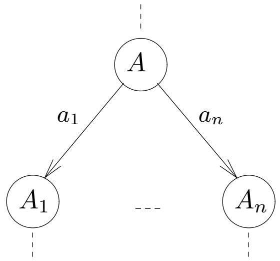
5 Characteristic Properties

First let us recall the characteristic formula construction for finite automata [8] [IS94, GS86, BCG88] (see Figure 3). The construction defines the characteristic formula $\Phi(A)$ of a node A in terms of similar characteristic formulas of the derivates $A_1 \ldots A_n$ of A: whenever A has an a_i–transition to A_i this is reflected in $\Phi(A)$ by addition of a conjunct $\langle a_i\rangle\Phi(A_i)$. To characterize A up to strong bisimilarity $\Phi(A)$ contains in addition a conjunct $[a]\Psi_a$ for each action a, where Ψ_a is a disjunction over all a–transitions out of A. In general the definitions of characteristic formulas $\Phi(A)$ constitutes a simultaneous recursive definition (as the automaton may have cycles), and to obtain the desired characterization the solution sought is the maximum one.

For timed automata the characteristic formula construction must necessarily take account of the time assignment in addition to the actual node. Thus, for a

[7] where $v \cdot u$ is the time assignment over $C \cup K$ such that $(v \cdot u)(x) = v(x)$ if $x \in C$ and $(v \cdot u)(x) = u(x)$ if $x \in K$.

[8] Alternatively you may think of finite automata as zero–clock timed automata.



$$\Phi(A) = \bigwedge_{i=1}^{n} \langle a_i \rangle \; \Phi(A_i) \;\wedge\; \bigwedge_{a} [a] \Big(\bigvee_{i.a_i = a} \Phi(A_i) \Big)$$

Figure 3: Characteristic formula for finite automata.

timed automaton $A = \langle \mathcal{A}, N, \eta_0, C, E \rangle$, we shall define characteristic formulas of the form $\Phi(\eta, \gamma)$, where η is a node of A and γ is a region over the clocks of A. The construction of $\Phi(\eta, \gamma)$ follows closely the pattern from the finite automa case. However, we first need to be able to determine the (a–) edges out of η which are enabled in the region γ. Given an edge $e = \langle \eta, \eta', a, r, b \rangle$ in E, η_e (resp. η'_e, a_e, r_e, b_e) denotes η (resp. η', a, r, b). Given $\eta \in N$ and $\gamma \in \mathcal{R}^C_{k_A}$, we define $E(\eta, \gamma) = \{ e \mid \eta_e = \eta \text{ and } b_e(\gamma) = \mathtt{tt} \}$ and $E(\eta, \gamma, a) = \{ e \in E(\eta, \gamma) \mid a_e = a \}$. Thus, $E(\eta, \gamma)$ (resp. $E(\eta, \gamma, a)$) is the set of all enabled transitions (resp. a-transitions) from $[\eta, \gamma]_A$.

We may now present the characteristic formula construction for timed automata:

Definition 6 *Let A be a timed automata $\langle \mathcal{A}, N, \eta_0, C, E \rangle$. For any region γ in $\mathcal{R}^C_{k_A}$, and node η in N, we introduce an identifier $\Phi(\eta, \gamma)$ (the characteristic formula) associated with the symbolic state $[\eta, \gamma]_A$. The definition (declaration) for $\Phi(\eta, \gamma)$ is:*

$$\Phi(\eta, \gamma) \;\stackrel{\text{def}}{=}\; \left(\begin{array}{c} \displaystyle\bigwedge_{e \in E(\eta,\gamma)} \langle a_e \rangle \Big(r_e \;\mathsf{in}\; \Phi(\eta'_e, r_e(\gamma)) \Big) \;\wedge\; \bigwedge_{a} [a] \Big(\bigvee_{e \in E(\eta,\gamma,a)} \Big(r_e \;\mathsf{in}\; \Phi(\eta'_e, r_e(\gamma)) \Big) \Big) \\ \displaystyle\wedge\; \forall\!\!\forall \Big(\bigwedge_{l = 0..l_\gamma} \beta(\gamma^l) \;\Rightarrow\; \Phi(\eta, \gamma^l) \Big) \end{array} \right)$$

We denote by Id_A the set of identifiers $\Phi(\eta, \gamma)$ and by $\mathcal{D}_A$ the corresponding declaration.

Note that the declaration for $\Phi(\eta, \gamma)$ is not quite a L_ν formula due to the presence of implication. However, it is easy to transform it into an equivalent L_ν formula because the negation of $\beta(\gamma)$ can be expressed in L_ν. Moreover $(r \,\mathsf{in}\, \varphi)$ is an abbreviation for $(c_1 \,\mathsf{in}\, (c_2 \,\mathsf{in}\, \ldots (c_n \,\mathsf{in}\, \varphi)))$ whenever r is $\{c_1, \ldots, c_n\}$. Finally $r(\gamma)$ denotes $[r \rightarrow 0]\gamma$. Note that $\mathcal{D}_A$ uses no more than $|C|$ formula clocks.

The declaration for $\Phi(\eta, \gamma)$ contains three groups of conjunctions the two first of which are closely related to the characteristic formula construction for finite automata. The first group contains a $\langle a_e \rangle$–formula for any edge e, which is enabled at η in the region γ. Following this edge clearly takes the automaton to the extended state $[\eta'_e, r_e(\gamma)]$. The second group of conjuncts contains for each action a a formula of the type $[a]\Psi_a$, where Ψ is a disjunction over all a–labelled edges being enabled at η in the region γ. Whereas the two first groups exhaustively characterizes the action behaviour of the extended state $[\eta, \gamma]$, the third conjunct is a $\forall\!\!\!\forall$–formula dealing with all delay transitions by requiring that any delay leading to a particular successor region γ^l should satisfy the corresponding characteristic formula.

Example 5 Reconsider the timed automata A described in Example 1 and the corresponding regions from Example 4. Below we give the declaration of some of the characteristic formulas. We define $\varphi_{nil} \overset{\text{def}}{=} \bigwedge_a [a]\,\mathbf{ff}$ [9] and we denote $\beta(\gamma_i)$ by β_i. We have:

$$\begin{aligned}
\Phi(\eta_0, \gamma_0) &\overset{\text{def}}{=} \varphi_{nil} \wedge \forall\!\!\!\forall\Big[(\beta_0 \;\Rightarrow\; \Phi(\eta_0, \gamma_0)) \wedge (\beta_6 \;\Rightarrow\; \Phi(\eta_0, \gamma_6)) \wedge (\beta_{14} \;\Rightarrow\; \Phi(\eta_0, \gamma_{14})) \\
&\qquad\qquad \wedge (\beta_{24} \;\Rightarrow\; \Phi(\eta_0, \gamma_{24}))\Big] \\
\Phi(\eta_0, \gamma_6) &\overset{\text{def}}{=} \langle a\rangle\, \Phi(\eta_1, \gamma_4) \wedge [a]\, \Phi(\eta_1, \gamma_4) \wedge [b]\, \mathbf{ff} \wedge [c]\, \mathbf{ff} \\
&\quad \wedge \forall\!\!\!\forall\Big[(\beta_6 \;\Rightarrow\; \Phi(\eta_0, \gamma_6)) \wedge (\beta_{14} \;\Rightarrow\; \Phi(\eta_0, \gamma_{14})) \wedge (\beta_{24} \;\Rightarrow\; \Phi(\eta_0, \gamma_{24}))\Big] \\
\Phi(\eta_1, \gamma_4) &\overset{\text{def}}{=} \langle b\rangle\, \Phi(\eta_2, \gamma_4) \wedge \langle c\rangle\, \Phi(\eta_3, \gamma_4) \wedge [b]\, \Phi(\eta_2, \gamma_4) \wedge [c]\, \Phi(\eta_3, \gamma_4) \wedge [a]\, \mathbf{ff} \\
&\quad \wedge \forall\!\!\!\forall\Big[(\beta_4 \;\Rightarrow\; \Phi(\eta_1, \gamma_4)) \wedge (\beta_5 \;\Rightarrow\; \Phi(\eta_1, \gamma_5)) \wedge (\beta_{13} \;\Rightarrow\; \Phi(\eta_1, \gamma_{13})) \\
&\qquad\qquad \wedge (\beta_{21} \vee \beta_{22} \vee \beta_{23}) \;\Rightarrow\; \varphi_{nil}\Big] \\
\Phi(\eta_1, \gamma_5) &\overset{\text{def}}{=} \langle b\rangle\, \Phi(\eta_2, \gamma_4) \wedge [b]\, \Phi(\eta_2, \gamma_4) \wedge [c]\, \mathbf{ff} \wedge [a]\, \mathbf{ff} \\
&\quad \wedge \forall\!\!\!\forall\Big[(\beta_5 \;\Rightarrow\; \Phi(\eta_1, \gamma_5)) \wedge (\beta_{13} \;\Rightarrow\; \Phi(\eta_1, \gamma_{13})) \\
&\qquad\qquad \wedge (\beta_{21} \vee \beta_{22} \vee \beta_{23}) \;\Rightarrow\; \varphi_{nil}\Big] \\
\Phi(\eta_2, \gamma) &\overset{\text{def}}{=} \forall\!\!\!\forall \varphi_{nil} \\
\Phi(\eta_3, \gamma) &\overset{\text{def}}{=} \forall\!\!\!\forall \varphi_{nil}
\end{aligned}$$

□

We have the following Main Theorem the proof of which is given in Appendix A.

Theorem 2 *Let $A = \langle \mathcal{A}, N, \eta_0, C, E\rangle$ and $B = \langle \mathcal{A}, M, \rho_0, K, F\rangle$ be two timed automata. Then for any $\rho \in M$, $\eta \in N$, $v \in \mathbf{R}^K$ and $u \in \mathbf{R}^C$:*

$$\langle \rho, v\rangle_B \;\sim\; \langle \eta, u\rangle_A \quad \textit{iff} \quad \langle \rho, v\,u\rangle_{B^+} \models_{\mathcal{D}_A} \Phi(\eta, [u])$$

where $\mathcal{D}_A$ corresponds to the previous definition of $\Phi(\eta, \gamma)$ for each $\eta \in N$ and $\gamma \in \mathcal{R}^C_{k_A}$.

As model–checking of L_ν is decidable we may use the above characteristic formula construction to decide timed bisimilarity between timed automata: to decide if two timed automata are timed bisimilar simply compare the one automaton to the characteristic formula of the other.

Corollary 1 *Timed bisimilarity between timed automata is decidable.*

[9] a state satisfies φ_{nil} whenever no action can be performed.

6 Model Construction

In this section we address the *satisfiability* problem for L_ν. That is we want to decide whether there exists a timed automaton A satisfying a given L_ν–formula φ. The hardness of this problem is illustrated by the following Proposition:

Proposition 1 *Let Ψ_l be the 1-clock formula defined as follows:*

$$\Psi_l \overset{\text{def}}{=} \Big(\underbrace{\exists\,]0;\infty[\,\langle a\rangle\ \cdots \exists\,]0;\infty[\,\langle a\rangle}_{l}\Big)\Big[\bigwedge_{i=1..l} \exists\,]0;1[\,\Big(\langle a_i\rangle\,\mathtt{tt} \wedge \bigwedge_{j\neq i}[a_j]\,\mathtt{ff}\Big)\Big]$$

where $l \in \mathbf{N}$. Then Ψ_l is satisfiable by some p-clock automata if and only if $l \leq 2p+1$.

As a consequence of this Proposition [10] we cannot deduce the number of clocks in the automata from the number of clocks in φ. In fact, similar to the results for TCTL and T_μ, we conjecture that the satisfiability problem for L_ν is undecidable [11].

Instead, we address the following more restricted *bounded satisfiability* problem in which bounds have been placed on both the number of automaton clocks as well as the size of the constants these clocks are compared to: given a formula φ (over a declaration $\mathcal{D}$), a set of clocks C and an integer M, we want to decide (and synthesize) whether there exists a (C, M)–automata [12] s.t. $A \models_{\mathcal{D}} \varphi$. We have the following main result:

Theorem 3 *The bounded satisfiability problem for* L_ν *is decidable.*

The remainder of this section is devoted to the proof of this theorem and to an example of bounded satisfiability checking. The decision procedure is closely related to the canonical model construction for modal logic [HC68].

Let φ be a given L_ν formula with k_φ as maximal constant. Let K be the set of formula clocks occurring in φ. Given C a set of clocks (with $C \cap K = \emptyset$) and M an integer, we want to decide if there exists a (C, M)–automaton satisfying φ.

Let $C^+ = C \cup K$. Let L_ν^φ be the set of all subformulae of φ [13]. Obviously L_ν^φ is finite.

A *problem* Π is a subset of $\mathcal{R}_k^{C^+} \times L_\nu^\varphi$ where $k = max(M, k_\varphi)$. A problem Π is said to be *satisfiable*[14] if there exists a (C, M)-automaton A and a node η of A such that for any $(\gamma, \psi) \in \Pi$ we have $[\eta, \gamma]_{A^+} \models_{\mathcal{D}} \psi$. We call A a solution to Π.

[10] the proof is given in appendix B

[11] due to the lack of a minimal fixed–point construct in L_ν and hence the lack of ability to express liveness properties we are unable to adopt the undecidability proofs for TCTL and T_μ.

[12] i.e. a $|C|$-clock automata A with maximum constant M.

[13] including φ and with $\mathcal{D}(Z)$ being a subformula of Z.

[14] or more precisely (C, M)–satisfiable.

A problem Π is said to be *maximal* if it satisfies the following closure conditions:

$$\begin{array}{lcl}
(\gamma, \psi) \in \Pi & \Rightarrow & (\gamma, \mathtt{tt}) \in \Pi \\
(\gamma, \psi_1 \wedge \psi_2) \in \Pi & \Rightarrow & (\gamma, \psi_1) \in \Pi \text{ and } (\gamma, \psi_2) \in \Pi \\
(\gamma, \psi_1 \vee \psi_2) \in \Pi & \Rightarrow & (\gamma, \psi_1) \in \Pi \text{ or } (\gamma, \psi_2) \in \Pi \\
(\gamma, \exists\psi) \in \Pi & \Rightarrow & \exists l.\ (\gamma^l, \psi) \in \Pi \\
(\gamma, \forall\psi) \in \Pi & \Rightarrow & \forall l.\ (\gamma^l, \psi) \in \Pi \\
(\gamma, x \ \mathsf{in}\ \psi) \in \Pi & \Rightarrow & ([\{x\} \rightarrow 0]\gamma, \psi) \in \Pi \\
(\gamma, Z) \in \Pi & \Rightarrow & (\gamma, \mathcal{D}(Z)) \in \Pi
\end{array}$$

We have the two following lemmas, the proofs of which are trivial:

Lemma 1 *If* $\Pi \subseteq \Pi'$ *and* Π' *is satisfiable then also* Π *is satisfiable.*

Lemma 2 *If* Π *is satisfiable then there exists a maximal problem* Π' *containing* Π *and being satisfiable.*

Thus it suffices to consider satisfiability of maximal problems. Given a problem Π, a region γ and an action a we define the problem $\Pi_a^{\gamma,r}$ as the set $\{(r(\gamma), \psi) \mid (\gamma, [a]\ \psi) \in \Pi\}$. Now we introduce a new notion about problems. Let $\mathcal{C}$ be a set of maximal problems. Then $\mathcal{C}$ is a *consistency relation* if whenever $\Pi \in \mathcal{C}$ then:

1− If $(\gamma, x + m \sim y + n) \in \Pi$ then $\gamma(x) + m \sim \gamma(y) + n$
2− For all γ, $(\gamma, \mathtt{ff}) \notin \Pi$
3− Whenever $(\gamma, \langle a \rangle\ \psi) \in \Pi$, there exists some $r \subseteq C, b \in \mathcal{B}_M(C)$ with $b(\gamma) = \mathtt{tt}$ and $\Pi' \in \mathcal{C}$ s.t. :
$i)$ $(r(\gamma), \psi) \cup \Pi_a^{\gamma,r} \subseteq \Pi'$
$ii)$ $\forall \gamma',\ b(\gamma') = \mathtt{tt} \Rightarrow \Pi_a^{\gamma',r} \subseteq \Pi'$

We say that a maximal problem is consistent if it belongs to some consistency relation. We have the following key lemma:

Lemma 3 *Let* Π *be a maximal problem. Then* Π *is consistent if and only if* Π *is satisfiable.*

Proof $\Leftarrow$ It's easy to show that $\mathcal{C} = \{\Pi \mid \Pi \text{ maximal and satisfiable}\}$ is a consistency relation.
$\Rightarrow$ Let $\mathcal{C}$ be a consistency relation (containing Π). Now construct the canonical automaton $A_{\mathcal{C}} = \langle \mathcal{A}, N, \eta_0, C, E \rangle$ s.t. :

- $N = \{\eta_\Pi \mid \Pi \in \mathcal{C}\}$
- η_0 is some $\eta_\Pi \in N$.
- $\langle \eta_\Pi, \eta_{\Pi'}, a, r, b \rangle \in E$ iff whenever $(\gamma, [a]\psi) \in \Pi$ and $b(\gamma) = \mathtt{tt}$ then $(r(\gamma), \psi) \in \Pi'$.

Now it can be shown that $A_{\mathcal{C}}$ solves all problems of $\mathcal{C}$. In particular whenever $(\gamma, \psi) \in \Pi$ for some $\Pi \in \mathcal{C}$, then $[\eta_\Pi, \gamma]_{A^+_{\mathcal{C}}} \models_{\mathcal{D}} \psi$. To prove this we show that the relation $\Vdash$ defined by: $[\eta_\Pi, \gamma] \Vdash \psi$ iff $(\gamma, \psi) \in \Pi$ with $\Pi \in \mathcal{C}$ is a symbolic satisfiability relation. That is, we must show that $\Vdash$ satisfies the eleven implications of definition 5:

- The implications $i)$, $iii) - vi)$, $x)$ and $xi)$ follow from maximality of any Π in $\mathcal{C}$.
- The implications $ii)$ and $ix)$ follow directly from consistency of $\mathcal{C}$.
- The implications $vii)$ and $viii)$ follow from the construction of E which is always possible thanks to the consistency of $\mathcal{C}$.

□

Finally we have:

Lemma 4 *It is decidable whether a maximal problem is consistent.*

Proof Let S_{Π_m} be the set of maximal problems over $\mathcal{R}_k^{C+} \times L_\nu^\varphi$. Clearly S_{Π_m} is finite (since L_ν^φ and $\mathcal{R}_k^{C+}$ are too). Thus the set of relations $\mathcal{C}$ over maximal problems is finite. Now given a relation $\mathcal{C}$ it is easy to check whether $\mathcal{C}$ is consistent since the choices for possible reset set r over C and the set $\mathcal{B}_M(C)$[15] are both finite. □

Thus given a formula φ and bounds C and M, we can consider the (finitely many) maximal problems Π over C and M containing (γ_0, φ). It follows that φ is (C, M)–satisfiable precisely if one of these maximal problems is consistent, which is decidable due to Lemma 4. Note that the proof of Theorem 2 is constructive: given a consistency relation it gives a (C, M)-timed automata satisfying φ.

Example 6 Consider the formula φ in Example 3:

$$\varphi = \exists\,]0;1[\,\langle a\rangle \left[\left(\langle c\rangle\, \mathtt{tt}\right) \wedge \left(\forall\,]0;1[\,[c]\,\mathtt{ff}\right) \wedge \left(\exists\,]0;1[\,\langle b\rangle\,\mathtt{tt}\right) \wedge \left(\exists\,]0;1[\,[b]\,\mathtt{ff}\right)\right]$$

We can use the model construction algorithm presented above to show that no $(1,1)$-automata satisfies φ. Since φ is a one-formula clock and $|C| = 1$, we have $C^+ = \{x, y\}$ where x denotes the automata clock and y the formula clock. Let ψ be the formula s.t. $\varphi = \exists\,]0;1[\,\langle a\rangle\,\psi$.

Consider the problem $\Pi = \{(\gamma_0, \exists]0;1[\langle a\rangle\,\psi)\}$, where γ_0 refers to the regions of Example 4. The maximal problem including Π is $\Pi_0 = \{(\gamma_0, \exists]0,;1[\langle a\rangle\psi), (\gamma_6, \langle a\rangle\,\psi), (\gamma_0, \mathtt{tt}), (\gamma_6, \mathtt{tt})\}$. If Π_0 is consistent, there exists a relation $\mathcal{C}$ containing a maximal problem Π_1 s.t. for some $r_1 \in \{\{x\}, \emptyset\}$ and $b_1 \in \mathcal{B}_1(\{x\})$ with $b_1(\gamma_6) = \mathtt{tt}$ we have: $(r_1(\gamma_6), \psi) \in \Pi_1$. We distinguish two cases depending on r_1:

- $r_1 = \emptyset : \Pi_1$ contains (γ_6, ψ). Since it's maximal, it also contains $(\gamma_6, \langle c\rangle\,\mathtt{tt})$ and $(\gamma_6, \forall]0;1[[c]\,\mathtt{ff}$. Then $\{(\gamma_6, [c]\,\mathtt{ff}), (\gamma_{14}, [c]\,\mathtt{ff}), (\gamma_{24}, [c]\,\mathtt{ff})\} \subset \Pi_1$. Thus Π_1 is not consistent since $(\gamma_6, \langle c\rangle\,\mathtt{tt})$ and $(\gamma_6, [c]\,\mathtt{ff})$ require the existence of a maximal problem containing $(\gamma_6, \mathtt{ff})$ or $(\gamma_1, \mathtt{ff})$. Thus $\mathcal{C}$ is not a consistency relation.

[15] modulo boolean reduction

- $r_1 = \{x\}$: Π_1 contains (γ_1, ψ), $(\gamma_1, \langle c\rangle\,\mathtt{tt})$, $(\gamma_1, \forall]0;1[\,[c]\,\mathtt{f})$, $(\gamma_1, \exists]0;1[\,\langle b\rangle\,\mathtt{tt})$ and $(\gamma_1, \exists\,]0;1[\,[b]\,\mathtt{ff})$. In fact there are several possibilities for Π_1 depending on which term among $(\gamma_7, \langle b\rangle\,\mathtt{tt})$, $(\gamma_8, \langle b\rangle\,\mathtt{tt})$, $\gamma_9, \langle b\rangle\,\mathtt{tt})$ and which term among $(\gamma_7, [b]\,\mathtt{ff})$, $(\gamma_8, [b]\,\mathtt{ff})$, $\gamma_9, [b]\,\mathtt{ff})$ are contained in Π_1 due to its maximality. In any case there are some $(\gamma, \langle b\rangle\,\mathtt{tt})$ and $(\gamma', [b]\,\mathtt{ff})$ in Π_1 with $\gamma, \gamma' \in \{\gamma_7, \gamma_8, \gamma_9\}$. Then there exists a maximal problem Π_2 s.t. for some $r_2 \in \{\{x\}, \emptyset\}$ and $b_2 \in \mathcal{B}_1(\{x\})$ with $b_2(\gamma) = \mathtt{tt}$ and $(r_2(\gamma), \mathtt{tt}) \in \Pi_2$. But for any condition $b \in \mathcal{B}_1(\{x\})$ we have: $b(\gamma_7) = b(\gamma_8) = b(\gamma_9)$, and thus $(\gamma', [b]\mathtt{ff}) \in \Pi_1$ requires that $(r_2(\gamma'), \mathtt{ff})$ is in Π_2. Thus $\mathcal{C}$ is not a consistency relation.

Thus no $(1,1)$-automata satisfies φ. □

Thus the formula in the above example is satisfiable by a 2–clock automaton but by no $(1,1)$–automata. Using the easily established fact that timed bisimilar automata satisfy the same L_ν–formulas it follows that the automaton of Example 3 is inequivalent to all $(1,1)$–automata with respect to timed bisimilarity. Now combining the above bounded model–construction algorithm with the characteristic property construction of the previous section we obtain an algorithm for deciding whether a timed automaton can be simplified in either its number clocks or the size of the constants these clocks are compared to. Using this combined method it can (constructively) be seen that the 2–clock automaton obtained by changing the c–edge enabling condition in Example 1 from $x = 0$ to $x > 0$ may endeed be simplified to an equivalent $(1,1)$–automaton.

Corollary 2 *Given a timed automaton A, a clock set C and a natural number M, it is decidable whether there exists a (C, M)–automaton being timed bisimilar to A.*

Conclusion

This paper has presented two main contributions: (1) a *characteristic formula* construction which for any given timed automaton give a logical formula uniquely characterizing it; and (2) a *model construction* algorithm, which given a logical formula will (if possible) synthesize a satisfying timed automaton within given bounds on the number of clocks and constants used.

The results presented may be pursued and improved in a number of directions: The notion of a characteristic formula construction may be applied to other behavioural preorders in order to obtain corresponding preorder checking algorithms. We have already shown that characteristic formula constructs also exists for the "faster–than"–relation in [FT91] and the time–abstracted equivalence in [LW93].

The results of this paper only solve (positively) the decidability of a *bounded* satisfiability problem for L_ν. However, it follows from this result that the unconstrainted satisfiability problem is at least r.e. though we conjecture that this problem is in fact undecidable. Decidability of the satisfiability problem with only bounds on the number of clocks is also left as an open (and interesting) problem.

Finally, future work includes study of the decidability of the satisfiability problems for L_ν extended with a minimal fixedpoint construction.

References

[AC88] A. Arnold and P. Crubille. A linear algorithm to solve fixed–point equations on transition systems. *Information Processing Letters*, 29, 1988.

[ACD90] R. Alur, C. Courcoubetis, and D. Dill. Model–checking for Real–Time Systems. In *Proceedings of Logic in Computer Science*, pages 414–425. IEEE Computer Society Press, 1990.

[AD94] R. Alur and D. Dill. Automata for Modelling Real–Time Systems. *Theoretical Computer Science*, 126(2):183–236, April 1994.

[Alu91] R. Alur. *Techniques for Automatic Verification of Real-time Systems.* PhD thesis, Stanford University, 1991.

[And92] H.R. Andersen. Model checking and boolean graphs. In *Proceedings of ESOP'92*, volume 582 of *Lecture Notes in Computer Science, Springer Verlag*, Berlin, 1992. Springer.

[BCG88] M. C. Browne, E. M. Clarke, and O. Grümberg. Characterizing finite Kripke structures in propositional temporal logic. *Theoretical Computer Science*, 59:115–131, 1988.

[CE81] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using Branching Time Temporal Logic. In *Proc. Workshop on Logics of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71, Berlin, 1981. Springer Verlag.

[Cer92] Karlis Cerans. Decidability of bisimulation equivalences for parallel timer processes. In *Proc. of CAV'92*, volume 663 of *Lecture Notes in Computer Science, Springer Verlag*, Berlin, 1992. Springer Verlag.

[CES86] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite state concurrent system using temporal logic. *ACM Trans. on Programming Languages and Systems*, 8(2):244–263, 1986.

[CS91] R. Cleaveland and B. Steffen. Computing behavioural relations, logically. In *Proceedings of 18th International Colloquium on Automata, Languages and Programming*, volume 510 of *Lecture Notes in Computer Science, Springer Verlag*, Berlin, 1991. Springer.

[EC82] E. A. Emerson and E. M. Clarke. Using Branching Time Temporal Logic to synthesize synchronization skeletons. *Sci. Comput. Programming*, 2:241–266, 1982.

[EH85] E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the Temporal Logic of Branching Time. *J. Comput. System Sci.*, 30(1):1–24, 1985.

[EL86] E.A. Emerson and C.L Lei. Efficient model checking in fragments of the propositional mu–calculus. In *Proceedings of Logic in Computer Science*, pages 267–278. IEEE Computer Society Press, 1986.

[FT91] F.Moller and C. Tofts. Relating Processes with Respect to Speed. Technical Report ECS–LFCS–91–143, Department of Computer Science, University of Edinburgh, 1991.

[GS86] S. Graf and J. Sifakis. A Modal Characterization of Observational Congruence on Finite Terms of CCS. *Information and Control*, 68:125–145, 1986.

[HC68] G.E. Hughes and M.J. Cresswell. *An Introduction to Modal Logic.* Methuen and Co., 1968.

[HLY92] U. Holmer, K.G. Larsen, and W. Yi. Decidability of bisimulation equivalence between regular timed processes. In *Proceedings of CAV'91*, volume 575 of *Lecture Notes in Computer Science, Springer Verlag*, Berlin, 1992.

[HNSY92] T. A. Henzinger, Z. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. In *Logic in Computer Science*, 1992.

[IS94] A. Ingolfsdottir and B. Steffen. Characteristic formulae. *Information and Computation*, 110(1), 1994. To appear.

[Koz82] D. Kozen. Results on the propositional mu–calculus. In *Proc. of International Colloquium on Algorithms, Languages and Programming 1982*, volume 140 of *Lecture Notes in Computer Science, Springer Verlag*, Berlin, 1982.

[KP83] D. Kozen and R. Parikh. A decision procedure for the propositional mu–calculus. *Lecture Notes in Computer Science*, 1983.

[LW93] K.G. Larsen and Y. Wang. Time Abstracted Bisimulation: Implicit Specifications and Decidability. *In Proceedings of MFPS'93*, 1993.

[Mil89] R. Milner. *Communication and Concurrency.* prentice, Englewood Cliffs, 1989.

[Par81] D. Park. Concurrency and automata on infinite sequences. In *Proceedings of 5th GI Conference*, volume 104 of *Lecture Notes in Computer Science, Springer Verlag*, Berlin, 1981. Springer.

[QS82] J. P. Queille and J. Sifakis. Specification and verification of concurrent programs in CESAR. In *Proc. 5th Internat. Symp. on Programming*, volume 137 of *Lecture Notes in Computer Science*, pages 195–220, Berlin, 1982. Springer Verlag.

[Tar55] A. Tarski. A lattice–theoretical fixpoint theorem and its applications. *Pacific Journal of Math.*, 5, 1955.

[Wol85] P. Wolper. The tableau method for Temporal Logic: an overview. *Logique et Anal.*, 28:119–136, 1985.

[Xin92] Liu Xinxin. *Specification and Decomposition in Concurrency.* PhD thesis, Aalborg University, 1992. R 92–2005.

A Proof of Theorem 2

Proof $\Leftarrow$ We are going to show that $B = \{(\langle \rho, v\rangle_B, \langle \eta, u\rangle_A) \mid \langle \rho, v\,u\rangle \models_{B,\mathcal{D}} \Phi(\eta, [u])\}$ is a timed strong bisimulation.

- Assume $\langle \eta, u\rangle_A \xrightarrow{a} \langle \eta', r(u)\rangle_A$. But as $\langle \rho, v\,u\rangle_{B^+} \models \Phi(\eta, [u])$ we have $\langle \rho, v\,u\rangle_{B^+} \models \langle a\rangle\ (r \textsf{ in } \Phi(\eta', r([u])))$. Thus $\langle \rho, v\,u\rangle_{B^+} \xrightarrow{a} \langle \rho', v'\,u\rangle_{B^+}$ s.t. $\langle \rho', v'\ r(u)\rangle_{B^+} \models \Phi(\eta', r([u]))$. Then $\langle \rho, v\rangle_B \xrightarrow{a} \langle \rho', v'\rangle_B$ and by i.h. $(\langle \rho', v'\rangle_B, \langle \eta', r(u)\rangle_A) \in B$.

- Assume $\langle \rho, v\rangle_B \xrightarrow{a} \langle \rho', v'\rangle_B$. Then $\langle \rho, v\,u\rangle_{B^+} \xrightarrow{a} \langle \rho', v'\,u\rangle_{B^+}$. And as $\langle \rho, v\,u\rangle_{B^+} \models \Phi(\eta, [u])$ we have $\langle \rho', v'\,u\rangle_{B^+} \models r_e \textsf{ in } \Phi(\eta'_e, r_e(r[u]))$ for some $e \in E(\eta, [u], a)$. Then $\langle \rho', v'\ r_e(u)\rangle_{B^+} \models \Phi(\eta'_e, [r_e(u)])$. Thus we have $(\langle \rho', v'\rangle_B, \langle \eta'_e, r_e(u)\rangle_A) \in B$.

- Finally to complete the proof we must show that for all $d \in \mathbf{R}$, we have: $(\langle \rho, v+d\rangle_B, \langle \eta, u+d\rangle_A) \in B$. There exists k s.t. $[u+d] = [u]^k$. As $\langle \rho, v\,u\rangle_{B^+} \models \Phi(\eta, [u])$ it follows that $\langle \rho, v+d\ u+d\rangle_{B^+} \models \beta([u]^k) \Rightarrow \Phi(\eta, [u]^k)$. Clearly it implies $\langle \rho, v+d\,u+d\rangle_{B^+} \models \Phi(\eta, [u]^k)$ or $\langle \rho, v+d\,u+d\rangle_{B^+} \models \Phi(\eta, [u+d])$. Thus $(\langle \rho, v+d\rangle_B, \langle \eta, u+d\rangle_A) \in B$.

$\Rightarrow$ Let $\mathcal{D}$ the declaration associating each $[\eta, \gamma]$ with $\Phi(\eta, \gamma)$. We define $\rhd_{\mathcal{D}}$ by structural induction as follows:

$$
\begin{array}{lcl}
\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \mathtt{tt} & \Leftrightarrow & \text{true} \\
\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \mathtt{ff} & \Leftrightarrow & \text{false} \\
\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \varphi \wedge \psi & \Leftrightarrow & \langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \varphi \ \text{ and } \ \langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \psi \\
\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \varphi \vee \psi & \Leftrightarrow & \langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \varphi \ \text{ or } \ \langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \psi \\
\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \exists\!\!\exists\, \varphi & \Leftrightarrow & \exists d \in \mathbf{R}.\ \ \langle \rho, v{+}d\ \ u{+}d\rangle_{B^+} \rhd_{\mathcal{D}} \varphi \\
\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \forall\!\!\forall\, \varphi & \Leftrightarrow & \forall d \in \mathbf{R}.\ \ \langle \rho, v{+}d\,u{+}d\rangle_{B^+} \rhd_{\mathcal{D}} \varphi \\
\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \langle a\rangle\, \varphi & \Leftrightarrow & \exists\, \langle \rho', v'\rangle_B.\ \ \langle \rho, v\rangle_B \xrightarrow{a} \langle \rho', v'\rangle_B \ \text{ and} \\
 & & \qquad\qquad\quad \langle \rho', v'\,u\rangle_{B^+} \rhd_{\mathcal{D}} \varphi \\
\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} [a]\, \varphi & \Leftrightarrow & \forall\, \langle \rho', v'\rangle_B.\ \ \langle \rho, v\rangle_B \xrightarrow{a} \langle \rho', v'\rangle_B \text{ implies} \\
 & & \qquad\qquad\quad \langle \rho', v'\,u\rangle_{B^+} \rhd_{\mathcal{D}} \varphi \\
\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} x{+}m{\sim}y{+}n & \Leftrightarrow & u(x) + m \sim u(y) + n \\
\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} x \textsf{ in } \varphi & \Leftrightarrow & \langle \rho, v\,u'\rangle_{B^+} \rhd_{\mathcal{D}} \varphi \ \text{ and } \ u' = [\{x\} \to 0]u \\
\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \Phi(\eta, [u]) & \Leftrightarrow & \langle \rho, v\rangle_B \sim \langle \eta, u\rangle_A
\end{array}
$$

We are going to prove that $\rhd_{\mathcal{D}}$ is a satisfiabilty relation. To show this it is sufficient to demonstrate that $\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \Phi(\eta, [u])$ implies $\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \mathcal{D}(\Phi(\eta, [u]))$:

- Consider $\langle \eta, u\rangle_A \xrightarrow{a_e} \langle \eta'_e, r_e(u)\rangle_A$. Since $\langle \rho, v\,u\rangle_{B^+} \rhd_{\mathcal{D}} \Phi(\eta, [u])$, we have $\langle \eta, u\rangle_A \sim \langle \rho, v\rangle_B$. Then there exists $\langle \rho, v\rangle_B \xrightarrow{a_e} \langle \rho', v'\rangle_B$ s.t. $\langle \eta'_e, r_e(u)\rangle_A \sim \langle \rho', v'\rangle_B$. Thus by def. of $\rhd_{\mathcal{D}}$ we have $\langle \rho', v'\ r_e(u)\rangle_{B^+} \rhd_{\mathcal{D}} \Phi(\eta'_e, [r_e(u)])$. Finally we have $\langle \rho, v u\rangle_{B^+} \rhd_{\mathcal{D}} \langle a_e\rangle\, (r_e \textsf{in} \Phi(\eta'_e, r([u])))$ for any $e \in E(\eta, [u])$.

- Consider $\langle \rho, v\rangle_B \xrightarrow{a} \langle \rho', v'\rangle_B$. Since $\langle \eta, u\rangle_A \sim \langle \rho, v\rangle_B$, there exists $\langle \eta, u\rangle_A \xrightarrow{a} \langle \eta'_e, r_e(u)\rangle_A$ s.t. $\langle \eta'_e, r_e(u)\rangle_A \sim \langle \rho', v'\rangle_B$. Thus $\langle \rho', v'\ r_e(u)\rangle_{B^+} \rhd_{\mathcal{D}} \Phi(\eta'_e, [r_e(u)])$, and we have $\langle \rho', v'\ u\rangle_{B^+} \rhd_{\mathcal{D}} r_e \textsf{ in } \Phi(\eta'_e, r([u]))$ for some $e \in E(\eta, [u], a)$.

- Consider $\langle \rho, v\rangle_B \xrightarrow{\epsilon(d)} \langle \rho, v+d\rangle_B$. Then we know $\langle \eta, u+d\rangle_A \sim \langle \rho, v+d\rangle_B$ and thus $\langle \rho, v+d\, u+d\rangle_{B^+} \rhd_{\mathcal{D}} \Phi(\eta, [u+d])$. Let $l \in \mathbf{N}$ s.t. $[u]^l = [u+d]$, We have $\langle \rho, v+d\, u+d\rangle_{B^+} \rhd_{\mathcal{D}} \beta([u]^l) \;\Rightarrow\; \Phi(\eta, [u]^l)$.

Then we have $\langle \rho, v u\rangle_{B^+} \rhd_{\mathcal{D}} \Phi(\eta, [u]) \;\;\Rightarrow\;\; \langle \rho, v u\rangle_{B^+} \rhd_{\mathcal{D}} \mathcal{D}(\Phi(\eta, [u]))$. Thus $\rhd_{\mathcal{D}}$ is a satisfiability relation and is included in $\models_{\mathcal{D}}$. Thus, we can conclude that $\langle \rho, v\, u\rangle_{B^+} \models_{\mathcal{D}} \Phi(\eta, [u])$ whenever $\langle \rho, v\rangle_B \sim \langle \eta, u\rangle_A$. □

B Proof of Proposition 1

Proof $\Rightarrow$ Given $A = \langle \mathcal{A}, N, \eta_0, C, E\rangle$ a timed automata s.t. $A \models \Psi_l$ (i.e. $\langle \eta_0, v_0\rangle_A \models \Psi_l$). Let Φ_l be the subformula $\Big[\bigwedge_{i=1..l} \exists\,]0;1[\, (\langle a_i\rangle\, \mathtt{tt} \wedge \bigwedge_{j \neq i} [a_j]\, \mathtt{ff}) \Big]$. Since $A \models \Psi_l$, there exists a state $\langle \eta, v\rangle_A$ satisfying Φ_l. But it requires that there exists at least l different reachable regions with a $]0;1[$ delay from $\langle \eta, v\rangle_A$ and then $2|C| + 1 \geq l$.

$\Leftarrow$ It is easy to build a p-clock automata (with $l \leq 2p+1$) satisfying Ψ_l: Consider the p-clock automata s.t. the first l a-transitions [16] allow to reach a state $\langle \eta, v\rangle_A$ s.t. $0 < v(x_p) < \ldots < v(x_1) < 1$. Moreover we build l transitions $\langle \eta, \eta'_i, a_i, \{\}, b_i\rangle$ with $b_1 = (x_1 < 1)$, $b_2 = (x_1 = 1)$, $b_3 = (x_2 < 1 \wedge x_1 > 1), \ldots$ □

[16] In fact p transitions suffices to reach such a state.

Recent Publications in the BRICS Report Series

RS-95-2 François Laroussinie, Kim G. Larsen, and Carsten Weise. *From Timed Automata to Logic - and Back*. January 1995. 21 pp.

RS-95-1 Gudmund Skovbjerg Frandsen, Thore Husfeldt, Peter Bro Miltersen, Theis Rauhe, and Søren Skyum. *Dynamic Algorithms for the Dyck Languages*. January 1995. 21 pp.

RS-94-48 Jens Chr. Godskesen and Kim G. Larsen. *Synthesizing Distinguishing Formulae for Real Time Systems*. December 1994. 21 pp.

RS-94-47 Kim G. Larsen, Bernhard Steffen, and Carsten Weise. *A Constraint Oriented Proof Methodology based on Modal Transition Systems*. December 1994. 13 pp.

RS-94-46 Amos Beimel, Anna Gál, and Mike Paterson. *Lower Bounds for Monotone Span Programs*. December 1994. 14 pp.

RS-94-45 Jørgen H. Andersen, Kåre J. Kristoffersen, Kim G. Larsen, and Jesper Niedermann. *Automatic Synthesis of Real Time Systems*. December 1994. 17 pp.

RS-94-44 Sten Agerholm. *A HOL Basis for Reasoning about Functional Programs*. December 1994. PhD thesis. viii+224 pp.

RS-94-43 Luca Aceto and Alan Jeffrey. *A Complete Axiomatization of Timed Bisimulation for a Class of Timed Regular Behaviours (Revised Version)*. December 1994. 18 pp. To appear in *Theoretical Computer Science*.

RS-94-42 Dany Breslauer and Leszek Gąsieniec. *Efficient String Matching on Coded Texts*. December 1994. 20 pp.

RS-94-41 Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. *On Data Structures and Asymmetric Communication Complexity*. December 1994. 17 pp.