Effective Definability of the Reachability Relation in Timed Automata

Martin Fränzle
University Of Oldenburg

Karin Quaas
Universität Leipzig

Mahsa Shirmohammadi
CNRS & IRIF

James Worrell
University of Oxford

Abstract

We give a new proof of the result of Comon and Jurski that the binary reachability relation of a timed automaton is definable in linear arithmetic.

1 Introduction

Comon and Jurski [6, 7] showed that the binary reachability relation of a given timed automaton is effectively definable by a first-order formula of linear arithmetic over the reals augmented with a unary predicate denoting the integers. The proof of this result, given in [7], is based on a syntactic transformation of arbitrary timed automata into equivalent timed automata satisfying a certain structural restriction, called flatness. The proof is relatively long and technical (running to over 40 pages) and there have been a number of subsequent attempts to both generalise the result and simplify its proof [5, 8, 9, 10]. The present note is a development of [10, Sections III and IV] and further simplifies the proof of Comon and Jurski's result therein. In particular, we avoid many technicalities of [10] by employing a simple "clock memorisation" trick to reduce computation of the binary reachability relation of a timed automaton to computation of the set of configurations reachable from a given location starting with the all-zeros clock valuation. We show how to recover the latter set of configurations as the commutative image of a certain regular language accepted by a variant of Alur and Dill's region automaton (cf. [1]).

2 Definitions and Main Result

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

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

where $k \in \mathbb{N}$ and $x \in \mathcal{X}$. A *clock valuation* is a mapping $\nu : \mathcal{X} \to \mathbb{R}_{\geq 0}$, where $\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers. Denote by $\mathbb{R}^{\mathcal{X}}_{\geq 0}$ the set of all clock valuations. We write $\nu \models \varphi$ to denote that $\nu \in \mathbb{R}^{\mathcal{X}}_{\geq 0}$ satisfies the constraint φ. We denote by $\mathbf{0}$ the valuation such that $\mathbf{0}(x) = 0$ for all $x \in \mathcal{X}$. Given $t \in \mathbb{R}_{\geq 0}$, 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 $x \notin \lambda$.

A *1-bounded zone* $Z \subseteq [0,1]^{\mathcal{X}}$ is a set of clock valuations that is defined by a conjunction of difference constraints $x_i \sim c$ and $x_i - x_j \sim c$, where $c \in \{-1, 0, +1\}$, $\sim \in \{<, =\}$, $i, j \in \{1, \ldots, n\}$. We write $\mathcal{Z}_1(\mathcal{X})$ for the set of 1-bounded zones. Given a 1-bounded zone Z and $\lambda \subseteq \mathcal{X}$, the following are also 1-bounded zones (see, e.g., [4]):

$$\begin{aligned} Z[\lambda \leftarrow 0] &:= \{\nu[\lambda \leftarrow 0] : \nu \in Z\} \\ \overrightarrow{Z} &:= \{\nu + t : \nu \in Z, t \geq 0\} \cap [0,1]^{\mathcal{X}}\,. \end{aligned}$$

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*. A *configuration* of $\mathcal{A}$ is a pair $\langle \ell, \nu\rangle$ consisting of a location ℓ and a clock valuation ν. Such a timed automaton $\mathcal{A}$ induces a ternary *transition relation*

$$\Longrightarrow \subseteq (L \times \mathbb{R}^{\mathcal{X}}_{\geq 0}) \times \mathbb{R} \times (L \times \mathbb{R}^{\mathcal{X}}_{\geq 0})$$

on the set of configurations as follows. Given configurations $\langle \ell, \nu\rangle$ and $\langle \ell', \nu'\rangle$, we postulate:

- a delay transition $\langle \ell, \nu\rangle \stackrel{d}{\Longrightarrow} \langle \ell', \nu'\rangle$ for some $d \geq 0$, if $\nu' = \nu + d$ and $\ell = \ell'$;
- a discrete transition $\langle \ell, \nu\rangle \stackrel{0}{\Longrightarrow} \langle \ell', \nu'\rangle$, 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* $q_0 \stackrel{d_1}{\Longrightarrow} q_1 \stackrel{d_2}{\Longrightarrow} q_2 \stackrel{d_3}{\Longrightarrow} \ldots \stackrel{d_m}{\Longrightarrow} q_m$ of $\mathcal{A}$ is a finite sequence of delay and discrete transitions.

Let $\mathcal{L}$ denote the set of first-order formulas over the structure $\mathcal{R} = (\mathbb{R}, \mathbb{Z}(\cdot), 0, 1, +, \leq)$, where $\mathbb{R}$ is the universe and $\mathbb{Z}(\cdot)$ is a unary predicate denoting the set of integers. We call $\mathcal{L}$ the language of *mixed linear arithmetic*. This language subsumes both Presburger arithmetic (linear arithmetic over $\mathbb{Z}$) and linear arithmetic over $\mathbb{R}$. It is shown in [3, Section 3] how to rewrite a given $\mathcal{L}$-sentence in polynomial time into an equivalent Boolean formula whose atoms are either sentences of real linear arithmetic or sentences of Presburger arithmetic. From known complexity bounds for the latter two theories [2], it follows that deciding the truth of $\mathcal{L}$-sentences can be carried out by an alternating Turing machine running in time $2^{2^{O(n)}}$ with n alternations (i.e., the same complexity as Presburger arithmetic).

The main contribution of the present paper is to prove the following result.

Theorem 1. *Given a timed automaton $\mathcal{A}$ with n clock variables and locations ℓ_0, ℓ of $\mathcal{A}$, we can compute an $\mathcal{L}$-formula $\varphi^{\mathcal{A}}_{\ell_0,\ell}(x_1, \ldots, x_n, y_1, \ldots, y_n)$ such that there is a run of $\mathcal{A}$ from configuration $\langle \ell_0, \nu_0\rangle$ to configuration $\langle \ell, \nu\rangle$ iff $\mathcal{R} \models \varphi^{\mathcal{A}}_{\ell_0,\ell}[\nu_0, \nu]$.*

3 Proofs

3.1 Clock Memorisation

We describe a simple trick that reduces the problem of computing the binary reachability relation on a given timed automaton $\mathcal{A}$ to that of computing the set of configurations reachable from a fixed initial configuration in a derived automaton $\mathcal{B}$. The idea is that $\mathcal{B}$ starts from the zero clock valuation, guesses an initial clock valuation ν_0 of $\mathcal{A}$, and then simulates a computation of $\mathcal{A}$ while "remembering" ν_0 as a set of differences between the values of some fresh clocks. Formally we derive Theorem 1 from the following result, which we prove later on.

Reachability set

Proposition 2. *Given a timed automaton $\mathcal{A}$ with n clocks and locations ℓ_0, ℓ of $\mathcal{A}$, we can compute an $\mathcal{L}$-formula $\psi^{\mathcal{A}}_{\ell_0,\ell}(x_1, \ldots, x_n)$ such that there is a run in $\mathcal{A}$ from $\langle \ell_0, \mathbf{0}\rangle$ to $\langle \ell, \nu\rangle$ if and only if $\mathcal{R} \models \psi^{\mathcal{A}}_{\ell_0,\ell}[\nu]$.*

Proof of Theorem 1. Given a timed automaton $\mathcal{A} = \langle L, \mathcal{X}, E\rangle$ with a distinguished location $\ell_0 \in L$, we define a new timed automaton $\mathcal{B} = \langle L', \mathcal{X}', E'\rangle$ as follows. The set of locations is $L' = L \cup \{\ell'_0\}$, where $\ell'_0 \notin L$ is a distinguished location in $\mathcal{B}$. The set of clocks is $\mathcal{X}' = \{x, x' : x \in \mathcal{X}\} \cup \{z\}$, where $z \notin \mathcal{X}$, that is, $\mathcal{B}$ has two copies of each clock of $\mathcal{A}$ plus an extra "reference clock" z. We obtain E' by adding the following edges to E: for each $x \in \mathcal{X}$ we have an edge $\langle \ell'_0, \mathbf{true}, \{x, x'\}, \ell'_0\rangle$ in E' (that is, a selfloop on ℓ'_0 that resets both copies of clock x); we also have a single additional edge $\langle \ell'_0, \mathbf{true}, \{z\}, \ell_0\rangle$ in E'. Notice that once $\mathcal{B}$ leaves ℓ'_0 then neither the reference clock z nor any clock in the set $\{x' \mid x \in \mathcal{X}\}$ is reset—intuitively when $\mathcal{B}$ exits ℓ'_0, the value of clock $x \in \mathcal{X}$ is stored in the difference of x' and z.

Observe that for all $\ell \in L$ there is a run from $\langle \ell_0, \nu_0\rangle$ to $\langle \ell, \nu\rangle$ in $\mathcal{A}$ if and only if there is a run in $\mathcal{B}$ from $\langle \ell'_0, \mathbf{0}\rangle$ to $\langle \ell, \nu'\rangle$ such that $\nu'(x) = \nu(x)$ and $\nu'(x') - \nu'(z) = \nu_0(x)$ for all $x \in \mathcal{X}$. In particular, a run of $\mathcal{A}$ from $\langle \ell_0, \nu_0\rangle$ to $\langle \ell, \nu\rangle$ can be simulated in $\mathcal{B}$ as follows. Automaton $\mathcal{B}$ starts in configuration $\langle \ell'_0, \mathbf{0}\rangle$; by taking selfloops on ℓ'_0 and then the edge $\langle \ell'_0, \mathbf{true}, \{z\}, \ell_0\rangle$, $\mathcal{B}$ may reach a configuration $\langle \ell_0, \nu''\rangle$ such

that $\nu''(x) = \nu''(x') = \nu_0(x)$ for all $x \in \mathcal{X}$, and $\nu''(z) = 0$; then $\mathcal{B}$ directly simulates the given run of $\mathcal{A}$ (without resetting the new clocks x', $x \in \mathcal{X}$, and z).

Let $\psi^{\mathcal{B}}_{\ell_0',\ell}(\boldsymbol{x}, \boldsymbol{x}', z)$ be the formula obtained in Proposition 2 for automaton $\mathcal{B}$. Then we define

$$\varphi^{\mathcal{A}}_{\ell_0,\ell}(\boldsymbol{x}, \boldsymbol{y}) := \exists \boldsymbol{x}' \exists z \,.\, (\boldsymbol{x} = \boldsymbol{x}' - z\mathbf{1} \wedge \psi^{\mathcal{B}}_{\ell_0',\ell}(\boldsymbol{y}, \boldsymbol{x}', z)) \,.$$

$\overline{X}' = \overline{X} + z$

$\rightsquigarrow \varphi^{A}_{\ell_0\ell}(\bar{x},\bar{y}) \equiv \exists z \cdot \square \; \psi^{B}_{\ell_0',\ell}(\bar{y}, \bar{x}+z, z)$

3.2 A Discrete-Time Automaton

Given a timed automaton $\mathcal{A} = \langle L, \mathcal{X}, E \rangle$, we define an (untimed, infinite-state) nondeterministic automaton $R(\mathcal{A})$. The set of states of $R(\mathcal{A})$ is

$$Q = L \times \mathbb{N}^{\mathcal{X}} \times \mathcal{Z}_1(\mathcal{X}) \times 2^{\mathcal{X}} \,.$$

Given a state $\langle \ell, \upsilon, Z, \gamma \rangle \in Q$, intuitively υ and Z respectively encode the integer and fractional parts of the clocks of $\mathcal{A}$, while $\gamma \subseteq \mathcal{X}$ is a "prophecy variable" denoting the set of clocks that will be reset at least once in the future. The alphabet of $R(\mathcal{A})$ is $\mathcal{X}$ and the set of transitions (which includes ε-transitions) is as follows:

1. For each state $\langle \ell, \upsilon, Z, \gamma \rangle \in Q$ there is a *delay transition* $\langle \ell, \upsilon, Z, \gamma \rangle \xrightarrow{\varepsilon} \langle \ell, \upsilon, \overrightarrow{Z}, \gamma \rangle$.

2. For each clock $x \in \mathcal{X}$ and state $\langle \ell, \upsilon, Z, \gamma \rangle$ of $R(\mathcal{A})$ there is a *wrapping transition* $\langle \ell, \upsilon, Z, \gamma \rangle \xrightarrow{\sigma} \langle \ell, \upsilon', Z', \gamma \rangle$ where $Z' := (Z \cap [\![x = 1]\!])[x \leftarrow 0]$, $\upsilon' = \upsilon[x \leftarrow x + 1]$, and $\sigma = \varepsilon$ if $x \in \gamma$ but otherwise $\sigma = x$. (Intuitively a wrapping transition for clock x has label ε if $x \in \gamma$ since x will be reset again in the future.)

3. Each edge $\langle \ell, \varphi, \lambda, \ell' \rangle$ of $\mathcal{A}$ yields a transition $\langle \ell, \upsilon, Z, \gamma \rangle \xrightarrow{\varepsilon} \langle \ell', \upsilon', Z', \gamma' \rangle$ of $R(\mathcal{A})$, where $\upsilon' = \upsilon[\lambda \leftarrow 0]$, $Z' := \{\nu \in Z : \upsilon + \nu \models \varphi\}[\lambda \leftarrow 0]$, and $\gamma' \cup \lambda = \gamma$. (Intuitively, γ' is obtained from γ by guessing some clocks that will not be reset again and removing them from γ.)

Given states $q, q' \in Q$, we write $q \xrightarrow{w} q'$ if there is a run in $R(\mathcal{A})$ from q to q' on word $w \in \mathcal{X}^*$. We furthermore write $q \longrightarrow^* q'$ if there exists a run from q to q'.

Proposition 3. *Automaton $\mathcal{A}$ has a run from $\langle \ell_0, \mathbf{0} \rangle$ to $\langle \ell, \nu \rangle$ along which the set of clocks that are reset is $\gamma \subseteq \mathcal{X}$ if and only if $R(\mathcal{A})$ has a run from $\langle \ell_0, \mathbf{0}, \{\mathbf{0}\}, \gamma \rangle$ to $\langle \ell, \upsilon, Z, \emptyset \rangle$ for some $\upsilon \in \mathbb{N}^{\mathcal{X}}$ and $Z \in \mathcal{Z}_1(\mathcal{X})$ such that $\nu - \upsilon \in Z$.*

Proof. For the "if" direction, suppose that

$$\langle \ell_0, \upsilon^{(0)}, Z_0, \gamma_0 \rangle \longrightarrow \langle \ell_1, \upsilon^{(1)}, Z_1, \gamma_1 \rangle \longrightarrow \ldots \longrightarrow \langle \ell_k, \upsilon^{(k)}, Z_k, \gamma_k \rangle$$

is a run of $R(\mathcal{A})$ with $\upsilon^{(0)} = \mathbf{0}$, $Z_0 = \{\mathbf{0}\}$, and $\gamma_0 = \gamma$. Given any valuation $\nu^{(k)} \in Z_k$, we construct a sequence of valuations $\nu^{(0)} \in Z_0, \ldots, \nu^{(k-1)} \in Z_{k-1}$ such that $\mathcal{A}$ has a run

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

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 type of the transition

$$\langle \ell_{j-1}, \upsilon^{(j-1)}, Z_{j-1}, \gamma_{j-1} \rangle \longrightarrow \langle \ell_j, \upsilon^{(j)}, Z_j, \gamma_j \rangle \,.$$

- Delay transition. We have $Z_j = \overrightarrow{Z_{j-1}}$, $\ell_j = \ell_{j-1}$, and $\upsilon^{(j)} = \upsilon^{(j-1)}$. Thus we can pick $\nu^{(j-1)} \in Z_{j-1}$ such that $\nu^{(j)} = \nu^{(j-1)} + d$ for some $d \geq 0$. Hence there is in $\mathcal{A}$ a delay transition

$$\langle \ell_{j-1}, \upsilon^{(j-1)} + \nu^{(j-1)} \rangle \stackrel{d}{\Longrightarrow} \langle \ell_j, \upsilon^{(j)} + \nu^{(j)} \rangle \,.$$

- Wrapping transition. We have $Z_j = (Z_{j-1} \cap [\![x = 1]\!])[x \leftarrow 0]$ for some clock $x \in \mathcal{X}$. Thus we can pick $\nu^{(j-1)} \in Z_{j-1} \cap [\![x = 1]\!]$ such that $\nu^{(j)} = \nu^{(j-1)}[x \leftarrow 0]$. In this case we have

$$\langle \ell_{j-1}, \upsilon^{(j-1)} + \nu^{(j-1)} \rangle = \langle \ell_j, \upsilon^{(j)} + \nu^{(j)} \rangle .$$

- Discrete transition. Let the corresponding edge of $\mathcal{A}$ be $\langle \ell_{j-1}, \varphi, \lambda, \ell_j \rangle$. Then we have $\upsilon^{(j)} = \upsilon^{(j-1)}[\lambda \leftarrow 0]$ and $Z_j = \{\nu \in Z_{j-1} : \nu + \upsilon^{(j-1)} \models \varphi\}[\lambda \leftarrow 0]$. Choose $\nu^{(j-1)} \in Z_{j-1}$ such that $\nu + \upsilon^{(j)} \models \varphi$ and $\nu^{(j)} = \nu^{(j-1)}[\lambda \leftarrow 0]$. Then there is in $\mathcal{A}$ a discrete transition

$$\langle \ell_{j-1}, \upsilon^{(j-1)} + \nu^{(j-1)} \rangle \xRightarrow{0} \langle \ell_j, \upsilon^{(j)} + \nu^{(j)} \rangle .$$

We now turn to the "only-if" direction of the proof. Suppose that we have a run

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

of $\mathcal{A}$, where $\nu^{(0)} = \mathbf{0}$. We first 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 $0 \leq j \leq k-1$ and all $x \in \mathcal{X}$ the open interval $(\nu^{(j)}(x), \nu^{(j+1)}(x))$ contains no integer. In other words, we break every delay step at every point at which some clock crosses an integer boundary. We thus obtain a corresponding run of $R(\mathcal{A})$ that starts from state $\langle \ell_0, \upsilon^{(0)}, Z_0, \gamma_0 \rangle$, where $Z_0 = \{\mathbf{0}\}$, $\upsilon^{(0)} = \mathbf{0}$, and ends in state $\langle \ell_k, \upsilon^{(k)}, Z_k, \emptyset \rangle$ such that $\nu^{(k)} \in \upsilon^{(k)} + Z_k$.

We build such a run of $R(\mathcal{A})$ by forward induction. In particular, we construct a sequence of intermediate states $\langle \ell_i, \upsilon^{(i)}, Z_i, \gamma_i \rangle$, $0 \leq i \leq k$, such that $\nu^{(i)} \in \upsilon^{(i)} + Z_i$ for each such i. Each discrete transition of $\mathcal{A}$ is simulated by a discrete transition of $R(\mathcal{A})$. A delay transition of $\mathcal{A}$ that ends with set of clocks $\lambda \subseteq \mathcal{X}$ being integer valued is simulated by a delay transition of $R(\mathcal{A})$, followed by wrapping transitions for all $x \in \lambda$. □

Proposition 4. *Let $\mathcal{A} = \langle L, \mathcal{X}, E \rangle$ be a timed automaton with distinguished locations $\ell_0, \ell \in L$. For every $Z \in \mathcal{Z}_1(\mathcal{X})$ and $\gamma \subseteq \mathcal{X}$ the set*

$$\left\{ \upsilon \in \mathbb{N}^{\mathcal{X}} : \exists w \in \mathcal{X}^* \cdot \langle \ell_0, \mathbf{0}, \{\mathbf{0}\}, \gamma \rangle \xrightarrow{w} \langle \ell, \upsilon, Z, \emptyset \rangle \right\} \tag{1}$$

is effectively semilinear.

Proof. Consider the following language of "wrapping transitions" in $R(\mathcal{A})$:

$$L_{\text{wrap}} := \left\{ w \in \mathcal{X}^* : \exists \upsilon \in \mathbb{N}^{\mathcal{X}} \cdot \langle \ell_0, \mathbf{0}, \{\mathbf{0}\}, \gamma \rangle \xrightarrow{w} \langle \ell, \upsilon, Z, \emptyset \rangle \right\} .$$

Define the function $\pi : \mathcal{X}^* \to \mathbb{N}^{\mathcal{X}}$ such that $\pi(w)(x)$ is the number of occurrences of letter x in word w. Since visible transitions of $R(\mathcal{A})$ correspond to wrapping transitions of clocks that will not be reset any more, the commutative image $\pi(L_{\text{wrap}})$ is precisely the set of vectors defined in (1).

We claim that L_{wrap} is regular, which suffices to show that the set (1) is semilinear. Indeed, denote by $c_{\max}$ the largest constant appearing in a transition constraint in $\mathcal{A}$, and consider the equivalence relation on states of $R(\mathcal{A})$ defined by $(\ell, \upsilon, Z, \gamma) \sim (\ell, \upsilon', Z, \gamma)$ iff for all $x \in \mathcal{X}$ either $\upsilon(x) = \upsilon'(x)$ or $\upsilon(x), \upsilon'(x) \geq c_{\max}$. Clearly if $(\ell, \upsilon, Z, \gamma) \sim (\ell, \upsilon', Z, \gamma)$ then for all $\nu \in Z$ and clock constraints φ appearing in $\mathcal{A}$ we have that $\upsilon + \nu \models \varphi$ iff $\upsilon' + \nu \models \varphi$. It follows that $\sim$ is a strong bisimulation (that moreover has finite index). Taking the quotient of $R(\mathcal{A})$ by $\sim$ it follows that L_{wrap} is accepted by a finite automaton and hence is regular. □

Proof of Proposition 2. By Proposition 3 we have that

$$\left\{ \nu \in \mathbb{R}^{\mathcal{X}}_{\geq 0} : \langle \ell_0, \mathbf{0} \rangle \Longrightarrow^* \langle \ell, \nu \rangle \right\} = \bigcup_{Z \in \mathcal{Z}_1(\mathcal{X})} \bigcup_{\gamma \subseteq \mathcal{X}} \left\{ \upsilon + Z : \langle \ell_0, \mathbf{0}, \{\mathbf{0}\}, \gamma \rangle \longrightarrow^* \langle \ell, \upsilon, Z, \emptyset \rangle \right\}$$

But by Proposition 4 the right-hand expression above is definable by an $\mathcal{L}$-formula in the sense of Proposition 2. □

4 Computational Complexity

In this section we briefly retrace the proof of Theorem 1 in order to give a complexity bound for computing the $\mathcal{L}$-formula $\varphi_{\ell_0,\ell}$ described therein. We sketch a proof that $\varphi_{\ell_0,\ell}$ is an existential formula that can be computed in time polynomial in the number of locations of the timed automaton and exponential in the number of clocks and bit length of the maximum clock constant.

Working backwards, we start by considering the regular language L_{wrap} in the proof of Proposition 4. From the bound $|\mathcal{Z}_1(\mathcal{X})| \leq (2|\mathcal{X}|+1)! \leq 2^{O(|\mathcal{X}|\log|\mathcal{X}|)}$ it is straightforward that there is a finite automaton accepting language L_{wrap} with number of states bounded by $\text{poly}(|L|, c_{\max}, 2^{|\mathcal{X}|\log|\mathcal{X}|})$ (where L and $\mathcal{X}$ are as in the statement of Proposition 4) and moreover this automaton can be computed in time polynomial in its size.

Lin [11] shows that the Parikh image of the language of an NFA with m states and alphabet size k is described by a quantifier-free formula of Presburger arithmetic that can be computed in time $2^{O(k^2 \log m)}$. We may thus refine the statement of Proposition 4 by specifying that the subset of $\mathbb{N}^{\mathcal{X}}$ in Equation (1) is described by a quantifier-free formula of Presburger arithmetic that can be computed in time bounded by $\text{poly}(|L|, c_{\max}, 2^{|\mathcal{X}|^2})$.

Moving now to the proof of Proposition 2, we see that the formula $\psi_{\ell_0,\ell}$ can likewise be computed in time bounded by $\text{poly}(|L|, c_{\max}, 2^{|\mathcal{X}|^2})$. Constructing formula $\psi_{\ell_0,\ell}$ from the Presburger-arithmetic formula referred to in Proposition 4 requires to introduce existentially quantified variables to denote the fractional parts of the clock values of the source and target configurations; but $\psi_{\ell_0,\ell}$ is otherwise quantifier-free.

Finally, recall that the formula $\varphi_{\ell_0,\ell}$ in Theorem 1 was obtained from the formula $\psi_{\ell_0,\ell}$ in Proposition 2 with only a polynomial blow up that introduced extra existentially quantified variables.

References

[1] R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.

[2] L. Berman. The complexitiy of logical theories. *Theor. Comput. Sci.*, 11:71–77, 1980.

[3] B. Boigelot, S. Jodogne, and P. Wolper. An effective decision procedure for linear arithmetic over the integers and reals. *ACM Trans. Comput. Log.*, 6(3):614–633, 2005.

[4] P. Bouyer. Untameable timed automata! In *Proceedings of STACS'03*, volume 2607 of *LNCS*, pages 620–631. Springer, 2003.

[5] L. Clemente and S. Lasota. Binary reachability of timed pushdown automata via quantifier elimination and cyclic order atoms. volume 107 of *LIPIcs*, pages 118:1–118:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[6] H. Comon and Y. Jurski. Timed automata and the theory of real numbers. In *Proceedings of CONCUR'99*, volume 1664 of *LNCS*, pages 242–257. Springer, 1999.

[7] H. Comon and Y. Jurski. Timed automata and the theory of real numbers. Technical Report LSV-99-6, LSV, ENS Cachan, July 1999.

[8] Z. Dang. Pushdown timed automata: a binary reachability characterization and safety verification. *Theor. Comput. Sci.*, 302(1-3):93–121, 2003.

[9] C. Dima. Computing reachability relations in timed automata. In *Proceedings of LICS'02)*, page 177. IEEE Computer Society, 2002.

[10] K. Quaas, M. Shirmohammadi, and J. Worrell. Revisiting reachability in timed automata. In *Proceedings of LICS'17*, pages 1–12. IEEE Computer Society, 2017.

[11] A. W. To. Parikh images of regular languages: Complexity and applications. *CoRR*, 2010. http://arxiv.org/abs/1002.1464.