Computing Accumulated Delays in Real-time Systems*

Rajeev Alur† Costas Courcoubetis‡ Thomas A. Henzinger§

Abstract. We present a verification algorithm for duration properties of real-time systems. While simple real-time properties constrain the total elapsed time between events, duration properties constrain the accumulated satisfaction time of state predicates. We formalize the concept of durations by introducing duration measures for timed automata. A duration measure assigns to each finite run of a timed automaton a real number —the *duration* of the run— which may be the accumulated satisfaction time of a state predicate along the run. Given a timed automaton with a duration measure, an initial and a final state, and an arithmetic constraint, the *duration-bounded reachability problem* asks if there is a run of the automaton from the initial state to the final state such that the duration of the run satisfies the constraint. Our main result is an (optimal) PSPACE decision procedure for the duration-bounded reachability problem.

1 Introduction

Over the past decade, model checking [CE81, QS81] has emerged as a powerful tool for the automatic verification of finite-state systems. Recently the model-checking paradigm has been extended to real-time systems [ACD93, HNSY94, AFH96]. Thus, given the description of a finite-state system together with its timing assumptions, there are algorithms that test whether the system satisfies a specification written in a real-time temporal logic. A typical property that can be specified in real-time temporal logics is the following *time-bounded causality* property:

> A response is obtained whenever a ringer has been pressed *continuously* for 2 seconds.

Standard real-time temporal logics [AH92], however, have limited expressiveness and cannot specify some timing properties we may want to verify of a given system. In particular, they do not allow us to constrain the accumulated satisfaction times of state predicates. As an example, consider the following *duration-bounded causality* property:

> A response is obtained whenever a ringer has been pressed, *possibly intermittently*, for a total duration of 2 seconds. $(*)$

*A preliminary version of this paper appeared in the Proceedings of the Fifth International Conference on *Computer-Aided Verification* (CAV 93), Springer-Verlag LNCS 818, pp. 181–193, 1993.

†Bell Laboratories, Murray Hill, New Jersey, U.S.A.

‡Department of Computer Science, University of Crete, and Institute of Computer Science, FORTH, Greece.
Partially supported by the BRA ESPRIT project REACT.

§Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, U.S.A.
Partially supported by the ONR YIP award N00014-95-1-0520, by the NSF CAREER award CCR-9501708, by the NSF grants CCR-9200794 and CCR-9504469, by the AFOSR contract F49620-93-1-0056, and by the ARPA grant NAG2-892.

To specify this duration property, we need to measure the accumulated time spent in the state that models "the ringer is pressed." For this purpose, the concept of duration operators on state predicates is introduced in the *Calculus of Durations* [CHR91]. There, an axiom system is given for proving duration properties of real-time systems.

Here we address the *algorithmic* verification problem for duration properties of real-time systems. We use the formalism of timed automata [AD94] for representing real-time systems. A timed automaton operates with a finite control and a finite number of fictitious time gauges called *clocks*, which allow the annotation of the control graph with timing constraints. The state of a timed automaton includes, apart from the location of the control, also the real-numbered values of all clocks. Consequently, the state space of a timed automaton is infinite, and this complicates its analysis. The basic question about a timed automaton is the following *time-bounded reachability* problem:

> Given an initial state σ, a final state τ, and an interval I, is there a run of the automaton starting in state σ and ending in state τ such that the total elapsed time of the run is in the interval I? (†)

The solution to this problem relies on a partition of the infinite state space into finitely many regions, which are connected with transition and time edges to form the *region graph* of the timed automaton [AD94]. The states within a region are equivalent with respect to many standard questions. In particular, the region graph can be used for testing the emptiness of a timed automaton [AD94], for checking time-bounded branching properties [ACD93], for testing the bisimilarity of states [Cer92], and for computing lower and upper bounds on time delays [CY91]. Unfortunately, the region graph is *not* adequate for checking duration properties such as the duration-bounded causality property $(*)$; that is, of two runs that start in different states within the same region, one may satisfy the duration-bounded causality property, whereas the other one does not. Hence a new technique is needed for analyzing duration properties.

To introduce the concept of durations in a timed automaton, we associate with every finite run a nonnegative real number, which is called the *duration* of the run. The duration of a run is defined inductively using a *duration measure*, which is a function that maps the control locations to nonnegative integers: the duration of an empty run is 0; and the duration measure of a location gives the rate at which the duration of a run increases while the automaton control resides in that location. For example, a duration measure of 0 means that the duration of the run stays unchanged (i.e., the time spent in the location is not accumulated), a duration measure of 1 means that the duration of the run increases at the rate of time (i.e., the time spent in the location is accumulated), and a duration measure of 2 means that the duration of the run increases at twice the rate of time. The time-bounded reachability problem (†) can now be generalized to the *duration-bounded reachability* problem:

> Given an initial state σ, a final state τ, a duration measure, and an interval I, is there a run of the automaton starting in state σ and ending in state τ such that the duration of the run is in the interval I?

We show that the duration-bounded reachability problem is PSPACE-complete, and we provide an optimal solution. Our algorithm can be used to verify duration properties of real-time systems that are modeled as timed automata, such as the duration-bounded causality property $(*)$.

Let us briefly outline our construction. Given a region R, a final state τ, and a path in the region graph from R to τ, we show that the lower and upper bounds on the durations of all runs that start at some state in R and follow the chosen path can be written as linear expressions over the variables that represent the clock values of the start state. In a first step, we provide a recipe for computing these so-called *bound expressions*. In the next step, we define an infinite graph, the *bounds graph*, whose vertices are regions tagged with bound expressions that specify the set of possible durations for any path to the final state. In the final step, we show that the infinite bounds graph can be collapsed into a finite graph for solving the duration-bounded reachability problem.

2 The Duration-bounded Reachability Problem

Timed automata

Timed automata are a formal model for real-time systems [Dil89, AD94]. Each automaton has a finite set of control locations and a finite set of real-valued clocks. All clocks proceed at the same rate, and thus each clock measures the amount of time that has elapsed since it was started. A transition of a timed automaton can be taken only if the current clock values satisfy the constraint that is associated with the transition. When taken, the transition changes the control location of the automaton and restarts one of the clocks.

Formally, a *timed automaton* A is a triple (S, X, E) with the following components:

- S is a finite set of *locations*;
- X is a finite set of *clocks*;
- E is a finite set of *transitions* of the form (s, t, φ, x), for a source location $s \in S$, a target location $t \in S$, a clock constraint φ, and a clock $x \in X$. Each *clock constraint* is a positive boolean combination of atomic formulas of the form $y \leq k$ or $y < k$ or $k \leq y$ or $k < y$, for a clock $y \in X$ and a nonnegative integer constant $k \in \mathbb{N}$.

A configuration of the timed automaton A is fully described by specifying the location of the control and the values of all clocks. A *clock valuation* $c \in \mathbb{R}^X$ is an assignment of nonnegative reals to the clocks in X. A *state* σ of A is a pair (s, c) consisting of a location $s \in S$ and a clock valuation c. We write Σ for the (infinite) set of states of A. As time elapses, the values of all clocks increase uniformly with time, thereby changing the state of A. Thus, if the state of A is (s, c), then after time $\delta \in \mathbb{R}$, assuming that no transition occurs, the state of A is $(s, c + \delta)$, where $c + \delta$ is the clock valuation that assigns $c(x) + \delta$ to each clock x. The state of A may also change because of a transition (s, t, φ, x) in E. Such a transition can be taken only in a state whose location is s and whose clock valuation satisfies the constraint φ. The transition is instantaneous. After the transition, the automaton is in a state with location t and the new clock valuation is $c[x := 0]$; that is, the clock x associated with the transition is reset to the value 0, and all other clocks remain unchanged.

The possible behaviors of the timed automaton A are defined through a successor relation on the states of A:

Transition successor For all states $(s, c) \in \Sigma$ and transitions $(s, t, \varphi, x) \in E$, if c satisfies φ, then $(s, c) \xrightarrow{0} (t, c[x := 0])$.

Time successor For all states $(s, c) \in \Sigma$ and time increments $\delta \in \mathbb{R}$, we have $(s, c) \xrightarrow{\delta} (s, c + \delta)$.

A state (t, d) is a *successor* of the state (s, c), written $(s, c) \rightarrow (t, d)$, iff there exists a nonnegative real δ such that $(s, c) \xrightarrow{\delta} (t, d)$. The successor relation defines an infinite graph $\mathcal{K}(A)$ on the state space Σ of A. The transitive closure $\rightarrow^*$ of the successor relation $\rightarrow$ is called the *reachability relation* of A.

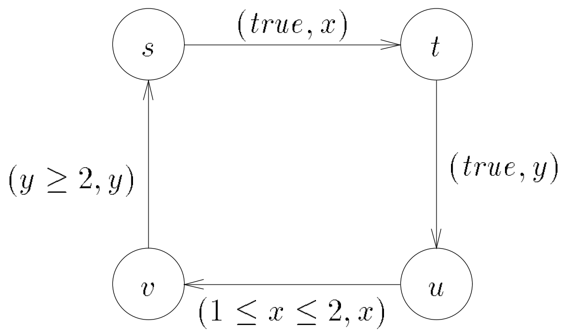

Figure 1: Sample timed automaton

Example 1 A sample timed automaton is shown in Figure 1. The automaton has four locations and two clocks. Each edge is labeled with a clock constraint and the clock to be reset. A state of the automaton contains a location and real-numbered values for the clocks x and y. Some sample pairs in the successor relation are shown below:

$$\begin{array}{llllllllll} (s,0,0) & \overset{1.2}{\to} & (s,1.2,1.2) & \overset{0}{\to} & (t,0,1.2) & \overset{1.3}{\to} & (t,1.3,2.5) & \overset{0}{\to} & (u,1.3,0) & \overset{0.5}{\to} \\ (u,1.8,0.5) & \overset{0}{\to} & (v,0,0.5) & \overset{1}{\to} & (v,1,1.5) & \overset{1}{\to} & (v,2,2.5) & \overset{0}{\to} & (s,2,0). & \end{array}$$

□

Depending on the application, a timed automaton may be augmented with additional components such as initial locations, accepting locations, transition labels for synchronization with other timed automata, and atomic propositions as location labels. It is also useful to label each location with a clock constraint that limits the amount of time spent in that location [HNSY94]. We have chosen a very simple definition of timed automata to illustrate the essential computational aspects of solving reachability problems. Also, the standard definition of a timed automaton allows a (possibly empty) set of clocks to be reset with each transition. Our requirement that precisely one clock is reset with each transition does not affect the expressiveness of timed automata.

Clock regions and the region graph

Let us review the standard method for analyzing timed automata. The key to solving many verification problems for a timed automaton is the construction of the so-called region graph [AD94]. The region graph of a timed automaton is a finite quotient of the infinite state graph that retains enough information for answering certain reachability questions.

Suppose that we are given a timed automaton A and an equivalence relation $\cong$ on the states Σ of A. For $\sigma \in \Sigma$, we write $[\sigma]_{\cong} \subseteq \Sigma$ for the equivalence class of states that contains the state σ. The successor relation $\to$ is extended to $\cong$-equivalence classes as follows: define $[\sigma]_{\cong} \to [\tau]_{\cong}$ iff there is a state $\sigma' \in [\sigma]_{\cong}$, a state $\tau' \in [\tau]_{\cong}$, and a nonnegative real δ such that $\sigma' \overset{\delta}{\to} \tau'$ and for all nonnegative reals $\varepsilon < \delta$, we have $(\sigma + \varepsilon) \in ([\sigma]_{\cong} \cup [\tau]_{\cong})$. The *quotient graph* of A with respect to the equivalence relation $\cong$, written $[\mathcal{K}(A)]_{\cong}$, is a graph whose vertices are the $\cong$-equivalence classes and whose edges are given by the successor relation $\to$. The equivalence relation $\cong$ is *stable* iff whenever $\sigma \to \tau$, then for all states $\sigma' \in [\sigma]_{\cong}$, there is a state $\tau' \in [\tau]_{\cong}$ such that $\sigma' \to \tau'$; and $\cong$ is *back stable* iff whenever $\sigma \to \tau$, then for all states $\tau' \in [\tau]_{\cong}$, there is a state $\sigma' \in [\sigma]_{\cong}$ such that

$\sigma' \rightarrow \tau'$. The quotient graph with respect to a (back) stable equivalence relation can be used for solving the *reachability problem* between equivalence classes: given two $\cong$-equivalence classes R_0 and R_f, is there a state $\sigma \in R_0$ and a state $\tau \in R_f$ such that $\sigma \rightarrow^* \tau$? If the equivalence relation $\cong$ is (back) stable, then the answer to the reachability problem is affirmative iff there is a path from R_0 to R_f in the quotient graph $[\mathcal{K}(A)]_{\cong}$.

The region graph of the timed automaton A is a quotient graph of A with respect to the particular equivalence relation defined below. For $x \in X$, let m_x be the largest constant that the clock x is compared to in any clock constraint of A. For $\delta \in \mathbb{R}$, let $\lfloor \delta \rfloor$ denote the integral part of δ, and let $\bar{\delta}$ denote the fractional part of δ (thus, $\delta = \lfloor \delta \rfloor + \bar{\delta}$). We freely use constraints like $\bar{x} \leq k$ and $\lfloor x \rfloor = k$, for a clock x and a nonnegative integer constant k (e.g., a clock valuation c satisfies the constraint $\lfloor x \rfloor \leq k$ iff $\lfloor c(x) \rfloor \leq k$). Two states (s, c) and (t, d) of A are *region-equivalent*, written $(s, c) \approx (t, d)$, iff the following four conditions hold:

1. $s = t$;
2. for each clock $x \in X$, either $\lfloor c(x) \rfloor = \lfloor d(x) \rfloor$, or both $c(x) > m_x$ and $d(x) > m_x$;
3. for all clocks $x, y \in X$, the valuation c satisfies $\bar{x} \leq \bar{y}$ iff the valuation d satisfies $\bar{x} \leq \bar{y}$;
4. for each clock $x \in X$, the valuation c satisfies $\bar{x} = 0$ iff the valuation d satisfies $\bar{x} = 0$.

A (*clock*) *region* $R \subseteq \Sigma$ is a $\approx$-equivalence class of states. Hence, a region is fully specified by a location, the integral parts of all clock values, and the ordering of the fractional parts of the clock values. For instance, if X contains three clocks, x, y, and z, then the region $R = [s, x = 1, y = 0.2, z = 1.3]_{\approx}$ contains all states (s, c) such that c satisfies $x = 1$, $\lfloor y \rfloor = 0$, $\lfloor z \rfloor = 1$, and $0 < \bar{y} < \bar{z} < 1$. For the region R, we write $[s, \lfloor x \rfloor = 1, \lfloor y \rfloor = 0, \lfloor z \rfloor = 1, 0 = \bar{x} < \bar{y} < \bar{z}]$, and we say that R has the location s and satisfies the constraints $\lfloor x \rfloor = 1$, etc. There are only finitely many regions, because the exact value of the integral part of a clock x is recorded only if it is smaller than m_x. The number of regions is bounded by $|S| \cdot 2^n \cdot n! \cdot \Pi_{x \in X}(m_x + 1)$, where $n = |X|$ is the number of clocks. The *region graph* $\mathcal{R}(A)$ of the timed automaton A is the (finite) quotient graph of A with respect to the region equivalence relation $\approx$. The region equivalence relation $\approx$ is stable as well as back-stable [AD94]. Hence the region graph can be used for solving reachability problems between regions, and also for solving time-bounded reachability problems [ACD93].

It is useful to define the edges of the region graph explicitly. A region R is a *boundary region* iff there is some clock x such that R satisfies the constraint $\bar{x} = 0$. A region that is not a boundary region is called an *open region*. For a boundary region R, we define its predecessor region $pred(R)$ to be the open region Q such that for all states $(s, c) \in Q$, there is a time increment $\delta \in \mathbb{R}$ such that $(s, c + \delta) \in R$ and for all nonnegative reals $\varepsilon < \delta$, we have $(s, c + \varepsilon) \in Q$. Similarly, we define the successor region $succ(R)$ of R to be the open region Q' such that for all states $(s, c) \in Q'$, there is a time increment $\delta \in \mathbb{R}$ such that $(s, c - \delta) \in R$ and for all nonnegative reals $\varepsilon < \delta$, we have $(s, c - \varepsilon) \in Q'$. The state of a timed automaton belongs to a boundary region R only instantaneously. Just before that instant the state belongs to $pred(R)$, and just after that instant the state belongs to $succ(R)$. For example, for the boundary region R given by

$$[s, \lfloor x \rfloor = 1, \lfloor y \rfloor = 0, \lfloor z \rfloor = 1, 0 = \bar{x} < \bar{y} < \bar{z}],$$

$pred(R)$ is the open region

$$[s, \lfloor x \rfloor = 0, \lfloor y \rfloor = 0, \lfloor z \rfloor = 1, 0 < \bar{y} < \bar{z} < \bar{x}],$$

and $succ(R)$ is the open region

$$[s, \lfloor x \rfloor = 1, \lfloor y \rfloor = 0, \lfloor z \rfloor = 1, 0 < \bar{x} < \bar{y} < \bar{z}].$$

The edges of the region graph $\mathcal{R}(A)$ fall into two categories:

Transition edges If $(s, c) \xrightarrow{0} (t, d)$, then there is an edge from the region $[s, c]_\approx$ to the region $[t, d]_\approx$.

Time edges For each boundary region R, there is an edge from $pred(R)$ to R, and an edge from R to $succ(R)$.

In addition, each region has a self-loop, which can be ignored for solving reachability problems.

Duration measures and duration-bounded reachability

A *duration measure* for the timed automaton A is a function p from the locations of A to the nonnegative integers. A *duration constraint* for A is of the form $\int p \in I$, where p is a duration measure for A and I is a bounded interval of the nonnegative real line whose endpoints are integers (I may be open, half-open, or closed).

Let p be a duration measure for A. We extend the state space of A to evaluate the integral $\int p$ along the runs of A. An *extended state* of A is a pair (σ, ε) consisting of a state σ of A and a nonnegative real number ε. The successor relation on states is extended as follows:

Transition successor For all extended states (s, c, ε) and all transitions (s, t, φ, x) such that c satisfies φ, define $(s, c, \varepsilon) \xrightarrow{0} (t, c[x := 0], \varepsilon)$.

Time successor For all extended states (s, c, ε) and all time increments $\delta \in \mathbb{R}$, define $(s, c, \varepsilon) \xrightarrow{\delta} (s, c + \delta, \varepsilon + \delta \cdot p(s))$.

We consider the *duration-bounded reachability problem* between regions: given two regions R_0 and R_f of a timed automaton A, and a duration constraint $\int p \in I$ for A, is there a state $\sigma \in R_0$, a state $\tau \in R_f$, and a nonnegative real $\delta \in I$ such that $(\sigma, 0) \rightarrow^* (\tau, \delta)$? We refer to this duration-bounded reachability problem using the tuple $(A, R_0, R_f, \int p \in I)$.

EXAMPLE **2** Recall the sample timed automaton from Figure 1. Suppose that the duration measure p is defined by $p(s) = p(u) = 0$ and $p(t) = p(v) = 1$. Let the initial region R_0 be the singleton $\{(s, x = 0, y = 0)\}$, and let the final region R_f be $\{(s, x = 1, y = 0)\}$. For the duration constraint $\int p = 2$, the answer to the duration-bounded reachability problem is in the affirmative, and the following sequence of successor pairs is a justification (the last component denotes the value of the integral $\int p$):

$$\begin{array}{llllllllll}
(s,0,0,0) & \xrightarrow{1} & (s,1,1,0) & \xrightarrow{0} & (t,0,1,0) & \xrightarrow{1} & (t,1,2,1) & \xrightarrow{0} & (u,1,0,1) & \xrightarrow{1} \\
(u,2,1,1) & \xrightarrow{0} & (v,0,1,1) & \xrightarrow{1} & (v,1,2,2) & \xrightarrow{0} & (s,1,0,2). & & &
\end{array}$$

On the other hand, for the duration constraint $\int p > 2$, the answer to the duration-bounded reachability problem is negative. The reader can verify that if $(s, 0, 0, 0) \rightarrow^* (s, 1, 0, \delta)$, then $1 \leq \delta \leq 2$. □

If the duration measure p is the constant function 1 (i.e., $p(s) = 1$ for all locations s), then the integral $\int p$ measures the total elapsed time, and the duration-bounded reachability problem between regions is called a *time-bounded reachability* problem. In this case, if $(\sigma, 0) \rightarrow^* (\tau, \delta)$ for some $\delta \in I$, then for all states $\sigma' \in [\sigma]_\approx$ there is a state $\tau' \in [\tau]_\approx$ and a real number $\delta' \in I$ such that $(\sigma', 0) \rightarrow^* (\tau', \delta')$. Hence, the region graph suffices to solve the time-bounded reachability problem. This, however, is not true for general duration measures.

3 A Solution to the Duration-bounded Reachability Problem

Bound-labeled regions and the bounds graph

Consider a timed automaton A, two regions R_0 and R_f, and a duration measure p. We determine the set I of possible values of δ such that $(\sigma, 0) \rightarrow^* (\tau, \delta)$ for some $\sigma \in R_0$ and $\tau \in R_f$. To compute the lower and upper bounds on the integral $\int p$ along a path of the region graph, we refine the graph by labeling all regions with expressions that specify the extremal values of the integral.

We define an infinite graph with vertices of the form (R, L, l, U, u), where R is a region, L and U are linear expressions over the clock variables, and l and u are boolean values. The intended meaning of the bound expressions L and U is that in moving from a state $(s, c) \in R$ to a state in the final region R_f, the set of possible values of the integral $\int p$ has the infimum L and the supremum U, both of which are functions of the current clock values c. If the bit l is 0, then the infimum L is included in the set of possible values of the integral, and if l is 1, then L is excluded. Similarly, if the bit u is 0, then the supremum U is included in the set of possible values of $\int p$, and if u is 1, then U is excluded. For example, if $l = 0$ and $u = 1$, then the left-closed right-open interval $[L, U)$ gives the possible values of the integral $\int p$.

The bound expressions L and U associated with the region R have a special form. Suppose that $X = \{x_1, \ldots, x_n\}$ is the set of clocks and that for all states $(s, c) \in R$, the clock valuation c satisfies $0 \leq \bar{x}_1 \leq \cdots \leq \bar{x}_n < 1$; that is, x_1 is the clock with the smallest fractional part in R, and x_n is the clock with the largest fractional part. The fractional parts of all n clocks partition the unit interval into $n+1$ subintervals represented by the expressions $e_0, \ldots, e_n$:

$$
\begin{aligned}
e_0 &= \bar{x}_1, \\
e_1 &= \bar{x}_2 - \bar{x}_1, \\
&\vdots \\
e_{n-1} &= \bar{x}_n - \bar{x}_{n-1}, \\
e_n &= 1 - \bar{x}_n.
\end{aligned}
$$

A *bound expression* for R is a positive linear combination of the expressions $e_0, \ldots, e_n$; that is, a bound expression for R has the form $a_0 \cdot e_0 + \cdots + a_n \cdot e_n$, where $a_0, \ldots, a_n$ are nonnegative integer constants. We denote bound expressions by $(n+1)$-tuples of coefficients and write $(a_0, \ldots, a_n)$ for the bound expression $a_0 \cdot e_0 + \cdots + a_n \cdot e_n$. For a bound expression e and a clock valuation c, we write $[\![e]\!]_c$ to denote the result of evaluating e using the clock values given by c. When time advances, the value of a bound expression changes at the rate $a_0 - a_n$. If the region R satisfies the constraint $\bar{x}_1 = 0$ (i.e., R is a boundary region), then the coefficient a_0 is irrelevant, and if R satisfies $\bar{x}_i = \bar{x}_{i+1}$, then the coefficient a_i is irrelevant. Henceforth, we assume that all irrelevant coefficients are set to 0.

A *bound-labeled region* (R, L, l, U, u) of the timed automaton A consists of a clock region R of A, two bound expressions L and U for R, and two bits $l, u \in \{0, 1\}$. We construct $\mathcal{B}_{p,R_f}(A)$, the *bounds graph* of A for the duration measure p and the final region R_f. The vertices of $\mathcal{B}_{p,R_f}(A)$ are the bound-labeled regions of A and the special vertex R_f, which has no outgoing edges. We first define the edges with the target R_f, and then the edges between bound-labeled regions.

The edges with the target R_f correspond to runs of the automaton that reach a state in R_f without passing through other regions. Suppose that R_f is an open region with the duration measure a (i.e., $p(s) = a$ for the location s of R_f). The final region R_f is reachable from a state $(s, c) \in R_f$ by remaining in R_f for at least 0 and at most $[\![1 - \bar{x}_n]\!]_c$ time units. Since the integral $\int p$ increases at the rate a, the lower bound on the integral value over all states $(s, c) \in R_f$ is 0,

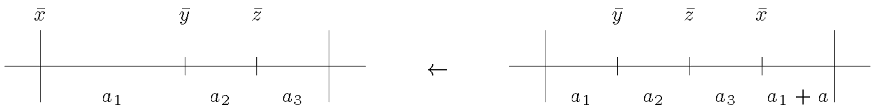

Figure 2: $(0 = \bar{x} < \bar{y} < \bar{z}) \;\leftarrow\; (0 < \bar{y} < \bar{z} < \bar{x})$

and the upper bound is $a \cdot (1 - \bar{x}_n)$. While the lower bound 0 is a possible value of the integral, if $a > 0$, then the upper bound is only a supremum of all possible values. Hence, we add an edge in the bounds graph to R_f from $(R_f, L, 0, U, u)$ for

$L = (0, \ldots, 0, 0)$ and $U = (0, \ldots, 0, a)$;
if $a = 0$ then $u = 0$, else $u = 1$.

If R_f is a boundary region, no time can be spent in R_f, and both bounds are 0. In this case, we add an edge to R_f from $(R_f, L, 0, U, 0)$ for $L = U = (0, \ldots, 0, 0)$.

Now let us look at paths that reach the final region R_f by passing through other regions. For each edge from R to R' in the region graph $\mathcal{R}(A)$, the bounds graph $\mathcal{B}_{p,R_f}(A)$ has exactly one edge to each bound-labeled region of the form (R', L', l', U', u'), from some bound-labeled region of the form (R, L, l, U, u). First, let us consider an example to understand the determination of the lower bound L and the corresponding bit l (the upper bound U and the bit u are determined similarly).

Suppose that $X = \{x, y, z\}$ and that the boundary region R_1, which satisfies $0 = \bar{x} < \bar{y} < \bar{z}$, is labeled with the lower bound $L_1 = (0, a_1, a_2, a_3)$ and the bit l_1. This means that starting from a state $(s, c) \in R_1$, the lower bound on the integral $\int p$ for reaching some state in R_f is

$$[\![a_1 \cdot \bar{y} + a_2 \cdot (\bar{z} - \bar{y}) + a_3 \cdot (1 - \bar{z})]\!]_c.$$

Consider the open predecessor region R_2 of R_1, which satisfies $0 < \bar{y} < \bar{z} < \bar{x}$. Let a be the duration measure of R_2. There is a time edge from R_2 to R_1 in the region graph. We want to compute the lower-bound label L_2 for R_2 from the lower-bound label L_1 of R_1. Starting in a state $(s, c) \in R_2$, the state $(s, c + \delta) \in R_1$ is reached after time $\delta = [\![1 - \bar{x}]\!]_c$, and

$$\begin{aligned}
[\![\bar{y}]\!]_{c+\delta} &= [\![\bar{y}]\!]_c + \delta = [\![\bar{y} + (1 - \bar{x})]\!]_c, \\
[\![\bar{z} - \bar{y}]\!]_{c+\delta} &= [\![\bar{z} - \bar{y}]\!]_c, \\
[\![1 - \bar{z}]\!]_{c+\delta} &= [\![1 - \bar{z}]\!]_c - \delta = [\![\bar{x} - \bar{z}]\!]_c.
\end{aligned}$$

Furthermore, from the state $(s, c) \in R_2$ the integral $\int p$ has the value $[\![a \cdot (1 - \bar{x})]\!]_c$ before entering the region R_1. Hence, the new lower bound is

$$[\![a_1 \cdot (\bar{y} + (1 - \bar{x})) + a_2 \cdot (\bar{z} - \bar{y}) + a_3 \cdot (\bar{x} - \bar{z}) + a \cdot (1 - \bar{x})]\!]_c$$

and the label L_2 is $(a_1, a_2, a_3, a_1 + a)$. See Figure 2. Whether the lower bound L_2 is a possible value of the integral depends on whether the original lower bound L_1 is a possible value of the integral starting in R_1. Thus, the bit l_2 labeling R_2 is the same as the bit l_1 labeling R_1.

Next, consider the boundary region R_3 such that R_2 is the successor region of R_3. The region R_3 satisfies $0 = \bar{y} < \bar{z} < \bar{x}$, and there is a time edge from R_3 to R_2 in the region graph. The reader can verify that the updated lower-bound label L_3 of R_3 is the same as L_2, namely $(a_1, a_2, a_3, a_1 + a)$, which can be simplified to $(0, a_2, a_3, a_1 + a)$, because R_3 is a boundary region. See Figure 3. The updated bit l_3 of R_3 is the same as l_2.

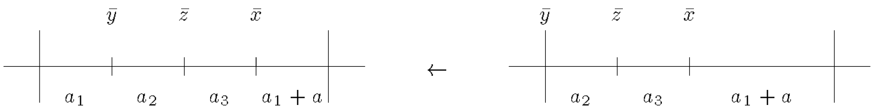

Figure 3: $(0 < \bar{y} < \bar{z} < \bar{x}) \ \leftarrow \ (0 = \bar{y} < \bar{z} < \bar{x})$

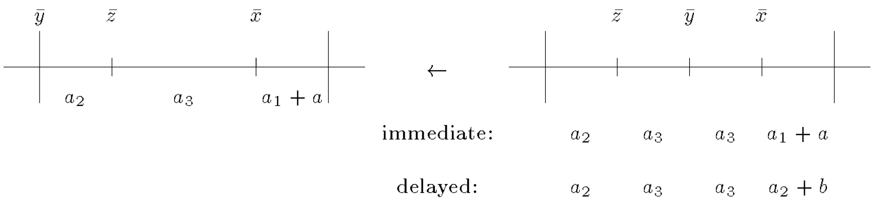

Figure 4: $(0 = \bar{y} < \bar{z} < \bar{x}) \ \leftarrow \ (0 = \bar{z} < \bar{y} < \bar{x})$

The process repeats if we consider further time edges, so let us consider a transition edge from region R_4 to region R_3, which resets the clock y. We assume that the region R_4 is open with duration measure b, and that R_4 satisfies $0 < \bar{z} < \bar{y} < \bar{x}$. Consider a state $(t,d) \in R_4$. Suppose that the transition happens after time δ; then $0 \le \delta < [\![1-\bar{x}]\!]_d$. If the state after the transition is $(s,c) \in R_3$, then $[\![\bar{x}]\!]_c = [\![\bar{x}]\!]_d + \delta$, $[\![\bar{y}]\!]_c = 0$, and $[\![\bar{z}]\!]_c = [\![\bar{z}]\!]_d + \delta$. The lower bound L_4 corresponding to this scenario is the value of the integral before the transition, which is $b \cdot \delta$, added to the value of the lower bound L_3 at the state (s,c), which is

$$[\![a_2 \cdot \bar{z} + a_3 \cdot (\bar{x} - \bar{z}) + (a_1 + a) \cdot (1 - \bar{x})]\!]_c.$$

To obtain the value of the lower bound L_4 at the state (t,d), we need to compute the infimum over all choices of δ, for $0 \le \delta < [\![1-\bar{x}]\!]_d$. Hence, the desired lower bound is

$$inf_{0 \le \delta < [\![1-\bar{x}]\!]_d} \ \{b \cdot \delta + [\![a_2 \cdot \bar{z} + a_3 \cdot (\bar{x} - \bar{z}) + (a_1 + a) \cdot (1 - \bar{x})]\!]_c\}.$$

After substituting $[\![\bar{x}]\!]_c = [\![\bar{x}]\!]_d + \delta$ and $[\![\bar{z}]\!]_c = [\![\bar{z}]\!]_d + \delta$, this simplifies to

$$[\![a_2 \cdot \bar{z} + a_3 \cdot (\bar{x} - \bar{z})]\!]_d + inf_{0 \le \delta < [\![1-\bar{x}]\!]_d} \ \{(a_2 + b) \cdot \delta + [\![(a_1 + a) \cdot (1 - \bar{x} - \delta)]\!]_d\}.$$

The infimum of the monotonic function in δ is reached at one of the two extreme points. When $\delta = 0$ (i.e., the transition occurs immediately), then the value of L_4 at (t,d) is

$$[\![a_2 \cdot \bar{z} + a_3 \cdot (\bar{x} - \bar{z}) + (a_1 + a) \cdot (1 - \bar{x})]\!]_d.$$

When δ approaches $[\![1-\bar{x}]\!]_d$ (i.e., the transition occurs as late as possible), then the value of L_4 at (t,d) is

$$[\![a_2 \cdot \bar{z} + a_3 \cdot (\bar{x} - \bar{z}) + (a_2 + b) \cdot (1 - \bar{x})]\!]_d.$$

Since R_4 satisfies $0 < \bar{z} < \bar{y} < \bar{x}$ and $(\bar{x} - \bar{z}) = (\bar{y} - \bar{z}) + (\bar{x} - \bar{y})$, the lower-bound label L_4 for R_4 is (a_2, a_3, a_3, a_4), where a_4 is the minimum of $a_1 + a$ and $a_2 + b$. See Figure 4. Finally, we need to

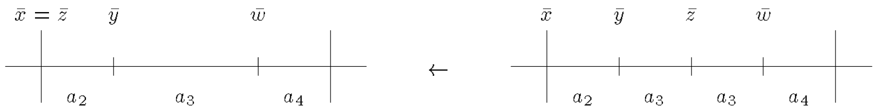

Figure 5: $(0 = \bar{x} = \bar{z} < \bar{y} < \bar{u}) \;\leftarrow\; (0 = \bar{x} < \bar{y} < \bar{z} < \bar{u})$

deduce the bit l_4, which indicates whether the lower bound L_4 is a possible value of the integral. If $a_1 + a \leq a_2 + b$, then the lower bound is obtained with $\delta = 0$, and L_4 is possible for R_4 iff L_3 is possible for R_3; so l_4 is the same as l_3. Otherwise, if $a_1 + a > a_2 + b$, then the lower bound is obtained with δ approaching $[\![1 - \bar{x}]\!]_d$, and L_4 is possible iff both $b = 0$ and l_3 is possible for R_3; so $l_4 = 0$ if $b = 0$ and $l_3 = 0$, and otherwise $l_4 = 1$.

We now formally define the edges between bound-labeled regions of the bounds graph $\mathcal{B}_{p,R_f}(A)$. Suppose that the region graph $\mathcal{R}(A)$ has an edge from R to R', and let a be the duration measure of R. Then the bounds graph has an edge from (R, L, l, U, u) to (R', L', l', U', u') iff the bound expressions $L = (a_0, a_1, \ldots, a_n)$, $L' = (a'_0, a'_1, \ldots, a'_n)$, $U = (b_0, b_1, \ldots, b_n)$, and $U' = (b'_0, b'_1, \ldots, b'_n)$, and the bits l, u, l', and u' are related as follows. There are various cases to consider, depending on whether the edge from R to R' is a time edge or a transition edge:

Time edge 1 R' is a boundary region and $R = \mathit{pred}(R')$ is an open region: let $1 \leq k \leq n$ be the largest index such that R' satisfies $\bar{x}_k = 0$, then

for all $0 \leq i \leq n-k$, we have $a_i = a'_{i+k}$ and $b_i = b'_{i+k}$;
for all $n-k < i < n$, we have $a_i = 0$ and $b_i = 0$;
$a_n = a'_k + a$ and $b_n = b'_k + a$;
$l = l'$ and $u = u'$.

Time edge 2 R is a boundary region and $R' = \mathit{succ}(R)$ is an open region:

$a_0 = 0$ and $b_0 = 0$;
for all $0 < i \leq n$, we have $a_i = a'_i$ and $b_i = b'_i$;
$l = l'$ and $u = u'$.

Transition edge 1 R' is a boundary region, R is an open region, and the clock with the k-th smallest fractional part in R is reset:

for all $0 \leq i < k$, we have $a_i = a'_{i+1}$ and $b_i = b'_{i+1}$;
for all $k \leq i < n$, we have $a_i = a'_i$ and $b_i = b'_i$;
if $a'_n \leq a'_1 + a$ then $a_n = a'_n$, else $a_n = a'_1 + a$;
if $b'_n \geq b'_1 + a$ then $b_n = b'_n$, else $b_n = b'_1 + a$;
if $a'_n > a'_1 + a$ and $a > 0$ then $l = 1$, else $l = l'$;
if $b'_n < b'_1 + a$ and $a > 0$ then $u = 1$, else $u = u'$.

Transition edge 2 Both R and R' are boundary regions, and the clock with the k-th smallest fractional part in R is reset:

for all $0 \leq i < k$, we have $a_i = a'_{i+1}$ and $b_i = b'_{i+1}$;
for all $k \leq i \leq n$, we have $a_i = a'_i$ and $b_i = b'_i$;
$l = l'$ and $u = u'$.

This case is illustrated in Figure 5.

This completes the definition of the bounds graph $\mathcal{B}_{p,R_f}(A)$.

Reachability in the bounds graph

Given a state $\sigma = (s, c)$, two bound expressions L and U, and two bits l and u, we define the (bounded) interval $I(\sigma, L, l, U, u)$ of the nonnegative real line as follows: the left endpoint is $[\![L]\!]_c$; the right endpoint is $[\![U]\!]_c$; if $l = 0$ then the interval is left-closed, else it is left-open; if $u = 0$ then the interval is right-closed, else it is right-open. The following lemma states the fundamental property of the bounds graph $\mathcal{B}_{p,R_f}(A)$.

LEMMA 1 *Let A be a timed automaton, let p be a duration measure for A, and let R_f be a region of A. For every state σ of A and every nonnegative real δ, there is a state $\tau \in R_f$ such that $(\sigma, 0) \rightarrow^* (\tau, \delta)$ iff in the bounds graph $\mathcal{B}_{p,R_f}(A)$, there is path to R_f from a bound-labeled region (R, L, l, U, u) with $\sigma \in R$ and $\delta \in I(\sigma, L, l, U, u)$.*

PROOF. Consider a state σ of A and a nonnegative real δ. Suppose $(\sigma, 0) \rightarrow^* (\tau, \delta)$ for some $\tau \in R_f$. Then, by the definition of the region graph $\mathcal{R}(A)$, we have a sequence

$$(\sigma_n, \delta_n) \rightarrow (\sigma_{n-1}, \delta_{n-1}) \rightarrow \cdots \rightarrow (\sigma_1, \delta_1) \rightarrow (\sigma_0, \delta_0) \rightarrow (\tau, \delta)$$

of successors of extended states with $\sigma_n = \sigma$, $\delta_n = 0$, $[\sigma_0] = R_f$, and for all $0 \leq i < n$, the region graph contains an edge from the region R_{i+1} containing σ_{i+1} to the region R_i containing σ_i. We claim that there exist bound-labeled regions $B_0, \ldots B_n$ such that (1) for all $0 \leq i \leq n$, the region component of B_i is R_i, (2) the bounds graph $\mathcal{B}_{p,R_f}(A)$ has an edge from B_0 to R_f and from B_{i+1} to B_i for all $0 \leq i < n$, and (3) for all $0 \leq i \leq n$, $\delta - \delta_i \in I(\sigma_i, L_i, l_i, U_i, u_i)$ where $B_i = (R_i, L_i, l_i, U_i, u_i)$. This claim is proved by induction on i, using the definition of the edges in the bounds graph.

Conversely, consider a sequence of bound-labeled regions $B_n, \ldots B_0$ such that the bounds graph $\mathcal{B}_{p,R_f}(A)$ has an edge from B_0 to R_f and from B_{i+1} to B_i for all $0 \leq i < n$. Let each B_i be $(R_i, L_i, l_i, U_i, u_i)$. We claim that for all $0 \leq i \leq n$, for all $\sigma \in R_i$, and for all $\delta \in I(\sigma, L_i, l_i, U_i, u_i)$, there exists $\tau \in R_f$ with $(\sigma, 0) \rightarrow^* (\tau, \delta)$. This is again proved by induction on i, using the definition of the edges in the bounds graph. $\square$

For a bound-labeled region $B = (R, L, l, U, u)$, let $I(B)$ denote the union $\bigcup_{\sigma \in R} I(\sigma, L, l, U, u)$ of intervals. It is not difficult to check that the set $I(B)$ is a bounded interval of the nonnegative real line with integer endpoints. The left endpoint ℓ of $I(B)$ is the infimum of $[\![L]\!]_c$ over all choices of clock valuations c that are consistent with R; that is, $\ell = inf_c\{[\![L]\!]_c \mid (s, c) \in R\}$. Since all irrelevant coefficients in the bound expression L are 0, the infimum ℓ is equal to the smallest nonzero coefficient in L (the left end-point is 0 if all coefficients are 0). Similarly, the right endpoint of $I(B)$ is the supremum of $[\![U]\!]_c$ over all choices of c that are consistent with R, and this supremum is equal to the largest coefficient in U. The type of the interval $I(B)$ can be determined as follows. Let $L = (a_0, \ldots, a_n)$ and $U = (b_0, \ldots, b_n)$.

- If $a_i = 0$ for all $0 \leq i < n$ and $l = 0$, then $I(B)$ is left-closed, and otherwise $I(B)$ is left-open.
- If $b_i = 0$ for all $0 \leq i < n$ and $u = 0$, then $I(B)$ is right-closed, and otherwise $I(B)$ is right-open.

For instance, consider the region R that satisfies $0 < \bar{x} < \bar{y} < \bar{z}$. Let $L = U = (2, 3, 1, 5)$ and $B = (R, L, l, U, u)$. Then $I(B)$ is the open interval $(1, 5)$, irrespective of the values of l and u. Lemma 1 implies

LEMMA **2** *Let A be a timed automaton, let $\int p \in I$ be a duration constraint for A, and let R_0, R_f be two regions of A. There are two states $\sigma \in R_0$ and $\tau \in R_f$ and a real number $\delta \in I$ such that $(\sigma, 0) \rightarrow^* (\tau, \delta)$ iff in the bounds graph $\mathcal{B}_{p,R_f}(A)$, there is path to R_f from a bound-labeled region B with region component R_0 and $I(B) \cap I \neq \emptyset$.*

Hence, to solve the duration-bounded reachability problem $(A, R_0, R_f, \int p \in I)$, we construct the portion of the bounds graph $\mathcal{B}_{p,R_f}(A)$ from which the special vertex R_f is reachable. This can be done in a backward breadth-first fashion starting from the final region R_f. On a particular path through the bounds graph, the same region may appear with different bound expressions. Although there are infinitely many distinct bound expressions, the backward search can be terminated within finitely many steps, because when the coefficients of the bound expressions become sufficiently large relative to I, then their actual values become irrelevant. This is shown in the following section.

Collapsing the bounds graph

Given a nonnegative integer constant m, we define an equivalence relation $\cong_m$ over bound-labeled regions as follows. For two nonnegative integers a and b, define $a \cong_m b$ iff either $a = b$, or both $a > m$ and $b > m$. For two bound expressions $e = (a_0, \ldots, a_n)$ and $f = (b_0, \ldots, b_n)$, define $e \cong_m f$ iff for all $0 \leq i \leq n$, we have $a_i \cong_m b_i$. For two bound-labeled regions $B_1 = (R_1, L_1, l_1, U_1, u_1)$ and $B_2 = (R_2, L_2, l_2, U_2, u_2)$, define $B_1 \cong_m B_2$ iff the following four conditions hold:

1. $R_1 = R_2$;
2. $L_1 \cong_m L_2$ and $U_1 \cong_m U_2$;
3. either $l_1 = l_2$ or some coefficient in L_1 is greater than m;
4. either $u_1 = u_2$ or some coefficient in U_1 is greater than m.

The following lemma states that the equivalence relation $\cong_m$ on bound-labeled regions is back stable.

LEMMA **3** *If the bounds graph $\mathcal{B}_{p,R_f}(A)$ contains an edge from a bound-labeled region B_1 to a bound-labeled region B_1', and $B_1' \cong_m B_2'$, then there exists a bound-labeled region B_2 such that $B_1 \cong_m B_2$ and the bounds graph contains an edge from B_2 to B_2'.*

PROOF. Consider two bound-labeled regions B_1' and B_2' such that $B_1' \cong_m B_2'$. Let R' be the clock region of B_1' and of B_2', and let $B_1' = (R', L_1', l_1', U_1', u_1')$ and $B_2' = (R', L_2', l_2', U_2', u_2')$. Let R be a clock region such that the region graph $\mathcal{R}(A)$ has an edge from R to R'. Then there is a unique bound-labeled region $B_1 = (R, L_1, l_1, U_1, u_1)$ such that the bounds graph $\mathcal{B}_{p,R_f}(A)$ has an edge from B_1 to B_1', and there is a unique bound-labeled region $B_2 = (R, L_2, l_2, U_2, u_2)$ such that the bounds graph has an edge from B_2 to B_2'. It remains to be shown that $B_1 \cong_m B_2$.

There are 4 cases to consider according to the rules for edges of the bounds graph. We consider only the case corresponding to **Transition edge 2**. This corresponds to the case when R' is a boundary region, R is an open region, and the clock with the k-th smallest fractional part in R is reset. Let the duration measure be a in R. We will establish that $L_1 \cong_m L_2$ and either $l_1 = l_2$ or some coefficient in L_1 is greater than m; the case of upper bounds is similar.

Let $L_1 = (a_0, \ldots a_n)$, $L'_1 = (a'_0, \ldots a'_n)$, $L_2 = (b_0, \ldots b_n)$ and $L'_2 = (b'_0, \ldots b'_n)$. According to the rule, for all $0 \leq i < k$, $a_i = a'_{i+1}$ and $b_i = b'_{i+1}$, and for all $k \leq i < n$, $a_i = a'_i$ and $b_i = b'_i$. Since $B'_1 \cong_m B'_2$, we have $a'_i \cong_m b'_i$ for all $0 \leq i \leq n$. It follows that for $0 \leq i < n$, $a_i \cong_m b_i$. We have $a_n = \min\,(a'_n, a'_1 + a)$ and $b_n = \min\,(b'_n, b'_1 + a)$. We have 4 cases to consider. (i) $a_n = a'_n$ and $b_n = b'_n$. Since $a'_n \cong_m b'_n$, we have $a_n \cong_m b_n$. In this case, $l_1 = l'_1$ and $l_2 = l'_2$. If $l'_1 = l'_2$, we have $l_1 = l_2$. Otherwise $a'_j > m$ for some $1 \leq j \leq n$ (recall $a'_0 = 0$ since R' is a boundary region). Each coefficient a'_j, for $1 \leq j \leq n$, equals either a_{j-1} or a_j, and thus some coefficient of L_1 also exceeds m. (ii) $a_n = a'_n$ and $b_n = b'_1 + a$. In this case, we have $a'_n \cong_m b'_n$, $a'_1 \cong_m b'_1$, $a'_n \leq a'_1 + a$, and $b'_n > b'_1 + a$. It follows that all the values a'_n, $a'_1 + a$, b'_n, and $b'_1 + a$ exceed m. Hence, $a_n > m$ and $b_n > m$. Since at least one coefficient of L_1 is at least m, there is no requirement that $l_1 = l_3$ (indeed, they can be different). The cases (iii) $a_n = a'_1 + a$ and $b_n = b'_n$, and (iv) $a_n = a'_1 + a$ and $b_n = b'_1 + a$ have similar analysis. □

Since the equivalence relation $\cong_m$ is back stable, for checking reachability between bound-labeled regions in the bounds graph $\mathcal{B}_{p,R_f}(A)$, it suffices to look at the quotient graph $[\mathcal{B}_{p,R_f}(A)]_{\cong_m}$. The following lemma indicates a suitable choice for the constant m for solving a duration-bounded reachability problem.

Lemma 4 *Consider two bound-labeled regions B_1 and B_2 and a bounded interval $I \subseteq \mathbb{R}$ with integer endpoints. If $B_1 \cong_m B_2$ for the right endpoint m of I, then $I \cap I(B_1) = \emptyset$ iff $I \cap I(B_2) = \emptyset$.*

Proof. Consider bound-labeled regions $B_1 = (R, L_1, l_1, U_1, u_1)$ and $B_2 = (R, L_2, l_2, U_2, u_2)$ such that $B_1 \cong_m B_2$. It is easy to check that the left end-points of $I(B_1)$ and $I(B_2)$ are either equal or both exceed m (see the rules for determining the left end-point). We need to show that when the left end-points are at most m, either both $I(B_1)$ and $I(B_2)$ are left-open or both are left-closed. If $l_1 = l_2$ this is trivially true. Suppose $l_1 \neq l_2$. Then we know that some coefficient of L_1 and of L_2 exceeds m. Since the left end-point is m or smaller, we know that both L_1 and L_2 have at least two nonzero coefficients. In this case, both the intervals are left-open irrespective of the bits l_1 and l_2. A similar analysis of right end-points shows that either both the right end-points exceed m, or both are at most m, are equal, and both the intervals are either right-open or right-closed. □

A bound expression e is *m-constrained*, for a nonnegative integer m, iff all coefficients in e are at most $m + 1$. Clearly, for every bound expression e, there exists a unique m-constrained bound expression $\gamma(e)$ such that $e \cong_m \gamma(e)$. A bound-labeled region $B = (R, L, l, U, u)$ is *m-constrained* iff (1) both L and U are m-constrained, (2) if some coefficient of L is $m + 1$, then $l = 0$, and (3) if some coefficient of U is $m + 1$, then $u = 0$. Then, for every bound-labeled region B, there exists a unique m-constrained bound-labeled region $\gamma(B)$ such that $B \cong_m \gamma(B)$. Since no two distinct m-constrained bound-labeled regions are $\cong_m$-equivalent, it follows that every $\cong_m$-equivalence class contains precisely one m-constrained bound-labeled region. We use the m-constrained bound-labeled regions to represent the $\cong_m$-equivalence classes.

The number of m-constrained expressions over n clocks is $(m+2)^{n+1}$. Hence, for a given region R, the number of m-constrained bound-labeled regions of the form (R, L, l, U, u) is $4 \cdot (m+2)^{2(n+1)}$. From the bound on the number of clock regions, we obtain a bound on the number of m-constrained bound-labeled regions of A, or equivalently, on the number of $\cong_m$-equivalence classes of bound-labeled regions.

Lemma **5** *Let A be a timed automaton with location set S and clock set X such that n is the number of clocks, and no clock x is compared to a constant larger than m_x. For every nonnegative integer m, the number of m-constrained bound-labeled regions of A is at most*

$$4 \cdot |S| \cdot n! \cdot 2^{n+2} \cdot (m+2)^{2(n+1)} \cdot \Pi_{x \in X}(m_x + 1).$$

Consider the duration-bounded reachability problem $(A, R_0, R_f, \int p \in I)$, and let $m \in \mathbb{N}$ be the right endpoint of the interval I. By Lemma 5, the number of m-constrained bound-labeled regions is exponential in the length of the problem description. By combining Lemmas 2, 3, and 4, we obtain the following exponential-time decision procedure for solving the given duration-bounded reachability problem.

Theorem **1** *Let $m \in \mathbb{N}$ be the right endpoint of the interval $I \subseteq \mathbb{R}$. The answer to the duration-bounded reachability problem $(A, R_0, R_f, \int p \in I)$ is affirmative iff there exists a finite sequence $B_0, \ldots, B_k$ of m-constrained bound-labeled regions of A such that*

1. *the bounds graph $\mathcal{B}_{p,R_f}(A)$ contains an edge to R_f from some bound-labeled region B with $\gamma(B) = B_0$;*
2. *for all $0 \leq i < k$, the bounds graph $\mathcal{B}_{p,R_f}(A)$ contains an edge to B_i from some bound-labeled region B with $\gamma(B) = B_{i+1}$;*
3. *$I(B_k) \cap I \neq \emptyset$ and the clock region of B_k is R_0.*

Hence, when constructing, in a backward breadth-first fashion, the portion of the bounds graph $\mathcal{B}_{p,R_f}(A)$ from which the special vertex R_f is reachable, we need to explore only m-constrained bound-labeled regions. For each m-constrained bound-labeled region B_i, we first construct all predecessors of B_i. The number of predecessors of B_i is finite, and corresponds to the number of predecessors of the clock region of B_i in the region graph $\mathcal{R}(A)$. Each predecessor B of B_i that is not an m-constrained bound-labeled region is replaced by the $\cong_m$-equivalent m-constrained region $\gamma(B)$. The duration-bounded reachability property holds if a bound-labeled region B with $I(B) \cap I \neq \emptyset$ is found. If the search terminates otherwise, by generating no new m-constrained bound-labeled regions, then the answer to the duration-bounded reachability problem is negative. The time complexity of the search is proportional to the number of m-constrained bound-labeled regions, which is given in Lemma 5. The space complexity of the search is Pspace, because the representation of an m-constrained bound-labeled region and its predecessor computation requires only space polynomial in the length of the problem description.

Corollary **1** *The duration-bounded reachability problem for timed automata can be decided in* Pspace.

The duration-bounded reachability problem for timed automata is Pspace-hard, because already the (unbounded) reachability problem between clock regions is Pspace-hard [AD94].

4 Discussion

We solved the duration-bounded reachability problem between two specified clock regions. Our construction can be used for solving many related problems. First, it should be clear that the initial and/or final region can be replaced either by a specific state with rational clock values, or by a specific location (i.e., a set of clock regions). For instance, suppose that we are given an initial state σ, a final state τ, a duration constraint $\int p \in I$, and we are asked to decide whether $(\sigma, 0) \rightarrow^* (\tau, \delta)$ for some real number $\delta \in I$. Assuming σ and τ assign *rational* values to all clocks, we can choose an appropriate time unit so that the regions $[\sigma]_\approx$ and $[\tau]_\approx$ are singletons. It follows that the duration-bounded reachability problem between rational states is also solvable in PSPACE.

A second example of a duration property we can decide is the following. Given a real-time system modeled as a timed automaton, and nonnegative integers m, a, and b, we sometimes want to verify that in every time interval of length m, the system spends at least a and at most b accumulated time units in a given set of locations. For instance, for a railroad crossing similar to the one that appears in various papers on real-time verification [AHH96], our algorithm can be used to check that "in every interval of 1 hour, the gate is closed for at most 5 minutes." The verification of this duration property, which depends on various gate delays and on the minimum separation time between consecutive trains, requires the accumulation of the time during which the gate is closed.

As a third, and final, example, recall the duration-bounded causality property $(*)$ from the introduction. Assume that each location of the timed automaton is labeled with atomic propositions such as q, denoting that the ringer is pressed, and r, denoting the response. The duration measure is defined so that $p(s) = 1$ if q is a label of s, and $p(s) = 0$ otherwise. The labeling of the locations with atomic propositions is extended to regions and bound-labeled regions. The desired duration-bounded causality property, then, *does not* hold iff there is an initial region R_0, a final region R_f labeled with r, and a bound-labeled region $B = (R_0, L, l, U, u)$ such that $I(B) \cap (2, \infty) = \emptyset$, and in the bounds graph B_{p,R_f}, there is a path from B to R_f that passes only through regions that are not labeled with r.

The duration-bounded reachability problem has been studied, independently, in [KPSY93] also. The approach taken there is quite different from ours. First, the problem is solved in the case of discrete time, where all transitions of a timed automaton occur at integer time values. Next, it is shown that the cases of discrete (integer-valued) time and dense (real-valued) time have the same answer, provided the following two conditions are met: (1) the clock constraints of timed automata use only positive boolean combinations of *non-strict* inequalities (i.e., inequalities involving $\leq$ and $\geq$ only); and (2) the duration constraint is *one-sided* (i.e., it has the form $\int p \sim k$ for $\sim \in \{\leq, <, \geq, >\}$ and $k \in \mathbb{N}$). The first requirement ensures that the runs of a timed automaton are closed under *digitization* (i.e., rounding of real-numbered transition times relative to an arbitrary, but fixed fractional part $\epsilon \in [0, 1)$ [HMP92]). The second requirement rules out duration constraints such as $\int p = 2$ and $2 \leq \int p \leq 3$. The approach of proving that the discrete-time and the dense-time answers to the duration-bounded reachability problem coincide gives a simpler solution than ours, and it also admits duration measures that assign negative integers to some locations. However, both requirements (1) and (2) are essential for this approach. We also note that for timed automata with a single clock, [KPSY93] gives an algorithm for checking more complex duration constraints, such as $\int p \in I \wedge \int p' \in I'$ for different duration measures p and p'.

Instead of equipping timed automata with duration measures, a more general approach extends timed automata with variables that measure accumulated durations. Such variables, which are called *integrators* or *stop watches*, may advance in any given location either with time derivative 1 (like a clock) or with time derivative 0 (not changing in value). Like clocks, integrators can be reset with transitions of the automaton, and the constraints guarding the automaton transitions can test integrator values. The reachability problem between the locations of a timed automaton with integrators, however, is undecidable [ACH+93, KPSY93]; indeed, a single integrator can cause undecidability [HKPV95]. Still, in many cases of practical interest, the reachability problem for timed automata with integrators can be answered by a symbolic execution of the automaton [ACH+93].

In contrast to the use of integrators, whose real-numbered values are part of the automaton state, we achieved decidability by separating duration constraints from the system and treating them as properties. This distinction between strengthening the model and strengthening the specification language with the duration constraints is essential for the decidability of the resulting verification problem. The expressiveness of specification languages can be increased further. For example, it is possible to define temporal logics with duration constraints or integrators. The decidability of the model-checking problem for such logics remains an open problem. For model checking a given formula, we need to compute the characteristic set, which contains the states that satisfy the formula. In particular, given an initial region R_0, a final state τ, and a duration constraint $\int p \in I$, we need to compute the set $Q_0 \subseteq R_0$ of states $\sigma \in R_0$ for which there exists a real number $\delta \in I$ such that $(\sigma, 0) \rightarrow^* (\tau, \delta)$. Each bound-labeled region (R_0, L, l, U, u) from which R_f is reachable in the bounds graph B_{p,R_f} contributes the subset $\{\sigma \in R_0 \mid I(\sigma, L, l, U, u) \cap I \neq \emptyset\}$ to Q_0. In general, there are infinitely many such contributions, possibly all singletons, and we know of no description of Q_0 that can be used to decide the model-checking problem. By contrast, over discrete time, the characteristic sets for formulas with integrators can be computed [BES93]. Also, over dense time, the characteristic sets can be approximated symbolically [AHH96].

Acknowledgements. We thank Sergio Yovine for a careful reading of the manuscript.

References

[ACD93] R. Alur, C. Courcoubetis, and D.L. Dill. Model checking in dense real time. *Information and Computation*, 104(1):2–34, 1993.

[ACH+93] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.

[AD94] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[AFH96] R. Alur, T. Feder, and T.A. Henzinger. The benefits of relaxing punctuality. *Journal of the ACM*, 43(1):116–146, 1996.

[AH92] R. Alur and T.A. Henzinger. Logics and models of real time: a survey. In J.W. de Bakker, K. Huizing, W.-P. de Roever, and G. Rozenberg, editors, *Real Time: Theory in Practice*, Lecture Notes in Computer Science 600, pages 74–106. Springer-Verlag, 1992.

[AHH96] R. Alur, T.A. Henzinger, and P.-H. Ho. Automatic symbolic verification of embedded systems. *IEEE Transactions on Software Engineering*, 22(3):181–201, 1996.

[BES93] A. Bouajjani, R. Echahed, and J. Sifakis. On model checking for real-time properties with durations. In *Proceedings of the Eighth Annual Symposium on Logic in Computer Science*, pages 147–159. IEEE Computer Society Press, 1993.

[CE81] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Workshop on Logic of Programs*, Lecture Notes in Computer Science 131. Springer-Verlag, 1981.

[Cer92] K. Cerāns. Decidability of bisimulation equivalence for parallel timer processes. In G. von Bochmann and D.K. Probst, editors, *CAV 92: Computer-aided Verification*, Lecture Notes in Computer Science 663, pages 302–315. Springer-Verlag, 1992.

[CHR91] Z. Chaochen, C.A.R. Hoare, and A.P. Ravn. A calculus of durations. *Information Processing Letters*, 40(5):269–276, 1991.

[CY91] C. Courcoubetis and M. Yannakakis. Minimum and maximum delay problems in real-time systems. In K.G. Larsen and A. Skou, editors, *CAV 91: Computer-aided Verification*, Lecture Notes in Computer Science 575, pages 399–409. Springer-Verlag, 1991.

[Dil89] D.L. Dill. Timing assumptions and verification of finite-state concurrent systems. In J. Sifakis, editor, *CAV 89: Automatic Verification Methods for Finite-state Systems*, Lecture Notes in Computer Science 407, pages 197–212. Springer-Verlag, 1989.

[HKPV95] T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? In *Proceedings of the 27th Annual Symposium on Theory of Computing*, pages 373–382. ACM Press, 1995.

[HMP92] T.A. Henzinger, Z. Manna, and A. Pnueli. What good are digital clocks? In W. Kuich, editor, *ICALP 92: Automata, Languages, and Programming*, Lecture Notes in Computer Science 623, pages 545–558. Springer-Verlag, 1992.

[HNSY94] T.A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994.

[KPSY93] Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Integration graphs: a class of decidable hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *Hybrid Systems*, Lecture Notes in Computer Science 736, pages 179–208. Springer-Verlag, 1993.

[QS81] J. Queille and J. Sifakis. Specification and verification of concurrent systems in CESAR. In M. Dezani-Ciancaglini and U. Montanari, editors, *Fifth International Symposium on Programming*, Lecture Notes in Computer Science 137, pages 337–351. Springer-Verlag, 1981.