The Observational Power of Clocks*

Rajeev Alur[†] Costas Courcoubetis[‡] Thomas A. Henzinger[§]

Abstract

We develop a theory of equivalences for timed systems. Two systems are equivalent iff external observers cannot observe differences in their behavior. The notion of equivalence depends, therefore, on the distinguishing power of the observers. The power of an observer to measure time results in untimed, clock, and timed equivalences: an untimed observer cannot measure the time difference between events; a clock observer uses a clock to measure time differences with finite precision; a timed observer is able to measure time differences with arbitrary precision.

We show that the distinguishing power of clock observers grows with the number of observers, and approaches, in the limit, the distinguishing power of a timed observer. More precisely, given any equivalence for untimed systems, two timed systems are k-clock congruent, for a nonnegative integer k, iff their compositions with every environment that uses k clocks are untimed equivalent. Both k-clock bisimulation congruence and k-clock trace congruence form strict decidable hierarchies that converge towards the corresponding timed equivalences. Moreover, k-clock bisimulation congruence and k-clock trace congruence provide an adequate and abstract semantics for branching-time and linear-time logics with k clocks.

Our results impact on the verification of timed systems in two ways. First, our decision procedure for k-clock bisimulation congruence leads to a new, symbolic, decision procedure for timed bisimilarity. Second, timed trace equivalence is known to be undecidable. If the number of environment clocks is bounded, however, then our decision procedure for k-clock trace congruence allows the verification of timed systems in a trace model.

*A preliminary version of this paper appeared in the *Proceedings of the International Conference on Concurrency Theory* (CONCUR 94), *Lecture Notes in Computer Science* **836**, Springer-Verlag, 1994, pp. 162–177.

[†]AT&T Bell Laboratories, Murray Hill, NJ.

[‡]Department of Computer Science, University of Crete, Greece. Supported in part by the BRA ESPRIT project REACT.

[§]Department of Computer Science, Cornell University, Ithaca, NY. Supported in part by the National Science Foundation under grant CCR-9200794 and by the United States Air Force Office of Scientific Research under contract F49620-93-1-0056.

1 Introduction

At the center of every theory of concurrency lies a notion of equivalence between systems: it indicates what aspects of a system behavior are considered to be observable. In the case of untimed systems, a variety of equivalences have been promoted—most notably, perhaps, bisimilarity and trace equivalence—and although there may be no agreement as to which equivalence is most appropriate, the relationship between different equivalences is well understood (see, for example, [13]). This is not the case for timed systems, where the introduction of time as a continuous quantity makes the question of what is observable even more subtle. This paper studies the relationship between several equivalences on timed systems that are induced by different capabilities of an observer to measure time.

A timed system always proceeds in one of two ways—by performing an action or by letting a certain amount of real time pass (see, for example, [4, 8, 12, 16]). The observability of actions and time delays are orthogonal issues, and as we are interested in the latter, we settle the former by studying the common untimed equivalences of bisimilarity (Section 3) and trace equivalence (Section 4).

From time-abstract and timed equivalences There are two extreme capabilities of an observer to measure time: (1) it may not be able to measure the duration of any delay, or (2) it may be able to measure the exact real-numbered duration of every delay. The first assumption leads to *time-abstract* (or *untimed*) *equivalences*; the second assumption, to *timed equivalences*. Both assumptions, however, have drawbacks. Time-abstract equivalences, on one hand, are unrealistically weak, because they do not admit timed systems themselves as observers; that is, time-abstract equivalences are not congruences under parallel composition. Timed equivalences, on the other hand, are unrealistically strong—what system can observe a delay of duration π?—uncomfortably strong—timed trace equivalence, for example, is undecidable—and, as we will demonstrate, unnecessarily strong.

To resource-bounded observational congruences We focus on the resources that an observer needs to distinguish timed systems and define a sequence of congruences that lie between time-abstract and timed equivalences. A *clock* is a resource that measures the length of a time interval with finite (integer) precision. The assumption that the observer has k clocks to measure time leads to *k-clock congruences*. We show that the k-clock congruences form, in both the bisimulation and trace worlds, a strict infinite hierarchy between the corresponding time-abstract and timed equivalences; that is, the distinguishing power of an observer increases with the number of clocks it uses. Alternatively, if every process has one clock, then the distinguishing power of an environment increases with the number of environment processes.

Decision procedures are known for time-abstract and timed bisimilarity and for time-abstract trace equivalence [2, 4, 14, 15, 17]. We present algorithms for deciding the k-clock congruences. In the bisimulation world, where the k-clock hierarchy collapses at $2n + 1$ clocks for a given pair of systems with n clocks, we obtain a new algorithm for deciding timed bisimilarity; unlike the decision procedure of [14], our algorithm can be executed symbolically. In the trace world, where the timed equivalence is undecidable, we obtain an algorithm for checking the correctness of substitutions in a hierarchical verification process. In both worlds, our results can be used to replace a timed system with a simpler system, provided the number of environment clocks is bounded.

Via automata, games, and logics Clocks provide a clean and general paradigm for specifying timed systems: clocks have been added to such diverse languages as temporal logic [5], ω-automata [4], and process algebra [11]. We develop our results using the model of *timed automata* (Section 2). A timed automaton operates with finite control—a finite set of control locations and a finite set of real-valued clocks. All clocks proceed at the same rate and measure the amount of time that has elapsed since they were started (or reset). Each automaton transition may compare some of the clock values with integer constants and reset some of the clocks. This model of timed automata has been widely and successfully used for the specification and verification of real-time systems [1, 3, 4, 6, 7, 9, 11, 12].

Our results on bisimulation equivalences in Section 3 are obtained by studying bisimulation games. We define *time-abstract* (or *untimed*), *k-clock*, and *timed bisimulation games*. Suppose that Player I attempts to distinguish two systems, while Player II tries to show bisimilarity. In a timed game, Player II must match a delay of Player I with a delay of the same duration; in a time-abstract game, Player II can match a delay of Player I with a delay of arbitrary duration; in a k-clock game, Player I may choose, in addition to a delay, any constraints on the values of k clocks, and Player II can match the delay of Player I with a delay of arbitrary duration as long as the clock constraints are satisfied.

We also provide logical characterizations of the k-clock congruences. For this purpose, we add clock variables to modal logics and prove Hennessy-Milner-like theorems. In the bisimulation world, two systems are k-clock congruent iff they cannot be distinguished by branching-time formulas with k clocks; in the trace world, iff they cannot be distinguished by linear-time formulas with k clocks. The limit of all k-clock congruences provides, then, an adequate and abstract semantics for clock logics. While in general this limit—*untimed bisimulation congruence* and *untimed trace congruence*—is weaker than the corresponding timed equivalence, it surprisingly coincides with the corresponding timed equivalence on initial system states, in which all clock values are 0.

Related work The relationship between timed bisimilarity and untimed bisimulation congruence is studied also in [10]. There, it is shown that timed bisimilarity coincides with untimed bisimulation congruence provided that the observer can compare clock values with arbitrary rational constants; indeed, in this scenario a single observer clock suffices. Furthermore, it is proved that if the observer is required to compare clock values with multiples of $1/n$, then the observational power grows with n, yielding a strict hierarchy of equivalences based on the time granularity of the observer. By contrast, we assume that the observer clocks have the same granularity as the system clocks and show that the power of the observer increases with the number of clocks it uses.

2 Timed Transition Systems

A *labeled transition system* consists of a set Q of states, a set L of labels, and a family $\{\stackrel{\alpha}{\to} \subseteq Q^2 \mid \alpha \in L\}$ of transition relations, one for each label in L. The transition relation can be extended to finite words over L: for a word $\overline{\alpha} = \alpha_0\alpha_1 \cdots \alpha_n$ in L^* and two states q and q', define $q \stackrel{\overline{\alpha}}{\to} q'$ iff there exist states $q_0, q_1, \ldots, q_{n-1}$ such that $q \stackrel{\alpha_0}{\to} q_0 \stackrel{\alpha_1}{\to} \cdots \stackrel{\alpha_{n-1}}{\to} q_{n-1} \stackrel{\alpha_n}{\to} q'$.

Let the *time domain* $\mathbb{R}_{\geq 0}$ be the set of nonnegative real numbers. A *timed transition system* is a labeled transition system with $\mathbb{R}_{\geq 0} \subseteq L$; that is, the label set includes all time increments, and the transition relation $\stackrel{\delta}{\to}$, for a time increment δ, represents a delay of duration δ. We define timed transition systems by timed automata.

Timed automata A *timed automaton* A is a tuple (Σ, Γ, V, E), where Σ is a finite input alphabet, Γ is a finite set of clocks, V is a finite set of locations, and E is a finite set of edges. Each edge is a tuple $(s, \sigma, \mu, \lambda, s')$ that represents a transition from location $s \in V$ to location $s' \in V$ on the input symbol $\sigma \in \Sigma$. The edge constraint μ is a boolean combination of atomic formulas of the form $x \leq c$ or $c \leq x$, for a clock x and a nonnegative integer c. The reset set $\lambda \subseteq \Gamma$ specifies the clocks that are reset,

A *state* of the timed automaton A is a pair (s, ν) consisting of a location $s \in V$ and a clock mapping $\nu\colon \Gamma \mapsto \mathbb{R}_{\geq 0}$ that assigns a time value to each clock. All clocks are initialized to zero: the state (s, ν) is *initial* iff $\nu(x) = 0$ for each clock x. We write Q_A for the state set of A, and Q_A^0 for the set of initial states of A. The timed automaton A proceeds from state to state in two ways.

- *Time successor.* For every time increment $\delta \in \mathbb{R}_{\geq 0}$ and every state (s, ν), let $(s, \nu) \stackrel{\delta}{\to} (s, \nu + \delta)$, where $\nu + \delta$ is the clock mapping that assigns the value $\nu(x) + \delta$ to each clock x.
- *Transition successor.* For every input symbol σ, every state (s, ν), and every edge $(s, \sigma, \mu, \lambda, s')$ such that ν satisfies the edge constraint μ, let $(s, \nu) \stackrel{\sigma}{\to} (s', \nu[\lambda := 0])$, where $\nu[\lambda := 0]$ is the clock mapping that assigns the value 0 to each clock $x \in \lambda$, and the value $\nu(x)$ to each clock $x \notin \lambda$.

We associate two labeled transition systems with the timed automaton A.

1. First, we observe both input symbols and time increments. The *timed transition system* of A, denoted by $\mathcal{S}_t(A)$, consists of the state set Q_A, the label set $\mathbb{R}_{\geq 0} \cup \Sigma$, and the transition relations $\xrightarrow{\alpha}$, for $\alpha \in \mathbb{R}_{\geq 0} \cup \Sigma$.

2. Second, we observe input symbols and hide time increments. For every input symbol σ and every pair of states $q, q' \in Q_A$, define $q \overset{\sigma}{\Rightarrow} q'$ iff $q \xrightarrow{\delta\sigma\delta'} q'$ for some time increments $\delta, \delta' \in \mathbb{R}_{\geq 0}$. The *time-abstract* (or *untimed*) *transition system* of A, denoted by $\mathcal{S}_u(A)$, consists of the state set Q_A, the label set Σ, and the transition relations $\overset{\sigma}{\Rightarrow}$, for $\sigma \in \Sigma$.

Given two timed automata $A = (\Sigma, \Gamma, V, E)$ and $A' = (\Sigma, \Gamma', V', E')$ over the same input alphabet Σ and disjoint sets Γ and Γ' of clocks, the *parallel composition* $A \otimes A'$ is the timed automaton $(\Sigma, \Gamma \cup \Gamma', V \times V', E'')$ such that $((s_1, s_1'), \sigma, \mu \wedge \mu', \lambda \cup \lambda', (s_2, s_2')) \in E''$ iff $(s_1, \sigma, \mu, \lambda, s_2) \in E$ and $(s_1', \sigma, \mu', \lambda', s_2') \in E'$. The timed transition system of $A \otimes A'$, then, represents the product of the timed transition systems of A and A': for all states $q_1, q_2 \in Q_A$, $q_1', q_2' \in Q_{A'}$, and all labels $\alpha \in \mathbb{R}_{\geq 0} \cup \Sigma$, we have $(q_1, q_1') \xrightarrow{\alpha} (q_2, q_2')$ for $\mathcal{S}_t(A \otimes A')$ iff $q_1 \xrightarrow{\alpha} q_2$ for $\mathcal{S}_t(A)$ and $q_1' \xrightarrow{\alpha} q_2'$ for $\mathcal{S}_t(A')$.

Equivalences and congruences on timed automata We define equivalence relations on systems as equivalence relations on system states: two systems are equivalent iff the initial states of the disjoint union of both systems are equivalent. Consider two equivalence relations $\equiv_1$ and $\equiv_2$ on the states of timed automata.

- We write $\equiv_1 \subseteq \equiv_2$, and call $\equiv_2$ *weaker than* $\equiv_1$ (on all states), iff for every timed automaton A and every pair of states $q, q' \in Q_A$, if $q \equiv_1 q'$ then $q \equiv_2 q'$.

- We write $\equiv_1 \subseteq_{init} \equiv_2$, and call $\equiv_2$ *weaker than* $\equiv_1$ *on initial states*, iff for every timed automaton A and every pair of initial states $q, q' \in Q_A^0$, if $q \equiv_1 q'$ then $q \equiv_2 q'$.

Notice that it may happen that $\equiv_2$ is weaker than $\equiv_1$, but $\equiv_1$ and $\equiv_2$ coincide on initial states. We say that $\equiv_2$ is *strictly weaker than* $\equiv_1$, written $\equiv_1 \prec \equiv_2$, iff $\equiv_2$ is weaker than $\equiv_1$ on all states and $\equiv_2 \not\subseteq_{init} \equiv_1$; that is, $\equiv_1$ distinguishes two $\equiv_2$-equivalent initial states.

The congruence induced by an equivalence relation on systems depends on the choice of operators used to build complex systems from simple systems. We study the parallel composition operator $\otimes$ on timed automata. For a given alphabet Σ, let TA_Σ denote the set of timed automata over Σ, and for $k \geq 0$, let TA_Σ^k denote the set of timed automata over Σ with at most k clocks. An equivalence relation $\equiv$ is a *congruence* iff whenever $q \equiv q'$ for two states q and q' of a timed automaton $A \in \mathrm{TA}_\Sigma$, then for all timed automata $B \in \mathrm{TA}_\Sigma$ and all initial states $q_0 \in Q_B^0$, the equivalence $(q, q_0) \equiv (q', q_0)$ holds for the product automaton $A \otimes B$. The congruence *induced by* the equivalence relation $\equiv$ is the weakest congruence relation that is stronger than $\equiv$.

If an equivalence relation $\equiv$ is not a congruence, then there are two $\equiv$-equivalent states q and q' of a timed automaton A—the *observed* automaton—and there is an initial state q_0 of a timed automaton B—the *observer* automaton—such that $(q, q_0) \not\equiv (q', q_0)$ for the product automaton $A \otimes B$. The observer automaton B, then, distinguishes the two states q and q' of the observed automaton A. Notice that our definition requires the observer automaton to be in the class TA_Σ of the observed automaton. Furthermore, the clock constraints of B compare clock values to integer constants; that is, the observer automaton has the same granularity for counting time as the observed automaton.

Region equivalence Emptiness checking and model checking algorithms for timed automata are based on an equivalence relation that partitions the states of a timed automaton A into so-called regions [1, 4]. Two clock mappings ν and ν' for A are region equivalent, written $\nu \equiv_r \nu'$, iff

1. Corresponding clock values agree on the integer parts: for every clock x, either $\lfloor \nu(x) \rfloor = \lfloor \nu'(x) \rfloor$, or both $\nu(x)$ and $\nu'(x)$ exceed the largest constant that is compared with x by the edge constraints of A;

2. Corresponding clock values agree on the ordering of the fractional parts: (i) for every clock x, $\langle \nu(x) \rangle = 0$ iff $\langle \nu'(x) \rangle = 0$, where $\langle \delta \rangle = \delta - \lfloor \delta \rfloor$; and ($ii$) for every pair x and y of clocks, $\langle \nu(x) \rangle \leq \langle \nu(y) \rangle$ iff $\langle \nu'(x) \rangle \leq \langle \nu'(y) \rangle$.

Two states (s, ν) and (s', ν') of A are *region equivalent* iff $s = s'$ and $\nu \equiv_r \nu'$; two states that belong to different timed automata are not region equivalent. An equivalence class of Q_A induced by $\equiv_r$ is called a *region*.

Two observations about region equivalence are in order. First, there are only finitely many regions (linear in the number of locations and edges, and singly exponential in the number of clocks and the length of edge constraints). Second, region equivalence is a congruence.

3 Bisimulation Equivalences

Consider a labeled transition system $\mathcal{S}$ with state set Q and label set L. An equivalence relation $\equiv \subseteq Q^2$ is a *bisimulation* for $\mathcal{S}$ iff $q_1 \equiv q_2$ implies for every label $\alpha \in L$ that

- If $q_1 \xrightarrow{\alpha} q_1'$, then there exists a state q_2' such that $q_1' \equiv q_2'$ and $q_2 \xrightarrow{\alpha} q_2'$;
- If $q_2 \xrightarrow{\alpha} q_2'$, then there exists a state q_1' such that $q_1' \equiv q_2'$ and $q_1 \xrightarrow{\alpha} q_1'$.

Two states q and q' are *bisimilar* with respect to $\mathcal{S}$ iff there exists a bisimulation $\equiv$ for $\mathcal{S}$ such that $q \equiv q'$; that is, bisimilarity with respect to $\mathcal{S}$ is the weakest bisimulation for $\mathcal{S}$.

Bisimulation can be viewed as a game between two players. Let q_1 and q_2 be two states of $\mathcal{S}$.

- *Move of Player I.* Player I chooses a side $i \in \{1,2\}$, a label $\alpha \in L$, and a state q_i' such that $q_i \xrightarrow{\alpha} q_i'$.
- *Move of Player II.* Let $j \in \{1,2\}$ such that $i \neq j$. Player II chooses a state q_j' such that $q_j \xrightarrow{\alpha} q_j'$. If no such state exists, Player I wins the game. Otherwise, the game continues on the two states q_1' and q_2'.

Thus, at every step Player I chooses a move, and Player II responds with a matching move. The goal of Player I is to distinguish the two starting states q and q' by enforcing a situation in which Player II cannot find a matching move. The starting states are bisimilar iff Player I cannot distinguish them; that is, iff Player I does not have a winning strategy.

The weakest bisimulation for $\mathcal{S}$ can be computed by an iterative approximation procedure that repeatedly refines a partition of the state set Q until a bisimulation is obtained. The initial partition Π_0 contains a single equivalence class. Now consider the partition Π_i after the i-th refinement. For two subsets $\pi, \pi' \subseteq Q$, and a label $\alpha \in L$, let $pre_\alpha(\pi, \pi')$ be the set of states $q \in \pi$ such that $q \xrightarrow{\alpha} q'$ for some $q' \in \pi'$. If there are two equivalence classes $\pi, \pi' \in \Pi_i$ and a label $\alpha \in L$ such that both $pre_\alpha(\pi, \pi')$ and $\pi - pre_\alpha(\pi, \pi')$ are nonempty, then we obtain the new partition Π_{i+1} by splitting π into the two equivalence classes $pre_\alpha(\pi, \pi')$ and $\pi - pre_\alpha(\pi, \pi')$. If no such splitting is possible, then the current partition is the weakest bisimulation for $\mathcal{S}$.

The iterative approximation procedure is a semidecision procedure for bisimilarity iff each partition is computable; that is, the label set is finite and the equivalence classes can be represented in an effective manner that supports the operations *pre* and set difference. The iterative approximation procedure is a decision procedure for bisimilarity iff it is a semidecision procedure that terminates. In the case of timed transition systems, the state set is infinite and, therefore, termination is not necessarily guaranteed.

3.1 Timed bisimilarity

Two states q and q' of a timed automaton A are *timed bisimilar*, written $q \equiv_{tb} q'$, iff q and q' are bisimilar with respect to the timed transition system $\mathcal{S}_t(A)$. If timed bisimulation is viewed as a game, whenever Player I takes a transition step, Player II must match it with a transition on the same input symbol; and whenever Player I lets time δ elapse, Player II must let time δ elapse, too.

Example 1 Consider a timed automaton with the single location s, the single clock x, and the single edge from s to itself labeled with the input symbol σ and the edge constraint $x = 1$. A state of the automaton is fully specified by the value $\delta \in \mathbb{R}_{\geq 0}$ of the clock x. Two states δ and δ' are timed bisimilar iff either $\delta = \delta'$ or both $\delta, \delta' > 1$. ∎

A few observations about timed bisimilarity are in order. First, Example 1 shows that, unlike in the case of region equivalence, the number of equivalence

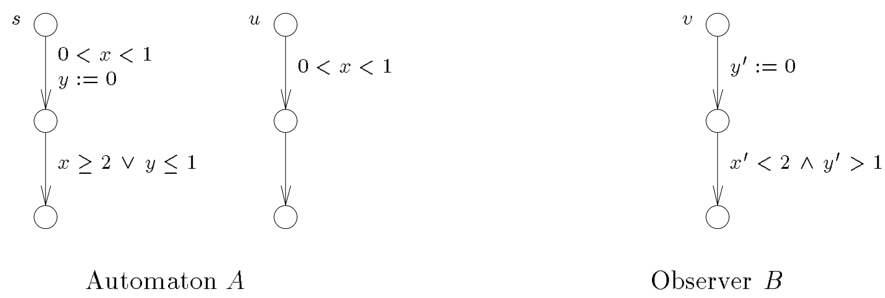

Figure 1: Example 2

classes of timed bisimilarity can be infinite. Indeed, the relations $\equiv_{tb}$ and $\equiv_r$ are incomparable; none is weaker than the other. Second, timed bisimilarity is a congruence. Third, timed bisimilarity is decidable: there is an EXPTIME algorithm to decide if two given states of a timed automaton are timed bisimilar [14].

3.2 Untimed bisimilarity

Two states q and q' of a timed automaton A are *untimed bisimilar*, written $q \equiv_{ub} q'$, iff q and q' are bisimilar with respect to the untimed transition system $\mathcal{S}_u(A)$. If untimed bisimulation is viewed as a game, Player II must match a transition step of Player I with a transition on the same input symbol, but whenever Player I lets time δ elapse, Player II may let any amount of time elapse.

Two states δ and δ' of Example 1 are untimed bisimilar iff either both $\delta, \delta' \leq 1$ or both $\delta, \delta' > 1$. Indeed, untimed bisimilarity always has finitely many equivalence classes, and each equivalence class is a union of regions [2, 10]. It follows that region equivalence is an untimed bisimulation ($\equiv_r \prec \equiv_{ub}$). Moreover, every timed bisimulation is an untimed bisimulation ($\equiv_{tb} \prec \equiv_{ub}$).

The iterative approximation procedure decides, in EXPTIME, if two given states of a timed automaton are untimed bisimilar: (1) every equivalence class that is computed by repeated refinement of the initial partition is a union of regions and, therefore, can be represented by a formula involving linear inequalities over the clock variables; (2) termination is guaranteed, because $\equiv_{ub}$ has finitely many equivalence classes.

The following example shows that untimed bisimilarity is not a congruence.

Example 2 Figure 1 shows a timed automaton A with two clocks x and y. The input alphabet is the singleton set $\{\sigma\}$. The two initial states $(s, \overline{0})$ and $(u, \overline{0})$ are untimed bisimilar. Now consider the product of A with the observer B with the two clocks x' and y'. The two states $((s, \overline{0}), (v, \overline{0}))$ and $((u, \overline{0}), (v, \overline{0}))$ of the product automaton $A \otimes B$ are not untimed bisimilar. ∎

3.3 Untimed bisimulation congruence

We now study the congruence induced by the untimed bisimulation $\equiv_{ub}$. The number of clocks of an observer increases its power to distinguish states. Consider, for instance, Example 2 again. The two states $(s, \overline{0})$ and $(u, \overline{0})$ can be distinguished using two clocks, but they cannot be distinguished by an observer with a single clock: for all timed automata B with one clock and all initial states q of B, the two states $((s, \overline{0}), q)$ and $((u, \overline{0}), q)$ of the product automaton are untimed bisimilar.

This observation prompts us to define a sequence of congruences. Let k be a nonnegative integer and let A be a timed automaton in TA_Σ. Two states q and q' of A are *k-clock congruent*, written $q \approx^k_{ub} q'$, iff for all timed automata $B \in \mathrm{TA}^k_\Sigma$ and all initial states q'' of B, the equivalence $(q, q'') \equiv_{ub} (q', q'')$ holds for the product automaton $A \otimes B$. Untimed bisimulation congruence is the intersection of all k-clock congruences: two states q and q' of A are *untimed bisimulation congruent*, written $q \approx_{ub} q'$, iff $q \approx^k_{ub} q'$ for all $k \geq 0$.

The k-clock congruences $\approx^k_{ub}$ can also be characterized using games. We modify the untimed game by introducing a set of environment clocks. If there are k environment clocks, then the state of the environment is represented by a k-tuple of real numbers. The game is played, instead of on states of a timed automaton, on pairs of the form (q, ν), where q is an automaton state and ν is a k-tuple of reals that provides time values for the environment clocks. We call such a pair an *augmented state*. With each time move, Player I chooses, in addition to a time increment, a constraint on the environment clocks. Player II may then choose any time increment provided it satisfies the clock constraint chosen by Player I.

Formally, a *k-clock move* α is a triple (σ, μ, λ), where σ is an input symbol, μ is a constraint on the environment clocks, and $\lambda \subseteq \{1, \ldots, k\}$ is a reset set. Given a timed automaton A, we thus obtain the *k-clock transition system* $\mathcal{S}^k_u(A)$. The states of $\mathcal{S}^k_u(A)$ are the augmented states $Q_A \times \mathbb{R}^k_{\geq 0}$; the labels of $\mathcal{S}^k_u(A)$ are the k-clock moves; and for each k-clock move, let $(q, \nu) \xrightarrow{\alpha} (q', \nu')$ iff there is a time increment δ such that $q \xrightarrow{\delta\sigma} q'$, and $\nu + \delta$ satisfies μ, and $\nu' = (\nu + \delta)[\lambda := 0]$.

The following lemma characterizes all k-clock congruences as bisimulations for k-clock transition systems.

Lemma 1 *Two states q and q' of a timed automaton A are k-clock congruent iff the augmented states $(q, \overline{0})$ and $(q', \overline{0})$ are bisimilar with respect to the k-clock transition system $\mathcal{S}^k_u(A)$.*

We can show that each equivalence class of $\approx_{ub}$ (and, hence, of $\approx^k_{ub}$ for each k) is a union of regions. It follows that the number of equivalence classes of untimed bisimulation congruence is finite and, thus, untimed bisimulation congruence

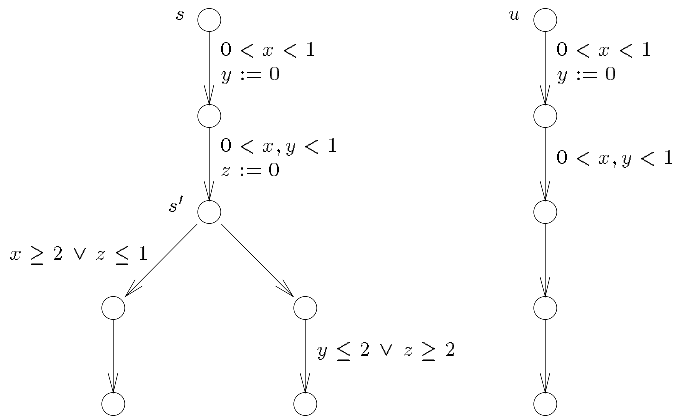

Figure 2: Example 3

differs from timed bisimilarity. Later, we will see that the two relations $\approx_{ub}$ and $\equiv_{tb}$ coincide on all *initial* states.

Theorem 1 *Untimed bisimulation congruence is strictly weaker than region equivalence* ($\equiv_r \prec \approx_{ub}$), *and is weaker than timed bisimilarity* ($\equiv_{tb} \subseteq \approx_{ub}$).

An infinite hierarchy of congruences Example 2 shows that 1-clock congruence is strictly weaker than the 2-clock congruence. By generalizing this example, we obtain a strict hierarchy of congruences; that is, each additional clock gives additional distinguishing power to an observer.

Example 3 Consider the timed automaton shown in Figure 2. It uses the three clocks x, y, and z. Consider the initial states $(s, \overline{0})$ and $(u, \overline{0})$. Using three environment clocks x', y', and z', we now show that Player I can distinguish these two states. Player I always moves on the right hand side. It resets y' in its first move, and resets z' in the second move. In the third move, it requires the environment clocks to satisfy the constraint $x' < 2 \wedge z' > 1$. This forces Player II to choose the left branch from location s'. In the fourth move, Player I uses the constraint $y' > 2 \wedge z' < 2$ to move to the final location on right hand side, and Player II cannot match this move.

On the other hand, it is not difficult to check that with only two environment clocks, Player I cannot distinguish $(s, \overline{0})$ and $(u, \overline{0})$. ■

Theorem 2 *The equivalence relations* $\approx_{ub}^k$, *for* $k \geq 0$, *form a strict hierarchy* ($\equiv_{ub} = \approx_{ub}^0 \succ \approx_{ub}^1 \succ \approx_{ub}^2 \succ \cdots \succ \approx_{ub}$).

The hierarchy of k-clock congruences collapses if we choose the natural numbers as time domain. In this case, a single observer clock suffices to distinguish any two noncongruent states ($\equiv_{ub} \succ \approx^1_{ub} = \approx_{ub}$).

Deciding k-clock congruence We now outline an algorithm for deciding k-clock congruence. The iterative approximation procedure on the k-clock transition system $\mathcal{S}^k_u(A)$ is not effective, as the number of possible k-clock moves is infinite. This is because the integer constants in a constraint on the environment clocks may be arbitrarily large.

We therefore modify the k-clock game in three ways. First, since the objective of Player I is to limit the possible choices of Player II, we require Player I to choose the tightest possible constraints on the environment clocks. Second, we require Player I to reset every environment clock when it reaches the value 1. Third, we allow Player I to employ ϵ-moves to reset a clock without choosing an input symbol. Formally, a *bounded k-clock ϵ-move* is of the form (σ, μ, λ), where $\sigma \in \Sigma \cup \{\epsilon\}$, the clock constraint μ is a conjunction of atomic constraints, one for each environment clock x, of the form $x < 1$ or $x = 1$, and the reset set λ contains the clock x iff μ contains the conjunct $x = 1$. We obtain the *bounded k-clock transition system* $\hat{\mathcal{S}}^k_u(A)$ with the state set $Q_A \times [0, 1]^k$ and with the bounded k-clock ϵ-moves as labels. The following lemma shows that this modification of the k-clock transition system for A does not change the induced bisimulation.

Lemma 2 *For all $k \geq 0$, all states q and q' of a timed automaton A, and all k-tuples $\nu, \nu' \in [0, 1]^k$ such that ν and ν' are region equivalent, the augmented states (q, ν) and (q', ν') are bisimilar with respect to the bounded k-clock transition system $\hat{\mathcal{S}}^k_u(A)$ iff they are bisimilar with respect to the k-clock transition system $\mathcal{S}^k_u(A)$.*

The k-clock congruence $\approx^k_{ub}$ can be computed, then, by iterative approximation on the bounded k-clock transition system $\hat{\mathcal{S}}^k_u(A)$. The procedure is effective, because the number of bounded k-clock ϵ-moves is finite (exponential in k) and each equivalence class that is computed is a union of extended regions (i.e., regions that contain both system and environment clocks); termination is guaranteed, because the number of extended regions is finite. The complexity of the algorithm is quadratic in the number of extended regions, and therefore exponential in both k and in the description of the timed automaton A.

Theorem 3 *Given two states q and q' of a timed automaton A and a nonnegative integer k, it can be decided in EXPTIME if $q \approx^k_{ub} q'$.*

Deciding untimed bisimulation congruence Suppose that two states q and q' of a timed automaton A with n clocks are not untimed bisimulation congruent. In the full paper, we show that Player I needs at most $2n + 1$ environment clocks to distinguish q and q'. Roughly speaking, this is because with $2n+1$ clocks, Player I can always keep an environment clock identical to each system clock on both sides of the game.

Lemma 3 *Let A be a timed automaton with n clocks. Two states of A are untimed bisimulation congruent iff they are $(2n+1)$-clock congruent.*

It follows that for any given timed automaton, the hierarchy of k-clock congruences collapses. This property, in conjunction with our decision procedure for k-clock congruence, allows us to decide untimed bisimulation congruence.

Theorem 4 *Given two initial states q and q' of a timed automaton A, it can be decided in EXPTIME if $q \approx_{ub} q'$.*

Deciding timed bisimilarity We have already seen that the relations $\equiv_{tb}$ and $\approx_{ub}$ are different: two states can be untimed bisimulation congruent but not timed bisimilar. Surprisingly, both relations coincide on initial states.

To see this, we introduce environment clocks in the timed game. We thus obtain a k-clock game on augmented states in which Player II must always choose the same time increment as Player I. In this game, the values of the environment clocks are identical on both sides; consequently, the state of the game is given by two augmented states of the form (q, ν) and (q', ν). If the game is started on initial states, then initially an environment clock is identical to each system clock, and provided there are enough environment clocks, this invariant can be maintained throughout the game. An augmented state (q, ν) in which each system clock equals some environment clock can be represented by a triple (s, γ, ν), where s is the location of q and γ is a mapping from system clocks to environment clocks. The following lemma is proved by induction on the number of moves in a game.

Lemma 4 *If Player I wins the timed game with environment clocks starting from the augmented states (s, γ, ν) and (s', γ', ν), then for all environment clock mappings ν', Player I wins the untimed game with environment clocks starting from (s, γ, ν) and (s', γ', ν').*

In other words, for augmented states in which all system clocks equal environment clocks, the timed game is equivalent to the untimed game. On initial states, untimed bisimulation congruence coincides therefore with timed bisimilarity.

Theorem 5 *Consider a timed automaton A with n clocks and two initial states q and q'. The following statements are equivalent:*

1. *The states q and q' are $(2n+1)$-clock congruent ($q \approx_{ub}^{2n+1} q'$).*
2. *The states q and q' are untimed bisimulation congruent ($q \approx_{ub} q'$).*

3. The states q and q' are timed bisimilar ($q \equiv_{tb} q'$).

This result gives us an EXPTIME iterative approximation algorithm for deciding timed bisimilarity of initial states (i.e., timed bisimilarity of initialized systems). Since the iterative approximation algorithm can be executed symbolically, by representing equivalence classes as formulas, we expect it to be more practical and flexible than the algorithm of [14].

3.4 Branching-time logics with clocks

The bisimilarity of states of a labeled transition system can be characterized by a modal next-state logic called Hennessy-Milner logic or, equivalently, by the branching-time temporal logic CTL: two states are bisimilar iff they satisfy the same CTL formulas.

In this section, we give a logical characterization of both k-clock congruence and untimed bisimulation congruence. For this purpose, we extend CTL with clock variables, thus obtaining the real-time logic TCTL [1]. The clock variables are bound by reset quantifiers, and they can be compared with nonnegative integer constants (the original definition of TCTL uses a freeze quantifier, which is equivalent to the reset quantifier). We use the modal operators $\exists\bigcirc\,\sigma.$ ("at the possible next input symbol σ") and $\exists\Diamond$ ("eventually along some word"); since we consider only finite words in this paper, the operator $\exists\Box$ is not useful.

Formally, the formulas of $\mathrm{TCTL}^{\diamond}$ are defined inductively as

$$\phi \;::=\; x \leq c \mid c \leq x \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \exists\bigcirc\,\sigma.\,\phi \mid \exists\Diamond\,\sigma.\,\phi \mid (x := 0).\,\phi$$

for clocks x, nonnegative integers c, and input symbols σ. Given a timed transition system with the state set Q and the label set $\mathbb{R}_{\geq 0} \cup \Sigma$, every formula of $\mathrm{TCTL}^{\diamond}$ defines a subset of Q. Let q be a state and let ν be a clock mapping. Then $q \models \phi$ iff $q \models_{\emptyset} \phi$ for the empty clock mapping $\emptyset$, and

- $q \models_{\nu} \exists\bigcirc\,\sigma.\,\phi$ iff there exist a state q' and a time increment δ such that $q \xrightarrow{\delta\sigma} q'$ and $q' \models_{\nu+\delta} \phi$;
- $q \models_{\nu} (x := 0).\,\phi$ iff $q \models_{\nu[x:=0]} \phi$.

The logic TCTL° is the fragment of $\mathrm{TCTL}^{\diamond}$ without the eventuality operator $\exists\Diamond$.

Example 4 Recall the timed automaton A of Figure 1. Let ϕ be the TCTL°-formula

$$(x := 0).\,\exists\bigcirc\,\sigma.\,(y := 0).\,\exists\bigcirc\,\sigma.\,(x < 2 \,\wedge\, y > 1).$$

Then $(s, \overline{0}) \not\models \phi$ and $(u, \overline{0}) \models \phi$. Thus the formula ϕ distinguishes the locations s and u. On the other hand, let ψ be the formula

$$((x := 0).\,\exists\bigcirc\,\sigma.\,\exists\bigcirc\,\sigma.\,x < 2) \,\wedge\, (\exists\bigcirc\,\sigma.\,(x := 0).\,\exists\bigcirc\,\sigma.\,x > 1).$$

Both states $(s, \overline{0})$ and $(u, \overline{0})$ satisfy ψ. Indeed, no TCTL$^{\diamond}$-formula that uses only one clock variable can distinguish the states $(s, \overline{0})$ and $(u, \overline{0})$. ■

The logics TCTL$^{\circ}$ and TCTL$^{\diamond}$ induce equivalence relations on the states of a timed automaton: two states q and q' are TCTL$^{\diamond}$-equivalent iff they satisfy the same TCTL$^{\diamond}$-formulas. We prove that TCTL$^{\circ}$-equivalence and TCTL$^{\diamond}$-equivalence coincide with each other and with untimed bisimulation congruence.

First we observe that the equivalence induced by TCTL$^{\diamond}$ is weaker than region equivalence [1]: if two augmented states (q, ν) and (q', ν') are region equivalent, then for all TCTL$^{\diamond}$-formulas ϕ, $q \models_{\nu} \phi$ iff $q' \models_{\nu'} \phi$. By induction on the number of moves that an observer needs to distinguish two augmented states in the k-clock game, we can show that if two augmented states are not bisimilar with respect to a k-clock transition system, then they can be distinguished by a TCTL$^{\circ}$-formula with at most k clock variables. On the other hand, if two augmented states are bisimilar with respect to a k-clock transition system, then we can prove, by induction on the structure of formulas, that they satisfy the same k-clock formulas of TCTL$^{\diamond}$. This leads to a logical characterization of k-clock congruence.

Theorem 6 *Consider a timed automaton A and two states q and q'. The following statements are equivalent:*

1. *The states q and q' are k-clock congruent ($q \approx_{ub}^{k} q'$).*
2. *For all* TCTL$^{\diamond}$*-formulas ϕ with at most k clock variables, $q \models \phi$ iff $q' \models \phi$.*
3. *For all* TCTL$^{\circ}$*-formulas ϕ with at most k clock variables, $q \models \phi$ iff $q' \models \phi$.*

It follows that untimed bisimulation congruence is an adequate and abstract semantics for both TCTL$^{\circ}$ and TCTL$^{\diamond}$.

4 Trace Equivalences

In the full paper, we develop the theory of trace equivalences for timed systems as carefully as the theory of bisimulation equivalences. Here we present only a few highlights.

Given a labeled transition system $\mathcal{S}$ with state set Q and label set L, and a state $q \in Q$, define the *language* $\mathcal{L}(\mathcal{S}, q)$ as the set $\{\overline{\alpha} \in L^* \mid \exists q'.\, q \xrightarrow{\overline{\alpha}} q'\}$ of finite words over L that are generated by $\mathcal{S}$ starting from q. Two states q and q' are *trace equivalent* with respect to $\mathcal{S}$ iff $\mathcal{L}(\mathcal{S}, q) = \mathcal{L}(\mathcal{S}, q')$. It is well-known that trace equivalence is strictly weaker than bisimilarity.

Timed trace equivalence Two states q and q' of a timed automaton A are *timed trace equivalent*, written $q \equiv_{tt} q'$, iff q and q' are trace equivalent with respect to the timed transition system $\mathcal{S}_t(A)$. That is, two states are timed trace equivalent iff they generate the same timed words—i.e., sequences of input symbols and time increments.

As expected, timed trace equivalence is strictly weaker than timed bisimilarity, but incomparable to region equivalence and incomparable to untimed bisimilarity. While timed trace equivalence is a congruence, it is computationally intractable. The undecidability proof for $\equiv_{tt}$ follows the proof that the language inclusion problem for timed automata over infinite words is undecidable [4].

Theorem 7 *The problem of deciding if two initial states of a timed automaton are timed trace equivalent is undecidable.*

Untimed trace equivalence Two states q and q' of a timed automaton A are *untimed trace equivalent*, written $q \equiv_{ut} q'$, iff q and q' are trace equivalent with respect to the untimed transition system $\mathcal{S}_u(A)$. That is, two states are untimed trace equivalent iff they generate the same untimed words—i.e., sequences of input symbols (all time increments are hidden).

Untimed trace equivalence is strictly weaker than region equivalence. Indeed, for all states q, the untimed language $\mathcal{L}(\mathcal{S}_u(A), q)$ can be characterized as a regular set over regions [4]. The problem of deciding untimed trace equivalence can then be reduced to the problem of deciding the language equivalence of two finite automata over regions.

Theorem 8 *There is an EXPSPACE algorithm that decides if two states of a timed automaton are untimed trace equivalent.*

Untimed trace equivalence is not a congruence. Hence we study the congruence induced by the equivalence relation $\equiv_{ut}$.

k-clock trace congruences As before, we consider a sequence of congruences. Let $k \geq 0$ and let A be a timed automaton in TA_Σ. Two states q and q' of A are *k-clock trace congruent*, written $q \approx_{ut}^k q'$, iff for all timed automata $B \in \mathrm{TA}_\Sigma^k$ and all initial states q'' of B, the equivalence $(q, q'') \equiv_{ut} (q', q'')$ holds for the product automaton $A \otimes B$. That is, two states are k-clock trace congruent iff in all environments with at most k clocks, they generate the same untimed words.

Lemma 5 *Two states q and q' of a timed automaton are k-clock trace congruent iff the augmented states $(q, \overline{0})$ and $(q', \overline{0})$ are trace equivalent with respect to the k-clock transition system $\mathcal{S}_u^k(A)$.*

As in the case of bisimulation, each additional clock increases the distinguishing power of an observer. Recall, for instance, Example 2. The initial states $(s, \overline{0})$ and $(u, \overline{0})$ generate different untimed words in the presence of an observer with two clocks and, therefore, are not 2-clock trace congruent. Both states, however, are bisimilar with respect to the 1-clock transition system $\mathcal{S}_u^1(A)$ and, hence, also 1-clock trace congruent.

Theorem 9 *The equivalence relations $\approx_{ut}^k$, for $k \geq 0$, form a strict hierarchy ($\equiv_{ut} = \approx_{ut}^0 \succ \approx_{ut}^1 \succ \approx_{ut}^2 \succ \cdots$).*

The k-clock trace congruences can be decided by a technique similar to the decision procedure for the k-clock bisimulation congruences. First, we show that $q \approx_{ut}^k q'$ iff the augmented states $(q, \overline{0})$ and $(q', \overline{0})$ are trace equivalent with respect to the bounded k-clock transition system $\hat{\mathcal{S}}_u^k(A)$. Then we decide trace equivalence with respect to $\hat{\mathcal{S}}_u^k(A)$ by constructing an equivalent finite automaton over extended regions.

Theorem 10 *Given two states q and q' of a timed automaton A and a nonnegative integer k, it can be decided in EXPSPACE if $q \approx_{ut}^k q'$.*

Untimed trace congruence Untimed trace congruence is the intersection of all k-clock trace congruences: two states q and q' of a timed automaton A are *untimed trace congruent*, written $q \approx_{ut} q'$, iff $q \approx_{ut}^k q'$ for all $k \geq 0$.

Theorem 11 *Untimed trace congruence is strictly weaker than region equivalence, is the same as timed trace equivalence on initial states, and is weaker than timed trace equivalence on all states.*

However, unlike in the case of bisimulation, timed trace congruence is still undecidable.

Theorem 12 *The problem of deciding if two initial states of a timed automaton are timed trace congruent is undecidable.*

It follows that the verification of timed systems in a trace model is computationally intractable. The decision procedures for k-clock trace congruences are therefore all the more important: they can be used for the compositional verification of timed systems in the trace model, provided the number of environment clocks is bounded.

Linear-time logics with clocks If we extend linear temporal logic with clock variables, we obtain the real-time logic TPTL [5]. Since we consider only finite words, we omit the temporal operator $\Box$. The formulas of TPTL$^\circ$ are defined inductively as

$$\phi \ ::= \ x \leq c \mid c \leq x \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \bigcirc\sigma.\, \phi \mid (x := 0).\, \phi$$

for clocks x, nonnegative integers c, and input symbols σ. Every formula ϕ of TPTL$^\circ$ defines a set of timed words. We write $q \models \phi$ iff ϕ defines a superset of the timed language $\mathcal{L}(\mathcal{S}_t(A), q)$.

As in the branching-time case, two states of a timed automaton can be distinguished by a TPTL$^\circ$-formula iff they are not untimed trace congruent. However, unlike in the branching-time case, if two states can be distinguished by a TPTL$^\circ$-formula, then they can be distinguished already by a TPTL$^\circ$-formula that uses a single clock variable.

Theorem 13 *Consider two states q and q' of a timed automaton A. The following statements are equivalent:*

1. *The states q and q' are untimed trace congruent ($q \approx_{ut} q'$).*
2. *For all* TPTL$^{\circ}$*-formulas ϕ, $q \models \phi$ iff $q' \models \phi$.*
3. *For all* TPTL$^{\circ}$*-formulas ϕ with at most 1 clock variable, $q \models \phi$ iff $q' \models \phi$.*

Further Work

(1) We studied timed and untimed equivalences and k-clock congruences in the bisimulation and trace cases. Corresponding relations can, of course, be defined for any equivalence relation on labeled transition systems, say, failures equivalence.

(2) We restricted ourselves to finite behaviors of systems, thus omitting liveness constraints. In the context of trace equivalences in particular, one typically considers automata on infinite words with acceptance conditions [4], and the full temporal logic TPTL [5]. We expect our results to generalize in a straightforward way.

(3) Our subject was the *distinguishing* power of clocks as observers. A complementary topic is the *expressive* power of clocks as specifiers. For example, it not difficult to show that if we measure the expressive power of timed automata by their ability to define languages of timed words, then the class of automata with $k+1$ clocks is more expressive than the class of automata with at most k clocks. Similarly, the expressive power of TPTL-formulas strictly increases with the number of clock variables.

References

[1] R. Alur, C. Courcoubetis, and D. Dill. Model checking in dense real time. *Information and Computation*, 104:2–34, 1993.

[2] R. Alur, C. Courcoubetis, N. Halbwachs, D. Dill, and H. Wong-Toi. Minimization of timed transition systems. In *3rd CONCUR*, 340–354. Springer LNCS 630, 1992.

[3] R. Alur, C. Courcoubetis, and T. Henzinger. Computing accumulated delays in real-time systems. In *5th CAV*, 181–193. Springer LNCS 697, 1993.

[4] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[5] R. Alur and T. Henzinger. A really temporal logic. *J. ACM*, 41:181–204, 1994.

[6] R. Alur, T. Henzinger, and M. Vardi. Parametric real-time reasoning. In *25th ACM STOC*, 592–601, 1993.

[7] C. Courcoubetis and M. Yannakakis. Minimum and maximum delay problems in real-time systems. In *3rd CAV*, 399–409. Springer LNCS 575, 1991.

[8] T. Henzinger, Z. Manna, and A. Pnueli. Temporal proof methodologies for real-time systems. In *18th ACM POPL*, 353–366, 1991.

[9] T. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. In *7th IEEE LICS*, 394–406, 1992.

[10] K. Laren and Y. Wang. Time-abstracting bisimulation: implicit specifications and decidability. In *Mathematical Foundations of Programming Semantics*, 1993.

[11] N. Lynch and F. Vaandrager. Action transducers and timed automata. In *3rd CONCUR*, 436–455. Springer LNCS 630, 1992.

[12] X. Nicollin, J. Sifakis, and S. Yovine. From ATP to timed graphs and hybrid systems. In *Real Time: Theory in Practice*, 549–572. Springer LNCS 600, 1991.

[13] R. van Glabbeek. *Comparative Concurrency Semantics and Refinement of Actions*. PhD thesis, Vrije Universiteit te Amsterdam, 1990.

[14] K. Čerāns. Decidability of bisimulation equivalence for parallel timer processes. In *4th CAV*, 302–315. Springer LNCS 663, 1992.

[15] K. Čerāns, J. Godskesen, and K. Larsen. Timed modal specification: theory and tools. In *5th CAV*, 253–267. Springer LNCS 697, 1993.

[16] Y. Wang. Real-time behavior of asynchronous agents. In *1st CONCUR*, 502–520. Springer LNCS 458, 1990.

[17] M. Yannakakis and D. Lee. An efficient algorithm for minimizing real-time transition systems. In *5th CAV*, 210–224. Springer LNCS 697, 1993.