Reachability in Two-Parametric Timed Automata with One Parameter Is EXPSPACE-Complete

Stefan Göller
School of Electrical Engineering and Computer Science, Universität Kassel, Germany

Mathieu Hilaire
Université Paris-Saclay, ENS Paris-Saclay, Laboratoire Spécification et Vérification (LSV), CNRS, Gif-sur-Yvette, France

Abstract

Parametric timed automata (PTA) have been introduced by Alur, Henzinger, and Vardi as an extension of timed automata in which clocks can be compared against parameters. The reachability problem asks for the existence of an assignment of the parameters to the non-negative integers such that reachability holds in the underlying timed automaton. The reachability problem for PTA is long known to be undecidable, already over three parametric clocks.

A few years ago, Bundala and Ouaknine proved that for PTA over two parametric clocks and one parameter the reachability problem is decidable and also showed a lower bound for the complexity class $\mathsf{PSPACE}^{\mathsf{NEXP}}$. Our main result is that the reachability problem for parametric timed automata over two parametric clocks and one parameter is EXPSPACE-complete.

For the EXPSPACE lower bound we make use of deep results from complexity theory, namely a serializability characterization of EXPSPACE (in turn based on Barrington's Theorem) and a logspace translation of numbers in Chinese Remainder Representation to binary representation due to Chiu, Davida, and Litow. It is shown that with small PTA over two parametric clocks and one parameter one can simulate serializability computations.

For the EXPSPACE upper bound, we first give a careful exponential time reduction from PTA over two parametric clocks and one parameter to a (slight subclass of) parametric one-counter automata over one parameter based on a minor adjustment of a construction due to Bundala and Ouaknine. For solving the reachability problem for parametric one-counter automata with one parameter, we provide a series of techniques to partition a fictitious run into several carefully chosen subruns that allow us to prove that it is sufficient to consider a parameter value of exponential magnitude only. This allows us to show a doubly-exponential upper bound on the value of the only parameter of a PTA over two parametric clocks and one parameter. We hope that extensions of our techniques lead to finally establishing decidability of the long-standing open problem of reachability in parametric timed automata with two parametric clocks (and arbitrarily many parameters) and, if decidability holds, determining its precise computational complexity.

2012 ACM Subject Classification Theory of computation → Timed and hybrid models; Theory of computation → Automata extensions

Keywords and phrases Parametric Timed Automata, Computational Complexity, Reachability

Digital Object Identifier 10.4230/LIPIcs.STACS.2021.36

Related Version *Full Version*: https://arxiv.org/abs/2011.07091

Funding *Stefan Göller*: The author was supported by the Agence Nationale de la Recherche grant no. ANR-17-CE40-0010.

Acknowledgements We thank Benedikt Bollig and Karin Quaas for discussions and feedback.

1 Introduction

Background. In the 1990's *timed automata* have been introduced by Alur and Dill [2]. They extend finite automata by clocks that can be compared against integer constants and provide a popular formalism to reason about the behavior of real-time systems with desirable algorithmic properties; for instance the reachability/emptiness problem is decidable and in fact PSPACE-complete [1].

For a more general means to specify the behavior of under-specified systems, such as embedded systems, Alur, Henzinger and Vardi [3] have introduced *parametric timed automata (PTA)* only a few years later. Here, the clocks can additionally be compared against parameters that can take unspecified non-negative integer values. Towards the verification of safety properties, or loosely speaking ruling out the existence of an execution to a bad state, the *reachability problem for PTA* in turn asks for the existence of an assignment of the parameters to the non-negative integers such that reachability holds in the resulting timed automaton.

A clock of a PTA that is being compared to at least one parameter is called *parametric*. On the negative side, it has been shown in [3] that already for PTA that contain *three parametric clocks* reachability is undecidable – even in the presence of one parameter [8]. On the positive side however, Alur, Henzinger and Vardi have shown in [3] that reachability is decidable for PTA that contain only *one parametric clock*, yet by an algorithm whose running time is non-elementary.

Reachability in PTA with two or less parametric clocks has not attracted much attention for many years, up until recently.

For PTA over *one parametric clock*, Bundala and Ouaknine have shown a first elementary complexity upper bound for the reachability problem; it is shown to be NEXP-hard and in 2NEXP [10]. The matching NEXP upper bound has been proven by Beneš et al. in [8] (also in the continuous time setting), we refer to [9] for an alternative proof by Bollig, Quaas and Sangnier using alternating two-way automata.

Bundala and Ouaknine [10] have recently advanced the decidability and complexity status of the reachability problem for PTA over *two parametric clocks* [10]: it is shown that in presence of *one parameter* the reachability problem is decidable and hard for the complexity class $\mathsf{PSPACE}^{\mathsf{NEXP}}$. To the best of our knowledge, this is in fact the largest subclass of PTA for which reachability is known to be decidable. For showing the above-mentioned decidability result [10] provides a reduction from PTA over two parametric clocks to a suitable formalism of *parametric one-counter automata*. Such an approach via parametric one-counter automata has already successfully been applied to model checking freeze-LTL as shown by Demri and Sangnier [12] and Lechner et al. [21], yet notably over a weaker model of parametric one-counter automata than the one introduced in [10]. On this note, it is worth mentioning that inter-reductions between the reachability problem of (non-parametric) timed automata involving two clocks and one-counter automata have already been established by Haase et al. [16, 17].

Decidability of reachability in PTA over two parametric clocks (without parameter restrictions) is still considered to be a challenging open problem to the best of our knowledge. For instance, as already remarked in [3], there is an easy reduction from the existential fragment of Presburger Arithmetic with divisibility to reachability in PTA over two parametric clocks.

Our contribution. Our main result (Theorem 4) states that reachability in parametric timed automata over two parametric clocks and one parameter is EXPSPACE-complete. Our contribution is two-fold.

Inspired by [13, 15], for the EXPSPACE lower bound we make use of deep results from complexity theory, namely a serializability characterization of EXPSPACE (in turn originally based on Barrington's Theorem [7]) and a logspace translation of numbers in Chinese Remainder Representation to binary representation due to Chiu, Davida, and Litow [11]. It is shown that with small PTA over two parametric clocks and one parameter one can simulate serializability computations.

For the EXPSPACE upper bound, we first give a careful exponential time reduction from PTA over two parametric clocks and one parameter to a (slight subclass of) parametric one-counter automata over one parameter based on a minor adjustment of a construction due to Bundala and Ouaknine [10]. In solving the reachability problem for parametric one-counter automata with one parameter, we provide a series of techniques to partition a fictitious run into several carefully chosen subruns that allow us to prove that it is sufficient to consider a parameter value of exponential magnitude. This allows us to show a doubly-exponential upper bound on the value of the only parameter of PTA with two parametric clocks and one parameter. We hope that extensions of our techniques lead to finally establishing decidability of the long-standing open problem of reachability in parametric timed automata with two parametric clocks (and arbitrarily many parameters) and, if decidability holds, determining its precise computational complexity.

As the results in [2], our results hold for PTA over discrete time. Indeed, for PTA with closed (i.e., non-strict) clock constraints and parameters ranging over integers, techniques [19, 22] exist that allow to reduce the reachability problem over continuous time to discrete time. There is a plethora of variants of PTA that have recently been studied, we refer to [4] for an extensive overview by André.

Overview of this paper. In Section 2 we introduce general notation, in particular PTA. Our EXPSPACE lower bound can be found in Section 3. Section 4 introduces parametric one-counter automata and states an exponential time reduction from PTA to this model. In Section 5 we introduce semiruns of parametric one-counter automata, a central notion of runs we make modifications on. Our upper bounds are the subject of Section 6. The full version of this paper is available on arXiv [14].

2 Preliminaries

By $\mathbb{N} = \{0, 1, \ldots\}$ we denote the *non-negative integers.* For every finite alphabet A we denote by A^* the set of finite words over A and the empty word by ε. For all $a \in A$ and all $w \in A^*$ let $|w|_a$ denote the number of occurrences of the letter a in w. For every finite set $M \subset \mathbb{N} \setminus \{0\}$ let $\mathsf{LCM}(M) = \min\{n \geq 1 \mid \forall m \in M \setminus \{0\} : m|n\}$ denote the least common multiple of the elements in M. For any $j \in \mathbb{N}$ let $\mathsf{LCM}(j) = \mathsf{LCM}([1, j])$ denote the least common multiple of the numbers $\{1, \ldots, j\}$.

A *guard* over a finite set of clocks Ω and a finite set of parameters P is a comparison of the form $g = \omega \bowtie e$, where $\omega \in \Omega$, $e \in P \cup \mathbb{N}$, and $\bowtie \in \{<, \leq, =, \geq, >\}$; in case $e \in P$ we call g *parametric* and *non-parametric* otherwise. We denote by $\mathcal{G}(\Omega, P)$ the *set of guards* over the set of clocks Ω and the set of parameters P. The *size* $|g|$ of a guard $g = \omega \bowtie e$ is defined as $\log(e)$ if $e \in \mathbb{N}$ and 1 otherwise. A *clock valuation* is a function from Ω to $\mathbb{N}$; we write $\vec{0}$ to denote the clock valuation $\omega \mapsto 0$. For each clock valuation v and each $t \in \mathbb{N}$ we denote by

$v + t$ the clock valuation $\omega \mapsto v(\omega) + t$. A *parameter valuation* is a function μ from P to $\mathbb{N}$. For every guard $g = \omega \bowtie p$ with $p \in P$ (resp. $g = \omega \bowtie k$ with $k \in \mathbb{N}$) we write $v \models_\mu g$ if $v(\omega) \bowtie \mu(p)$ (resp. $v(\omega) \bowtie k$).

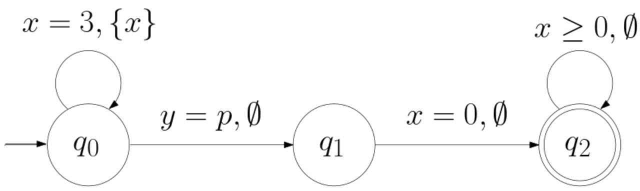

Figure 1 An example of a PTA. The automaton consists of three states, the set of clocks is $\{x, y\}$, the set of parameters is $\{p\}$. The edges are represented by arrows labeled with the corresponding guard and the set of clocks U to be reset. A parameter valuation μ witnesses that reachability holds for this PTA if, and only, if $\mu(p) \in 3\mathbb{Z}$.

A parametric timed automaton as introduced in [3] is a finite automaton extended with a finite set of parameters P and a finite set of clocks Ω that all progress at the same rate and that can be individually reset to zero. Moreover, every transition is labeled by a guard over Ω and P and by a set of clocks to be reset. Formally, a *parametric timed automaton* (*PTA* for short) is a tuple $\mathcal{A} = (Q, \Omega, P, R, q_{init}, F)$, where

- Q is a non-empty finite *set of control states*,
- Ω is a non-empty finite *set of clocks*,
- P is a finite *set of parameters*,
- $R \subseteq Q \times \mathcal{G}(\Omega, P) \times \mathcal{P}(\Omega) \times Q$ is a finite *set of rules*,
- $q_{init} \in Q$ is an *initial control state*, and
- $F \subseteq Q$ is a *set of final control states*.

A clock $\omega \in \Omega$ is called *parametric* if there exists some $(q, g, U, q') \in R$ such that the guard g is parametric. We also refer to $\mathcal{A}$ as an (m, n)-PTA if $m = |\{\omega \in \Omega \mid \omega \text{ is parametric}\}|$ is the number of parametric clocks and $n = |P|$ is the number of parameters of $\mathcal{A}$ – sometimes we also just write $(m, *)$-PTA (resp. $(*, n)$-PTA) when n (resp. m) is a priori not fixed.

The *size* of $\mathcal{A}$ is defined as $|\mathcal{A}| = |Q| + |\Omega| + |P| + |R| + \sum_{(q,g,U,q') \in R} |g|$. Let $\mathsf{Consts}(\mathcal{A})$ denote the set of constants that appear in the guards of the rules of $\mathcal{A}$. By $\mathsf{Conf}(\mathcal{A}) = Q \times \mathbb{N}^\Omega$ we denote the set of *configurations* of $\mathcal{A}$. We prefer however to denote a configuration by $q(v)$ instead of (q, v).

For each parameter valuation $\mu : P \to \mathbb{N}$ and each $(\delta, t) \in R \times \mathbb{N}$ with $\delta = (q, g, U, q')$, let $\xrightarrow{\delta,t,\mu}$ denote the binary relation on $\mathsf{Conf}(\mathcal{A})$, where $q(v) \xrightarrow{\delta,t,\mu} q'(v')$ if $v + t \models_\mu g$, $v'(\omega) = 0$ for all $\omega \in U$ and $v'(\omega) = v(\omega) + t$ for all $\omega \in \Omega \setminus U$. A *$\mu$-run* from $q_0(v_0)$ to $q_n(v_n)$ is a sequence $q_0(v_0) \xrightarrow{\delta_1,t_1,\mu} q_1(v_1) \cdots \xrightarrow{\delta_n,t_n,\mu} q_n(v_n)$. In case $P = \{p\}$ is a singleton and $\mu(p) = N$ we prefer to say *N-run* instead of μ-run and write $q(v) \xrightarrow{N} q'(v')$ to denote $q(v) \xrightarrow{\mu} q'(v')$. We say *reachability holds* for $\mathcal{A}$ if there is a μ-run from $q_{init}(\vec{0})$ to some configuration $q(v)$ for some $q \in F$, some $v \in \mathbb{N}^\Omega$, and some $\mu \in \mathbb{N}^P$. We refer to Figure 1 for an instance of a PTA for which reachability holds.

It is worth mentioning that there are further modes of time valuations and guards which exist in the literature, we refer to [5] for a recent overview.

We are interested in the following decision problem.

(m, n)-PTA-Reachability

INPUT: A (m, n)-PTA $\mathcal{A}$.

QUESTION: Does reachability hold for $\mathcal{A}$?

Alur et al. have already shown in their seminal paper that PTA-Reachability is in general undecidable, already in the presence of only three parametric clocks [3], Beneš et al. strengthened this when only one parameter is present [8].

▶ **Theorem 1** ([8]). $(3,1)$-PTA-Reachability *is undecidable.*

To the contrary, $(1,*)$-PTA-Reachability has recently been shown to be complete for NEXP, where a non-elementary upper bound was initially given by Alur et al. [3].

▶ **Theorem 2** ([10, 8, 9]). $(1,*)$-PTA-Reachability *is* NEXP-*complete.*

On the other end, decidability of $(2,*)$-PTA-Reachability is still open to the best of our knowledge. In presence of precisely one parameter the following is known.

▶ **Theorem 3** ([10]). $(2,1)$-PTA-Reachability *is decidable and* $\mathsf{PSPACE}^{\mathsf{NEXP}}$-*hard.*

The following theorem states our main result.

▶ **Theorem 4.** $(2,1)$-PTA-Reachability *is* EXPSPACE-*complete.*

3 An EXPSPACE lower bound via serializability

In this section we show an EXPSPACE lower bound for $(2,1)$-PTA-Reachability. We show that on small PTA with two parametric clocks x and y and one parameter p one can perform both (i) PSPACE computations and (ii) compute $x - y \bmod p$ modulo numbers that are dynamically given in binary. Building upon these auxiliary gadgets we show how to implement bottleneck computations in a leaf language characterization of EXPSPACE [13]. We assume the reader is familiar with Turing machines and standard complexity classes such as LOGSPACE, PSPACE and EXPSPACE. We refer to [23, 6] for further details on complexity. We also assume the reader is familiar with (deterministic) finite automata and regular languages, we refer to [18] for more details on this.

For each $a, b \in \mathbb{Z}$ we define $[a,b] = \{k \in \mathbb{Z} \mid a \leq k \leq b\}$. For each $i, n \in \mathbb{N}$ let $\textsc{Bit}_i(n)$ denote the i-th least significant bit of the binary presentation of n, i.e. $n = \sum_{i \in \mathbb{N}} 2^i \cdot \textsc{Bit}_i(n)$. For each $m \geq 1$, by $\textsc{Bin}_m(n) = \textsc{Bit}_0(n) \cdots \textsc{Bit}_{m-1}(n)$ we denote the sequence of the first m least significant bits of the binary presentation of n, i.e. the least significant bit is on the left. Conversely, given a binary string $w = w_0 \cdots w_{n-1} \in \{0,1\}^n$ of length n we denote by $\textsc{Val}(w) = \sum_{i=0}^{n-1} 2^i \cdot w_i \in [0, 2^n - 1]$ the value of w interpreted as a non-negative integer.

Let $\mathcal{A}$ be a parametric timed automaton over a set of clocks Ω with two parametric clocks x and y. We say a valuation $v : \Omega \to \mathbb{N}$ is *bit-compatible* if $v(\omega) \in \{0,1\}$ for all non-parametric clocks $\omega \in \Omega$. Assume moreover that Ω contains *non-parametric clocks* $\Theta_+ \cup \Theta_-$, where Θ is some set and $\Theta_+ = \{\vartheta^+ \mid \vartheta \in \Theta\}$ and $\Theta_- = \{\vartheta^- \mid \vartheta \in \Theta\}$ are two disjoint corresponding copies of Θ; in this case, for any valuation $v : \Omega \to \mathbb{N}$ we define the mapping $\widehat{v} : \Theta \to \{0,1\}$ as $\widehat{v}(\vartheta) = 0$ if $v(\vartheta^+) = v(\vartheta^-)$ and $\widehat{v}(\vartheta) = 1$ otherwise. In the following we call such non-parametric clocks $\{\vartheta^+, \vartheta^- \mid \vartheta \in \Theta\}$, appearing as implicit pairs, *bit clocks* since they are used to encode bits.

▶ **Definition 5.** *A* $(2,1)$-*PTA* $\mathcal{A} = (Q, \Omega, \{p\}, R, q_{init}, \{q_{fin}\})$ *whose parametric clocks are* x *and* y *and whose one parameter is* p computes a function $f : \mathbb{N} \times \{0,1\}^n \to \{0,1\}^m$ *if its set of clocks* Ω *contains two disjoint sets of*

- *non-parametric "input" bit clocks* $\{{in_0}^+, {in_0}^-, \ldots, in_{n-1}^+, in_{n-1}^-\}$ *and*
- *non-parametric "output" bit clocks* $\{{out_0}^+, {out_0}^-, \ldots, {out_{m-1}}^+, {out_{m-1}}^-\}$

such that for all $N \in \mathbb{N}$ *and all bit-compatible* $v_0 : \Omega \to [0, N-1]$ *we have*

1. *$q_{init}(v_0) \xrightarrow{N}^{*} q_{fin}(v')$ for some bit-compatible $v' : \Omega \to [0, N-1]$ and*
2. *for all $v' : \Omega \to \mathbb{N}$ for which $q_{init}(v_0) \xrightarrow{N}^{*} q_{fin}(v')$ we have*
   - *$v' \in [0, N-1]^{\Omega}$ is bit-compatible,*
   - *$\widehat{v'}(in_i) = \widehat{v_0}(in_i)$ for all $i \in [0, n-1]$,*
   - *$v'(x) - v'(y) \equiv v_0(x) - v_0(y)$ mod N, and*
   - *$\prod_{j=0}^{m-1} \widehat{v'}(out_j) = f(v_0(x) - v_0(y) \text{ mod } N, \prod_{i=0}^{n-1} \widehat{v_0}(in_i))$, where $\prod$ denotes concatenation.*

The following lemma essentially has its roots in the PSPACE-hardness proof for the emptiness problem for timed automata (without parameters) introduced by Alur and Dill [2], however constructed to satisfy the carefully chosen interface given by Definition 5.

▶ **Lemma 6.** *For every* PSPACE*-computable function $g : \{0,1\}^n \to \{0,1\}^m$ one can compute in polynomial time in $n + m$ a $(2,1)$-PTA computing the function $f : \mathbb{N} \times \{0,1\}^n \to \{0,1\}^m$, where $f(k, w) = g(w)$ for all $(k, w) \in \mathbb{N} \times \{0,1\}^n$.*

The following lemma shows that PTA with two parametric clocks and one parameter can compute modulo dynamically given numbers in binary.

▶ **Lemma 7.** *One can compute in polynomial time in $n+m$ a $(2,1)$-PTA with two parametric clocks and one parameter that computes the function $f : \mathbb{N} \times \{0,1\}^n \to \{0,1\}^m$, where $f(k, w) = \textsc{Bin}_m(k \text{ mod } \textsc{Val}(w))$.*

We are now ready to state the main result of this section.

▶ **Theorem 8.** $(2,1)$-PTA-Reachability *is* EXPSPACE*-hard.*

For each language $L \subseteq A^*$ let $\chi_L : A^* \to \{0,1\}$ denote the characteristic function of L. By $\preceq_n$ we denote the lexicographic order on n-bit strings, thus $w \preceq_n v$ if $\textsc{Val}(w) \leq \textsc{Val}(v)$, e.g. $0101 \preceq_4 0011$.

Our EXPSPACE lower bound proof makes use of the following characterization of EXPSPACE, which is a slight padded adjustment of the leaf-language characterization of PSPACE from [20], which in turn has its roots in Barrington's Theorem [7].

▶ **Theorem 9** (Theorem 2 in [13]). *For every language $L \subseteq \{0,1\}^*$ in* EXPSPACE *there exists a polynomial $s : \mathbb{N} \to \mathbb{N}$, a regular language $R \subseteq \{0,1\}^*$, and a language $U \in$* LOGSPACE *such that for all $w \in \{0,1\}^n$ we have*

$$w \in L \iff \prod_{m=0}^{2^{2^{s(n)}}-1} \chi_U(w \cdot \textsc{Bin}_{2^{s(n)}}(m)) \in R, \tag{1}$$

where $\cdot$ and $\prod$ denote string concatenation.

Let us fix any language L in EXPSPACE and assume $L \subseteq \{0,1\}^*$ without loss of generality.

Applying Theorem 9, let us fix the regular language $R \subseteq \{0,1\}^*$ along with some fixed deterministic finite automaton (DFA for short) $D = (Q_D, \{0,1\}, q_0, \delta_D, F_D)$ with $L(D) = R$, the fixed polynomial s and the fixed language $U \in$ LOGSPACE.

Let us moreover fix an input $w \in \{0,1\}^n$ of length n for L. Figure 2 rephrases the characterization (1) in Theorem 9 in terms of an execution of a program that returns 1 if, and only if, $w \in L$.

The following lemma gives us a helpful initial gadget PTA that allows us to enforce that the parameter p can only be evaluated to numbers that are larger than $2^{2^{s(n)}}$, thus being helpful for storing variables up to the value 2^{2^n}.

```
(1)   var $q \in Q_D$
(2)   var $b \in \{0,1\}$
(3)   var $B \in \mathbb{N}$
(4)   $q := q_0$;
(5)   $B := 0$;
(6)   while $B < 2^{2^n}$ loop
(7)       $b := \chi_U(w \cdot \textsc{Bin}_{2^{s(n)}}(B))$
(8)       $q := \delta_D(q, b)$
(9)       $B := B + 1$
(10)  end loop
(11)  return $q \in F$
```

Figure 2 A program returning 1 if, and only if, $w \in L$ (using the charactization in Theorem 9), where $D = (Q_D, \{0,1\}, q_0, \delta_D, F_D)$ is some deterministic finite automaton such that $L(D) = R$.

▶ **Lemma 10.** *One can compute in polynomial time in n some parametric timed automaton $\mathcal{A}_{big} = (Q_{big}, \Omega_{big}, \{p\}, R_{big}, q_{big,init}, \{q_{big,fin}\})$ with two parametric clocks $x, y \in \Omega_{big}$ and one parameter p such that*

1. *$q_{big,init}(\vec{0}) \xrightarrow{N} q_{big,fin}(v')$ for some $v' : \Omega_{big} \to \mathbb{N}$ for some $N \in \mathbb{N}$, and*
2. *for all $N \in \mathbb{N}$ and all $v' : \Omega_{big} \to \mathbb{N}$ we have $q_{big,init}(\vec{0}) \xrightarrow{N}{}^{*} q_{big,fin}(v')$ implies $N > 2^{2^{s(n)}}$.*

Using the above gadgets one can show that the program in Figure 2 can indeed be simulated by small $(2,1)$-PTA, whose proof we sketch below.

▶ **Lemma 11.** *One can compute in polynomial time in n a $(2,1)$-PTA $\mathcal{A}$ for which reachability holds, if and only if, the execution of the program depicted in Figure 2 returns* 1.

Proof (sketch). The initial part of $\mathcal{A}$ will consist of the gadget PTA $\mathcal{A}_{big}$ from Lemma 10 and allow us to enforce an assignment of $\mathcal{A}$'s only parameter p to some value $N > 2^{2^{s(n)}}$. We store the variable B of the program in Figure 2 as the difference between $\mathcal{A}$'s two parametric clocks x and y modulo N. We only sketch the most crucial program line (7), namely computing the bit $\chi_U(w \cdot \textsc{Bin}_{2^{s(n)}}(B))$, where we recall that U is in LOGSPACE.

For simulating a suitable logspace Turing machine on this exponentially large input our PTA $\mathcal{A}$ will use $O(\log(n + 2^{s(n)})) = \text{poly}(n)$ auxiliary bit clocks, say $\mathcal{J}$, to store in binary the position of the input head and further $O(\log(n + 2^{s(n)})) = \text{poly}(n)$ auxiliary bit clocks for storing the working tape. Reading and writing on the working tape as well as updating the position of the input head can be done quite straightforwardly using polynomially many bit clocks. The main challenge is to access the cell content $\textsc{Bit}_j(w \cdot \textsc{Bin}_{2^{s(n)}}(B))$, where the address j can directly (in binary) be stored using the above-mentioned bit clocks $\mathcal{J}$.

To compute $\textsc{Bit}_j(w \cdot \textsc{Bin}_{2^{s(n)}}(B))$ we access B on the fly via its Chinese Remainder Representation $\mathsf{CRR}(B)$ that we define next: Let p_i denote the i-th prime number and assume $\prod_{i=1}^{m} p_i > B$ for some $m \in \mathbb{N}$, then $\mathsf{CRR}_m(B)$ denotes the bit tuple $(b_{i,r})_{i \in [1,m], r \in [0, p_i - 1]}$, where $b_{i,r} = 1$ if $B \bmod p_i = r$ and $b_{i,r} = 0$ otherwise. The individual input bits to $\mathsf{CRR}(B)$ can be sub-computed via our modulo gadget from Lemma 7. The individual input bits to $\textsc{Bin}_{2^{s(n)}}(B)$ can be obtained by a composition of the latter access to $\mathsf{CRR}(B)$ and simulating a logspace Turing machine that computes $\textsc{Bin}_{2^{s(n)}}(B)$ from $\mathsf{CRR}(B)$ by a result by Chiu, Davida, and Litow [11]. ◀

4 From two-parametric timed automata with one parameter to parametric one-counter automata

Being introduced by Bundala and Ouaknine in [10], we define parametric one-counter automata. These are automata that can manipulate a counter that can be incremented or decremented, parametrically or not, compared against constants or parameters, and with divisibility tests modulo constants. It is worth mentioning that the notion of parametric one-counter automata from [10] is slightly more expressive than ours, allowing more operations.

After introducing parametric one-counter automata we mention Theorem 13, proven essentially already in [10] – again, however for a slightly more expressive model of parametric one-counter automata – that states that $(2,1)$-PTA-Reachability can be reduced in exponential time to the reachability problem of parametric one-counter automata over one parameter.

Given a set of parameters P we denote by $\mathsf{Op}(P)$ the *set of operations*, namely $\mathsf{Op}(P) = \mathsf{Op}_{\pm} \cup \mathsf{Op}_{\pm P} \cup \mathsf{Op}_{\mathsf{mod}\,\mathbb{N}} \cup \mathsf{Op}_{\bowtie\mathbb{N}} \cup \mathsf{Op}_{\bowtie P}$, where

- $\mathsf{Op}_{\pm} = \{-1, 0, +1\}$, $\mathsf{Op}_{\pm P} = \{+p, -p \mid p \in P\}$,
- $\mathsf{Op}_{\mathsf{mod}\,\mathbb{N}} = \{\mathsf{mod}\ c \mid c \in \mathbb{N}\}$,
- $\mathsf{Op}_{\bowtie\mathbb{N}} = \{\bowtie c \mid \bowtie \in \{<, \leq, =, \geq, >\}, c \in \mathbb{N}\}$, and $\mathsf{Op}_{\bowtie P} = \{\bowtie p \mid \bowtie \in \{<, \leq, =, \geq, >\}, p \in P\}$.

The *size* $|op|$ of an operation op is defined as $|op| = \log(c)$ if $op = \mathsf{mod}\ c$ or $op = \bowtie c$ with $c \in \mathbb{N}$ and $|op| = 1$ otherwise. We denote by *updates* those operations that lie in $\mathsf{Op}_{\pm} \cup \mathsf{Op}_{\pm P}$ and by *tests* those operations that lie in $\mathsf{Op}_{\mathsf{mod}\,\mathbb{N}} \cup \mathsf{Op}_{\bowtie\mathbb{N}} \cup \mathsf{Op}_{\bowtie P}$.

A *parametric one-counter automaton* (POCA for short) is a tuple $\mathcal{C} = (Q, P, R, q_{init}, F)$, where Q is a non-empty finite *set of control states*, P is a non-empty finite *set of parameters* that can take non-negative integer values, $R \subseteq Q \times \mathsf{Op}(P) \times Q$ is a finite *set of rules*, q_{init} is an *initial control state*, and $F \subseteq Q$ is a *set of final control states*. The *size* of $\mathcal{C}$ is defined as $|\mathcal{C}| = |Q| + |P| + |R| + \sum_{(q,op,q') \in R} |op|$. Let $\mathsf{Consts}(\mathcal{C})$ denote the constants that appear in the operations $op \in \mathsf{Op}_{\mathsf{mod}\,\mathbb{N}} \cup \mathsf{Op}_{\bowtie\mathbb{N}}$ for some rule (q, op, q') in R. By $\mathsf{Conf}(\mathcal{C}) = Q \times \mathbb{Z}$ we denote the set of *configurations* of $\mathcal{C}$. We prefer however to denote a configuration of $\mathsf{Conf}(\mathcal{C})$ by $q(z)$ instead of (q, z).

Being slightly non-standard we define configurations to take counter values over $\mathbb{Z}$ rather than over $\mathbb{N}$ for notational convenience. This does not cause any loss of generality as we allow guards that enable us to test if the value of the counter is greater or equal to zero.

▶ **Definition 12** (transition). *For every $op \in \mathsf{Op}(P)$, for every parameter valuation $\mu : P \to \mathbb{N}$, for every POCA $\mathcal{C}$, and for every two configurations $q(z)$ and $q'(z')$ in $\mathsf{Conf}(\mathcal{C})$ we define the* transition *$q(z) \xrightarrow{op,\mu} q'(z')$ if there exists some $(q, op, q') \in R$ such that either of the following holds*

(1) *$op = c \in \mathsf{Op}_{\pm}$ and $z' = z + c$,*

(2) *$op \in \mathsf{Op}_{\pm P}$, and either $op = +p$ and $z' = z + \mu(p)$, or $op = -p$ and $z' = z - \mu(p)$.*

(3) *$op = \mathsf{mod}\ c \in \mathsf{Op}_{\mathsf{mod}\,\mathbb{N}}$, $z = z'$ and $z' \equiv 0\ \mathsf{mod}\ c$,*

(4) *$op = \bowtie c \in \mathsf{Op}_{\bowtie\mathbb{N}}$, $z = z'$ and $z' \bowtie c$, or*

(5) *$op = \bowtie p \in \mathsf{Op}_{\bowtie P}$, $z = z'$ and $z' \bowtie \mu(p)$.*

Let $\mu : P \to \mathbb{N}$ be a parameter valuation. We just write $q(z) \xrightarrow{\mu} q'(z')$ if $q(z) \xrightarrow{op,\mu} q'(z')$ for some operation op. A *μ-run (or just run)* in $\mathcal{C}$ (from $q_0(z_0)$ to $q_n(z_n)$) is a sequence, possibly empty (i.e. $n = 0$), of the form $\pi = q_0(z_0) \xrightarrow{op_0,\mu} q_1(z_1) \cdots \xrightarrow{op_{n-1},\mu} q_n(z_n)$.

We say π is *accepting* if $q_0 = q_{init}$, $z_0 = 0$, and $q_n \in F$. We say *reachability holds* (for the POCA $\mathcal{C}$) if there exists an accepting μ-run for some $\mu \in \mathbb{N}^P$. We refer to Figure 3 for an

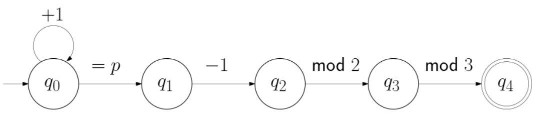

Figure 3 An example of a POCA. The automaton consists of five states and the set of parameters is $\{p\}$. The edges are represented by arrows labeled with the corresponding operations. A parameter valuation $\mu : \{p\} \to \mathbb{N}$ witnesses that reachability holds for the above POCA if, and only, if $\mu(p) \equiv 1 \bmod 6$.

instance of a POCA for which reachability holds. For any two $c, d \in [0, n]$ we define the *subrun* $\pi[c, d]$ *from* $q_c(z_c)$ *to* $q_d(z_d)$ *of* π as the μ-run $q_c(z_c) \xrightarrow{\pi_c,\mu} q_{c+1}(z_{c+1}) \cdots \xrightarrow{\pi_{d-1},\mu} q_d(z_d)$.

As expected, a *prefix* (resp. *suffix*) of π is an μ-run of the form $\pi[0, d]$ (resp. $\pi[d, n]$). In the particular case where $P = \{p\}$ is a singleton for some parameter p and $\mu(p) = N$, we write $q(z) \xrightarrow{op,N} q'(z')$ to denote $q(z) \xrightarrow{op,\mu} q'(z')$ and prefer to call a μ-run an N*-run*.

We define $\Delta(\pi) = z_n - z_0$ as the *counter effect* of the run π and for each $i \in [0, n-1]$ we define $\Delta(\pi, i) = \Delta(\pi[i, i+1])$ to denote the counter effect of the i-th transition of π. Its *length* is defined as $|\pi| = n$. As expected, let $\textsc{Values}(\pi) = \{z_i \mid i \in [0, n]\}$ denote the set of counter values of the configurations of the run π. The run π's *maximum* is defined as $\max(\pi) = \max(\textsc{Values}(\pi))$ and its *minimum* as $\min(\pi) = \min(\textsc{Values}(\pi))$.

The next theorem states an exponential time reduction from $(2, 1)$-PTA-Reachability to the reachability problem of POCA over one parameter whose counter values are bound by a linear function in the parameter value and its size. It has already been proven in the more general setting over an arbitrary number of parameters in [10], however using a POCA model allowing more operations.

▶ **Theorem 13.** *The following is computable in exponential time:*
INPUT: A $(2, 1)$-PTA $\mathcal{A}$.
OUTPUT: A POCA $\mathcal{C}$ over one parameter p such that

1. *for all $N \in \mathbb{N}$ all accepting N-runs π in $\mathcal{C}$ satisfy $\textsc{Values}(\pi) \subseteq [0, 4 \cdot \max(N, |\mathcal{C}|)]$, and*
2. *reachability holds for $\mathcal{A}$ if, and only if, reachability holds for $\mathcal{C}$.*

5 Semiruns, their bracket projection, and embeddings

In this section we motivate and introduce the notion of semiruns by loosening the conditions on runs, and define basic operations on them. These basic operations possibly change their counter values, length, or counter effect. We finally introduce the notion of embeddings, which provide a formal means to express when a semirun can structurally be found as a subsequence of another.

▶ **Definition 14** (semitransition). *Let $\mathcal{C} = (Q, P, R, q_{init}, F)$ be a POCA. For every operation $op \in \mathsf{Op}(P)$ and every $N \in \mathbb{N}$ and for every two configurations $q(z)$ and $q'(z')$ in $\mathsf{Conf}(\mathcal{C})$ we define the* semitransition $q(z) \xRightarrow{op,\mu} q'(z')$ *if there exists some $(q, op, q') \in R$ such that conditions (1),(2), and (3) hold and where conditions (4) and (5) are loosened by*

(4') *$op = \bowtie c \in \mathsf{Op}_{\bowtie\mathbb{N}}$ and $z = z'$, and*
(5') *$op = \bowtie p \in \mathsf{Op}_{\bowtie P}$, and $z = z'$.*

Thus, in a nutshell, when writing $q(z) \xRightarrow{op,\mu} q'(z')$ we do not require that the comparison tests against parameters or against constants hold; however the updates and the modulo tests against constants must be respected.

This naturally gives rise to the definition of a *μ-semirun*, which is defined as expected. Note that in particular every μ-run is a μ-semirun. The notion of an N-semirun, the relation $q(z) \xRightarrow{op,N} q'(z')$, the counter effect Δ, Values, min, max, subsemirun, prefix, suffix are defined as for runs.

Importantly, note also that semitransitions involving comparison tests are still syntactically present in semiruns. By a careful analysis, one can therefore possibly perform operations on N-semiruns in order to show that they are in fact N-runs.

Let Γ be any integer that is divisible by all constants in $\mathsf{Consts}(\mathcal{C})$ in some POCA $\mathcal{C}$. We define the *shifting* of an N-semirun π by Γ as $\pi + \Gamma \ = \ q_0(z_0 + \Gamma) \xRightarrow{\pi_0,N} q_1(z_1 + \Gamma) \cdots \xRightarrow{\pi_{n-1},N} q_n(z_n + \Gamma)$. Since there are no effective comparison tests and Γ is an integer that is divisible by all constants appearing in modulo tests in $\mathcal{C}$, it is clear that $\pi + \Gamma$ is again an N-semirun.

For two configurations $q_i(z_i)$ and $q_j(z_j)$ with $0 \leq i < j \leq n$, $q_i = q_j$, and $z_j - z_i = \Gamma$ we define the *gluing* of the configurations as

$$\pi - [i,j] \ = \ q_0(z_0) \ \cdots \xRightarrow{\pi_{i-1},N} q_i(z_i) \xRightarrow{\pi_j,N} q_{j+1}(z_{j+1} - \Gamma) \ \cdots \xRightarrow{\pi_{n-1},N} q_n(z_n - \Gamma).$$

When gluing the leftmost and rightmost configurations of pairwise non-intersecting intervals $I_1 = [a_1, b_1], \ldots, I_k = [a_k, b_k] \subseteq [0, n]$, assuming $b_i < a_{i+1}$ for all $1 \leq i < k$, and $q_{a_i} = q_{b_i}$ and $z_{b_i} - z_{a_i}$ is divisible by all constants in $\mathsf{LCM}(\mathsf{Consts}(\mathcal{C}))$ for all $1 \leq i \leq k$, we will use $\pi - I_1 - I_2 \cdots - I_k$ to denote the result corresponding to gluing each interval successively while shifting the others accordingly, formally $\pi^{(k)}$, where $\pi^{(1)} = \pi - [a_1, b_1]$ and inductively, $\pi^{(s)} = \pi^{(s-1)} - [a_s - \Sigma_{1 \leq j < s}(|I_j| - 1), b_s - \Sigma_{1 \leq j < s}(|I_j| - 1)]$ for all $s \in [2, k]$.

We define the projection ϕ of a semitransition $\tau = q(z) \xRightarrow{op,N} q'(z')$ to a word over the binary alphabet $\{[,]\}$, where transitions with $op = +p$ are mapped to $[$, transitions with $op = -p$ are mapped to $]$, and all other transitions are mapped to the empty word ε. The mapping ϕ is naturally extended to a morphism on semiruns.

We are particularly interested in N-semiruns whose projection by ϕ contains as many opening as closing brackets and only a few pending (when read from left to right) opening or closing brackets. To make this formal, for all $k \in \mathbb{N}$ we define regular language

$$\Lambda_k = \left\{ w \in \{[,]\}^* : |w|_{[} = |w|_{]}, \forall u, v \in \{[,]\}^*.\ uv = w \implies |u|_{[} - |u|_{]} \in [-k, k] \right\}.$$

We will often prefer to view N-runs as N-semiruns. Indeed, in case N is sufficiently large we first view any N-run as an N-semirun, apply certain of the above-mentioned operations on them to obtain some $(N - \Gamma)$-semirun, where Γ is divisible by all constants appearing in the underlying POCA. However, we would then like to claim that the resulting $(N - \Gamma)$-semirun is in fact an $(N - \Gamma)$-run as desired, in particular the comparison tests need to hold. To do so, we introduce a notion when an N-semirun can be embedded into an M-semirun (possibly $N \neq M$) in the sense that operations are being preserved, source and target control states are being preserved and that with respect to some line $\ell \in \mathbb{Z}$ the counter value of each configuration of the embedding has the same orientation with respect to ℓ as the counter value of the configuration it corresponds to.

▶ **Definition 15** (ℓ-embedding). *Let $\ell \in \mathbb{Z}$. An N-semirun $\sigma = s_0(y_0) \ \cdots \xRightarrow{\sigma_{n-1},N} s_n(y_n)$ is an* ℓ-embedding *of an M-semirun $\pi = q_0(z_0) \ \cdots \xRightarrow{\pi_{m-1},M} q_m(z_m)$ if $s_0 = q_0$, $s_n = q_m$ and there exists an order-preserving injective mapping $\psi : [0, n] \to [0, m]$ such that*

- *$\sigma_i = \pi_{\psi(i)}$ for all $i \in [0, n-1]$, and*
- *$\ell \bowtie y_i$ if, and only if, $\ell \bowtie z_{\psi(i)}$ for all $\bowtie \in \{<, =, >\}$ and all $i \in [0, n]$.*

Moreover we say σ is max-falling, *resp.* min-rising, *w.r.t. π if $\max(\sigma) \leq \max(\pi)$, resp. if $\min(\sigma) \geq \min(\pi)$.*

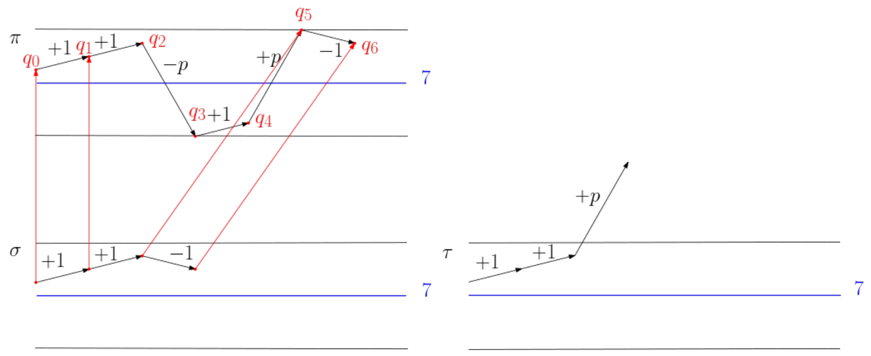

Figure 4 Example of a semirun σ that could possibly be a 7-embedding of the semirun π and a semirun τ that cannot.

Consider the semiruns π, σ and τ in Figure 4, where neither concrete counter values nor the control states of σ nor τ are mentioned. The semirun σ can possibly be an 7-embedding of π (if its source control control is q_0 and its target control state is q_6). However, τ cannot be a 7-embedding of π. Indeed, for every possible injection ψ such that $\tau_2 = +p = \pi_{\psi(2)}$, the counter value of τ at position 2 is strictly larger than 7, whereas the counter value of π at position $\psi(2)$ is strictly below 7.

It is immediate that an ℓ-embedding of an ℓ-embedding is again an ℓ-embedding. Moreover, if the target configuration of σ equals the source configuration of τ and σ and τ are ℓ-embeddings of σ' and τ' respectively, and $\sigma'\tau'$ is a semirun, then so is their concatenation $\sigma\tau$ an ℓ-embedding of $\sigma'\tau'$. Such basic properties will be used extensively in our proofs.

6 Upper bounds

In this section we state the Small Parameter Theorem which states that every POCA over one parameter and every sufficiently large parameter value N, accepting N-runs with counter values all in $[0, 4N]$ can be turned into accepting N'-runs for some smaller N'. After having stated the theorem one can show that together with Theorem 13 it implies an EXPSPACE upper bound for $(2,1)$-PTA-Reachability.

For each POCA $\mathcal{C} = (Q, P, R, q_{init}, F)$ we define the following constants:

$$\begin{array}{|cc|}\hline Z_{\mathcal{C}} = \mathsf{LCM}(\mathsf{Consts}(\mathcal{C})) & \Gamma_{\mathcal{C}} = \mathsf{LCM}(17 \cdot |Q|) \cdot Z_{\mathcal{C}} \\ \Upsilon_{\mathcal{C}} = 17 \cdot |Q| \cdot \mathsf{LCM}(17 \cdot |Q|) \cdot (17 \cdot |Q| \cdot Z_{\mathcal{C}} + 2) & M_{\mathcal{C}} = 30 \cdot (\Upsilon_{\mathcal{C}} + \Gamma_{\mathcal{C}} + 1) \\ \hline\end{array}$$

Since for every non-empty finite set $U \subseteq \mathbb{N} \setminus \{0\}$ we have $\mathsf{LCM}(U) \leq \max(U)^{|U|}$, all of the above constants are asymptotically bounded by $2^{\mathsf{poly}(|\mathcal{C}|)}$.

The main result of this section is the following theorem.

▶ **Theorem 16** (Small Parameter Theorem). *Let $\mathcal{C} = (Q, \{p\}, R, q_{init}, F)$ be a POCA with one parameter. If there exists an accepting N-run in $\mathcal{C}$ with values in $[0, 4N]$ for some $N > M_{\mathcal{C}}$, then there exists an accepting $(N - \Gamma_{\mathcal{C}})$-run in $\mathcal{C}$.*

The Small Parameter Theorem has the following consequence for $(2,1)$-PTA-Reachability.

▶ **Corollary 17.** $(2,1)$-PTA-Reachability *is in* EXPSPACE.

Overview of the proof of the Small Parameter Theorem

The Small Parameter Theorem (Theorem 16) states that, in case N is sufficiently large, accepting N-runs whose configurations have counter values all inside $[0,4N]$ can be turned into accepting $(N-\Gamma_{\mathcal{C}})$-runs. For its proof we proceed as follows. As mentioned already in Section 5 we prefer to view N-runs as N-semiruns.

Manipulating only N-semiruns, the following Depumping Lemma can turn N-semiruns whose Δ is either sufficiently large (resp. sufficiently small) again into N-semiruns whose Δ is less large (resp. small). It requires however an N-run whose ϕ-projection has a nice bracketing property, namely a ϕ-projection that lies in the regular language Λ_8.

▶ **Lemma 18** (Depumping Lemma). *For all N-semiruns π satisfying $\phi(\pi) \in \Lambda_8$ and $|\Delta(\pi)| > \Upsilon_{\mathcal{C}}$ there exists an N-semirun π' such that either*

- $\Delta(\pi) > \Upsilon_{\mathcal{C}}$ *and* $\Delta(\pi') = \Delta(\pi) - \Gamma_{\mathcal{C}}$, *or*
- $\Delta(\pi) < -\Upsilon_{\mathcal{C}}$ *and* $\Delta(\pi') = \Delta(\pi) + \Gamma_{\mathcal{C}}$.

Moreover, $\pi' = \pi - I_1 - I_2 \ \cdots \ - I_k$ for pairwise disjoint intervals $I_1, \ldots, I_k \subseteq [0, |\pi|]$ such that we have $\phi(\pi[I_i]) \in \Lambda_{16}$ for all $i \in [1,k]$, and either $\Delta(\pi[I_i]) > 0$ for all $i \in [1,k]$ or $\Delta(\pi[I_i]) < 0$ for all $i \in [1,k]$.

Proof. Let $\pi = q_0(z_0) \xRightarrow{\pi_0,N} q_1(z_1) \xRightarrow{\pi_1,N} \ \cdots \ \xRightarrow{\pi_{n-1},N} q_n(z_n)$ be an N-semirun such that $\phi(\pi) \in \Lambda_8$. We will assume without loss of generality that $\Delta(\pi) > \Upsilon_{\mathcal{C}}$. The dual case when $\Delta(\pi) < -\Upsilon_{\mathcal{C}}$ can be proven analogously.
For every position $i \in [0,n]$ let us define

$$\lambda(i) = |\phi(\pi[0,i])|_{[} - |\phi(\pi[0,i])|_{]} \qquad \text{and} \qquad \mathsf{pot}(i) = z_i - z_0 - \lambda(i) \cdot N \qquad .$$

Note that since $\phi(\pi) \in \Lambda_8$ by assumption we have for all $i \in [0,n]$,

$$\lambda(i) \in [-8, 8], \tag{2}$$

and moreover

$$\phi(\pi[0,i]) \in \Lambda_8 \iff \lambda(i) = 0. \tag{3}$$

We note the following important properties of pot,

1. $|\mathsf{pot}(i-1) - \mathsf{pot}(i)| \leq 1$ for all $i \in [1,n]$,
2. $\mathsf{pot}(0) = 0$,
3. for all $0 \leq i < j \leq n$, if $\lambda(i) = \lambda(j)$, then $\mathsf{pot}(j) - \mathsf{pot}(i) = z_j - z_i$, and
4. $\mathsf{pot}(n) = z_n - z_0 = \Delta(\pi)$ since $\lambda(0) = \lambda(n) = 0$.

The following claim states that if in a subsemirun pot increases sufficiently, one can find a subsemirun therein that can potentially be glued.

▷ Claim 19. For each subsemirun $\pi[a,b]$ that satisfies $\mathsf{pot}(b) - \mathsf{pot}(a) > 17 \cdot |Q| \cdot Z_{\mathcal{C}}$ there exist positions $a \leq s < t \leq b$, such that

- $q_s = q_t$,
- $\lambda(s) = \lambda(t)$, and
- $z_t - z_s = dZ_{\mathcal{C}}$ for some $d \in [1, 17 \cdot |Q|]$.

Proof of the Claim. Since by assumption $\mathsf{pot}(b) - \mathsf{pot}(a) > 17 \cdot |Q| \cdot Z_{\mathcal{C}}$, by the pigeonhole principle and Point 1 above, there exist two indices $a \leq s < t \leq b$ such that $q_s = q_t$, $\lambda(s) \in [-8,8]$ and $\lambda(t) \in [-8,8]$ are equal, and $\mathsf{pot}(t) - \mathsf{pot}(s) = dZ_{\mathcal{C}}$ for some $d \in [1, 17 \cdot |Q|]$. By Point 3 above, from $\lambda(t) = \lambda(s)$, it follows $z_t - z_s = \mathsf{pot}(t) - \mathsf{pot}(s) = dZ_{\mathcal{C}}$. ◁

Since $\mathsf{pot}(i) - \mathsf{pot}(i-1) \leq 1$ for all $i \in [1, n]$ by Point 1 above and

$$\begin{aligned} \mathsf{pot}(n) - \mathsf{pot}(0) &= z_n - z_0 \\ &= \Delta(\pi) \\ &> \Upsilon_{\mathcal{C}} \\ &\overset{\text{page 11}}{=} 17 \cdot |Q| \cdot \mathsf{LCM}(17 \cdot |Q|) \cdot (17 \cdot |Q| \cdot Z_{\mathcal{C}} + 2)\,, \end{aligned}$$

by the pigeonhole principle, there exist at least

$$17 \cdot |Q| \cdot \mathsf{LCM}(17 \cdot |Q|)$$

pairwise disjoint subsemiruns $\pi[a, b]$ satisfying $\mathsf{pot}(b) - \mathsf{pot}(a) > 17 \cdot |Q| \cdot Z_{\mathcal{C}}$. Let

$$L = \mathsf{LCM}(17 \cdot |Q|),$$

and let $\pi[a_1, b_1], \ldots, \pi[a_{17\cdot|Q|\cdot L}, b_{17\cdot|Q|\cdot L}]$ be an enumeration of these latter subsemiruns. We apply the above Claim to all of these $\pi[a_i, b_i]$: there exist positions $a_i \leq s_i \leq t_i \leq b_i$ such that $\lambda(s_i) = \lambda(t_i)$, $q_{s_i} = q_{t_i}$, and $z_{t_i} = z_{s_i} + d_i Z_{\mathcal{C}}$ for some $d_i \in [1, 17 \cdot |Q|]$. From $\lambda(s_i) = \lambda(t_i)$ and (2) it follows $\phi(\pi[s_i, t_i]) \in \Lambda_{16}$. Recall that $\Gamma_{\mathcal{C}} = \mathsf{LCM}(17 \cdot |Q|) \cdot Z_{\mathcal{C}} = L \cdot Z_{\mathcal{C}}$ by definition on page 11. By the pigeonhole principle, among these $17 \cdot |Q| \cdot L$ pairwise disjoint subsemiruns $\pi[a_i, b_i]$, there exists some $d \in [1, 17 \cdot |Q|]$ such that there are L/d many different $\pi[a_i, b_i]$ all satisfying $d_i = d$. Let $\pi[a_{i_1}, b_{i_1}], \ldots, \pi[a_{i_{L/d}}, b_{i_{L/d}}]$ be an enumeration of these latter $\pi[a_i, b_i]$. Note that for all of these $\pi[a_i, b_i]$ we have $\Delta(\pi[s_{i_j}, t_{i_j}]) = d \cdot Z_{\mathcal{C}}$. Since moreover $q_{s_{i_j}} = q_{t_{i_j}}$ we know that, for all $j \in [1, L/d]$, the gluing $\pi - [s_{i_j}, t_{i_j}]$ is an N-semirun with $\Delta(\pi - [s_{i_j}, t_{i_j}]) = \Delta(\pi) - dZ_{\mathcal{C}}$. Thus,

$$\pi' \quad = \quad \pi - [s_{i_1}, t_{i_1}] - \ldots - [s_{i_{L/d}}, t_{i_{L/d}}]$$

is an N-semirun satisfying $\Delta(\pi') = \Delta(\pi) - d \cdot (L/d) \cdot Z_{\mathcal{C}} = \Delta(\pi) - \Gamma_{\mathcal{C}}$ as required. ◀

Assuming N to be sufficiently large the Bracket Lemma shows that for every $(N - \Gamma_{\mathcal{C}})$-semirun whose Δ is again sufficiently large (resp. sufficiently small) and whose ϕ-projection satisfies a majority condition, the existence of a subsemirun whose Δ is again sufficiently large (resp. sufficiently small) but which has a ϕ-projection that lies in the regular language Λ_8. Since the resulting subsemiruns have a desirable ϕ-projection, the Depumping Lemma can be applied to these. Combining these remarks allows us to construct a new semirun whose Δ is slightly smaller (resp. bigger) than the Δ of the original semirun the Bracket Lemma was applied to.

▶ **Lemma 20** (Bracket Lemma). *For all $N > M_{\mathcal{C}}$ and for all $(N - \Gamma_{\mathcal{C}})$-semiruns π satisfying $\textsc{Values}(\pi) \subseteq [0, 4N]$, $\Delta(\pi) < -\Upsilon_{\mathcal{C}}$ (resp. $\Delta(\pi) > \Upsilon_{\mathcal{C}}$) and where $\phi(\pi)$ contains at least as many occurrences of* [*as occurrences of*] *(resp. at least as many occurrences of*] *as occurrences of* [*) there exists a subsemirun $\pi[c, d]$ satisfying $\phi(\pi[c, d]) \in \Lambda_8$ and $\Delta(\pi[c, d]) < -\Upsilon_{\mathcal{C}}$ (resp. $\Delta(\pi[c, d]) > \Upsilon_{\mathcal{C}}$).*

Note that trivially, every N-semirun ρ with $\phi(\rho) = \varepsilon$ is already an $(N - \Gamma_{\mathcal{C}})$-semirun. Let us exemplify an interplay between the Bracket Lemma and the Depumping Lemma. For instance, assume we are to turn the following N-semirun into an $(N - \Gamma_{\mathcal{C}})$-semirun with the same source and target configuration, namely an N-semirun of the form $\tau\rho$, where τ is a $+p$-transition (thus a length one N-semirun), $\phi(\rho) = \varepsilon$, and $\Delta(\rho) < -\Upsilon_{\mathcal{C}}$: indeed, firstly one can explicitly turn the $+p$-transition τ into the $+p$-transition $\widehat{\tau}$ with $\Delta(\widehat{\tau}) = N - \Gamma_{\mathcal{C}}$ (thus a length one $(N - \Gamma_{\mathcal{C}})$-semirun) and secondly apply (by observing that ρ is already

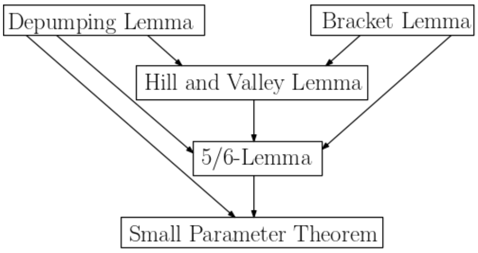

Figure 5 Illustration of the dependencies between the lemmas. The presence of an arrow going from a lemma to another means that the lemma in question is used inside the proof of the lemma the arrow points to.

an $(N - \Gamma_{\mathcal{C}})$-semirun) the Bracket Lemma to ρ. Using the interplay between the Bracket Lemma and the Depumping Lemma one can obtain an $(N - \Gamma_{\mathcal{C}})$-semirun $\widehat{\rho}$ (obtained by gluing and shifting) such that $\widehat{\tau}\,\widehat{\rho}$ is an $(N - \Gamma_{\mathcal{C}})$-semirun with the same source and target configuration as $\tau\rho$.

The following notion of hills and valleys provides a more general class of semiruns to which the above-mentioned reasoning in the previous paragraph can be applied. B-hills are semiruns that start and end in configurations with low counter values but where all intermediate configurations have counter values above these source and target configurations, and where moreover $+p$-transitions (resp. $-p$-transitions) are followed (resp. preceded) by semiruns with counter effect strictly smaller than $-\Upsilon_{\mathcal{C}}$ (resp. strictly larger than $\Upsilon_{\mathcal{C}}$). B-valleys are defined dually.

Formally let $q_0(z_0) \xRightarrow{\pi_0, N} q_1(z_1) \cdots \xRightarrow{\pi_{n-1}, N} q_n(z_n)$ be an N-semirun. It is a *B-hill* if $z_0, z_n < B$, $z_i \geq B$ for all $i \in [1, n-1]$, $\pi_i = -p$ implies $z_i > z_0 + \Upsilon_{\mathcal{C}}$ for all $i \in [0, n-1]$, and $\pi_i = +p$ implies $z_{i+1} > z_n + \Upsilon_{\mathcal{C}}$ for all $i \in [0, n-1]$. Dually, it is a *B-valley* if $z_0, z_n > B$, $z_i \leq B$ for all $i \in [1, n-1]$, $\pi_i = -p$ implies $z_{i+1} < z_n - \Upsilon_{\mathcal{C}}$ for all $i \in [0, n-1]$, and $\pi_i = +p$ implies $z_i < z_0 - \Upsilon_{\mathcal{C}}$ for all $i \in [0, n-1]$.

The Hill and Valley Lemma (Lemma 21) allows us to transform N-semiruns that are B-hills (resp. B-valleys) into $(N - \Gamma_{\mathcal{C}})$-semiruns with the same source and target configurations.

▶ **Lemma 21** (Hill and Valley Lemma). *For all $N, B \in \mathbb{N}$ and all N-semiruns π from $q_0(z_0)$ to $q_n(z_n)$ with $N > M_{\mathcal{C}}$ and $\text{VALUES}(\pi) \subseteq [0, 4N]$ such that moreover π is either a B-hill or a B-valley, there exists an $(N - \Gamma_{\mathcal{C}})$-semirun from $q_0(z_0)$ to $q_n(z_n)$ that is both a min-rising and max-falling $(B - \Upsilon_{\mathcal{C}} - \Gamma_{\mathcal{C}} - 1)$-embedding of π (in case π is a B-hill), or both a min-rising and max-falling $(B + \Upsilon_{\mathcal{C}} + \Gamma_{\mathcal{C}} + 1)$-embedding of π (in case π is a B-valley).*

By carefully factorizing N-semiruns with a Δ smaller than $5/6 \cdot N$ into suitably chosen hills and valleys, one can turn them into $(N - \Gamma_{\mathcal{C}})$-semiruns that are moreover ℓ-embeddings for every ℓ that is not far away from the counter values of both the source and target configuration. The following lemma makes this more formal.

▶ **Lemma 22** (5/6-Lemma). *For all $N > M_{\mathcal{C}}$ and all $\ell \in \mathbb{Z}$ and all N-semiruns π from $q_0(z_0)$ to $q_n(z_n)$ with $\text{VALUES}(\pi) \subseteq [0, 4N]$ satisfying $\max(z_0, z_n, \ell) - \min(z_0, z_n, \ell) \leq 5/6 \cdot N$ there exists an $(N - \Gamma_{\mathcal{C}})$-semirun π' from $q_0(z_0)$ to $q_n(z_n)$ that is an ℓ-embedding of π such that $\text{VALUES}(\pi') \subseteq [\min(\pi) - \Gamma_{\mathcal{C}}, \max(\pi) + \Gamma_{\mathcal{C}}]$.*

Figure 5 provides an overview of the dependencies of the above-mentioned lemmas.

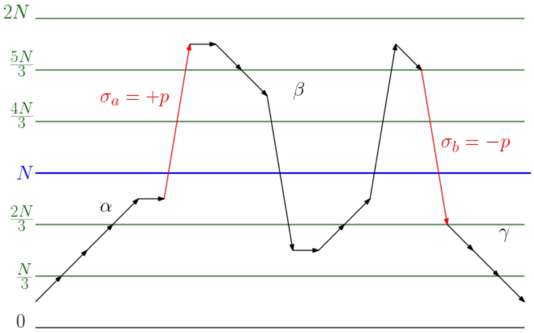

Figure 6 Application of the 5/6-Lemma to the subrun $\sigma[a, b+1]$.

Let us exemplify how the 5/6-Lemma is used in proving the Small Parameter Theorem. For this let us fix some POCA $\mathcal{C}$ over a parameter p, some $N > M_{\mathcal{C}}$ and an accepting N-run π with $\textsc{Values}(\pi) \subseteq [0, 4N]$, where π is of the form $\pi = r_0(x_0) \xrightarrow{\pi_0,N} r_1(x_1) \cdots \xrightarrow{\pi_{n-1},N} r_n(x_n)$ and where $r_n \in F$. We need to show the existence of an accepting $(N - \Gamma_{\mathcal{C}})$-run. We may assume $x_n = 0$ w.l.o.g. (by simply requiring a final zero test in a new PTA).

Since $\frac{N}{3} < N - \Gamma_{\mathcal{C}}$, by definition of the constants on page 11, every subrun ρ of π with $\textsc{Values}(\rho) \subseteq [0, \frac{N}{3}[$ is already an $(N - \Gamma_{\mathcal{C}})$-run. One can therefore uniquely factorize π as $\pi = \rho^{(0)}\sigma^{(1)}\rho^{(1)} \cdots \sigma^{(m)}\rho^{(m)}$, where each $\rho^{(j)}$ satisfies $\textsc{Values}(\rho^{(j)}) \subseteq [0, \frac{N}{3}[$ and each $\sigma^{(j)}$ is some subrun $\pi[c, d]$ with $x_c < \frac{N}{3}$, $x_d < \frac{N}{3}$ and $x_e \geq \frac{N}{3}$ for all $e \in [c+1, d-1]$, where moreover $[c+1, d-1] \neq \emptyset$.

Thus, it suffices to show that for every N-run $\sigma = q_0(z_0) \cdots \xrightarrow{\sigma_{m-1},N} q_m(z_m)$ satisfying $\textsc{Values}(\sigma) \subseteq [0, 4N]$, $z_0, z_m < \frac{N}{3}$ and $z_i \geq \frac{N}{3}$ for all $i \in [1, m-1]$, there exists an $(N - \Gamma_{\mathcal{C}})$-run from $q_0(z_0)$ to $q_m(z_m)$. Let us assume $\max(\sigma) \geq N$ (the case $\max(\sigma) < N$ is even easier) and let $a \in [0, m]$ be minimal such that $z_a < N$ and $z_{a+1} \geq N$ and let $b \in [0, m]$ be maximal such that $z_b \geq N$ and $z_{b+1} < N$. That is, one can factorize σ as $\sigma = \alpha\,\sigma[a, a+1]\,\beta\,\sigma[b, b+1]\,\gamma$, where $\alpha = \sigma[0, a]$, $\beta = \sigma[a+1, b]$ and $\gamma = \sigma[b+1, m]$. The situation is depicted in Figure 6.

For $i \in [1, 5]$ let $\mathcal{I}_i = \left\{ z \in [0, 4N] \mid \frac{iN}{3} \leq z < \frac{(i+1)N}{3} \right\}$. Our proof involves a careful case distinction on which of the $\mathcal{I}_i$ the counter values z_a, z_{a+1}, z_b and z_{b+1} lie in, respectively. Let us here only treat the case $z_{a+1}, z_b \in \mathcal{I}_5$; thus σ_a (resp. σ_b) is a $+p$-transition (resp. $-p$-transition) and therefore $z_a, z_{b+1} \in \mathcal{I}_2$. We apply the 5/6-Lemma to the subrun $\sigma[a, b+1]$ for $\ell = N$, hereby obtaining an $(N - \Gamma_{\mathcal{C}})$-semirun $\sigma\widehat{[a, b+1]}$ that is an N-embedding of $\sigma[a, b+1]$ also from $q_a(z_a)$ to $q_{b+1}(z_{b+1})$. It follows from $\textsc{Values}(\sigma\widehat{[a, b+1]}) \subseteq [\min(\sigma[a, b+1]) - \Gamma_{\mathcal{C}}, \max(\sigma[a, b+1]) + \Gamma_{\mathcal{C}}]$ that $\sigma\widehat{[a, b+1]} - \Gamma_{\mathcal{C}}$ is in fact an $(N - \Gamma_{\mathcal{C}})$-run from $q_a(z_a - \Gamma_{\mathcal{C}})$ to $q_{b+1}(z_{b+1} - \Gamma_{\mathcal{C}})$. By definition of a and b it follows that $\phi(\alpha) = \phi(\gamma) = \varepsilon$. Thus, by exploiting that $z_a, z_{b+1} \in \mathcal{I}_2$ one can, by suitably applying the Depumping Lemma (possibly several times), obtain an $(N - \Gamma_{\mathcal{C}})$-run $\widehat{\alpha}$ from $q_0(z_0)$ to $q_a(z_a - \Gamma_{\mathcal{C}})$, and dually an $(N - \Gamma_{\mathcal{C}})$-run $\widehat{\gamma}$ from $q_{b+1}(z_{b+1} - \Gamma_{\mathcal{C}})$ to $q_m(z_m)$. The concatenation of $\widehat{\alpha}$, $\sigma\widehat{[a, b+1]} - \Gamma_{\mathcal{C}}$ and $\widehat{\gamma}$ is the desired $(N - \Gamma_{\mathcal{C}})$-run from $q_0(z_0)$ to $q_m(z_m)$.

7 Conclusion

In this paper we have shown that the reachability problem for parameteric timed automata with two parametric clocks and one parameter is complete for exponential space.

For the lower bound proof, inspired by [13, 15], we made use of two results from complexity theory. First, we made use of a serializability characterization of EXPSPACE from [13] which is a padded version of the serializability characterization of PSPACE from [20], which in turn has its roots in Barrington's Theorem [7]. Second, we made use of a result of Chiu, Davida, Litow that states that numbers in Chinese Remainder Representation can be translated into binary representation in NC^1 (and thus in logarithmic space). We are convinced that it is worthwhile to develop a suitable programming language that serves as a unifying framework in that it provides an interface for proving lower bounds for various problems involving automata. In a sense, we have developed the corresponding interface "by hand" when defining how parametric timed automata can compute functions (Definition 5).

For the EXPSPACE upper bound we first followed the approach of Bundala and Ouaknine [10] by providing an exponential time translation from reachability in parametric timed automata with two parametric clocks and one parameter (i.e. $(2,1)$-PTA) to reachability in parametric one-counter automata (POCA) over one parameter, yet on a slightly less expressive POCA model as introduced in [10]. We then studied the reachability in POCA with one parameter p. A repeated application of our Small Parameter Theorem (Theorem 16) allows to conclude that such a POCA has an accepting N-run all of whose counter values lie in $[0,4N]$ if, and only if, there exists such an accepting N-run for some N that is at most exponential in the size of the POCA. Since the translation from $(2,1)$-PTA to POCA is computable in exponential time, this gives a doubly exponential upper bound on the parameter value of the original $(2,1)$-PTA and hence an EXPSPACE upper bound for $(2,1)$-PTA-Reachability (Corollary 17).

In proving the Small Parameter Theorem we introduced the notion of semiruns and gave several techniques for manipulating them. The Depumping Lemma (Lemma 18) allowed us to construct from semiruns with large absolute counter effect new semiruns with a smaller absolute counter effect. The Bracket Lemma (Lemma 20) allowed us to find in semiruns having a sufficiently large absolute counter effect and satisfying some majority condition on the number of occurrences of $+p$-transitions and $-p$-transitions some subsemirun that has again a large absolute counter effect and moreover some bracketing properties. Our Hill and Valley Lemma (Lemma 21) allowed to turn, for sufficiently large N, any N-semirun that is either a hill or a valley into an N'-semirun for some $N' < N$. Our 5/6-Lemma (Lemma 22) allowed to turn for sufficiently large N any N-semirun with an absolute counter effect of at most $5/6 \cdot N$ into an N'-semirun for some $N' < N$.

We hope that extensions of our techniques provide a line of attack for finally showing decidability (and the precise complexity) of $(2,*)$-PTA-Reachability. For these however, it seems that the reduction to POCA indeed requires the presence of so-called $+[0,p]$-transitions. When analyzing runs in the corresponding more general POCA model that in turn also involves an arbitrary number of parameters, it will become necessary to "de-scale" semiruns in the following sense. Already in the presence of two parameters one can see that it becomes necessary to decrease the value of both parameters simultaneously proportionally: for instance one can build a $(2,2)$-PTA for which reachability holds only if the first parameter is a multiple of the second parameter. How our techniques can be extended to handle such obstacles remains yet to be explored.

References

1 Rajeev Alur, Costas Courcoubetis, and David L. Dill. Model-checking for real-time systems. In *Proceedings of the Fifth Annual Symposium on Logic in Computer Science (LICS '90), Philadelphia, Pennsylvania, USA, June 4-7, 1990*, pages 414–425. IEEE Computer Society, 1990. doi:10.1109/LICS.1990.113766.

2 Rajeev Alur and David L Dill. A theory of timed automata. *Theoretical computer science*, 126(2):183–235, 1994.

3 Rajeev Alur, Thomas A Henzinger, and Moshe Y Vardi. Parametric real-time reasoning. In *Proc. STOC'93*, pages 592–601. ACM, 1993.

4 Étienne André. What's decidable about parametric timed automata? *International Journal on Software Tools for Technology Transfer*, 21(2):203–219, 2019.

5 Étienne André. What's decidable about parametric timed automata? *Int. J. Softw. Tools Technol. Transf.*, 21(2):203–219, 2019.

6 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambrdige University Press, 2009.

7 D. A. M. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *Journal of Computer and System Sciences*, 38:150–164, 1989.

8 Nikola Benes, Peter Bezdek, Kim Guldstrand Larsen, and Jirí Srba. Language emptiness of continuous-time parametric timed automata. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 69–81. Springer, 2015.

9 Benedikt Bollig, Karin Quaas, and Arnaud Sangnier. The complexity of flat freeze ltl. *Logical Methods in Computer Science*, 15(3):32:1–32:26, 2019. arXiv:1609.06124.

10 Daniel Bundala and Joel Ouaknine. On parametric timed automata and one-counter machines. *Information and Computation*, 253:272–303, 2017.

11 Andrew Chiu, George Davida, and Bruce Litow. Division in logspace-uniform NC^1. *Theoretical Informatics and Applications. Informatique Théorique et Applications*, 35(3):259–275, 2001.

12 Stéphane Demri and Arnaud Sangnier. When model-checking freeze LTL over counter machines becomes decidable. In C.-H. Luke Ong, editor, *Foundations of Software Science and Computational Structures, 13th International Conference, FOSSACS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*, volume 6014 of *Lecture Notes in Computer Science*, pages 176–190. Springer, 2010.

13 Stefan Göller, Christoph Haase, Joël Ouaknine, and James Worrell. Model Checking Succinct and Parametric One-Counter Automata. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part II*, volume 6199 of *Lecture Notes in Computer Science*, pages 575–586. Springer, 2010.

14 Stefan Göller and Mathieu Hilaire. Reachability in two-parametric timed automata with one parameter is EXPSPACE-complete. *CoRR*, abs/2011.07091, 2020. arXiv:2011.07091.

15 Stefan Göller and Markus Lohrey. Branching-time model checking of one-counter processes and timed automata. *SIAM J. Comput.*, 42(3):884–923, 2013.

16 Christoph Haase. *On the complexity of model checking counter automata*. PhD thesis, Oxford University, 2012.

17 Christoph Haase, Joël Ouaknine, and James Worrell. Relating reachability problems in timed and counter automata. *Fundam. Informaticae*, 143(3-4):317–338, 2016.

18 Michael A. Harrison. *Introduction to Formal Language Theory*. Addison-Wesley, 1978.

19 Thomas A. Henzinger, Zohar Manna, and Amir Pnueli. What good are digital clocks? In Werner Kuich, editor, *Automata, Languages and Programming, 19th International Colloquium, ICALP92, Vienna, Austria, July 13-17, 1992, Proceedings*, volume 623 of *Lecture Notes in Computer Science*, pages 545–558. Springer, 1992.

20 Ulrich Hertrampf, Clemens Lautemann, Thomas Schwentick, Heribert Vollmer, and Klaus W. Wagner. On the power of polynomial time bit-reductions. In *Proceedings of the Eighth Annual Structure in Complexity Theory Conference*, pages 200–207. IEEE Computer Society Press, 1993.

21 Antonia Lechner, Richard Mayr, Joël Ouaknine, Amaury Pouly, and James Worrell. Model checking flat freeze LTL on one-counter automata. *Log. Methods Comput. Sci.*, 14(4), 2018.

22 Joël Ouaknine and James Worrell. Universality and Language Inclusion for Open and Closed Timed Automata. In Oded Maler and Amir Pnueli, editors, *Hybrid Systems: Computation and Control, 6th International Workshop, HSCC 2003 Prague, Czech Republic, April 3-5, 2003, Proceedings*, volume 2623 of *Lecture Notes in Computer Science*, pages 375–388. Springer, 2003. doi:10.1007/3-540-36580-X_28.

23 C. H. Papadimitriou. *Computational Complexity.* Addison Wesley, 1994.