Revisiting Reachability in Timed Automata

Karin Quaas*, Mahsa Shirmohammadi†, James Worrell†
*Universität Leipzig, Germany
†University of Oxford, UK

***Abstract*—We revisit a fundamental result in real-time verification, namely that the binary reachability relation between configurations of a given timed automaton is definable in linear arithmetic over the integers and reals. In this paper we give a new and simpler proof of this result, building on the well-known reachability analysis of timed automata involving difference bound matrices. Using this new proof, we give an exponential-space procedure for model checking the reachability fragment of the logic parametric TCTL. Finally we show that the latter problem is NEXPTIME-hard.**

***Index Terms*—Timed automata, Reachability, Difference Bound Matrices, Linear Arithmetic, Model Checking**

I. Introduction

The PSPACE-completeness of the reachability problem for timed automata is arguably the most fundamental result in real-time verification. This theorem was established by Alur and Dill in paper [1] for which they were awarded the Alonzo Church award in 2016. The reachability problem has been intensively studied in the intervening 20 years, leading to practical algorithms and generalisations to more expressive models. As of now, [1] is the most cited paper that has appeared in the journal *Theoretical Computer Science*.

Properly speaking, Alur and Dill considered reachability between *control states* (also called *locations*). The problem of computing the binary reachability relation over *configurations* (both control states and clock valuations) is more involved. Here the main result is due to Comon and Jurski [2], who showed that the reachability relation of a given timed automaton is effectively definable by a formula of first-order linear arithmetic over the reals augmented with a unary predicate denoting the integers. Importantly, this fragment of mixed linear arithmetic has a decidable satisfiability problem, e.g., by translation to S1S.

Despite its evident utility, particularly for parametric verification, it is fair to say that the result of Comon and Jurski has proven less influential than that of Alur and Dill. We believe that this is due both to the considerable technical complexity of the proof, which runs to over 40 pages in [3], as well as the implicit nature of their algorithm, making it hard to extract complexity bounds.

In this paper we revisit the result of Comon and Jurski. Our two main contributions as follows:

- We give a new and conceptually simpler proof that generalises the classical reachability algorithm for timed automata involving difference bound matrices and standard operations thereon. The key new idea is to carry out the algorithm on a symbolically presented initial configuration. This approach is fundamentally different from that of [2], the main part of which involves a syntactic transformation showing that every timed automaton can be effectively emulated by a *flat* timed automaton, i.e., one that does not contain nested loops in its control graph.
- We apply our strengthened formulation of the Comon-Jurski result to parametric model checking. We show that the formula representing the reachability relation can be computed in time singly exponential in the size of the timed automaton. Using this bound on the formula size and utilising results of [4], [5] on quantifier-elimination for first-order logic over the reals and integers, we show that the model checking problem for the reachability fragment of the temporal logic *parametric TCTL* is decidable in exponential space. We show in the main body of the paper that this problem is NEXPTIME-hard and sketch in the conclusion how to obtain matching upper and lower bounds.

There are two main steps in our approach to computing a formula representing the reachability relation. First, given a timed automaton $\mathcal{A}$ and a configuration $\langle \ell, \nu \rangle$ of $\mathcal{A}$, we construct a version of the region automaton of [1] that represents all configurations reachable from $\langle \ell, \nu \rangle$. Unlike [1] we do not identify all clock values above the maximum clock constant; so our version of the region automaton is a counter machine rather than a finite state automaton. The counters are used to store the integer parts of clock valuations of reachable configurations, while the fractional parts of the clock valuations are aggregated into zones that are represented within the control states of the counter machine by difference bound matrices. Since the counters mimic clocks they are monotonic and so the reachability relation on such a counter machine is definable in a weak fragment of Presburger arithmetic.

The second step of our approach is to make the previous construction parametric: we show that the form of the counter machine does not depend on the precise numerical values of the clocks in the initial valuation ν, just on a suitable logical *type* of ν. Given such a type, we develop a parametric version of the counter-machine construction. Combining this construction with the fact that the reachability relation for the considered class of counter machines is definable in a fragment of Presburger arithmetic, we obtain a formula that represents the full reachability relation of the timed automaton $\mathcal{A}$.

A. Related Work

Dang [6] has generalised the result of Comon and Jurski, showing that the binary reachability relation for pushdown timed automata is definable in linear arithmetic. The approach

in [6] relies on a finite partition of the fractional parts of clock valuations into so-called *patterns*, which play a role analogous to types in our approach. The notion of pattern is ad-hoc and, as remarked by Dang, relatively complicated. In particular, patterns lack the simple characterisation in terms of difference constraints that is possessed by types. The latter is key to our result that the reachability relation can be expressed by a Boolean combination of difference constraints.

Dima [7] gives an automata theoretic representation of the reachability relation of a timed automaton. To this end he introduces a class of automata whose runs encode tuples in such a relation. The main technical result of [7] is to show that this class of automata is effectively closed under relational reflexive-transitive closure.

The model checking problem for parametric TCTL was studied by Bruyère *et al.* [8], [9] in the case of integer-valued parameters. Here we allow real-valued parameters, which leads to a strictly more expressive semantics.

Parametric DBMs have been used in [10], [11] to analyse reachability in parametric timed automata. These are related to but different from the parametric DBMs occurring in Subsection III-C.

B. Organisation

We introduce and state our main results in the body of the paper. The central constructions underlying our proofs are also given in the body, along with illustrative examples. Many of the proof details are relegated to the appendix.

II. Main Definitions and Results

A. Timed Automata

Given a set $\mathcal{X} = \{x_1, \ldots, x_n\}$ of *clocks*, the set $\Phi(\mathcal{X})$ of *clock constraints* is generated by the grammar

$$\varphi ::= \mathtt{true} \mid x < k \mid x = k \mid x > k \mid \varphi \wedge \varphi \,,$$

where $k \in \mathbb{N}$ is a natural number and $x \in \mathcal{X}$. A *clock valuation* is a mapping $\nu : \mathcal{X} \to \mathbb{R}_{\geqslant 0}$, where $\mathbb{R}_{\geqslant 0}$ is the set of non-negative real numbers. We denote by $\mathbf{0}$ the valuation such that $\mathbf{0}(x) = 0$ for all $x \in \mathcal{X}$. Let $\mathbb{R}_{\geqslant 0}^{\mathcal{X}}$ be the set of all clock valuations. We write $\nu \models \varphi$ to denote that ν satisfies the constraint φ. Given $t \in \mathbb{R}_{\geqslant 0}$, we let $\nu + t$ be the clock valuation such that $(\nu + t)(x) = \nu(x) + t$ for all clocks $x \in \mathcal{X}$. Given $\lambda \subseteq \mathcal{X}$, let $\nu[\lambda \leftarrow 0]$ be the clock valuation such that $\nu[\lambda \leftarrow 0](x) = 0$ if $x \in \lambda$, and $\nu[\lambda \leftarrow 0](x) = \nu(x)$ if $x \notin \lambda$. We typically write ν_i as shorthand for $\nu(x_i)$, and by convention we define $\nu_0 = 0$. For all $r \in \mathbb{R}$, let $\mathrm{frac}(r)$ be the fractional part of r, and $\lfloor r \rfloor$ be the integer part. Denote by $\mathrm{frac}(\nu)$ and $\lfloor \nu \rfloor$ the valuations such that $(\mathrm{frac}(\nu))(x_i) = \mathrm{frac}(\nu_i)$ and $\lfloor \nu \rfloor(x_i) = \lfloor \nu_i \rfloor$ for all clocks $x_i \in \mathcal{X}$.

A *timed automaton* is a tuple $\mathcal{A} = \langle L, \mathcal{X}, E \rangle$, where L is a finite set of *locations*, $\mathcal{X}$ is a finite set of *clocks* and $E \subseteq L \times \Phi(\mathcal{X}) \times 2^{\mathcal{X}} \times L$ is the set of *edges*.

The semantics of a timed automaton $\mathcal{A} = \langle L, \mathcal{X}, E \rangle$ is given by a labelled transition system $\langle Q, \Rightarrow \rangle$ with set of *configurations* $Q = L \times \mathbb{R}_{\geqslant 0}^{\mathcal{X}}$ and set of *transition labels* $\mathbb{R}_{\geqslant 0}$. A configuration $\langle \ell, \nu \rangle$ consists of a location ℓ and a clock valuation ν. Given two configurations $\langle \ell, \nu \rangle$ and $\langle \ell', \nu' \rangle$, we postulate:

- a delay transition $\langle \ell, \nu \rangle \overset{d}{\Rightarrow} \langle \ell', \nu' \rangle$ for some $d \geqslant 0$, if $\nu' = \nu + d$ and $\ell = \ell'$;
- a discrete transition $\langle \ell, \nu \rangle \overset{0}{\Rightarrow} \langle \ell', \nu' \rangle$, if there is an edge $\langle \ell, \varphi, \lambda, \ell' \rangle$ of $\mathcal{A}$ such that $\nu \models \varphi$ and $\nu' = \nu[\lambda \leftarrow 0]$.

A *run* $\rho = q_0 \overset{d_1}{\Rightarrow} q_1 \overset{d_2}{\Rightarrow} q_2 \overset{d_3}{\Rightarrow} \ldots$ of $\mathcal{A}$ is a (finite or infinite) sequence of delay and discrete transitions in $\langle Q, \Rightarrow \rangle$. We require infinite runs to have infinitely many discrete transitions and to be *non-zeno*, that is, we require $\sum_{i=1}^{\infty} d_i$ to diverge.

Henceforth we assume that in any given timed automaton with set $\mathcal{X}$ of clocks, x_n is a special reference clock that is never reset. Clearly this assumption is without loss of generality for encoding the reachability relation.

Note that we consider timed automata without *diagonal constraints*, that is, guards of the form $x_i - x_j \sim k$, for k an integer. It is known that such constraints can be removed without affecting the reachability relation (see [1], [12]).

B. Linear Arithmetic

In this section we introduce a first-order language $\mathcal{L}_{\mathbb{R},\mathbb{Z}}$ in which to express the reachability relation of a timed automaton.

Language $\mathcal{L}_{\mathbb{R},\mathbb{Z}}$ has two sorts: a real-number sort and an integer sort. The collection $\mathcal{T}_{\mathbb{R}}$ of terms of real-number sort is specified by the grammar

$$t ::= c \mid r \mid t + t \mid t - t \,,$$

where $c \in \mathbb{Q}$ is a constant and $r \in \{r_0, r_1, \ldots\}$ is a real-valued variable. Given terms $t, t' \in \mathcal{T}_{\mathbb{R}}$, we have an atomic formula $t \leqslant t'$. The collection $\mathcal{T}_{\mathbb{Z}}$ of terms of integer sort is specified by the grammar

$$t ::= c \mid z \mid t + t \mid t - t \,,$$

where $c \in \mathbb{Z}$ is a constant and $z \in \{z_0, z_1, \ldots\}$ is an integer variable. Given terms $t, t' \in \mathcal{T}_{\mathbb{Z}}$, we have atomic formulas $t \leqslant t'$ and $t \equiv t' \pmod{m}$, where $m \in \mathbb{Z}$. Formulas of $\mathcal{L}_{\mathbb{R},\mathbb{Z}}$ are constructed from atomic formulas using Boolean connectives and first-order quantifiers.

Throughout the paper we consider a fixed semantics for $\mathcal{L}_{\mathbb{R},\mathbb{Z}}$ over the two-sorted structure in which the real-number sort is interpreted by $\mathbb{R}$, the integer sort by $\mathbb{Z}$, and with the natural interpretation of addition and order on each sort.

The sublanguage $\mathcal{L}_{\mathbb{R}}$ of $\mathcal{L}_{\mathbb{R},\mathbb{Z}}$ involving only terms of real-number sort is called *real arithmetic*. The sublanguage $\mathcal{L}_{\mathbb{Z}}$ involving only terms of integer sort is called *Presburger arithmetic*. Optimal complexity bounds for deciding satisfiability of sentences of real arithmetic and Presburger arithmetic are given in [13] with, roughly speaking, real arithmetic requiring single exponential space and Presburger arithmetic double exponential space.

Proposition 1. *Deciding the truth of a sentence in the existential fragment of $\mathcal{L}_{\mathbb{R},\mathbb{Z}}$ can be done in NP.*

Proof. The respective decision problems for the existential fragment of real arithmetic and the existential fragment of Presburger arithmetic are in NP [14], [15]. Deciding the truth of a sentence in the existential fragment of $\mathcal{L}_{\mathbb{R},\mathbb{Z}}$ is therefore also in NP, since we can guess truth values for the Presburger and real-arithmetic subformulas, and separately check realisability of the guessed truth values in non-deterministic polynomial time. □

For the purpose of model checking, it will be useful to establish complexity bounds for a language $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$, intermediate between $\mathcal{L}_{\mathbb{R}}$ and the full language $\mathcal{L}_{\mathbb{R},\mathbb{Z}}$. The language $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$ arises from $\mathcal{L}_{\mathbb{R},\mathbb{Z}}$ by restricting the atomic formulas over terms of integer sort to have the form

$$z - z' \leqslant c \mid z \leqslant c \mid z - z' \equiv c \pmod{d} \tag{1}$$

for integer variables z, z' and integers c, d.

Proposition 2. *Deciding the truth of a prenex-form sentence $Q_1 x_1 \ldots Q_n x_n\, \varphi$ in $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$ can be done in space exponential in n and polynomial in φ.*

Proof. The proposition is known to hold separately for $\mathcal{L}_{\mathbb{R}}$ [4] and for the fragment of $\mathcal{L}_{\mathbb{Z}}$ in which atomic formulas have the form shown in (1) [5, Section 4]. The respective arguments of [4] and [5] can be straightforwardly combined to prove the proposition; see Section A for details. □

C. Definability of the Reachability Relation

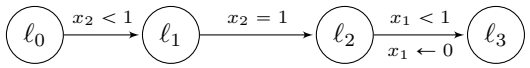
Given a timed automaton $\mathcal{A}$ with n clock variables, we express the reachability relation between every pair of locations ℓ, ℓ' by a formula

$$\varphi_{\ell,\ell'}(z_1, \ldots, z_n, r_1, , \ldots, r_n, z'_1, \ldots, z'_n, r'_1, , \ldots, r'_n),$$

in the existential fragment of $\mathcal{L}_{\mathbb{R},\mathbb{Z}}$ where $z_1, z'_1, \ldots, z_n, z'_n$ are integer variables and $r_1, r'_1, \ldots, r_n, r'_n$ are real variables ranging over the interval $[0, 1]$. Our main result, Theorem 10, shows that there is a finite run in $\mathcal{A}$ from configuration $\langle \ell, \nu \rangle$ to configuration $\langle \ell', \nu' \rangle$ just in case

$$\langle \lfloor \nu_1 \rfloor, \ldots, \lfloor \nu_n \rfloor, \mathrm{frac}(\nu_1), \ldots, \mathrm{frac}(\nu_n),$$
$$\lfloor \nu'_1 \rfloor, \ldots, \lfloor \nu'_n \rfloor, \mathrm{frac}(\nu'_1), \cdots, \mathrm{frac}(\nu'_n) \rangle \models \varphi_{\ell,\ell'}.$$

Example 1. *Consider the following timed automaton:*

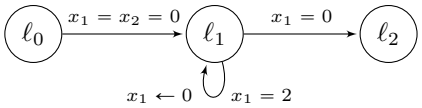


A brief inspection reveals that location ℓ_3 can be reached from a configuration $\langle \ell_0, \binom{\nu_1}{\nu_2} \rangle$ if and only if $\nu_1 < \nu_2 < 1$. The reachability relation between locations ℓ_0 and ℓ_3 is expressed by the formula

$$\begin{aligned}\varphi_{\ell_0,\ell_3}(z_1, z_2, r_1, r_2, z'_1, z'_2, r'_1, r'_2) \stackrel{\text{def}}{=}\ & (z_1 = z_2 = 0) \\ & \wedge (r_1 < r_2 < 1) \\ & \wedge ((z'_2 - z'_1 = 1 \wedge 0 \leqslant r'_2 - r'_1 < r_2 - r_1) \\ & \vee (z'_2 - z'_1 = 2 \wedge 0 \leqslant 1 + r'_2 - r'_1 < r_2 - r_1)),\end{aligned}$$

where the real-valued variables r_1, r_2, r'_1, r'_2 range over the interval $[0, 1]$.

Example 2. *Consider the following timed automaton:*



We have

$$\begin{aligned}\varphi_{\ell_0,\ell_3}(z_1, z_2, r_1, r_2, z'_1, z'_2, r'_1, r'_2) \stackrel{\text{def}}{=}\ & \\ & (r_1 = r_2 = 0) \wedge (r'_1 = r'_2) \wedge \\ & (z_1 = z_2 = 0) \wedge (z'_2 - z'_1 \equiv 0 \pmod 2).\end{aligned}$$

D. Parametric Timed Reachability Logic

Timed computation tree logic (TCTL) is an extension of computation tree logic for specifying real-time properties [16]. In [8] TCTL was generalised to allow parameters within timing constraints, yielding the logic *parametric TCTL*. In this paper we consider the fragment of parametric TCTL generated by the reachability modality $\exists\Diamond$, which we call *parametric timed reachability logic (PTRL)*.

Let AP be a set of atomic propositions and Θ a set of parameters. Formulas of PTRL of the *first type* are given by the grammar

$$\varphi ::= p \mid \varphi \wedge \varphi \mid \neg\varphi \mid \exists\Diamond_{\sim\alpha}\, \varphi\,, \tag{2}$$

where $p \in AP$, $\sim \in \{<, \leqslant, =, \geqslant, >\}$, and $\alpha \in \mathbb{Q} \cup \Theta$. Formulas of PTRL of the *second type* are given by grammar

$$\psi ::= \varphi \mid \theta - \theta' \sim c \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid \exists\theta\, \psi\,, \tag{3}$$

where φ is a formula of the first type, $\theta, \theta' \in \Theta$, $\sim \in \{<, \leqslant, =, \geqslant, >\}$, and $c \in \mathbb{Q}$. In the sequel we use $\forall\, \Box_{\sim\alpha}\, \varphi$ as abbreviation for $\neg\exists\Diamond_{\sim\alpha}\neg\varphi$.

Formulas of PTRL are interpreted with respect to a timed automaton $\mathcal{A} = \langle L, \mathcal{X}, E \rangle$ and *labelling function* $LB : L \to 2^{AP}$. A *parameter valuation* is a function $\xi : \Theta \to \mathbb{R}_{\geqslant 0}$. Such a function is extended to the rational numbers by writing $\xi(c) = c$ for $c \in \mathbb{Q}$. Given a parameter valuation ξ, we define a satisfaction relation $\models_\xi$ between configurations of $\mathcal{A}$ and PTRL formulas by induction over the structure of formulas. The Boolean connectives are handled in the expected way, and we define

- $q \models_\xi \theta - \theta' \sim c$ iff $\xi(\theta) - \xi(\theta') \sim c$.
- $q \models_\xi \exists\Diamond_{\sim\alpha}\, \varphi$ iff there exists some infinite non-zeno run $\rho = q_0 \stackrel{d_1}{\Rightarrow} q_1 \stackrel{d_2}{\Rightarrow} q_2 \stackrel{d_3}{\Rightarrow} \ldots$ of $\mathcal{A}$ and $i \in \mathbb{N}$ such that $q_0 = q$, $d_1 + \ldots + d_i \sim \xi(\alpha)$, and $q_i \models \varphi$.
- $q \models_\xi \exists\theta\, \psi$ iff there exists a parameter valuation ξ' such that $q \models_{\xi'} \psi$ and ξ, ξ' agree on $\Theta \backslash \{\theta\}$.

Example 3. *The PTRL-formula $\forall\theta(\exists\Diamond_{<\theta} p_1 \to \exists\Diamond_{<\theta} p_2)$ expresses that some p_2-state is reachable in at most the same time as any p_1-state is reachable.*

The paper [8] considered a semantics for parametric TCTL in which parameters range over naturals $\mathbb{N}$. Here we have given

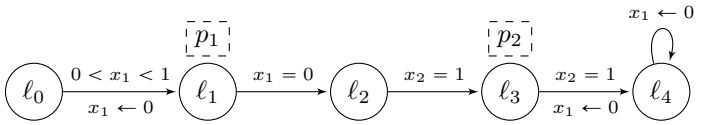
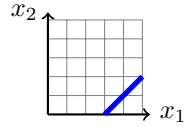


Fig. 1. A timed automaton where the satisfaction relation of PTRL with parameters ranging over non-negative real numbers is different from the relation when parameters are restricted to naturals. The locations ℓ_1 and ℓ_3 are labelled by propositions p_1 and p_2, respectively. The set λ of clocks that are reset by a transitions are shown by $\lambda \leftarrow 0$; for example, the transition from ℓ_3 to ℓ_4 is guarded by $x_2 = 1$ and resets x_1. For all $0 < \theta < 1$, we have $(\ell_0, \mathbf{0}) \models \exists \Diamond (p_1 \wedge \exists \Diamond_{=\theta} p_2)$, whereas there exists no $n \in \mathbb{N}$ such that $(\ell_0, \mathbf{0}) \models \exists \Diamond (p_1 \wedge \exists \Diamond_{=n} p_2)$.



$$\begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \end{array} \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{bmatrix} (\leqslant, 0) & (\leqslant, -0.6) & (\leqslant, 0) \\ (\leqslant, 1) & (\leqslant, 0) & (\leqslant, 0.6) \\ (\leqslant, 0.4) & (\leqslant, -0.6) & (\leqslant, 0) \end{bmatrix} \end{array}$$

Fig. 2. A DBM M with a zone $Z = [\![M]\!]$.

a more general semantics in which parameters range over non-negative real numbers $\mathbb{R}_{\geqslant 0}$. The following example shows that the satisfaction relation changes under this extension.

Example 4. *Consider the timed automaton in Figure 1 with two clocks x_1, x_2. Clock valuations ν are denoted by vectors $\binom{\nu_1}{\nu_2}$. Let $\varphi = \exists \Diamond (p_1 \wedge \exists \Diamond_{=\theta} p_2)$. All non-zeno infinite runs of the timed automaton, from configuration $\langle \ell_0, \mathbf{0} \rangle$, start with the following prefix*

$$(\ell_0, \tbinom{0}{0}) \overset{t}{\Rightarrow} (\ell_1, \tbinom{0}{t}) \overset{0}{\Rightarrow} (\ell_2, \tbinom{0}{t}) \overset{1-t}{\Rightarrow} (\ell_3, \tbinom{1-t}{1}) \overset{0}{\Rightarrow} (\ell_4, \tbinom{0}{1})$$

where $0 < t < 1$. Now we have that $(\ell_1, \binom{0}{t}) \models (p_1 \wedge \exists \Diamond_{=1-t} p_2)$. As a result, $(\ell_0, \mathbf{0}) \models \exists \Diamond (p_1 \wedge \forall \Diamond_{=\theta} p_2)$ only for $0 < \theta < 1$. Thus $(\ell_0, \mathbf{0}) \models \exists \theta \, \varphi$ when the parameter θ ranges over $\mathbb{R}_{\geqslant 0}$ but not when θ ranges over $\mathbb{N}$.

Let $\mathcal{A} = \langle L, \mathcal{X}, E \rangle$ be a timed automaton augmented with a labelling function $LB : L \to 2^{AP}$. Let φ be a PTRL formula in which all occurrences of parameters are bound. The model checking problem of $\mathcal{A}$ against φ asks, given a configuration $\langle \ell, \nu \rangle$ of $\mathcal{A}$, whether $\langle \ell, \nu \rangle \models \varphi$.

The model checking procedure for parametric TCTL with integer-valued parameters, developed in [8], relies on the region abstraction. In particular, formulas in this logic have the same truth value for all configurations in a given region. However, as the following example shows, region invariance fails when parameters range over the set of real numbers.

Example 5. *Consider the timed automaton in Figure 1. Let $\varphi = \exists \theta \, \exists \Diamond_{=\theta} (p_1 \wedge \exists \Diamond_{=\theta} p_2)$. Then a configuration $(\ell_0, \binom{t_1}{t_2})$ satisfies φ just in case $t_1, t_2 < 1$ and $2t_1 - t_2 < 1$, for $\theta = (1 - t_2)/2$.*

In Section V we show that model checking PTRL over real-valued parameters is decidable in EXPSPACE and it is NEXPTIME-hard.

III. Difference Bound Matrices

A. Basic Definitions

In this section we review the notions of clock zones and difference bound matrices; see [17], [18] for further details.

Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be a set of clock variables. A *zone* $Z \subseteq \mathbb{R}^{\mathcal{X}}_{\geqslant 0}$ is a set of valuations defined by a conjunction of *difference constraints* $x_j - x_i \prec c$ for $c \in \mathbb{R}$ and $\prec \in \{<, \leqslant\}$. Note that we allow real-valued constants in difference constraints.

Zones and operations thereon can be efficiently represented using *difference bound matrices* (DBMs). A DBM is an $(n+1) \times (n+1)$ matrix M with entries in the set

$$\mathbb{V} = (\{<, \leqslant\} \times \mathbb{R}) \cup \{(<, \infty)\}.$$

A DBM $M = (\prec_{i,j}, m_{i,j})$ can be interpreted as a conjunction of constraints $x_i - x_j \prec_{i,j} m_{i,j}$, where x_0 is a special clock that symbolically represents zero. Formally, the semantics of DBM M is the zone

$$[\![M]\!] = \left\{ \nu \in \mathbb{R}^{\mathcal{X}}_{\geqslant 0} : \bigwedge_{0 \leqslant i,j \leqslant n} \nu_i - \nu_j \prec_{i,j} m_{i,j} \right\},$$

where $\nu_0 = 0$. Figure 2 depicts a zone $Z \subseteq [0,1]^2$ containing a line segment and a DBM M with $[\![M]\!] = Z$.

An *atomic* DBM M' is one that represents a single constraint $x_i - x_j \sim c$, where $\sim \in \{<, \leqslant\}$ and $c \in \mathbb{R}$. Note that all but one entry of an atomic DBM is the trivial constraint $(<, \infty)$. We often denote DBMs by the constraints that they represent.

Define a total order $\leqslant_{\mathbb{V}}$ on $\mathbb{V}$ by writing $(\prec, m) \leqslant_{\mathbb{V}} (\prec', m')$ if $m < m'$ or if $m = m'$ and either $\prec = <$ or $\prec' = \leqslant$. Define addition on $\mathbb{V}$ by $(\prec, m) + (\prec', m') = (\prec'', m + m')$, where

$$\prec'' = \begin{cases} \leqslant & \text{if } \prec = \leqslant \text{ and } \prec' = \leqslant, \\ < & \text{otherwise.} \end{cases}$$

Here we adopt the convention that $m + \infty = \infty + m = \infty$ for all $m \in \mathbb{R}$. A DBM $M = (M_{i,j})$ is in *canonical form* if $M_{i,k} \leqslant_{\mathbb{V}} M_{i,j} + M_{j,k}$ for all $0 \leqslant i, j, k \leqslant n$. One can transform an arbitrary DBM into an equivalent canonical-form DBM using the Floyd-Warshall algorithm. For all non-empty clock zones Z, there is a unique DBM M in canonical form with $[\![M]\!] = Z$. A DBM M is said to be *consistent* if $[\![M]\!] \neq \varnothing$. If M is in canonical form, then it is consistent if and only if $(\leqslant, 0) \leqslant_{\mathbb{V}} M_{i,i}$ for all $0 \leqslant i \leqslant n$.

We now define operations on DBMs that correspond to time elapse, projection, and intersection on zones.

Time Elapse. The image of a DBM M under *time elapse* is the DBM $\overrightarrow{M}$ defined by

$$\overrightarrow{M}_{i,j} = \begin{cases} (<, \infty) & \text{if } i \neq 0, j = 0 \\ M_{i,j} & \text{otherwise.} \end{cases}$$

If M is canonical, then $\overrightarrow{M}$ is also canonical and we have $[\![\overrightarrow{M}]\!] = \{\nu + t : \nu \in [\![M]\!] \text{ and } t \geqslant 0\}$.

Reset. The image of a DBM M under *resetting clock* x_ℓ is $M[x_\ell \leftarrow 0]$, given by $M[x_\ell \leftarrow 0]_{i,j} = M_{i_\ell,j_\ell}$, where for any index k,

$$k_\ell = \begin{cases} k & \text{if } k \neq \ell \\ 0 & \text{otherwise.} \end{cases}$$

If M is canonical, then $M[x_\ell \leftarrow 0]$ is also canonical and $[\![M]\!] = \{\nu[x_\ell \leftarrow 0] : \nu \in [\![M]\!]\}$.

Intersection. Our presentation of intersection of DBMs is slightly non-standard. First, we only consider intersection with atomic DBMs. (Clearly this is without loss of generality since any DBM can be written as an intersection of atomic DBMs.) Under this restriction we combine intersection and canonisation, so that our intersection operation yields a DBM in canonical form if the input DBM is in canonical form. Specifically, let M' be an atomic DBM with non-trivial constraint $M'_{p,q}$. The DBM $M'' = M \cap M'$ is given by

$$M''_{i,j} = \min(M_{i,j}, M_{i,p} + M'_{p,q} + M_{q,j})$$

for all i, j. Then M'' is canonical and $[\![M'']\!] = [\![M]\!] \cap [\![M']\!]$.

B. Closure of a DBM

We will use zones to represent the fractional parts of clocks in a given set of valuations. For this reason we are solely interested in zones contained in $[0,1]^n$. We say that a DBM M is 1-*bounded* if for all entries $(\prec, m)$ of M we have $-1 \leqslant m \leqslant 1$. It is clear that if M is 1-bounded then $[\![M]\!] \subseteq [0,1]^n$. Conversely the unique DBM in canonical form that represents a zone $Z \subseteq [0,1]^n$ is necessarily 1-bounded since the constraints in a canonical DBM cannot be tightened.

Given a 1-bounded DBM M, define the *closure* M to be the smallest set $\mathit{closure}(M)$ of DBMs such that $M \in \mathit{closure}(M)$, and if $N \in \mathit{closure}(M)$ then

- $N \cap M' \in \mathit{closure}(M)$ for all atomic DBMs M' with numerical entries in $\mathbb{Z} \cup \{\infty\}$.
- $\overrightarrow{N} \cap \bigcap_{i=1}^{n}(x_i \leqslant 1) \in \mathit{closure}(M)$,
- $N[x_i \leftarrow 0] \in \mathit{closure}(M)$ for $0 \leqslant i \leqslant n-1$,
- $(N \cap (x_n = 1))[x_n \leftarrow 0] \in \mathit{closure}(M)$.

We make three observations about this definition. First, notice that in the first item we only require closure with respect to intersection with constraints with integer constants. Observe also that in the second item the time elapse operation has been relativized to $[0,1]^n$. This ensures that every DBM $N \in \mathit{closure}(M)$ denotes a subset of $[0,1]^n$. It follows that any consistent DBM in $\mathit{closure}(M)$ is 1-bounded. Finally, note that the clock x_n is treated in a special way (in keeping with our assumptions about timed automata in Section II-A): it is only reset when it reaches 1.

Let $\nu \in [0,1]^n$ be a clock valuation, and recall that, by convention, $\nu_0 = 0$. We write M_ν for the 1-bounded DBM $M_\nu = (\prec_{i,j}, m_{i,j})$, where $\prec_{i,j} = \leqslant$ and $m_{i,j} = \nu_j - \nu_i$ for all $0 \leqslant i, j \leqslant n$. Then M_ν is in canonical form and $[\![M_\nu]\!] = \{\nu\}$.

We say a DBM $M = (\prec_{i,j}, m_{i,j}) \in \mathit{closure}(M_\nu)$ is *well-supported*, if each entry $m_{i,j}$ can be written in the form $c + \nu_{j'} - \nu_{i'}$ for some $c \in \{-1, 0, 1\}$ and indices $0 \leqslant i', j' \leqslant n$. Clearly M_ν is well-supported.

The following is the main technical result in this section. See Appendix B for the full proof.

Lemma 3. *Let $\nu \in [0,1]^n$ be a clock valuation. Then every consistent DBM lying in $\mathit{closure}(M_\nu)$ is well-supported.*

Proof Sketch. We show by induction on the structure of $\mathit{closure}(M_\nu)$ that any consistent DBM $M \in \mathit{closure}(M_\nu)$ is well-supported. The key case is for intersection (see Section III-A), which does not immediately preserve well-supportedness due to the possibility that $M''_{i,j} = M_{i,p} + M'_{p,q} + M_{q,j}$. However we show that in this case at least one of $m_{i,p}$ or $m_{q,j}$ lies in $\mathbb{Z}$, which ensures well-supportedness of M''. □

C. Parametric DBMs

In this subsection we observe that the construction of $\mathit{closure}(M_\nu)$ can be carried out parametrically, based on the logical *type* of the clock valuation $\nu \in [0,1]^n$ (to be defined below). In particular, if $\nu, \nu' \in [0,1]^n$ have the same type, then $\mathit{closure}(M_\nu)$ and $\mathit{closure}(M_{\nu'})$ can both be seen as instances of a common parametric construction.

Recall from Subsection II-B the definition of the set of terms $\mathcal{T}_\mathbb{R}$ of real arithmetic. Given $n \in \mathbb{N}$, let us further write $\mathcal{T}_\mathbb{R}(n)$ for the set of terms in variables $r_0, \ldots, r_n$. A valuation $\nu \in [0,1]^n$ extends in a natural way to a function $\nu : \mathcal{T}_\mathbb{R}(n) \to \mathbb{R}$ mapping r_i to ν_i (recalling the convention that $\nu_0 = 0$).

Given a clock valuation $\nu \in [0,1]^n$, the *type* of ν is the set of atomic $\mathcal{L}_\mathbb{R}$-formulas $t \leqslant t'$, with $t, t' \in \mathcal{T}_\mathbb{R}(n)$ that are satisfied by the valuation ν. A collection of atomic formulas τ is said to be an n-*type* if it is the type of some clock valuation $\nu \in [0,1]^n$. Note that every type contains the inequalities $r_0 \leqslant 0$ and $0 \leqslant r_0$.

Given an n-type τ, we define an equivalence relation on the set of terms $\mathcal{T}_\mathbb{R}(n)$ that relates terms t and t' just in case the formulas $t \leqslant t'$ and $t' \leqslant t$ both lie in τ. We write $[t]$ for the equivalence class of term t and denote by $\mathcal{T}_\mathbb{R}(\tau)$ the set of equivalence classes of $\mathcal{T}_\mathbb{R}(n)$. We can define a linear order on $\mathcal{T}_\mathbb{R}(\tau)$ by writing $[t] \leqslant [t']$ if and only if formula $t \leqslant t'$ lies in τ. We define an addition operation on $\mathcal{T}_\mathbb{R}(\tau)$ by writing $[t] + [t'] = [t + t']$.

Given an n-type τ, a *parametric DBM* of dimension n over $\mathcal{T}_\mathbb{R}(\tau)$ is an $(n+1) \times (n+1)$ matrix with entries in

$$(\{<, \leqslant\} \times \mathcal{T}_\mathbb{R}(\tau)) \cup \{(<, \infty)\}.$$

We use letters in calligraphic font to denote parametric DBMs, and roman font for concrete DBMs. Given a parametric DBM $\mathcal{M}$, we obtain a concrete DBM $\nu(\mathcal{M})$ by applying ν pointwise to the entries of $\mathcal{M}$.

The time elapse and reset operations on DBMs, defined in Section III-A, formally carry over to parametric DBMs. Since the notions of addition and minimum are well-defined

on $\mathcal{T}_{\mathbb{R}}(\tau)$, we can also formally carry over the definition of intersection to parametric DBMs.

Proposition 4. *Let $\nu \in [0,1]^n$ be a clock valuation with type τ and let $\mathcal{M}$ be a parametric DBM over $\mathcal{T}_{\mathbb{R}}(\tau)$. Then*

1) $\nu(\overrightarrow{\mathcal{M}}) = \overrightarrow{\nu(\mathcal{M})}$.
2) $\nu(\mathcal{M}[x_i \leftarrow 0]) = \nu(\mathcal{M})[x_i \leftarrow 0]$.
3) $\nu(\mathcal{M} \cap N) = \nu(\mathcal{M}) \cap N$ *for all atomic DBMs* N.

Proof. Suppose that ν has type τ. Then $\nu : \mathcal{T}_{\mathbb{R}}(\tau) \to \mathbb{R}$ is an order embedding ($[t] \leqslant [t']$ if and only if $\nu(t) \leqslant \nu(t')$) and a homomorphism ($\nu([t]+[t']) = \nu([t])+\nu([t])$). In particular, ν preserves all operations used to define time elapse, projection, and intersection of DBMs. The result follows. □

Since the basic operations on DBMs are all defined for parametric DBMs, we can also formally carry over the definition of the closure of a DBM to parametric DBMs. In particular, given an n-type τ, we consider the closure of the parametric DBM $\mathcal{M}_\tau = (\prec_{i,j}, m_{i,j})$ over $\mathcal{T}_{\mathbb{R}}(\tau)$, where $\prec_{i,j} = \leqslant$ and $m_{i,j} = [r_i - r_j]$. Note that $\nu(\mathcal{M}_\tau) = M_\nu$ for any clock valuation $\nu \in [0,1]^n$. Then, by Proposition 4, we have the following result:

Proposition 5. *Let $\nu \in [0,1]^n$ be a clock valuation with type τ. Then*

$$\{\nu(\mathcal{M}) : \mathcal{M} \in \mathit{closure}(\mathcal{M}_\tau)\} = \mathit{closure}(M_\nu)\,.$$

Define the set $\mathcal{DT}_{\mathbb{R}}(n)$ of *difference terms* to be the subset of $\mathcal{T}_{\mathbb{R}}(n)$ comprising those terms of the form $c + r_i - r_j$, where $c \in \{-1, 0, 1\}$ is a constant and r_i, r_j are variables with $0 \leqslant i, j \leqslant n$. From Lemma 3 and Proposition 5 we now have:

Corollary 6. *Fix an n-type τ. Then every DBM in $\mathit{closure}(\mathcal{M}_\tau)$ has all its entries of the form $(\prec, [t])$, where $\prec \in \{<, \leqslant\}$ and $t \in \mathcal{DT}_{\mathbb{R}}(n)$.*

The significance of Corollary 6 is that the only part of the type τ required to determine $\mathit{closure}(\mathcal{M}_\tau)$ is the *finite* collection of formulas $t \leqslant t'$ in τ such that $t, t' \in \mathcal{DT}_{\mathbb{R}}(n)$. Thus $\mathit{closure}(\mathcal{M}_\tau)$ is finite. Indeed it is not hard to see from Corollary 6 that $|\mathit{closure}(\mathcal{M}_\tau)| \leqslant 2^{\mathrm{poly}(n)}$.

IV. A Family of Region Automata

Let $\mathcal{A}$ be a timed automaton. Our aim in this section is to define a finite collection of counter machines that represents the reachability relation on $\mathcal{A}$. Intuitively the counters in these machines are used to store the integer parts of clock valuations of reachable configurations, while the fractional parts of the clock valuations are aggregated into zones which are represented by difference bound matrices encoded within control states.

A. Monotonic Counter Machine

In this subsection we introduce the class of *monotonic counter machines* and show that the reachability relation for a machine in this class is definable in Presburger arithmetic. The proof is straightforward, and is related to the fact that the reachability relation of every reversal-bounded counter machine is Presburger definable [19].

Let $C = \{c_1, \ldots, c_n\}$ be a finite set of *counters*. The collection of *guards*, denoted $\Phi(C)$, is given by the grammar

$$\varphi ::= \mathtt{true} \mid c < k \mid c = k \mid c > k \mid \varphi \wedge \varphi\,,$$

where $c \in C$ and $k \in \mathbb{Z}$. The set of *counter operations* is

$$\mathrm{Op}(C) = \{\mathit{reset}(c), \mathit{inc}(c) : c \in C\} \cup \{\mathit{nop}\}\,.$$

A *monotonic counter machine* is a tuple $\mathcal{C} = \langle S, C, \Delta\rangle$, where S is a finite set of *states*, C is a finite set of *counters*, and $\Delta \subseteq S \times \Phi(C) \times \mathrm{Op}(C) \times S$ is a set of *edges*.

The set of *configurations* of $\mathcal{C}$ is $S \times \mathbb{N}^n$. A configuration $\langle s, \upsilon\rangle$ consists of a state $s \in S$ and a *counter valuation* $\upsilon \in \mathbb{N}^n$, where υ_i represents the value of counter c_i for $i = 1, \ldots, n$. The satisfaction relation $\models$ between counter valuations and guards is defined in the obvious way. The *transition relation*

$$\to \subseteq (S \times \mathbb{N}^n) \times (S \times \mathbb{N}^n)$$

is specified by writing $\langle s, \upsilon\rangle \to \langle s', \upsilon'\rangle$ just in case at least one of the following holds:

- there is an edge $\langle s, \varphi, \mathit{nop}, s'\rangle \in \Delta$ such that $\upsilon \models \varphi$ and $\upsilon = \upsilon'$;
- there is an edge $\langle s, \varphi, \mathit{reset}(c_i), s'\rangle \in \Delta$ such that $\upsilon \models \varphi$, $\upsilon'_i = 0$, and $\upsilon'_j = \upsilon_j$ for $i \neq j$;
- there is an edge $\langle s, \varphi, \mathit{inc}(c_i), s'\rangle \in \Delta$ such that $\upsilon \models \varphi$, $\upsilon'_i = \upsilon_i + 1$, and $\upsilon'_j = \upsilon_j$ for $i \neq j$.

The reachability relation on $\mathcal{C}$ is the reflexive transitive closure of $\to$.

The proof of the following result is given in Appendix C.

Proposition 7. *Let $\mathcal{C}$ be a monotonic counter machine with n counters. Given states s, s' of $\mathcal{C}$, the reachability relation*

$$\{\langle \upsilon, \upsilon'\rangle \in \mathbb{N}^{2n} : \langle s, \upsilon\rangle \longrightarrow^* \langle s', \upsilon'\rangle\}$$

is definable by a formula in the existential fragment of Presburger arithmetic that has size exponential in $\mathcal{C}$.

B. Concrete Region Automata

Let $\mathcal{A} = \langle L, \mathcal{X}, E\rangle$ be a timed automaton and $\langle \ell, \nu\rangle$ a configuration of $\mathcal{A}$. We define a monotonic counter machine $\mathcal{C}_{\langle \ell, \nu\rangle}$ whose configuration graph represents all configurations of $\mathcal{A}$ that are reachable from $\langle \ell, \nu\rangle$.

Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be the set of clocks in $\mathcal{A}$. Recall from Section II-A the assumption that clock x_n is never reset by the timed automaton. To simplify the construction, we also assume that each transition in $\mathcal{A}$ resets at most one clock. This is without loss of generality with respect to reachability.

Given a clock constraint $\varphi \in \Phi(\mathcal{X})$, we decompose φ into an integer constraint $\varphi_{\mathsf{int}} \in \Phi(C)$ and a real constraint $\varphi_{\mathsf{frac}} \in \Phi(\mathcal{X})$ such that for every clock valuation $\nu' \in \mathbb{R}^{\mathcal{X}}_{\geqslant 0}$,

$$\nu' \models \varphi \quad \text{iff} \quad \lfloor \nu' \rfloor \models \varphi_{\mathsf{inc}} \text{ and } \mathrm{frac}(\nu') \models \varphi_{\mathsf{frac}}$$

The definition of φ_{int} and φ_{frac} is by induction on the structure of φ. The details are given in Figure 4.

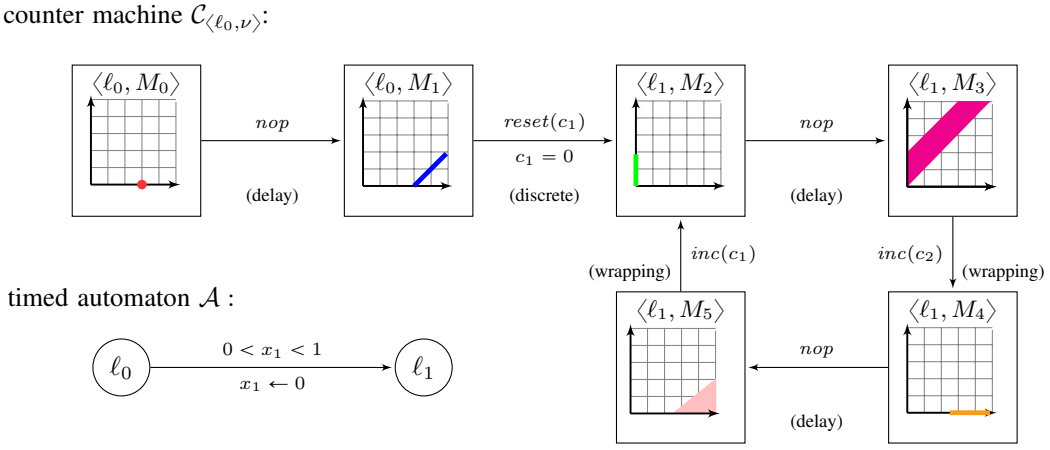


Fig. 3. A timed automaton $\mathcal{A}$ together with the fragment of counter machine $\mathcal{C}_{\langle \ell_0,\nu\rangle}$ relevant to expressing the reachability relation of ℓ_0 and ℓ_1. The valuation ν is such that $\nu_1 = 0.6$ and $\nu_2 = 0$. States $\langle \ell, M\rangle$ of the counter machine are illustrated by ℓ and the zone that M represents. The initial state is $\langle \ell_0, M_0\rangle$, where $M_0 = M_\nu$.

φ	$x < k$	$x = k$	$k < x < k+1$	$x \geqslant k$
φ_{int}	$c \leqslant k-1$	$c = k$	$c = k$	$c \geqslant k$
φ_{frac}	$x < 1$	$x = 0$	$0 < x < 1$	$x \geqslant 0$

Fig. 4. Base cases of the inductive definition of φ_{inc} and φ_{frac}, where x is a clock variable and c is a counter variable. (Note any guard $\varphi \in \Phi(X)$ can be expressed as a Boolean combination of the basic guards in the table.) For the inductive step we have $(\varphi \wedge \varphi')_{\mathsf{int}} = \varphi_{\mathsf{int}} \wedge \varphi'_{\mathsf{int}}$ and $(\varphi \wedge \varphi')_{\mathsf{frac}} = \varphi_{\mathsf{frac}} \wedge \varphi'_{\mathsf{frac}}$.

The construction of the counter machine $\mathcal{C}_{\langle \ell,\nu\rangle} = \langle S, C, \Delta\rangle$ is such that the set S of states comprises all pairs $\langle \ell', M\rangle$ such that $\ell' \in L$ is a location of $\mathcal{A}$ and $M \in \mathit{closure}(M_{\mathrm{frac}(\nu)})$ is a consistent DBM. The set of counters is $C = \{c_1, \ldots, c_n\}$, where n is the number of clocks in $\mathcal{A}$. Intuitively the purpose of counter c_i is to store the integer part of clock x_i, for $i = 1, \ldots, n$.

We classify the transitions of $\mathcal{C}_{\langle \ell,\nu\rangle}$ into three different types: From all states $\langle \ell_1, M_1\rangle$ to a state $\langle \ell_1, M_2\rangle$, there is

- a *delay transition* if $M_2 = \overrightarrow{M_1} \cap \bigcap_{i=1}^{n}(x_i \leqslant 1)$. Such a transition has guard `true` and operation nop;
- a *wrapping transition* if $M_2 = (M_1 \cap (x_i = 1))[x_i \leftarrow 0]$ for some clock x_i. Such a transition has guard `true` and operation $inc(c_i)$.

Suppose that $(\ell, \varphi, \{x_i\}, \ell')$ is a transition of $\mathcal{A}$. Decompose the guard φ into φ_{int} and φ_{frac}. Then from all states $\langle \ell_1, M_1\rangle$ to a state $\langle \ell_2, M_2\rangle$, there is

- a *discrete transition* if $M_2 = (M_1 \cap \varphi_{\mathsf{frac}})[x_i \leftarrow 0]$. Such a transition has guard φ_{int} and operation $reset(c_i)$.

The following proposition describes how the set of reachable configurations in $\mathcal{C}_{\langle \ell,\nu\rangle}$ represents the set of configurations reachable from $\langle \ell, \nu\rangle$ in the timed automaton $\mathcal{A}$. The proposition is a straightforward variant of the soundness and completeness of the DBM-based forward reachability algorithm for timed automata, as shown, e.g., in [20, Theorem 1]. We give a proof in Appendix D.

Proposition 8. *Configuration $\langle \ell', \nu'\rangle$ is reachable from $\langle \ell, \nu\rangle$ in $\mathcal{A}$ if and only if there exists some DBM $M' \in \mathit{closure}(M_{\mathrm{frac}(\nu)})$ such that the configuration $\langle\langle \ell', M'\rangle, \lfloor \nu'\rfloor\rangle$ is reachable from $\langle\langle \ell, M_{\mathrm{frac}(\nu)}\rangle, \lfloor \nu\rfloor\rangle$ in the counter machine $\mathcal{C}_{\langle \ell,\nu\rangle}$ and $\mathrm{frac}(\nu') \in [\![M']\!]$.*

We illustrate the translation from timed automata to counter machines with the following example.

Example 6. *Consider the timed automaton $\mathcal{A}$ in Figure 3 with clocks $\mathcal{X} = \{x_1, x_2\}$, where x_2 is the reference clock. Let the configuration $\langle \ell_0, \nu\rangle$ be such that $\nu = \binom{0.6}{0}$. Also shown in Figure 3 is the counter machine $\mathcal{C}_{\langle \ell_0,\nu\rangle}$ that is constructed from $\mathcal{A}$ and $\langle \ell_0, \nu\rangle$ in the manner described above. The control states of this machine are pairs $\langle \ell, M\rangle$, where ℓ is a location of $\mathcal{A}$ and M is a consistent DBM in $\mathit{closure}(M_\nu)$. The machine $\mathcal{C}_{\langle \ell_0,\nu\rangle}$ has two counters, respectively denoted by c_1 and c_2.*

The initial state of $\mathcal{C}_{\langle \ell_0,\nu\rangle}$ is $\langle \ell_0, M_0\rangle$, where $M_0 = M_\nu$. Note that $[\![M_0]\!] = \{\binom{0.6}{0}\}$. The counter-machine state $\langle \ell_0, M_0\rangle$ in tandem with counter valuation $\binom{0}{0}$ represents the configuration $\langle \ell_0, \nu\rangle$ of $\mathcal{A}$.

There is a delay edge in $\mathcal{C}_{\langle \ell_0,\nu\rangle}$ from $\langle \ell_0, M_0\rangle$ to $\langle \ell_0, M_1\rangle$, where $M_1 = \overrightarrow{M_0} \cap \bigcap_{i=1}^{2}(x_i \leqslant 1)$. We then have $[\![M_1]\!] = \{\binom{0.6}{0} + t : 0 \leqslant t \leqslant 0.4\}$.

The single transition of $\mathcal{A}$ yields a discrete edge in $\mathcal{C}_{\langle \ell_0,\nu\rangle}$ from $\langle \ell_0, M_1\rangle$ to $\langle \ell_1, M_2\rangle$. This transition in $\mathcal{A}$ has guard $\varphi \stackrel{\text{def}}{=} 0 < x_1 < 1$. This decomposes into separate constraints on the integer and fractional parts, respectively given by

$$\varphi_{\mathsf{int}} \stackrel{\text{def}}{=} (c_1 = 0) \qquad \text{and} \qquad \varphi_{\mathsf{frac}} \stackrel{\text{def}}{=} (0 < x_1 < 1).$$

The integer part φ_{int} becomes the guard of the corresponding edge in $\mathcal{C}_{\langle \ell_0,\nu\rangle}$. The fractional part φ_{frac} is incorporated into the DBM M_2, which is defined as

$$M_2 = (M_1 \cap (0 < x_1 < 1))[x_1 \leftarrow 0],$$

where $[\![M_2]\!] = \{\binom{0}{y} : 0 \leqslant y < 0.4\}$. There is a further delay edge in $\mathcal{C}_{\langle \ell_0,\nu\rangle}$ from $\langle \ell_1, M_2\rangle$ to $\langle \ell_1, M_3\rangle$.

There is a wrapping edge from $\langle \ell_1, M_3 \rangle$ to $\langle \ell_1, M_4 \rangle$, where $M_4 = (M_3 \cap (x_2 = 1))[x_2 \leftarrow 0]$. The counter c_2 is incremented along this edge, corresponding to the integer part of clock x_2 increasing by 1 as time progresses.

The remaining states and edges of $\mathcal{C}_{\langle \ell_0, \nu \rangle}$ are illustrated in Figure 3. Note that we only represent states that are relevant to expressing reachability from ℓ_0 to ℓ_1.

An important fact about the collection of counter machines $\mathcal{C}_{\langle \ell, \nu \rangle}$, as $\mathrm{frac}(\nu)$ varies over $[0,1]^{\mathcal{X}}$, is that there are only finitely many such machines up to isomorphism. This essentially follows from Proposition 5, which shows that $closure(M_{\mathrm{frac}(\nu)})$ is determined by the type of $\mathrm{frac}(\nu)$. In the next section we develop this intuition to build a symbolic counter machine that embodies $\mathcal{C}_{\langle \ell, \nu \rangle}$ for all valuations ν of the same type.

C. Parametric Region Automata

Consider a timed automaton $\mathcal{A}$ with n clocks, a location ℓ of $\mathcal{A}$, and an n-type τ. In this section we define a monotonic counter machine $\mathcal{C}_{\langle \ell, \tau \rangle}$ that can be seen as a parametric version of the counter machine $\mathcal{C}_{\langle \ell, \nu \rangle}$ from the previous section, where valuation ν has type τ.

First recall that $\mathcal{M}_\tau = (\prec_{i,j}, m_{i,j})$ is the parametric DBM over $\mathcal{T}_{\mathbb{R}}(\tau)$ such that $\prec_{i,j} = \leqslant$ and $m_{i,j} = [r_i - r_j]$ for $0 \leqslant i, j \leqslant n$.

The construction of the counter machine $\mathcal{C}_{\langle \ell, \tau \rangle}$ is formally very similar to that of $\mathcal{C}_{\langle \ell, \nu \rangle}$. Specifically, the set S of states of $\mathcal{C}_{\langle \ell, \tau \rangle}$ comprises all pairs $\langle \ell', \mathcal{M}' \rangle$ such that $\ell' \in L$ is a location in $\mathcal{A}$ and $\mathcal{M}' \in closure(\mathcal{M}_\tau)$ is a consistent parametric DBM. The set of counters is $C = \{c_1, \ldots, c_n\}$, where n is the number of clocks in $\mathcal{A}$. The transitions of $\mathcal{C}_{\langle \ell, \tau \rangle}$ are defined in a formally identical way to those of $\mathcal{C}_{\langle \ell, \nu \rangle}$; we simply replace operations on concrete DBMs with the corresponding operations on parametric DBMs.

With the above definition, it follows from Proposition 4 that the counter machine $\mathcal{C}_{\langle \ell, \tau \rangle}$ and $\mathcal{C}_{\langle \ell, \nu \rangle}$ are isomorphic via the map sending a control state $\langle \ell, \mathcal{M} \rangle$ of $\mathcal{C}_{\langle \ell, \tau \rangle}$ to the control state $\langle \ell, \nu(\mathcal{M}) \rangle$ of $\mathcal{C}_{\langle \ell, \nu \rangle}$. Proposition 8 then yields:

Theorem 9. *Consider states $\langle \ell, \nu \rangle$ and $\langle \ell', \nu' \rangle$ of a timed automaton $\mathcal{A}$ such that $\mathrm{frac}(\nu)$ has type τ. Then $\langle \ell', \nu' \rangle$ is reachable from $\langle \ell, \nu \rangle$ in $\mathcal{A}$ if and only if there exists some DBM $\mathcal{M}' \in closure(\mathcal{M}_\tau)$ such that the configuration $\langle \langle \ell', \mathcal{M}' \rangle, \lfloor \nu' \rfloor \rangle$ is reachable from $\langle \langle \ell, \mathcal{M}_\tau \rangle, \lfloor \nu \rfloor \rangle$ in the counter machine $\mathcal{C}_{\langle \ell, \tau \rangle}$ and $\mathrm{frac}(\nu') \in [\![\mathrm{frac}(\nu)(\mathcal{M}')]\!]$.*

D. Reachability Formula

We are now in a position to state our main result.

Theorem 10. *Given a timed automaton $\mathcal{A}$ with n clocks and locations ℓ, ℓ', we can compute in exponential time a formula*

$$\varphi_{\ell,\ell'}(z_1, \ldots, z_n, r_1, , \ldots, r_n, z'_1, \ldots, z'_n, r'_1, , \ldots, r'_n)$$

in the existential fragment[1] *of $\mathcal{L}_{\mathbb{R},\mathbb{Z}}$ such that there is a finite run in $\mathcal{A}$ from state $\langle \ell, \nu \rangle$ to state $\langle \ell', \nu' \rangle$ just in case*

$$\langle \lfloor \nu \rfloor, \mathrm{frac}(\nu), \lfloor \nu' \rfloor, \mathrm{frac}(\nu') \rangle \models \varphi_{\ell,\ell'} .$$

Proof. We give the definition of $\varphi_{\ell,\ell'}$ below and justify the complexity bound in Appendix E.

For simplicity we write formula $\varphi_{\ell,\ell'}$ as a disjunction over the collection Tp_n of all n-types. However each disjunct only depends on the restriction of the type τ to the (finite) set of atomic formulas $t \leqslant t'$ with $t, t' \in \mathcal{DT}_{\mathbb{R}}(n)$; so $\varphi_{\ell,\ell'}$ can equivalently be written as a finite disjunction. We define

$$\varphi_{\ell,\ell'} \stackrel{\mathrm{def}}{=} \bigvee_{\tau \in \mathrm{Tp}_n} \alpha^\tau \wedge \chi^\tau_{\ell,\ell'} \tag{4}$$

where the subformulas α^τ and $\chi^\tau_{\ell,\ell'}$ are defined below.

The Hintikka formula $\alpha^\tau(r_1, \ldots, r_n)$[2] is defined by

$$\alpha^\tau \stackrel{\mathrm{def}}{=} \bigwedge_{\substack{t,t' \in \mathcal{DT}_{\mathbb{R}}(n) \\ (t \leqslant t') \in \tau}} t \leqslant t' \wedge \bigwedge_{\substack{t,t' \in \mathcal{DT}_{\mathbb{R}}(n) \\ (t \leqslant t') \notin \tau}} \neg(t \leqslant t') .$$

Given a valuation $\nu \in \mathbb{R}^{\mathcal{X}}_{\geqslant 0}$, $\mathrm{frac}(\nu) \models \alpha^\tau$ just in case the set of difference formulas satisfied by $\mathrm{frac}(\nu)$ is identical to the set of difference formulas in τ.

Formula $\chi^\tau_{\ell,\ell'}$ is defined by writing

$$\chi^\tau_{\ell,\ell'} \stackrel{\mathrm{def}}{=} \bigvee_{\substack{\mathcal{M} \in closure(\mathcal{M}_\tau) \\ \mathcal{M} = (\prec_{i,j}, m_{i,j})}} \Big(\psi_{\langle \ell, \mathcal{M}_\tau \rangle, \langle \ell', \mathcal{M} \rangle}(z_1, \ldots, z_n, z'_1, \ldots, z'_n) \wedge \bigwedge_{0 \leqslant i,j \leqslant n} r'_i - r'_j \prec_{i,j} m_{i,j} \Big) .$$

Here the subformula $\psi_{\langle \ell, \mathcal{M}_\tau \rangle, \langle \ell', \mathcal{M} \rangle}$, expresses the reachability relation in the counter machine $\mathcal{C}_{\langle \ell, \tau \rangle}$ between control states $\langle \ell, \mathcal{M}_\tau \rangle$ and $\langle \ell', \mathcal{M} \rangle$, as per Proposition 7. Recall from Corollary 6 that each $m_{i,j}$ is a difference term involving variables $r_0, \ldots, r_n$. The correctness of $\varphi_{\ell,\ell'}$ is immediate from Proposition 7 and Theorem 9. ☐

Example 7. *Consider the timed automaton $\mathcal{A}$ in Figure 3. Fix the type τ_1 for the valuation $\binom{0.6}{0}$. We illustrate the relevant part of the counter machine $\mathcal{C}_{\langle \ell_0, \tau_1 \rangle}$ in Figure 5. States $\langle \ell, \mathcal{M} \rangle$ of the machine comprise a location ℓ and parametric DBM $\mathcal{M}$. Moreover, $\mathcal{M}_0 = \mathcal{M}_{\tau_1}$. The placement of a transition between $\langle \ell_1, \mathcal{M}_5 \rangle$ and $\langle \ell_1, \mathcal{M}_2 \rangle$ relies on the fact that terms $-r_2$ and 0 are equivalent with respect to the equivalence relation on terms induced by τ_1.*

Let α^{τ_1} be the Hintikka formula of the type τ_1. Clearly, $\langle 0.6, 0 \rangle \models \alpha^{\tau_1}$. We define $\chi^\tau_{\ell_0,\ell_1}$ as follows:

$$\chi^{\tau_1}_{\ell_0,\ell_1} \stackrel{\mathrm{def}}{=} (z_1 = 0) \wedge \Big[\big[(z'_2 - z'_1 = z_2 - z_1) \wedge (\psi_2 \vee \psi_3) \big] \vee \big[(z'_2 - z'_1 = -1 + z_2 - z_1) \wedge (\psi_4 \vee \psi_5) \big] \Big],$$

[1] We claim that this result can be strengthened to state that the reachability relation can be expressed by a *quantifier-free* formula, again computable in exponential time. To do this one can exploit structural properties of the class of monotonic counter machine that arise from timed automata. We omit details.

[2] Recall that by convention $[r_0] = [0]$, thus we treat variable r_0 as synonymous with the constant 0.

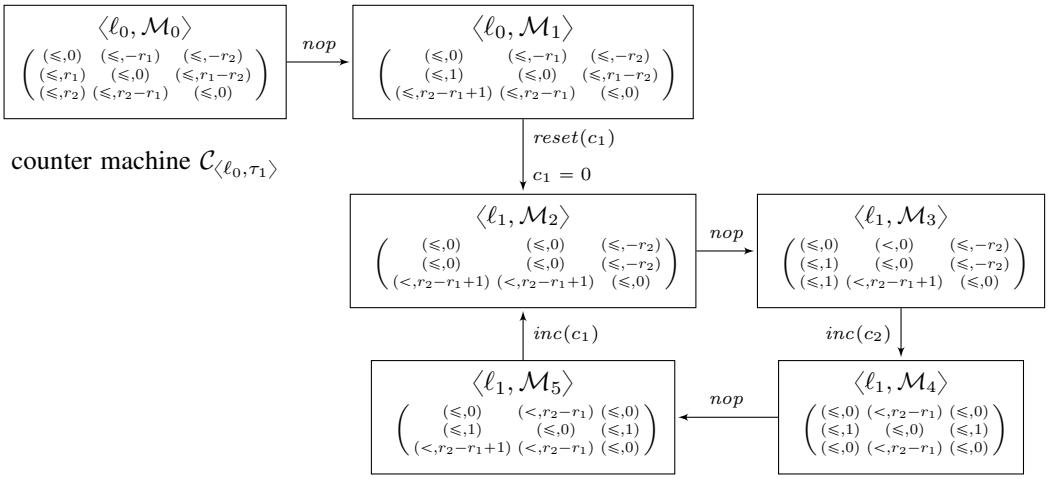


Fig. 5. The (relevant part of the) counter machine $\mathcal{C}_{\langle \ell, \tau_1 \rangle}$ constructed from the timed automaton in Figure 3, where τ_1 is the type of the valuation ν with $\nu_1 = 0.6$ and $\nu_2 = 0$. The placement of a transition between $\langle \ell_1, \mathcal{M}_5 \rangle$ and $\langle \ell_1, \mathcal{M}_2 \rangle$ relies on the fact that terms $-r_2$ and 0 are equivalent under the preorder induced by τ_1.

where ψ_1, ψ_2, ψ_3 and ψ_4 are given in the following:

$$\psi_2 \equiv (r_1' = 0) \wedge (r_2 \leqslant r_2' < r_2 - r_1 + 1),$$

$$\psi_3 \equiv (0 < r_1') \wedge (r_2 \leqslant r_2') \wedge (r_2 \leqslant r_2' - r_1' < r_2 - r_1 + 1),$$

$$\psi_4 \equiv (r_2 - r_1 < r_1') \wedge (r_2' = 0),$$

$$\psi_5 \equiv (r_2 - r_1 < r_1') \wedge (r_2' < r_2 - r_1 + 1) \wedge (-1 \leqslant r_2' - r_1' < r_2 - r_1).$$

The formulae ψ_i (with $i \in \{2,3,4,5\}$) summarise the constraints placed on r_1' and r_2' by the parametric DBMs $\mathcal{M}_i$ in the counter machine $\mathcal{C}_{\langle \ell_0, \tau_1 \rangle}$. See Figure 5 for the given constraints in the parametric DBMs $\mathcal{M}_i$. Recall that real-valued variables r_i, r_i' range over the interval $[0,1]$.

Let τ_2 be the type for the valuation $\binom{0}{0.2}$. In comparison with $\mathcal{C}_{\langle \ell_0, \tau_1 \rangle}$, we present the counter machine $\mathcal{C}_{\langle \ell_0, \tau_2 \rangle}$ in Figure 6 in Appendix F.

The formula φ_{ℓ_0, ℓ_1}, expressing the set of valuations ν and ν' such that $\langle \ell_1, \nu' \rangle$ is reachable from $\langle \ell_0, \nu \rangle$, is then the disjunction of all formulas $\alpha^\tau \wedge \chi^\tau_{\ell_0, \ell_1}$ for types $\tau \in \mathrm{Tp}_n$:

$$\varphi_{\ell_0, \ell_1} = (\alpha^{\tau_1} \wedge \chi^{\tau_1}_{\ell_0, \ell_1}) \vee (\alpha^{\tau_2} \wedge \chi^{\tau_2}_{\ell_0, \ell_1}) \vee \cdots .$$

V. Parametric Timed Reachability Logic

Let $\mathcal{A} = \langle L, \mathcal{X}, E \rangle$ be a timed automaton augmented with a labelling function $LB : L \to 2^{AP}$. Let φ be a sentence of PTRL. Recall that the model checking problem of $\mathcal{A}$ against φ asks, given a state $\langle \ell, \nu \rangle$ of $\mathcal{A}$, whether $\langle \ell, \nu \rangle \models \varphi$.

In this section we prove the following result.

Theorem 11. *The model-checking problem for PTRL is decidable in EXPSPACE and is NEXPTIME-hard.*

For membership in EXPSPACE, given a timed automaton $\mathcal{A}$, a configuration $\langle \ell, \nu \rangle$ of $\mathcal{A}$, and a sentence ψ of PTRL, we construct in exponential time a sentence $\widetilde{\psi}$ of $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$ that is true if and only if $\langle \ell, \nu \rangle \models \psi$. We thereby obtain an exponential space algorithm for the model checking problem. We then prove NEXPTIME-hardness by a reduction from SUCCINCT 3-SAT.

A. Reduction of Model Checking to Satisfiability

The model checking procedure for PTRL relies on a "cut-down" version of Theorem 10, concerning the logical definability of the reachability relation. In this version, given as Lemma 12 below, we do not represent the full reachability relation, but instead abstract the integer parts of all clocks except the reference clock x_n. This abstraction is sufficient for model-checking PTRL, and moreover allows us to obtain a formula that lies in the sub-logic $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$, which has better complexity bounds than the full logic $\mathcal{L}_{\mathbb{R},\mathbb{Z}}$.

Given $N \in \mathbb{N}$, define the set $\mathcal{R}_N$ of *regions* to be $\mathcal{R}_N = \{0, \ldots, N\} \cup \{\infty\}$. A counter valuation $\upsilon \in \mathbb{N}^n$ is abstracted to $Reg(\upsilon) \in \mathcal{R}_N^n$, where

$$Reg(\upsilon)_i = \begin{cases} \upsilon_i & \text{if } \upsilon_i \leqslant N \\ \infty & \text{otherwise} \end{cases}$$

The following lemma is proved in Appendix C.

Lemma 12. *Let $\mathcal{A}$ be a timed automaton with n clocks and maximum clock constant N. Given two locations ℓ, ℓ' of $\mathcal{A}$ and $R, R' \in \mathcal{R}_N^n$, we can compute in exponential time a quantifier-free $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$-formula*

$$\varphi_{\ell,R,\ell',R'}(z, r_1, \ldots, r_n, z', r_1', \ldots, r_n')$$

such that there is a finite run in $\mathcal{A}$ from state $\langle \ell, \nu \rangle$ to state $\langle \ell', \nu' \rangle$, where $Reg(\lfloor \nu \rfloor) = R$ and $Reg(\lfloor \nu' \rfloor) = R'$, just in case

$$\langle \lfloor \nu_n \rfloor, \mathrm{frac}(\nu), \lfloor \nu_n' \rfloor, \mathrm{frac}(\nu') \rangle \models \varphi_{\ell,R,\ell',R'} .$$

Let ψ be a formula of PTRL of the first type, involving the set of parameters $\theta_1, \ldots, \theta_k$, and let $\mathcal{A}$ be a timed automaton with n clocks and maximum clock constant N. For each

location ℓ of $\mathcal{A}$ and $R \in \mathcal{R}_N^n$ such that $R_n = 0$, we obtain a $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$-formula

$$\widetilde{\psi}_{\ell,R}(r_1, \ldots, r_n, w_1, \ldots, w_k, s_1, \ldots, s_k)$$

in real variables $\boldsymbol{r} = (r_1, \ldots, r_n)$ and $\boldsymbol{s} = (s_1, \ldots, s_k)$ and integer variables $\boldsymbol{w} = (w_1, \ldots, w_k)$ such that

$$\langle \mathrm{frac}(\nu), \lfloor \xi \rfloor, \mathrm{frac}(\xi) \rangle \models \widetilde{\psi}_{\ell,R} \quad \text{iff} \quad \langle \ell, \nu \rangle \models_\xi \psi$$

for all parameter valuations $\xi \in \mathbb{R}^k_{\geqslant 0}$ and all clock valuations $\nu \in \mathbb{R}^n_{\geqslant 0}$ such that $Reg(\lfloor \nu \rfloor) = R$ and $\nu_n = 0$.

To keep things simple, we assume that every configuration of $\mathcal{A}$ can generate an infinite non-zeno run. It is not difficult to drop this assumption since the collection of configurations from which there exists such a run is a union of clock regions and hence is definable in $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$. We also assume, without loss of generality, that the reference clock x_n is not mentioned in any guard of $\mathcal{A}$.

The construction of $\widetilde{\psi}_{\ell,R}$ is by induction on the structure of ψ. The induction cases for the Boolean connectives are straightforward and we concentrate on the induction step for the connective $\exists \Diamond_{\sim\theta}$. In fact we only consider the case that $\sim$ is the equality relation $=$, the cases for $<$ and $>$ being very similar.

Suppose that $\psi \equiv \exists \Diamond_{=\theta_i} \psi'$ for some PTRL-formula ψ' and $i \in \{1, \ldots, k\}$. Then we define

$$\widetilde{\psi}_{\ell,R}(\boldsymbol{r}, \boldsymbol{w}, \boldsymbol{s}) \stackrel{\text{def}}{=} \bigvee_{\ell', R'} \exists \boldsymbol{r}' \exists z' \, \varphi_{\ell,R,\ell',R'}(0, \boldsymbol{r}, z', \boldsymbol{r}') \\ \wedge (r'_n = s_i \wedge z' = w_i) \wedge \widetilde{\psi}'_{\ell',R'}(r'_1 \ldots, r'_{n-1}, 0, \boldsymbol{w}, \boldsymbol{s})$$

where $\varphi_{\ell,R,\ell',R'}$ is the reachability formula defined in Lemma 12. Note that this definition relies on the assumption that the clock x_n is never reset by the timed automaton and hence can be used to keep track of global time.

This completes the translation of PTRL-formulas of the first type to formulas of $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$. Extending this inductive translation to PTRL-formulas of the second type is straightforward, bearing in mind that we represent each parameter θ_i by a variable w_i for its integer part and a variable s_i for its fractional part. Thus, e.g., the PTRL-formula $\exists \theta_i \psi$ is translated as $\exists w_i \exists s_i (0 \leqslant s_i < 1 \wedge \widetilde{\psi})$.

Given a sentence ψ of PTRL, location ℓ of $\mathcal{A}$, and $R \in \mathcal{R}_N$, our translation yields a formula $\widetilde{\psi}_{\ell,R}(r_1, \ldots, r_n)$ such that for any valuation ν with $Reg(\lfloor \nu \rfloor) = R$ we have $\langle \ell, \nu \rangle \models \psi$ if and only if $\mathrm{frac}(\nu) \models \widetilde{\psi}_{\ell,R}$. By Lemma 12, formula $\widetilde{\psi}_{\ell,R}$ has size singly exponential in the size of ψ and $\mathcal{A}$ and quantifier-depth linear in the size of ψ.

The model checking problem then reduces to determining the truth of $\widetilde{\psi}_{\ell,R}$ on $\mathrm{frac}(\nu)$, where $Reg(\lfloor \nu \rfloor) = R$. Since satisfiability for sentences of $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$ can be decided in polynomial space in the formula size and exponential space in the number of quantifiers (by Proposition 2), the model checking problem of PTRL lies in EXPSPACE.

B. NEXPTIME-Hardness

In this section we show that model checking timed automata against the fixed PTRL sentence $\exists \theta \, \forall \Box_{=\theta} \, p$ is NEXPTIME-hard. We remark that, due to the punctual constraint $=\theta$, the above formula expresses a synchronization property—*there exists a duration θ such that all runs are in a p-state after time exactly θ.*

Recall that a *Boolean circuit* is a finite directed acyclic graph, whose nodes are called *gates*. An *input gate* is a node with indegree 0. All other gates have label either $\vee$, $\wedge$, or $\neg$. An *output gate* is a node with outdegree 0.

We show NEXPTIME-hardness by reduction from the SUCCINCT 3-SAT problem. The input of SUCCINCT 3-SAT is a Boolean circuit C, representing a 3-CNF formula φ_C, and the output is whether or not φ_C is satisfiable. Specifically, C has 2 output gates, and the input gates are partitioned into two nonempty sets of respective cardinalities n and m. The formula φ_C has 2^n variables and 2^m clauses (in particular, the number of variables and clauses in φ_C can be exponential in the size of C). The first n inputs of C represent the binary encoding of the index i of a variable, and the remaining m inputs of C represent the binary encoding of the index j of a clause in φ_C. The output of C indicates whether the i-th variable occurs positively, negatively, or not at all in the j-th clause of φ_C. The SUCCINCT 3-SAT problem is NEXPTIME-complete [21].

Given an instance of SUCCINCT 3-SAT, that is, a Boolean circuit C as described above, we construct a timed automaton $\mathcal{A}$ augmented with a labelling function LB such that the 3-CNF formula φ_C encoded by circuit C is satisfiable if and only if $(\ell, \mathbf{0}) \models \exists \theta \forall \, \Box_{=\theta} \, p$ for some designated location ℓ.

There are two ideas behind the reduction. First we construct a linear bounded automaton $\mathcal{B}$ from the circuit C such that, roughly speaking, the 3-CNF formula φ_C is satisfiable if and only if there exists an integer N such that, starting from an initial configuration, all length-N paths in the configuration graph of $\mathcal{B}$ end in a configuration with label p. The second part of the reduction is to simulate encode the configuration graph of $\mathcal{B}$ as the configuration graph of a timed automaton $\mathcal{A}$.

We construct $\mathcal{B}$ such that its number of control states is polynomial in the size of C, and we fix an initial tape configuration of $\mathcal{B}$ of length likewise bounded by a polynomial in the size of C. We designate certain transitions of $\mathcal{B}$ as ✓-transitions. In every computation of $\mathcal{B}$, the sequence of steps between the i-th and $(i+1)$-st ✓-transitions, for $i \in \mathbb{N}$, is referred to as the *i-th phase* of the computation. We design $\mathcal{B}$ so that the number of steps in the i-th phase is independent of the nondeterministic choices along the run.

The definition of $\mathcal{B}$ is predicated on a numerical encoding of propositional valuations. Suppose that $X_1, \ldots, X_{2^n}$ are the variables occurring in φ_C, and write $p_1, \ldots, p_{2^n}$ for the first 2^n prime numbers in increasing order. Given a positive integer N, we obtain a Boolean valuation of $X_1, \ldots, X_{2^n}$ in which X_j is false if, and only if, $N \bmod p_j = 0$. With this encoding in hand, we proceed to define $\mathcal{B}$:

1) In the first phase, $\mathcal{B}$ guesses three n-bit numbers $1 \leqslant i_1, i_2, i_3 \leqslant 2^n$ and a single m-bit number $1 \leqslant j \leqslant 2^m$ and writes them on its tape.
2) In the second phase, $\mathcal{B}$ computes the three prime numbers $p_{i_1}, p_{i_2}, p_{i_3}$ and writes them on its tape.
3) In the third phase, by simulating the circuit C, $\mathcal{B}$ determines whether the propositional variables $X_{i_1}, X_{i_2}, X_{i_3}$ appear in the j-th clause of φ_C, henceforth denoted ψ_j. If one of them does not appear at all, then $\mathcal{B}$ moves into an accepting self-loop. Otherwise, $\mathcal{B}$ remembers in its state whether $X_{i_1}, X_{i_2}, X_{i_3}$ appear positively or negatively in ψ_j, and then $\mathcal{B}$ proceeds to the next phase.
4) From phase four onwards, $\mathcal{B}$ maintains on its tape three counters, respectively counting modulo $p_{i_1}, p_{i_2}, p_{i_3}$. In every successive phase, each of these counters is incremented by one. At the end of each phase, $\mathcal{B}$ checks whether the values of the counters encode a satisfying valuation of clause ψ_j. If this is the case, then $\mathcal{B}$ moves into an accepting state. Otherwise $\mathcal{B}$ proceeds to the next phase.

By construction, $N \in \mathbb{N}$ encodes a satisfying valuation of φ_C if and only if all computation paths of $\mathcal{B}$ reach an accepting state at the end of the $(N+3)$-rd phase.

It remains to explain how from $\mathcal{B}$ one can define a timed automaton $\mathcal{A}$ whose configuration graph embeds the configuration graph of $\mathcal{B}$. The construction is adapted from the PSPACE-hardness proof for reachability in timed automata [1]. We refer to Appendix G for details of this construction. In the end, the initial configuration $(\ell, \mathbf{0})$ of $\mathcal{A}$ satisfies $\exists \theta', \forall \, \Box_{=\theta} \, p$ if and only if φ_C is satisfiable.

VI. Conclusion

We have given a new proof of the result of Comon and Jurski that the reachability relation of a timed automaton is definable in linear arithmetic. In addition to making the result more accessible, our main motivations in revisiting this result concerned potential applications and generalisations. With regard to applications, we have already put the new proof to work in deriving complexity bounds for model checking the reachability fragment of parametric TCTL. In future work we would like to see whether ideas from this paper can be applied to give a more fine-grained analysis of extensions of timed automata, such as timed games and priced timed automata.

We claim that a finer analysis of the complexity of our decision procedure for model checking PTRL yields membership of the problem in the complexity class $\mathrm{STA}(*, 2^{O(n)}, n)$, i.e., the class of languages accepted by alternating Turing machines running in time $2^{O(n)}$ and making at most n alternations on an input of length n. This improved upper bound follows from a refinement of the statement of Proposition 2, on the complexity of the decision problem for $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$, to state that the truth of a prenex-form sentences of size n and with k quantifier alternations can decided by a polynomial time alternating Turing machine, making at most k alternations.

We claim also that our NEXPTIME-hardness result can be strengthened to match the new upper bound. The idea here would be to reduce a version of SUCCINCT 3-SAT with quantifier alternation to model checking $PTRL$ formulas of the form $Q_1\theta_1 \ldots Q_k\theta_k \forall \, \Box_{=\theta_1} \ldots \forall \, \Box_{=\theta_k} \, p$ for $Q_1, \ldots, Q_k$ a sequence of quantifiers with k alternations.

Details of the improved upper and lower complexity bound will appear in a subsequent version of this paper.

Acknowledgements. This work was partially supported by the EPSRC through grants EP/M012298/1 and EP/M003795/1 and by the German Research Foundation (DFG), project QU 316/1-2.

References

[1] R. Alur and D. L. Dill, "A theory of timed automata," *Theor. Comput. Sci.*, vol. 126, no. 2, pp. 183–235, 1994. [Online]. Available: http://dx.doi.org/10.1016/0304-3975(94)90010-8

[2] H. Comon and Y. Jurski, "Timed automata and the theory of real numbers," in *CONCUR '99: Concurrency Theory, 10th International Conference, Eindhoven, The Netherlands, August 24-27, 1999, Proceedings*, ser. Lecture Notes in Computer Science, vol. 1664. Springer, 1999, pp. 242–257.

[3] ——, "Timed automata and the theory of real numbers," LSV, ENS Cachan, Tech. Rep. LSV-99-6, July 1999.

[4] J. Ferrante and C. Rackoff, "A decision procedure for the first order theory of real addition with order," *SIAM J. Comput.*, vol. 4, no. 1, pp. 69–76, 1975. [Online]. Available: http://dx.doi.org/10.1137/0204006

[5] A. W. To, "Model checking FO(R) over one-counter processes and beyond," in *Computer Science Logic, 23rd international Workshop, CSL, Proceedings*, ser. Lecture Notes in Computer Science, vol. 5771. Springer, 2009, pp. 485–499.

[6] Z. Dang, "Pushdown timed automata: a binary reachability characterization and safety verification," *Theor. Comput. Sci.*, vol. 302, no. 1-3, pp. 93–121, 2003.

[7] C. Dima, "Computing reachability relations in timed automata," in *17th IEEE Symposium on Logic in Computer Science (LICS 2002), Proceedings*. IEEE Computer Society, 2002, p. 177.

[8] V. Bruyère, E. Dall'Olio, and J. Raskin, "Durations and parametric model-checking in timed automata," *ACM Trans. Comput. Log.*, vol. 9, no. 2, 2008. [Online]. Available: http://doi.acm.org/10.1145/1342991.1342996

[9] ——, "Durations, parametric model-checking in timed automata with presburger arithmetic," in *STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany, February 27 - March 1, 2003, Proceedings*, ser. Lecture Notes in Computer Science, vol. 2607. Springer, 2003, pp. 687–698.

[10] A. Annichini, E. Asarin, and A. Bouajjani, "Symbolic techniques for parametric reasoning about counter and clock systems," in *Computer Aided Verification, 12th International Conference, CAV 2000, Proceedings*, ser. Lecture Notes in Computer Science, vol. 1855. Springer, 2000, pp. 419–434.

[11] T. Hune, J. Romijn, M. Stoelinga, and F. W. Vaandrager, "Linear parametric model checking of timed automata," *J. Log. Algebr. Program.*, vol. 52-53, pp. 183–220, 2002.

[12] B. Bérard, A. Petit, V. Diekert, and P. Gastin, "Characterization of the expressive power of silent transitions in timed automata," *Fundam. Inform.*, vol. 36, no. 2-3, pp. 145–182, 1998.

[13] L. Berman, "The complexitiy of logical theories," *Theor. Comput. Sci.*, vol. 11, pp. 71–77, 1980. [Online]. Available: http://dx.doi.org/10.1016/0304-3975(80)90037-7

[14] E. D. Sontag, "Real addition and the polynomial hierarchy," *INFORMATION PROCESSING LETTERS*, vol. 20, pp. 115–120, 1985.

[15] V. Weispfenning, "The complexity of linear problems in fields," *J. Symb. Comput.*, vol. 5, no. 1-2, pp. 3–27, Feb. 1988.

[16] R. Alur, C. Courcoubetis, and D. L. Dill, "Model-checking in dense real-time," *Inf. Comput.*, vol. 104, no. 1, pp. 2–34, 1993. [Online]. Available: http://dx.doi.org/10.1006/inco.1993.1024

[17] E. M. Clarke, Jr., O. Grumberg, and D. A. Peled, *Model Checking*. Cambridge, MA, USA: MIT Press, 1999.

[18] P. Bouyer, "Untameable timed automata!" in *STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany, February 27 - March 1, 2003, Proceedings*, ser. Lecture Notes in Computer Science, vol. 2607. Springer, 2003, pp. 620–631.

[19] A. Finkel and A. Sangnier, "Reversal-bounded counter machines revisited," in *Mathematical Foundations of Computer Science 2008, 33rd International Symposium, MFCS 2008, Proceedings*, ser. Lecture Notes in Computer Science, vol. 5162. Springer, 2008, pp. 323–334.

[20] J. Bengtsson and W. Yi, "Timed automata: Semantics, algorithms and tools," ser. Lecture Notes in Computer Science, vol. 3098. Springer, 2003, pp. 87–124.

[21] C. H. Papadimitriou and M. Yannakakis, "A note on succinct representations of graphs," *Information and Control*, vol. 71, no. 3, pp. 181–185, 1986.

[22] D. C. Kozen, *Automata and Computability*, 1st ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.

[23] M. Chrobak, "Finite automata and unary languages," *Theor. Comput. Sci.*, vol. 47, no. 3, pp. 149–158, 1986. [Online]. Available: http://dx.doi.org/10.1016/0304-3975(86)90142-8

[24] A. Martinez, "Efficient computation of regular expressions from unary nfas," in *Fourth International Workshop on Descriptional Complexity of Formal Systems - DCFS 2002*, vol. Report No. 586. Department of Computer Science, The University of Western Ontario, Canada, 2002, pp. 174–187.

APPENDIX

A. Proof of Proposition 2

We first recall that the language $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$ has terms of both real-number sort and integer sort, where the atomic formulas:

- if integer sort, have form

$$z - z' \leqslant c \mid z \leqslant c \mid z - z' \equiv c \pmod{d} \tag{5}$$

  for integer variables z, z' and integers c, d.
- if real-number sort, have form $t \leqslant t'$ where t, t' are derived by the grammar

$$t ::= c \mid r \mid t + t \mid t - t\,,$$

  where $c \in \mathbb{Q}$ is a constant and $r \in \{r_0, r_1, \ldots\}$ is a real-valued variable.

One can prove Proposition 2 by combining the quantifier-elimination procedures of Ferante and Rackoff [4], [22] for $\mathcal{L}_{\mathbb{R}}$ and To [5, Section 4] for the fragment of Presburger arithmetic in which atomic formulas have the form shown in (5).

To eliminate quantifiers in formulas of real arithmetic, Ferante and Rackoff [4] define an equivalence relation $\mathcal{R}^k_m$ on k-tuples of real numbers. The relation is such that $\mathcal{R}^k_m$-equivalent k-tuples agree on all quantifier-free formulas in which all constants have the largest (absolute) constant at most m. We refer the reader to [22] for the definition of $\mathcal{R}^k_m$; here we just recall the key results.

Let A^k_m be the set of all affine functions $f : \mathbb{R}^k \to \mathbb{R}$ with integer coefficients, where all constants and coefficients have the largest (absolute) constant at most m.

Lemma 13 (Lemma 22.3 and 22.4 from [22])**.** *Given two k-tuples $\boldsymbol{a} = (a_1, \cdots, a_k)$ and $\boldsymbol{b} = (b_1, \cdots, b_k)$ of real numbers such that $\boldsymbol{a}\ \mathcal{R}^k_{2m^2}\ \boldsymbol{b}$ for some $m \in \mathbb{Z}_{>0}$, then for all $c \in \mathbb{R}$ there exists $d \in \mathbb{R}$ such that $(\boldsymbol{a}, c)\ \mathcal{R}^{k+1}_m\ (\boldsymbol{b}, d)$. Moreover, d can be chosen to have the form $f(\boldsymbol{b})/e$ where $f \in A^k_{2m^2}$ and $|e| \leqslant 2m^2$.*

To eliminate quantifiers in formulas of the above fragment of Presburger arithmetic, analogue to the relation $\mathcal{R}^k_m$, To [5, Definition 6] has defined an equivalence relation $\mathcal{Z}^k_{p,m}$ on k-tuples of integers, where $p, m \in \mathbb{Z}_{>0}$. The relation is such that $\mathcal{Z}^k_{p,m}$-equivalent k-tuples agree on all quantifier-free formulas, where all constants have the largest (absolute) constant at most m and the period of the formula is p. The period of the formula is the least common multiple of the periods e of each atomic term $z - z' \equiv c \pmod{e}$. We refer the reader to [5] for the definition of $\mathcal{Z}^k_{p,m}$; here we just recall the key results.

Lemma 14 (Lemma 7 and 8 from [5])**.** *Given two $(k+1)$-tuples $\boldsymbol{a} = (a_0, \cdots, a_k)$ and $\boldsymbol{b} = (b_0, \cdots, b_k)$ of integers such that $a_0 = b_0 = 0$ and $\boldsymbol{a}\ \mathcal{Z}^k_{p,3m}\ \boldsymbol{b}$ for some $p, m > 0$, then for all $c \in \mathbb{N}$ there exists $b \in \mathbb{N}$ such that $(\boldsymbol{a}, c)\ \mathcal{Z}^{k+1}_{p,m}\ (\boldsymbol{b}, d)$. Moreover, d can be chosen such that $0 \leqslant d \leqslant max(b_0, \cdots, b_k) + pm + p$.*

Fix $m \in \mathbb{Z}_{>0}$. For all $n \in \mathbb{N}$, define $g(0, m) \stackrel{\text{def}}{=} m$ and $g(n+1, m) \stackrel{\text{def}}{=} 2g(n, m)^2$, moreover, define $h(0, m) \stackrel{\text{def}}{=} m$ and $h(n+1, m) \stackrel{\text{def}}{=} 3h(n, m)$.

Lemma 15. *Let $\varphi(r_1, \cdots, r_k, z_0, \cdots, z_{k'})$ be a formula in $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$, with k free real-valued variables r_i and k' free integer variables z_i, and with n quantifiers over real-valued variables and n' quantifiers over integer-valued variables, where m is the largest (absolute) constant and p is the period of the formula. Suppose $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^k$ are k-tuples of real numbers such that $\boldsymbol{a}\ \mathcal{R}^k_{g(n,m)}\ \boldsymbol{b}$. Suppose $\boldsymbol{a}', \boldsymbol{b}' \in \mathbb{N}^{k'}$ are k'-tuples of integers such that $\boldsymbol{a}'\ \mathcal{Z}^{k'}_{p,h(n',m)}\ \boldsymbol{b}'$. Then, we have*

$$\varphi(\boldsymbol{a}, \boldsymbol{a}') \text{ holds} \Leftrightarrow \varphi(\boldsymbol{b}, \boldsymbol{b}') \text{ holds.}$$

Proof. The proof is by an induction on the structure of the formula. For atomic formulas, each sort, the result is immediate from the definition of equivalence relations $\mathcal{R}^k_m$ and $\mathcal{Z}^k_{p,m}$. For the Boolean connectives, the result is straightforward using the induction hypothesis for each subformula.

- For formulas $\exists r\, \varphi(\boldsymbol{a}, \boldsymbol{a}', r)$, where r is a real-valued variable: suppose $\exists r\, \varphi(\boldsymbol{a}, \boldsymbol{a}', r)$ holds, and let $c \in R$ be such that $\varphi(\boldsymbol{a}, \boldsymbol{a}', c)$ holds. Since $\boldsymbol{a}\ \mathcal{R}^k_{g(n,m)}\ \boldsymbol{b}$, by Lemma 13, there exists some d such that $(\boldsymbol{a}, c)\ \mathcal{R}^{k+1}_{g(n-1,m)}\ (\boldsymbol{b}, d)$. Applying the induction hypothesis, $\exists r\, \varphi(\boldsymbol{b}, \boldsymbol{b}', r)$ holds, too.
- For formulas $\exists z\, \varphi(\boldsymbol{a}, \boldsymbol{a}', z)$, where z is a integer-valued variable: suppose $\exists z\, \varphi(\boldsymbol{a}, \boldsymbol{a}', z)$ holds, and let $c' \in R$ be such that $\varphi(\boldsymbol{a}, \boldsymbol{a}', c')$ holds. Since $\boldsymbol{a}'\ \mathcal{Z}^k_{p,h(n',m)}\ \boldsymbol{b}'$, by Lemma 13, there exists some d' such that $(\boldsymbol{a}', c')\ \mathcal{Z}^{k+1}_{p,h(n'-1,m)}\ (\boldsymbol{b}', d')$. Applying the induction hypothesis, $\exists z\, \varphi(\boldsymbol{b}, b', z)$ holds, too.

The step of induction for formulas $\forall r\, \varphi(\boldsymbol{a}, a', r)$ and $\forall z\, \varphi(\boldsymbol{a}, a', z)$ are similar. □

Given a prenex-form sentence φ of $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$, using Lemma 15 we derive an equivalent formula in which all quantifiers range over finite domains. Specifically, if φ has n quantifiers over real variables and n' quantifiers over integer variables, maximum constant m, and period p, then the real-valued quantifiers can be restricted to range over rationals whose numerator and

denominator is at most $g(n,m) = 2^{2^n-1}m^{2^n} = 2^{2^{O(n+\log\log m)}}$ and the integer quantifiers can be restricted to range over numbers of the largest (absolute) constant at most $p(n'+1)h(n',m)+p(n'+1) = p(n'+1)3^{n'}(m+1) = 2^{O(n'+\log m+\log p)}$. Thus the truth of φ can be established by an alternating Turing machine using space exponential in $n+n'$ and polynomial in the size of the quantifier-free part of φ. This concludes the proof of Proposition 2.

B. Proof of Lemma 3

In this section we prove Lemma 3 from Subsection III-B.

Recall that DBMs have entries in $\mathbb{V} = (\{<,\leqslant\}\times\mathbb{R})\cup\{(<,\infty)\}$. In this section we denote the order $\leqslant_{\mathbb{V}}$ simply by $\leqslant$ (and the corresponding strict order by $<$). Recall that a DBM is *atomic* if all but at most one entry is the trivial constraint $(<,\infty)$. Recall also that DBM M is *consistent* if $(\leqslant,0)\leqslant M_{i,i}$ for all $0\leqslant i\leqslant n$. Write $\mathbb{Z}_\infty$ for $\mathbb{Z}\cup\{\infty\}$.

1) Tightness: In order to prove Lemma 3, we first introduce the concept of *tightness* for DBMs and prove that, for a clock valuation $\nu\in[0,1]^n$, every DBM in $closure(M_\nu)$ is tight.

Let M be a DBM of dimension $(n+1)\times(n+1)$. We say that M is *tight* if $M_{i,j} = M_{i,n}+M_{n,j}$ for every pair of indices i,j with $m_{i,j}\notin\mathbb{Z}_\infty$.

Proposition 16. *If M is tight, then $\overrightarrow{M}$ is tight.*

Proof. Write $M' = \overrightarrow{M}$ and assume that $m'_{i,j}\notin\mathbb{Z}_\infty$ for some $0\leqslant i,j\leqslant n$. We show that $M'_{i,j} = M'_{i,n}+M'_{n,j}$. Indeed, since $m'_{i,j}\notin\mathbb{Z}_\infty$ we have $j\neq 0$ and thus

$$\begin{array}{rcll} M'_{i,j} & = & M_{i,j} & \\ & = & M_{i,n}+M_{n,j} & (M \text{ is tight}) \\ & = & M'_{i,n}+M'_{n,j} & (\text{since } n,j\neq 0). \end{array}$$

□

Proposition 17. *Suppose that M is tight and M' is atomic. Then $M'' = M\cap M'$ is tight.*

Proof. Suppose that $m''_{i,j}\notin\mathbb{Z}_\infty$. We show that $M''_{i,j} = M''_{i,n}+M''_{n,j}$. There are two main cases. First suppose that $M''_{i,j} = M_{i,j}$. Then

$$\begin{array}{rcll} M''_{i,j} & \leqslant & M''_{i,n}+M''_{n,j} & (M'' \text{ canonical}) \\ & \leqslant & M_{i,n}+M_{n,j} & (M''\leqslant M \text{ pointwise}) \\ & = & M_{i,j} & (m_{i,j}\notin\mathbb{Z}_\infty,\ M \text{ tight}). \end{array}$$

Since we assume that $M''_{i,j} = M_{i,j}$, all the inequalities above are tight and we conclude that $M''_{i,j} = M''_{i,n}+M''_{n,j}$.

The second case is that $M''_{i,j} < M_{i,j}$. Then by definition of M'' we have $M''_{i,j} = M_{i,p}+M'_{p,q}+M_{q,j}$. Since $m''_{i,j}\notin\mathbb{Z}_\infty$, we must have either $m_{i,p}\notin\mathbb{Z}_\infty$ or $m_{q,j}\notin\mathbb{Z}_\infty$. We will handle the first of these two subcases; the second follows by symmetric reasoning.

If $m_{i,p}\notin\mathbb{Z}_\infty$ then

$$\begin{array}{rcll} M''_{i,j} & = & M_{i,p}+M'_{p,q}+M_{q,j} & (\text{definition of } M'') \\ & = & M_{i,n}+M_{n,p}+M'_{p,q}+M_{q,j} & (m_{i,p}\notin\mathbb{Z}_\infty,\ M \text{ tight}) \\ & \geqslant & M''_{i,n}+M''_{n,p}+M''_{p,q}+M''_{q,j} & (M''\leqslant M, M' \text{ pointwise}) \\ & \geqslant & M''_{i,n}+M''_{n,j} & (M'' \text{ canonical}) \end{array}$$

But $M''_{i,j}\leqslant M''_{i,n}+M''_{n,j}$ by canonicity of M''. Hence $M''_{i,j} = M''_{i,n}+M''_{n,j}$. □

Proposition 18. *Suppose that M is tight.*

1) *If $\ell\neq n$ then $M[x_\ell\leftarrow 0]$ is tight.*
2) *$(M\cap(x_n=1))[x_n\leftarrow 0]$ is tight.*

Proof. 1) Write $M' = M[x_\ell\leftarrow 0]$, where $\ell\neq n$, and assume that $m'_{i,j}\notin\mathbb{Z}_\infty$. We show that $M'_{i,j} = M'_{i,n}+M'_{n,j}$. Indeed we have

$$\begin{array}{rcll} M'_{i,j} & = & M_{i_\ell,j_\ell} & (\text{definition of } M') \\ & = & M_{i_\ell,n}+M_{n,j_\ell} & (M \text{ is tight, } m_{i_\ell,j_\ell} = m'_{i,j}\notin\mathbb{Z}_\infty) \\ & = & M_{i_\ell,n_\ell}+M_{n_\ell,j_\ell} & (n_\ell = n) \\ & = & M'_{i,n}+M'_{n,j} & (\text{definition of } M'). \end{array}$$

2) Write $M' = M \cap (x_n = 1)$. We know from Proposition 17 that M' is tight. Moreover we have $M'_{n,0} = (\leqslant, 1)$ and $M'_{0,n} = (\leqslant, -1)$. Now write $M'' = M'[x_n \leftarrow 0]$ and assume that $m''_{i,j} \notin \mathbb{Z}_\infty$. We show that $M''_{i,j} = M''_{i,n} + M''_{n,j}$. The equality is trivial if $i = n$ or $j = n$, so we may suppose that $i, j \neq n$.
Then we have

$$\begin{aligned} M''_{i,j} &= M'_{i,j} \quad \text{(definition of } M'' \text{ and } i,j \neq n) \\ &= M'_{i,n} + M'_{n,j} \quad (M' \text{ is tight, } m'_{i,j} \notin \mathbb{Z}_\infty) \\ &= M'_{i,n} + M'_{n,0} + M'_{0,n} + M'_{n,j} \quad (M'_{n,0} = (\leqslant, 1) \text{ and } M'_{0,n} = (\leqslant, -1)) \\ &= M'_{i,0} + M'_{0,j} \quad (M \text{ tight}) \\ &= M''_{i,n} + M''_{n,j} \quad \text{(definition of } M''). \end{aligned}$$

□

Proposition 19. *Let $\nu \in [0,1]^n$ be a valuation. Then every DBM $M \in closure(M_\nu)$ is tight.*

Proof. M_ν is obviously tight. Then by induction, using Propositions 16, 17, and 18, every DBM in $closure(M_\nu)$ is tight. □

2) DBM Operators Preserve Well-Supportedness:

Proof of Lemma 3. Assume that $\nu \in [0,1]^n$ is a clock valuation. We prove that all consistent DBMs $M \in closure(M_\nu)$ are well-supported. To this end, define

$$\mathrm{Supp}_\nu = \{c + \nu_i - \nu_j \mid c \in \mathbb{Z}, 0 \leqslant i, j \leqslant n\} \cup \{\infty\}.$$

It suffices to show that every consistent DBM in $closure(M_\nu)$ has entries in Supp_ν. Indeed we have already noted that all consistent DBMs in $closure(M_\nu)$ are 1-bounded; but an entry of Supp_ν lies in the interval $[-1,1]$ only if it has the form $c + \nu_i - \nu_j$ for $c \in \{-1, 0, 1\}$ and $0 \leqslant i, j \leqslant n$.

We prove that every consistent DBM in $closure(M_\nu)$ has entries in Supp_ν by induction on the sequence of operations producing such a DBM.

Base case. The DBM M_ν is obviously well-supported, since its (i,j)-th entry is $\nu_i - \nu_j \in \mathrm{Supp}_\nu$ for all $0 \leqslant i, j \leqslant n$.

Induction step. Let $M(\prec_{i,j}, m_{i,j}) \in closure(M_\nu)$ be a DBM and assume that each entry $m_{i,j}$ lies in Supp_ν. We prove that all entries of the DBMs $\overrightarrow{M} \cap \bigcap_{i=1}^n (x_i \leqslant 1)$, $M[x_\ell \leftarrow 0]$, and $M \cap M'$, for M' atomic, also lie in Supp_ν provided that these DBMs are consistent.

It is clear that each entry of $M[x_\ell \leftarrow 0]$ lies in Supp_ν since reset only permutes the entries of a DBM and introduces 0 as a new entry. Likewise it is clear that each entry of $\overrightarrow{M}$ also lies in Supp_ν. Thus to complete the inductive argument it suffices to show that for any DBM M with entries in Supp_ν and any atomic DBM M', all entries of $M \cap M'$ are contained in $\mathrm{Supp}(\nu)$ if $M \cap M'$ is consistent.

Let $M' = \{(\prec'_{i,j}, m'_{i,j})\}$ be an atomic DBM whose single non-trivial constraint is $M'_{p,q}$ for some indices p, q (i.e., all other entries are $(<, \infty)$). Then $m'_{p,q} \in \mathbb{Z}$ by definition of atomic DBMs. Recall that the DBM $M'' = M \cap M'$ is given by

$$M''_{i,j} = \min(M_{i,j}, M_{i,p} + M'_{p,q} + M_{q,j})$$

for all indices i, j. Suppose $M'' = M \cap M'$ is consistent and recall by Proposition 19 that M is tight.

Fix indices $0 \leqslant i, j \leqslant n$. We show that $m''_{i,j} \in \mathrm{Supp}_\nu$. If $M''_{i,j} = M_{i,j}$ then $m''_{i,j} \in \mathrm{Supp}_\nu$ by the induction hypothesis. So we may suppose that

$$M''_{i,j} = M_{i,p} + M'_{p,q} + M_{q,j} < M_{i,j} \tag{6}$$

By the induction hypothesis, $m_{i,p}, m_{q,j} \in \mathrm{Supp}_\nu$. From (6) we must have $m_{i,p}, m_{q,j} < \infty$. We now consider three cases.

1) Suppose that $m_{i,p} \in \mathbb{Z}$. Then $m''_{i,j}$ has the form $d + m_{q,j}$ for some integer d, and hence $m''_{i,j} \in \mathrm{Supp}_\nu$ by the induction hypothesis.
2) Suppose that $m_{q,j} \in \mathbb{Z}$. Then $m''_{i,j}$ has the form $d + m_{i,p}$ for some integer d, and hence $m''_{i,j} \in \mathrm{Supp}_\nu$ by the induction hypothesis.
3) The final case is that $m_{i,p}, m_{q,j} \notin \mathbb{Z}_\infty$. Then

$$\begin{aligned} M_{i,p} + M'_{p,q} + M_{q,j} &= M_{i,n} + M_{n,p} + M'_{p,q} + M_{q,n} + M_{n,j} \quad (M \text{ tight}) \\ &\geqslant M_{i,n} + M''_{n,p} + M''_{p,q} + M''_{q,n} + M_{n,j} \quad (M, M' \geqslant M'' \text{ pointwise}) \\ &\geqslant M_{i,n} + M''_{n,n} + M_{n,j} \quad (M'' \text{ canonical}) \\ &\geqslant M_{i,n} + M_{n,j} \quad (M'' \text{ consistent}) \\ &\geqslant M_{i,j} \quad (M \text{ canonical}). \end{aligned}$$

But this contradicts (6) and so this case cannot hold.

□

C. Proof of Propositions 7 and Lemma 12

Let $\Sigma = \{a_1, \ldots, a_n\}$ be a finite alphabet. Define a function $\pi : \Sigma^* \to \mathbb{N}^n$ such that $\pi(w)_i$ is the number of occurrences of letter a_i in w for $i = 1, \ldots, n$. The image of a language $L \subseteq \Sigma^*$ under π is called the *Parikh image* (or *commutative image*) of L. It is well known that the Parikh image of any regular language (indeed any context-free language) is definable in Presburger arithmetic. In particular, the Parikh image of the language of an NFA over a unary alphabet is a union of arithmetic progressions. Chrobak and Martinez [23], [24] show that the Parikh image of the language of an n-state NFA $\mathcal{A}$ over a unary alphabet comprises $O(n^2)$ many arithmetic progressions which can be explicitly computed from $\mathcal{A}$ in polynomial time.

Consider a monotonic counter machine $\mathcal{C} = (S, C, \Delta)$. Let N be the maximum constant appearing in a transition guard. Define the set $\mathcal{R}_N$ of *regions* to be $\mathcal{R}_N = \{0, \ldots, N\} \cup \{\infty\}$. A counter valuation $v \in \mathbb{N}^n$ defines a tuple $Reg(v) \in \mathcal{R}_N^n$ by

$$Reg(v)_i = \begin{cases} v_i & \text{if } v_i \leqslant N \\ \infty & \text{otherwise} \end{cases}$$

Intuitively ∞ represents any counter value strictly greater than N. The satisfaction relation $\models$ between regions and guards is defined in the obvious way. Below we define a finite automaton $[\mathcal{C}]$ that simulates $\mathcal{C}$.

The alphabet of $[\mathcal{C}]$ is $\Sigma = \{inc_1, \ldots, inc_n\}$. Intuitively $[\mathcal{C}]$ performs an inc_i-transition when simulating an increment on counter c_i. A state of $[\mathcal{C}]$ is a tuple $\langle s, R, \lambda \rangle$, where $s \in S$, $R \in \mathcal{R}_N^n$ is a region of $\mathcal{C}$, and $\lambda \subseteq C$. With a configuration $\langle s, v \rangle$ in a run ρ of $\mathcal{C}$ we associate a state $\langle s, Reg(v), \lambda \rangle$ of $[\mathcal{C}]$. Intuitively, λ represents the set of counters that will be reset along the suffix of the run ρ starting from $\langle s, v \rangle$.

The transition relation of $[\mathcal{C}]$ is defined as follows:

- For each edge $\langle s, \varphi, reset(c_i), s' \rangle \in \Delta$ we add a transition $\langle s, R, \lambda \rangle \xrightarrow{\varepsilon} \langle s', R', \lambda' \rangle$ if $R \models \varphi$, $R'_i = 0$, $R'_j = R_j$ for $j \neq i$, and $\lambda' \cup \{c_i\} = \lambda$.
- For each edge $\langle s, \varphi, nop, s' \rangle \in \Delta$ we add a transition $\langle s, R, \lambda \rangle \xrightarrow{\varepsilon} \langle s', R, \lambda \rangle$ if $R \models \varphi$.
- For each edge $\langle s, \varphi, inc(c_i), s' \rangle \in \Delta$ we add a transition $\langle s, R, \lambda \rangle \xrightarrow{\sigma} \langle s', R', \lambda \rangle$ if $R \models \varphi$, $R'_i = R_i + 1$, and $R'_j = R_j$ for $j \neq i$. The label σ is inc_i if $c_i \notin \lambda$ and otherwise σ is ε.

By construction of $[\mathcal{C}]$, there is a run of $\mathcal{C}$ from $\langle s, v \rangle$ to $\langle s', v' \rangle$ along which the collection of counters that are reset is $\lambda = \{c_1, \ldots, c_m\}$ only if there is a run of $[\mathcal{C}]$ from $\langle s, Reg(v), \lambda \rangle$ to $\langle s', Reg(v'), \varnothing \rangle$. If $w \in \Sigma^*$ is the word read along such a run then we have

$$\begin{aligned} v'_i &= \pi(w)_i \quad i = 1, \ldots, m \\ v'_i - v_i &= \pi(w)_i \quad i = m+1, \ldots, n\,. \end{aligned} \tag{7}$$

Fix states $\langle s, R, \lambda \rangle$ and $\langle s', R', \varnothing \rangle$ of $[\mathcal{C}]$. Let $L_{\langle s,R,\lambda \rangle, \langle s',R',\varnothing \rangle}$ be the set of words w on which $[\mathcal{C}]$ has a run from $\langle s, R, \lambda \rangle$ to $\langle s', R', \varnothing \rangle$. Then the Parikh image $\pi(L_{\langle s,R,\lambda \rangle, \langle s',R',\varnothing \rangle})$ is expressible by a formula $\psi(z_1, \ldots, z_n)$ of Presburger arithmetic.

Returning to the counter machine $\mathcal{C}$, we wish to express the reachability relation of $\mathcal{C}$ between two controls states s and s'. The idea is that for each initial counter valuation v and each run of $\mathcal{C}$ from $\langle s, v \rangle$ to s', we need to specify the total number of increments for each counter that is never reset along the run and the total number of increments since the last reset for all other counters. With this in mind, using Equation (7), the $\mathcal{L}_{\mathbb{Z}}$-formula

$$\varphi(v, v') \stackrel{\text{def}}{=} (Reg(v) = R) \wedge \psi(v'_1, \ldots, v'_m, v'_{m+1} - v_{m+1}, \ldots, v'_n - v_n)$$

describes the subset of the reachability relation arising from the runs of $\mathcal{C}$ whose projection on $[\mathcal{C}]$ goes from state $\langle s, R, \lambda \rangle$ to $\langle s', R', \varnothing \rangle$, for $\lambda = \{c_1, \ldots, c_m\}$. The reachability relation of $\mathcal{C}$ can clearly be described as a finite disjunction of such formulas. This concludes the proof of Proposition 7.

The following specialisation of Proposition 7 is used in the proof of Lemma 12.

Proposition 20. *Let $\mathcal{C}$ be a monotonic counter machine with n counters and with N the maximum integer constant appearing in a transition guard. Given states s, s' of $\mathcal{C}$ and $R, R' \in \mathcal{R}_N^n$, the set*

$$\begin{aligned} \{(u, u') \in \mathbb{N}^2 : \exists \langle s, v \rangle \longrightarrow^* \langle s', v' \rangle \textit{ s.t.} \\ Reg(v) = R, Reg(v') = R', v_n = u, v'_n = u'\} \end{aligned}$$

is definable by a quantifier-free formula of $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$ (involving only integer terms) that is computable in time polynomial in (the largest (absolute) constant of) N and the number of states and counters of $\mathcal{C}$.

Proof. We start by defining an NFA $\mathcal{B}$, over a singleton alphabet $\{inc_n\}$. Automaton $\mathcal{B}$ can be seen as a "sub-automaton" of the NFA $[\mathcal{C}]$ from the proof of Proposition 7. Specifically the states of $\mathcal{B}$ are those states $\langle s'', R'', \lambda \rangle$ of $[\mathcal{C}]$ such that either

$\lambda = C$ or $\lambda = C\backslash\{c_n\}$. (This last condition means that all increments of counters other than c_n are represented in $\mathcal{B}$ by ε-transitions.) For the fixed states and regions s, R, s', R', as in the statement of the proposition, the initial states of $\mathcal{B}$ are those of the form $\langle s, R, \lambda\rangle$, where $\lambda = C$ or $\lambda = C\backslash\{c_n\}$, and the accepting states those of the form $\langle s', R', C\backslash\{c_n\}\rangle$.

Then the Parikh image of the language of $\mathcal{B}$ is equal to

$$\{(v_n, v'_n) \in \mathbb{N}^2 : Reg(v) = R, Reg(v') = R', \langle s, v\rangle \,(\longrightarrow)^* \langle s', v'\rangle\}\,. \tag{8}$$

We can now appeal to the above-mentioned result of Chrobak and Martinez [23], [24] to get that the set (8) is definable by a quantifier-free formula of Presburger arithmetic that is computable in time polynomial in the number of states of $\mathcal{B}$, that is, polynomial in the largest (absolute) constant of N and the number of states and counters of $\mathcal{C}$. □

The proof of Lemma 12 is exactly the same as the proof Theorem 10, except that we replace the use of Proposition 7 by Proposition 20, so as to obtain a quantifier-free formula in $\mathcal{L}^*_{\mathbb{R},\mathbb{Z}}$.

D. Proof of Proposition 8

We first give the "soundness" direction of the proof, that is, from runs of the counter machine $\mathcal{C}_{\langle\ell,\nu\rangle}$ to runs of $\mathcal{A}$.

Suppose that

$$\langle\langle\ell_0, M_0\rangle, v^{(0)}\rangle \longrightarrow \langle\langle\ell_1, M_1\rangle, v^{(1)}\rangle \longrightarrow \ldots \longrightarrow \langle\langle\ell_k, M_k\rangle, v^{(k)}\rangle$$

is a run of $\mathcal{C}_{\langle\ell,\nu\rangle}$ with $\ell_0 = \ell$, $v^0 = \lfloor\nu\rfloor$ and $[\![M_0]\!] = \{\mathrm{frac}(\nu)\}$. Given any valuation $\nu^{(k)} \in [\![M_k]\!]$, we construct a sequence of valuations $\nu^{(0)}, \ldots, \nu^{(k-1)}$, with $\nu^{(j)} \in [\![M_j]\!]$ for $j = 0, \ldots, k-1$, such that

$$\langle\ell_0, v^{(0)} + \nu^{(0)}\rangle \Longrightarrow \langle\ell_1, v^{(1)} + \nu^{(1)}\rangle \Longrightarrow \ldots \Longrightarrow \langle\ell_k, v^{(k)} + \nu^{(k)}\rangle$$

is a run of $\mathcal{A}$. Note that then we must have $\nu^{(0)} = \mathrm{frac}(\nu)$.

The construction of $\nu^{(j)}$ is by backward induction on j. The base step, valuation $\nu^{(k)}$, is given. The induction step divides into three cases according to the nature of the transition $\langle\langle\ell_{j-1}, M_{j-1}\rangle, v^{(j-1)}\rangle \longrightarrow \langle\langle\ell_j, M_j\rangle, v^{(j)}\rangle$. (Recall the classification of transitions in the definition of $\mathcal{C}_{\langle\ell,\nu\rangle}$.)

- $\langle\langle\ell_{j-1}, M_{j-1}\rangle, v^{(j-1)}\rangle \longrightarrow \langle\langle\ell_j, M_j\rangle, v^{(j)}\rangle$ is a delay transition. Then we have $M_j = \overrightarrow{M_{j-1}} \cap [0,1]^n$, $\ell_j = \ell_{j-1}$, and $v^{(j)} = v^{(j-1)}$. Thus we can pick $\nu^{(j-1)} \in [\![M_{j-1}]\!]$ such that $\nu^{(j)} = \nu^{(j-1)} + d$ for some $d \geqslant 0$. Thus there is a delay transition
$$\langle\ell_{j-1}, v^{(j-1)} + \nu^{(j-1)}\rangle \stackrel{d}{\Longrightarrow} \langle\ell_j, v^{(j)} + \nu^{(j)}\rangle$$
in $\mathcal{A}$.
- $\langle\langle\ell_{j-1}, M_{j-1}\rangle, v^{(j-1)}\rangle \longrightarrow \langle\langle\ell_j, M_j\rangle, v^{(j)}\rangle$ is a wrapping transition. Then we have $M_j = (M_{j-1} \cap (x_i = 1))[x_i \leftarrow 0]$ for some index i. Thus we can pick $\nu^{(j-1)} \in [\![M_{j-1} \cap (x_i = 1)]\!]$ such that $\nu^{(j)} = \nu^{(j-1)}[x_i \leftarrow 0]$. In this case we have
$$\langle\ell_{j-1}, v^{(j-1)} + \nu^{(j-1)}\rangle = \langle\ell_j, v^{(j)} + \nu^{(j)}\rangle\,.$$
- $\langle\langle\ell_{j-1}, M_{j-1}\rangle, v^{(j-1)}\rangle \longrightarrow \langle\langle\ell_j, M_j\rangle, v^{(j)}\rangle$ is a discrete transition. Let the corresponding edge of $\mathcal{A}$ be $\langle\ell_{j-1}, \varphi, \{x_i\}, \ell_j\rangle$. Then we have $M_j = (M_{j-1} \cap \varphi_{\mathsf{frac}})[x_i \leftarrow 0]$. Thus we may pick $\nu^{(j-1)} \in [\![M_{j-1} \cap \varphi_{\mathsf{frac}}]\!]$ such that $\nu^{(j-1)}[x_i \leftarrow 0] = \nu^{(j)}$. Since $v^{(j-1)} \models \varphi_{\mathsf{int}}$ we have that $v^{(j-1)} + \nu^{(j-1)} \models \varphi$. Thus there is a discrete transition
$$\langle\ell_{j-1}, v^{(j-1)} + \nu^{(j-1)}\rangle \stackrel{0}{\Longrightarrow} \langle\ell_j, v^{(j)} + \nu^{(j)}\rangle$$
in $\mathcal{A}$.

We now give the "completeness" direction of the proof: from runs of the timed automaton $\mathcal{A}$ to runs of the counter machine $\mathcal{C}_{\langle\ell,\nu\rangle}$.

Suppose that we have a run

$$\langle\ell_0, \nu^{(0)}\rangle \stackrel{d_1}{\Longrightarrow} \langle\ell_1, \nu^{(1)}\rangle \stackrel{d_2}{\Longrightarrow} \ldots \stackrel{d_k}{\Longrightarrow} \langle\ell_k, \nu^{(k)}\rangle$$

of $\mathcal{A}$, where $\langle\ell_0, \nu^{(0)}\rangle = \langle\ell, \nu\rangle$. We can transform such a run, while keeping the same initial and final configurations, by decomposing each delay step into a sequence of shorter delays, so that for all $0 \leqslant j \leqslant k-1$ and all $x \in \mathcal{X}$ the open interval $(\nu^{(j)}(x), \nu^{(j+1)}(x))$ contains no integer. In other words, we break a delay step at any point at which some clock crosses an integer boundary. We can now obtain a corresponding run of $\mathcal{C}_{\langle\ell,\nu\rangle}$ that starts from state $\langle\langle\ell_0, M_0\rangle, v^{(0)}\rangle$, where $[\![M_0]\!] = \{\mathrm{frac}(\nu)\}$ and $v^{(0)} = \lfloor\nu^{(0)}\rfloor$, and ends in state $\langle\langle\ell_k, M_k\rangle, v^{(k)}\rangle$ such that $\nu^{(k)} \in v^{(k)} + [\![M_k]\!]$.

We build such a run of $\mathcal{C}_{\langle\ell,\nu\rangle}$ by forward induction. In particular, we construct a sequence of intermediate states $\langle\langle\ell_i, M_i\rangle, v^{(i)}\rangle$, $0 \leqslant i \leqslant k$, such that $\nu^{(i)} \in v^{(i)} + [\![M_i]\!]$ for each such i. Each discrete transition of $\mathcal{A}$ is simulated by a discrete transition of $\mathcal{C}_{\langle\ell,\nu\rangle}$. A delay transition of $\mathcal{A}$ that ends with set of clocks $\lambda \subseteq \{x_1, \ldots, x_n\}$ being integer valued is simulated by a delay transition of $\mathcal{C}_{\langle\ell,\nu\rangle}$, followed by wrapping transitions for all counters c_i for which $x_i \in \lambda$.

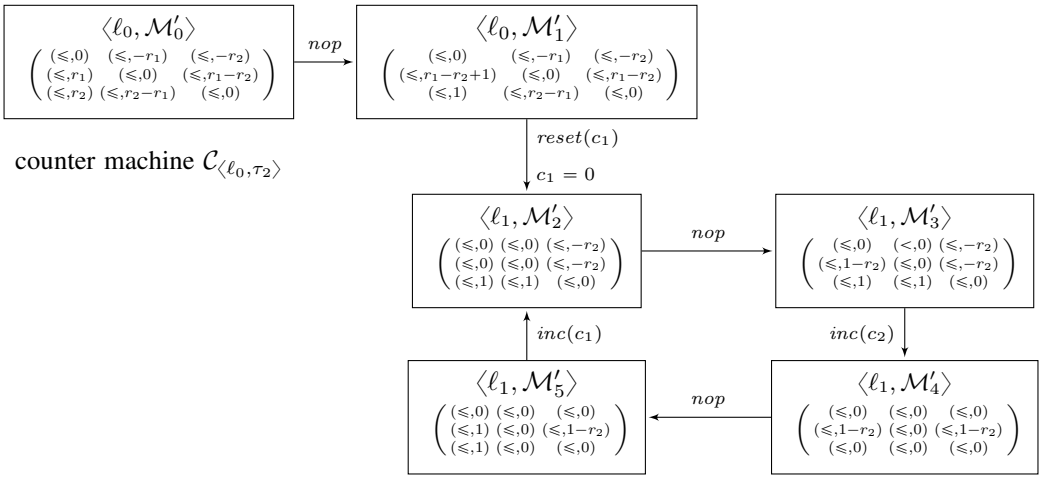


Fig. 6. The counter machine $\mathcal{C}_{\langle \ell,\tau_2 \rangle}$ constructed from the timed automaton in Figure 3, where τ_2 is the type of the valuation ν with $\nu_1 = 0$ and $\nu_2 = .2$.

E. Proof of Theorem 10

Let $\mathcal{A} = \langle L, \mathcal{X}, E \rangle$ be a timed automaton with maximum clock constant N. We first transform $\mathcal{A}$ so that all guards are conjunctions of atoms of the type appearing in Figure 4. This transformation may lead to an exponential blow-up in the number of edges. In any case, it can be accomplished in time at most $2^{\mathrm{poly}(n)} \cdot \mathrm{poly}(L)$.

Let τ be an n-type. Following Corollary 6 we have observed that $|closure(\mathcal{M}_\tau)| \leqslant 2^{\mathrm{poly}(n)}$. It follows that for a location $\ell \in L$ and n-type τ, the monotonic counter automaton $\mathcal{C}_{\langle \ell,\tau \rangle}$ can be computed in time at most $2^{\mathrm{poly}(n)} \cdot \mathrm{poly}(|L|)$.

Applying Proposition 7, we get that the formula $\chi^{\tau}_{\ell,\ell'}$ can be computed in time at most $2^{\mathrm{poly}(n)} \cdot \mathrm{poly}(|L|, N^n)$. Furthermore, given τ, the formula α^{τ} can be computed in time $\mathrm{poly}(n)$.

Finally, the number of disjuncts in (4), i.e., the number of different n-types when restricting to formulas $t \leqslant t'$ for $t, t' \in \mathcal{DT}_{\mathbb{R}}(n)$, is bounded by $2^{\mathrm{poly}(n)}$.

Putting everything together, the formula $\varphi_{\ell,\ell'}$ can be computed in time at most $2^{\mathrm{poly}(n)} \cdot \mathrm{poly}(|L|, N^n)$, that is, exponential in the size of the original timed automaton $\mathcal{A}$.

F. Symbolic Counter Machines

In this section we illustrate Figure 6 used in Example 7.

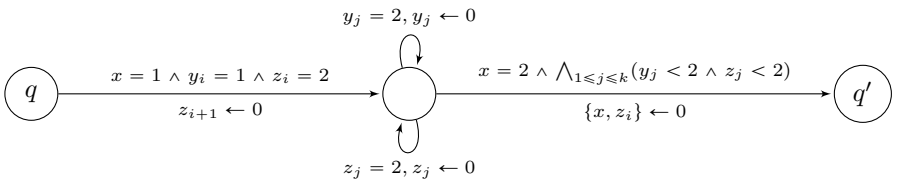
G. Proof of Theorem 11

This section we continue the argument of Section V-B showing that model checking parametric timed reachablity logic is NEXPTIME-hard.

It remains to explain how from linear bounded automaton $\mathcal{B}$ one can define a timed automaton $\mathcal{A}$ whose configuration graph embeds the configuration graph of $\mathcal{B}$. The construction is adapted from the PSPACE-hardness proof for reachability in timed automata [1]. We assume that $\mathcal{B}$ uses a binary input alphabet and a fixed tape length of k. The main idea is as follows: $\mathcal{A}$ uses $2k + 1$ clocks: one clock y_i and z_i for each tape cell i, and one extra clock x. The clocks y_i and z_i, respectively, are used to encode the current tape content and the position of the pointer of $\mathcal{B}$, respectively. The clock x is an auxiliary clock that helps to encode this information correctly into the other clocks. Technically, x is used to measure out cycles of two time units, i.e., x is reset to 0 whenever it reaches 2. The construction is such that the values of y_i and z_i obey the following policy: whenever x takes value 0, y_i takes value 1 (0, respectively) if there is a 1 (0, respectively) in the i-th cell of the tape; and z_i takes value 1 if the position of the pointer is the i-th cell, otherwise, z_i takes value 0. We can set these bits appropriately by resetting clocks y_i and z_i either when $x = 1$ or $x = 2$, and we can preserve the values of a clock y_i or z_i between successive cycles by resetting it when it reaches value 2, see below for more details. Using this idea, $\mathcal{A}$ can be defined such that it only takes transitions at integer times and such that a configuration of $\mathcal{A}$ after $2t$ time steps encodes a configuration of $\mathcal{B}$ after t computation steps for each $t \in \mathbb{N}$.

More formally, the set of locations of $\mathcal{A}$ contains one copy location q for each state q of $\mathcal{B}$, plus some additional auxiliary locations, one of which being an initial location ℓ_0. In the initialization phase, we encode the initial configuration $(q_0, \sigma_1)\sigma_2 \ldots \sigma_k$ of $\mathcal{B}$, where q_0 is the initial state of $\mathcal{B}$, and $\sigma_i \in \{0, 1\}$. For this, we define a transition from ℓ_0 to q_0, with guard $x = 1$, and resetting x, $z_2, \ldots, z_k$, and we further reset clock y_i iff $\sigma_i = 0$. One can easily observe that if $\mathcal{A}$ reaches q_0 with clock value $x = 0$, then $z_1 = 1$, and $y_i = 1$ iff the i-th cell contains a 1, while all other clocks have value 0. This

correctly encodes the initial configuration of $\mathcal{B}$. We now proceed with the simulation phase. From locations q that correspond to states of $\mathcal{B}$, we simulate the computation steps from $\mathcal{B}$. Assume, for instance, that the transition relation of $\mathcal{B}$ contains the tuple $(q, 0, q', 0, R)$, i.e., when reading letter 0 in state q, $\mathcal{B}$ goes to state q', leaves the symbol on the tape as it is, and moves the pointer one position to the right. According to the encoding described above, this means that if $\mathcal{A}$ reaches q with $x = 0$, we need to test whether $y_i = 0$ for the unique $1 \leqslant i \leqslant k$ such that $z_i = 1$, and whether $i < k$ (because we want to move the position of the pointer one cell to the right). If this is the case, $\mathcal{A}$ should go to location q' and the bit of z_i should be reset to z_{i+1}. We thus define for every $1 \leqslant i < k$ a transition as shown in the following, where the loops in the auxiliary location in the middle are defined for every $1 \leqslant j \leqslant k$.



Transitions of $\mathcal{B}$ of other forms can be simulated in a similar way. We finally augment $\mathcal{A}$ with a label function LB that assigns p to a location q iff q is an accepting state of $\mathcal{B}$.

This finishes the proof for NEXPTIME-hardness.