Binary Reachability of Timed Pushdown Automata via Quantifier Elimination and Cyclic Order Atoms

Lorenzo Clemente[1]
University of Warsaw
clementelorenzo@gmail.com
https://orcid.org/0000-0003-0578-9103

Sławomir Lasota[2]
University of Warsaw
sl@mimuw.edu.pl
https://orcid.org/0000-0001-8674-4470

Abstract

We study an expressive model of timed pushdown automata extended with modular and fractional clock constraints. We show that the binary reachability relation is effectively expressible in hybrid linear arithmetic with a rational and an integer sort. This subsumes analogous expressibility results previously known for finite and pushdown timed automata with untimed stack. As key technical tools, we use quantifier elimination for a fragment of hybrid linear arithmetic and for cyclic order atoms, and a reduction to register pushdown automata over cyclic order atoms.

2012 ACM Subject Classification Theory of computation → Timed and hybrid models, Theory of computation → Logic and verification, Theory of computation → Formal languages and automata theory

Keywords and phrases timed automata, reachability relation, timed pushdown automata, linear arithmetic

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.118

1 Introduction

Timed automata (TA) are one of the most studied models of reactive timed systems. The fundamental result that paved the way to automatic verification of timed systems is decidability (and PSPACE-completeness) of the reachability problem for TA [2]. However, in certain applications, such as in parametric verification, deciding reachability is insufficient, and one needs to construct the more general *binary reachability relation*, i.e., the entire (possibly infinite) set of of pairs of configurations (c_i, c_f) s.t. there is an execution from c_i to c_f. The reachability relation for TA has been shown to be effectively expressible in hybrid linear arithmetic with rational and integer sorts [11, 13, 15, 18]. Since hybrid logic is decidable, this yields an alternative proof of decidability of the reachability problem.

In this paper, we compute the reachability relation for timed automata extended with a stack. An early model of *pushdown timed automata* (PTDA) extending TA with a (classical, untimed) stack has been considered by Bouajjani *et al.* [5]. More recently, *dense-timed pushdown automata* (dtPDA) have been proposed by Abdulla *et al.* [1] as an extension of PTDA. In dtPDA, stack symbols are equipped with rational *ages*, which initially are 0 and increase with the elapse of time at the same rate as global clocks; when a symbol is popped,

[1] Partially supported by Polish NCN grant 2017/26/D/ST6/00201.
[2] Partially supported by Polish NCN grant 2016/21/B/ST6/01505.

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 118; pp. 118:1–118:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

its age is tested for membership in an interval. While dtPDA syntactically extend PTDA by considering a timed stack, timed constraints can in fact be removed while preserving the timed language recognised by the dtPDA, and thus they semantically collapse to PTDA [8]. This motivates the quest for a strictly more expressive generalisation of PTDA and dtPDA with a truly timed stack. It has been observed in [21] that adding fractional stack constraints prevents the stack from being untimed, and thus strictly enriches the expressive power[3].

We embrace this observation and propose the model of *timed pushdown automata* (TPDA), which extends timed automata with a timed stack and integer, fractional, and modulo diagonal/non-diagonal constraints. The model features local clocks and stack clocks. As time elapses, all clocks increase their values, and they do so at the same rate. Local clocks can be reset and compared according to the generalised constraints above. At the time of a push operation, new stack clocks are created whose values are initialised, possibly non-deterministically, as to satisfy a given push constraint between stack clocks and local clocks; similarly, a pop operation requires that stack clocks to be popped satisfy a given pop constraint of analogous form. Stack push/pop constraints are also of the form of diagonal/non-diagonal integer, modulo, and fractional constraints.

Contributions. We compute the *binary reachability relation* of TPDA, i.e., the family of binary relations $\{\leadsto_{\ell r}\} \subseteq \mathbb{Q}_{\geq 0}^X \times \mathbb{Q}_{\geq 0}^X$ for control locations ℓ, r s.t. from the initial clock valuation $\mu \in \mathbb{Q}_{\geq 0}^X$ and control location ℓ we can reach the final clock valuation $\nu \in \mathbb{Q}_{\geq 0}^X$ and control location r, written $\mu \leadsto_{\ell r} \nu$. The stack is empty at the beginning and at the end of the computation. The main contribution of the paper is the effective computation of the TPDA reachability relation in the existential fragment of *linear arithmetic* $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$, a two-sorted logic combining Presburger arithmetic $(\mathbb{Z}, \leq, (\equiv_m)_{m\in\mathbb{N}}, +, 0)$ and linear rational arithmetic $(\mathbb{Q}, \leq, +, 0)$. As a byproduct of our constructions, we actually characterise the more general *ternary reachability relation* $\mu \overset{\pi}{\leadsto}_{\ell r} \nu$, where μ, ν are as above and $\pi : \mathbb{N}^\Sigma$ additionally counts the number of occurrences of input letters over a finite alphabet Σ, i.e., the Parikh image of the run. To our knowledge, the ternary reachability relation was not previously considered. As an application of ternary reachability, we can model, for instance, letter counts of initial and final, possibly non-empty, stack contents. Thus, ternary reachability is an expressive extension of binary reachability.

The computation of the ternary reachability relation is achieved by two consecutive translations. First, we transform a TPDA into a *fractional* TPDA, which uses only fractional constraints. In this step we exploit *quantifier elimination* for a fragment of linear arithmetic corresponding to clock constraints. Quantifier elimination is a pivotal tool in this work, and to our knowledge its use in the study of timed models is novel. The final integer value of clocks is reconstructed by letting the automaton input special tick symbol $\checkmark_x$ every time clock x reaches an integer value (provided it is not reset anymore later); it is here that ternary reachability is more suitable than binary reachability.

Secondly, a fractional TPDA is transformed into a PDA with registers (RPDA) over the so called *cyclic order atoms* $(\mathbb{Q} \cap [0, 1), K)$ [7], where K is the ternary cyclic order relation

$$K(a, b, c) \equiv a < b < c \vee b < c < a \vee c < a < b, \qquad \text{for } a, b, c \in \mathbb{Q} \cap [0, 1). \tag{1}$$

In other words, $K(a, b, c)$ holds if, distributing a, b, c on the unit circle and going clockwise from a, then we fist visit b and afterwards c. Since fractional values are wrapped around 0 when time increases, K is invariant under time elapse. We use registers to store the fractional

[3] For TA, fractional constraints can be handled by the original region construction and do not make the model harder to analyse [2].

parts of absolute times of last clock resets; fractional constraints on clocks are simulated by constraints on registers using K. In order to compute the reachability relation for RPDA we use again quantifier elimination, this time over cyclic order atoms. The latter property holds since cyclic order atoms constitute a homogeneous structure [16]. Therefore, another contribution of this work is the solution of a nontrivial problem such as computing the reachability relation for TPDA, which is a clock model, as an application of RPDA, which is a register model. The analysis of RPDA is substantially easier than a direct analysis of (fractional) TPDA.

From the complexity standpoint, the formula characterising the reachability relation of a TPDA is computable in double exponential time. However, when cast down to TA or TPDA with timeless stack (which subsume PTDA and, a posteriori, dtPDA), the complexity drops to singly exponential, matching the previously known complexity for TA [18]. For PTDA, no complexity was previously given in [12], and thus the result is new. For TPDA, the binary reachability problem has not been studied before. Since the existential fragment of $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$ is decidable in NP (because so is existential linear rational arithmetic [19] and existential Presburger arithmetic [23]), we can solve the reachability problem of TPDA in 2NEXP by reduction to satisfiability for $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$. Since our constructions preserve the languages of all the models involved, untimed TPDA languages are context-free.

Discussion. From a syntactic point of view, TPDA significantly lifts the restrictions of dtPDA– which allow only classical non-diagonal constraints, i.e., interval tests, and thus has neither diagonal, nor modulo, nor fractional constraints – and of the model of [21] – which additionally allows diagonal/non-diagonal fractional tests, and thus does not have modulo constraints. Since classical diagonal constraints reduce to classical non-diagonal constraints, and, *in the presence of fractional constraints*, integer and modulo constraints can be removed altogether (cf. Sec. 4), TPDA are expressively equivalent to [21]. However, while [21] solves the control state reachability problem, we solve the more general problem of computing the binary reachability relation. Our reduction technique not only preserves reachability, like [21], but additionally enables the reconstruction of the reachability relation.

Our expressivity result generalises analogous results for TA [11, 13, 15, 18] and PTDA [12]. The proof of [11] for TA has high technical difficulty and does not yield complexity bounds. The proof of [13] for TA uses an automata representation for sets of clock valuations; the idea of *reset-point semantics* employed in [13] is analogous to using registers instead of clocks. The paper [15] elegantly expresses the reachability relation for TA with *clock difference relations* (CDR) over the fractional values of clocks. It is remarkable that the formulas expressing the reachability relations that we obtain are of the same shape as CDR. The recent paper [18] shows that the TA binary reachability relation can be expressed in the same fragment of hybrid linear arithmetic that we use for TPDA, which we find very intriguing. Their proof converts the integer value of clocks into counters, and then observes that, thanks to the specific reset policy of clocks, these counter machines have a semilinear reachability relation; the latter is proved by encoding the value of counters into the language. In our proof, we bring the encoding of the integer value of clocks into the language to the forefront, via the introduction of the ternary reachability relation. The proof of [12] for PTDA also separates clocks into their integer and fractional part. It is not clear how any of the previous approaches could handle a timed stack.

Another approach for computing the reachability relation for TPDA would be to reduce it directly to a more expressive register model, such as *timed register pushdown automata* (TRPDA) [8, 9], which considers both integer $(\mathbb{Z}, \leq, +1)$ and rational registers $(\mathbb{Q}_{\geq 0}, \leq)$. While such a reduction for the reachability problem is possible since (the integer part of) large clock values can be "forgotten", e.g., along the lines of [8], this does not hold anymore if we want to preserve the reachability relation. For this reason, in the present work we first remove the integer part of clocks (by encoding it in the untimed language) and then we reduce to RPDA, which have only fractional registers and no integer register, and are thus easier to analyse than TRPDA[4]. The method of quantifier elimination was recently applied to the analysis of another timed model, namely *timed communicating automata* [6].

Finally, another expressive extension of TA, called *recursive timed automata* (RTA), has been proposed [20, 3]. RTA use a timed stack to store the current clock valuation, which does not evolve as time elapses and can be restored at the time of pop. This facility makes RTA expressively incomparable to all models previously mentioned.

Missing proofs can be found in the technical report [10].

Notations. Let $\mathbb{Q}$, $\mathbb{Q}_{\geq 0}$, $\mathbb{Z}$, and $\mathbb{N}$ denote the rationals, the non-negative rationals, the integers, and the natural numbers; let $\mathbb{I} = \mathbb{Q}_{\geq 0} \cap [0, 1)$ be the unit rational interval. Let $\equiv_m$ denote the congruence modulo $m \in \mathbb{N} \setminus \{0\}$ in $\mathbb{Z}$. For $a \in \mathbb{Q}$, let $\lfloor a \rfloor \in \mathbb{Z}$ denote the largest integer k s.t. $k \leq a$, and let $\{a\} = a - \lfloor a \rfloor$ denote its fractional part. Let $\mathbb{1}_{C?}$, for a condition C, be 1 if C holds, and 0 otherwise.

2 Linear arithmetic and quantifier elimination

Consider the two-sorted structure $\mathcal{A} = \mathcal{A}_{\mathbb{Z}} \uplus \mathcal{A}_{\mathbb{Q}}$, where $\mathcal{A}_{\mathbb{Z}} = (\mathbb{Z}, \leq, (\equiv_m)_{m\in\mathbb{N}}, +, (k)_{k\in\mathbb{Z}})$ and $\mathcal{A}_{\mathbb{Q}} = (\mathbb{Q}, \leq, +, (k)_{k\in\mathbb{Q}})$. We consider "+" as a binary function, and we have a constant k for every integer/rational number. By *linear arithmetic*, denoted $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$, we mean the two-sorted first-order language in the vocabulary of $\mathcal{A}$. Restriction to the integer sort yields Presburger arithmetic $\mathcal{L}_{\mathbb{Z}}$ (*integer formulas*), and restriction to the rational sort yields linear rational arithmetic $\mathcal{L}_{\mathbb{Q}}$ (*rational formulas*). We assume constants are encoded in binary.

Two formulas are *equivalent* if they are satisfied by the same valuations. It is well-known that the theories of $\mathcal{A}_{\mathbb{Z}}$ [17] and $\mathcal{A}_{\mathbb{Q}}$ [14] admit effective elimination of quantifiers: Every formula can effectively be transformed in an equivalent quantifier-free one. Therefore, the theory of $\mathcal{A}$ also admits quantifier elimination, by the virtue of the following general fact (when speaking of a structure admitting quantifier elimination, we have in mind its theory).

▶ **Lemma 1.** *If the structures $\mathcal{A}_1$ and $\mathcal{A}_2$ admit (effective) elimination of quantifiers, then the two-sorted structure $\mathcal{A}_1 \uplus \mathcal{A}_2$ also does so. For conjunctive formulas, the complexity is the maximum of the two complexities.*

For clock constraints, we will use the first-order language over the two sorted structure $\mathcal{A}^c = \mathcal{A}^c_{\mathbb{N}} \uplus \mathcal{A}^c_{\mathbb{I}}$, where the integer sort is restricted to $\mathcal{A}^c_{\mathbb{N}} = (\mathbb{N}, \leq, (\equiv_m)_{m\in\mathbb{N}}, +1, 0)$ – the domain is now $\mathbb{N}$ and full addition "+" is replaced by the unary successor operation "+1") – and the rational sort to $\mathcal{A}^c_{\mathbb{I}} = (\mathbb{I}, \leq, 0)$ – the domain is now the unit interval, there is no addition, and the only constant is 0. Let $\mathcal{L}^c_{\mathbb{N},\mathbb{I}}$ be such a sub-logic. (As syntactic sugar we allow to use addition of arbitrary, even negative, integer constants in integer formulas, e.g. $x - 4 \leq y + 2$.) As before, $\mathcal{L}^c_{\mathbb{N}}$ and $\mathcal{L}^c_{\mathbb{I}}$ are the restrictions to the respective sorts. All the sub-logics above admit effective elimination of quantifiers.

---

[4] TRPDA are more general than RPDA– cyclic order atoms can be interpreted into $(\mathbb{Q}_{\geq 0}, \leq)$. The binary reachability relation for TRPDA can be computed by refining the reductions of [9] used for deciding the reachability problem. However, we do not know how to use the reachability relation of TRPDA to compute that of TPDA.

▶ **Lemma 2.** *The structures $\mathcal{A}^{c}_{\mathbb{N}}$ and $\mathcal{A}^{c}_{\mathbb{I}}$ admit effective elimination of quantifiers. For $\mathcal{A}^{c}_{\mathbb{N}}$ the complexity is singly exponential for conjunctive formulas, while for $\mathcal{A}^{c}_{\mathbb{I}}$ is quadratic.*

Notice that since $\mathcal{L}^{c}_{\mathbb{N}}$ is a fragment of Presburger arithmetic $\mathcal{L}_{\mathbb{Z}}$, we could apply the quantifier elimination for $\mathcal{L}_{\mathbb{Z}}$ to get a quantifier-free $\mathcal{L}_{\mathbb{Z}}$ formula. Our result is stronger since we get a quantifier-free formula of the more restrictive fragment $\mathcal{L}^{c}_{\mathbb{N}}$.

▶ **Corollary 3.** *The structure $\mathcal{A}^{c}$ admits effective quantifier elimination. The complexity is exponential for conjunctive formulas.*

3 Timed pushdown automata

Clock constraints. Let X be a finite set of clocks. We consider constraints which can separately speak about the integer $\lfloor x \rfloor$ and fractional value $\{x\}$ of a clock $x \in X$. A *clock constraint* over X is a boolean combination of *atomic clock constraints* of one of the forms

	(integer)	(modular)	(fractional)
(non-diagonal)	$\lfloor x \rfloor \leq k$	$\lfloor x \rfloor \equiv_m k$	$\{x\} = 0$
(diagonal)	$\lfloor x \rfloor - \lfloor y \rfloor \leq k$	$\lfloor x \rfloor - \lfloor y \rfloor \equiv_m k$	$\{x\} \leq \{y\}$

where $x, y \in X$, $m \in \mathbb{N}$,a and $k \in \mathbb{Z}$. Since we allow arbitrary boolean combinations, we consider also the constraint **true**, which is always satisfied, and variants with any $\sim \in \{\leq, <, \geq, >\}$ in place of $\leq$. A *clock valuation* is a mapping $\mu \in \mathbb{Q}^X_{\geq 0}$ assigning a non-negative rational number to every clock in X; we write $\lfloor \mu \rfloor$ for the valuation in $\mathbb{N}^X$ s.t. $\lfloor \mu \rfloor(x) := \lfloor \mu(x) \rfloor$ and $\{\mu\}$ for the valuation in $\mathbb{I}^X$ s.t. $\{\mu\}(x) := \{\mu(x)\}$. For a valuation μ and a clock constraint φ we say that μ *satisfies* φ if φ is satisfied when integer clock values $\lfloor x \rfloor$ are evaluated according to $\lfloor \mu \rfloor$ and fractional values $\{x\}$ according to $\{\mu\}$.

▶ Remark (Clock constraints as quantifier-free $\mathcal{L}^{c}_{\mathbb{N},\mathbb{I}}$ formulas). Up to syntactic sugar, a clock constraint over clocks $\{x_1, \ldots, x_n\}$ is the same as a quantifier-free $\mathcal{L}^{c}_{\mathbb{N},\mathbb{I}}$ formula $\varphi(\lfloor x_1 \rfloor, \ldots, \lfloor x_n \rfloor, \{x_1\}, \ldots, \{x_n\})$ over n integer and n rationals variables.

▶ Remark (Classical clock constraints). Integer and fractional constraints subsume classical ones. For clocks x, y, since $x = \lfloor x \rfloor + \{x\}$ (and similarly for y)[5], $x - y \leq k$ for an integer k is equivalent to $(\lfloor x \rfloor - \lfloor y \rfloor \leq k \wedge \{x\} \leq \{y\}) \vee \lfloor x \rfloor - \lfloor y \rfloor \leq k - 1$, and similarly for $x \leq k$. On the other hand, the fractional constraint $\{x\} = 0$ is not expressible as a classical constraint.

▶ Remark ($\lfloor x \rfloor - \lfloor y \rfloor$ versus $\lfloor x - y \rfloor$). In the presence of fractional constraints, the expressive power would not change if, instead of atomic constraints $\lfloor x \rfloor - \lfloor y \rfloor \equiv_m k$ and $\lfloor x \rfloor - \lfloor y \rfloor \leq k$ speaking of the *difference of the integer parts*, we would choose $\lfloor x - y \rfloor \equiv_m k$ and $\lfloor x - y \rfloor \leq k$ speaking of the *integer part of the difference*, since the two are inter-expressible:

$$\lfloor x - y \rfloor = \lfloor x \rfloor - \lfloor y \rfloor - \mathbb{1}_{\{x\}<\{y\}?} \qquad \text{and} \qquad \{x - y\} = \{x\} - \{y\} + \mathbb{1}_{\{x\}<\{y\}?}. \tag{2}$$

The model. A *timed pushdown automaton* (TPDA) is a tuple $\mathcal{P} = \langle \Sigma, \Gamma, L, X, Z, \Delta \rangle$ where Σ is a finite input alphabet, Γ is a finite stack alphabet, L is a finite set of control locations, X is a finite set of *global clocks*, and Z is a finite set of *stack clocks* disjoint from X. The last item Δ is a set of transition rules $\langle \ell, \mathsf{op}, r \rangle$ with $\ell, r \in L$ control locations, where op determines the type of transition:

[5] We often identify a clock x with its value for simplicity of notation.

- *time elapse* $\mathsf{op} = \mathsf{elapse}$,
- *input* $\mathsf{op} = a \in \Sigma_\varepsilon := \Sigma \cup \{\varepsilon\}$ an input letter,
- *test* $\mathsf{op} = \varphi$ a *transition constraint* over clocks X,
- *reset* $\mathsf{op} = \mathsf{reset}(Y)$ with $Y \subseteq X$ a set of clocks to be reset,
- *push* $\mathsf{op} = \mathsf{push}(\alpha : \psi)$ with $\alpha \in \Gamma$ a stack symbol to be pushed on the stack under the *stack constraint* ψ over clocks $X \cup Z$, or
- *pop* $\mathsf{op} = \mathsf{pop}(\alpha : \psi)$ similarly as push.

We assume that every atomic constraint in a stack constraint contains some stack variable from Z. Throughout the paper, let x_0 be a global clock that is never reset (and thus measures the total elapsed time), and let z_0 be a stack clock that is 0 when pushed. A TPDA has *untimed stack* if the only stack constraint is **true**. Without push/pop operations, we obtain nondeterministic timed automata (TA).

▶ Remark (Complexity). For complexity estimations, we assume that constraints are conjunctions of atomic constraints, that constants therein are encoded in binary, that M is the maximal constant, and that all modular constraints use the same modulus M.

▶ Remark (Time elapse). The standard semantics of timed automata where time can elapse freely in every control location is simulated by adding explicit time elapse transitions $\langle \ell, \mathsf{elapse}, \ell \rangle$ for suitable locations ℓ. Our explicit modelling of the elapse of time will simplify the constructions in Sec. 4.

▶ Remark (Comparison with dtPDA). The dtPDA model [1] allows only one stack clock $Z = \{z\}$ and stack constraints of the form $z \sim k$. As shown in [8], this model is equivalent to TPDA with untimed stack. Our extension is two-fold. First, our definition of stack constraint is more liberal, since we allow more general *diagonal stack constraints* of the form $z - x \sim k$. Second, we also allow *modular* $\lfloor y \rfloor - \lfloor x \rfloor \equiv_m k$ and *fractional constraints* $\{x\} \sim \{y\}$, where clocks x, y can be either global or stack clocks. As demonstrated in Example 4 below, this model is not reducible to untimed stack, and thus TPDA are more expressive than dtPDA.

Semantics. Every stack symbol is equipped with a fresh copy of clocks from Z. At the time of $\mathsf{push}(\alpha : \psi)$, the push constraint ψ specifies possibly nondeterministically the initial value of all clocks in Z w.r.t. global clocks in X. Both global and stack clocks evolve at the same rate when a time elapse transition is executed. At the time of $\mathsf{pop}(\alpha : \psi)$, the pop constraint ψ specifies the final value of all clocks in Z w.r.t. global clocks in X. A *timed stack* is a sequence $w \in (\Gamma \times \mathbb{Q}_{\geq 0}^Z)^*$ of pairs (γ, μ), where γ is a stack symbol and μ is a valuation for stack clocks in Z. For a clock valuation μ and a set of clocks Y, let $\mu[Y \mapsto 0]$ be the same as μ except that clocks in Y are mapped to 0. For $\delta \in \mathbb{Q}_{\geq 0}$, let $\mu + \delta$ be the clock valuation which adds δ to the value of every clock, i.e., $(\mu + \delta)(x) := \mu(x) + \delta$, and for a timed stack $w = (\gamma_1, \mu_1) \cdots (\gamma_k, \mu_k)$, let $w + \delta$ be $(\gamma_1, \mu_1 + \delta) \cdots (\gamma_k, \mu_k + \delta)$. A *configuration* is a triple $\langle \ell, \mu, w \rangle \in L \times \mathbb{Q}_{\geq 0}^X \times (\Gamma \times \mathbb{Q}_{\geq 0}^Z)^*$ where ℓ is a control location, μ is a clock valuation over the global clocks X, and w is a timed stack. Let $\langle \ell, \mu, u \rangle, \langle r, \nu, v \rangle$ be two configurations. For every input symbol or time increment $a \in (\Sigma_\varepsilon \cup \mathbb{Q}_{\geq 0})$ we have a transition $\langle \ell, \mu, u \rangle \xrightarrow{a} \langle r, \nu, v \rangle$ whenever there exists a rule $\langle \ell, \mathsf{op}, r \rangle \in \Delta$ s.t. one of the following holds:

- $\mathsf{op} = \mathsf{elapse}$, $a \in \mathbb{Q}_{\geq 0}$, $\nu = \mu + a$, $v = u + a$.
- $\mathsf{op} = a \in \Sigma_\varepsilon$, $\nu = \mu$, $u = v$.
- $\mathsf{op} = \varphi$, $a = \varepsilon$, $\mu \models \varphi$, $\nu = \mu$, $u = v$.
- $\mathsf{op} = \mathsf{reset}(Y)$, $a = \varepsilon$, $\nu = \mu[Y \mapsto 0]$, $v = u$.
- $\mathsf{op} = \mathsf{push}(\gamma : \psi)$, $a = \varepsilon$, $\mu = \nu$, $v = u \cdot \langle \gamma, \mu_1 \rangle$ if $\mu_1 \in \mathbb{Q}_{\geq 0}^Z$ satisfies $(\mu, \mu_1) \models \psi$, where $(\mu, \mu_1) \in \mathbb{Q}_{\geq 0}^{X \cup Z}$ is the unique clock valuation that agrees with μ on X and with μ_1 on Z.
- $\mathsf{op} = \mathsf{pop}(\gamma : \psi)$, $a = \varepsilon$, $\mu = \nu$, $u = v \cdot \langle \gamma, \mu_1 \rangle$ provided that $\mu_1 \in \mathbb{Q}_{\geq 0}^Z$ satisfies $(\mu, \mu_1) \models \psi$.

A *timed word* is a sequence $w = \delta_1 a_1 \cdots \delta_n a_n \in (\mathbb{Q}_{\geq 0}\Sigma_\varepsilon)^*$ of alternating time elapses and input symbols; the one-step transition relation $\langle \ell, \mu, u \rangle \xrightarrow{a} \langle r, \nu, v \rangle$ is extended on timed words w as $\langle \ell, \mu, u \rangle \xrightarrow{w} \langle r, \nu, v \rangle$ in the natural way. The *timed language* from location ℓ to r is $L(\ell, r) := \left\{ \pi_\varepsilon(w) \in (\mathbb{Q}_{\geq 0}\Sigma)^* \;\middle|\; \langle \ell, \mu_0, \varepsilon \rangle \xrightarrow{w} \langle r, \mu_0, \varepsilon \rangle \right\}$ where $\pi_\varepsilon(w)$ removes the ε's from w and μ_0 is the valuation that assigns $\mu_0(x) = 0$ to every clock x. The corresponding *untimed language* $L^{\mathrm{un}}(\ell, r)$ is obtained by removing the time elapses from $L(\ell, r)$.

▶ **Example 4.** Let L be the timed language of even length palindromes s.t. the time distance between every pair of matching symbols is an integer:

$$L = \{\delta_1 a_1 \cdots \delta_{2n} a_{2n} \mid \forall (1 \leq i \leq n) \cdot a_i = a_{2n-i+1} \wedge \delta_{i+1} + \cdots + \delta_{2n-i+1} \in \mathbb{N}\}.$$

L can be recognised by a TPDA over input and stack alphabet $\Sigma = \Gamma = \{a, b\}$, with locations ℓ, r, no global clock, one stack clock $Z = \{z\}$, and the following transition rules (omitting some intermediate states), where α ranges over $\{a, b\}$:

$$\langle \ell, \alpha; \mathsf{push}(\alpha : \{z\} = 0), \ell \rangle \qquad \langle \ell, \varepsilon, r \rangle$$
$$\langle r, \alpha; \mathsf{pop}(\alpha : \{z\} = 0), r \rangle \qquad \langle \ell, \mathsf{elapse}, \ell \rangle, \langle r, \mathsf{elapse}, r \rangle$$

We have $L = L(\ell, r)$. Since L cannot be recognised by TPDA with untimed stack (cf. [21]), fractional stack constraints strictly increase the expressive power of the model.

The reachability relation. The *Parikh image* of a timed word w is the mapping $\mathrm{PI}_w \in \mathbb{N}^\Sigma$ s.t. $\mathrm{PI}_w(a)$ is the number of a's in w, ignoring the elapse of time and ε's. For two control locations ℓ, r, clock valuations $\mu, \nu \in \mathbb{Q}^X_{\geq 0}$, and a timed word $w \in (\mathbb{Q}_{\geq 0}\Sigma_\varepsilon)^*$, we write $\mu \overset{w}{\rightsquigarrow}_{\ell r} \nu$ if $\langle \ell, \mu, \varepsilon \rangle \xrightarrow{w} \langle r, \nu, \varepsilon \rangle$. We overload the notation and, for $\pi \in \mathbb{N}^\Sigma$, we write $\mu \overset{\pi}{\rightsquigarrow}_{\ell r} \nu$ if there exists a timed word w s.t. $\mu \overset{w}{\rightsquigarrow}_{\ell r} \nu$ and $\pi = \mathrm{PI}_w$. We see $\{\rightsquigarrow_{\ell r}\}_{\ell, r \in L}$ as a family of subsets of $\mathbb{Q}^X_{\geq 0} \times \mathbb{N}^\Sigma \times \mathbb{Q}^X_{\geq 0}$ and we call it the *ternary reachability relation*.

Let $\left\{\psi_{\ell r}(\lfloor \overline{x} \rfloor, \{\overline{x}\}, \overline{f}, \lfloor \overline{y} \rfloor, \{\overline{y}\})\right\}_{\ell, r \in L}$ be a family of $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$ formulas, where $\lfloor \overline{x} \rfloor, \lfloor \overline{y} \rfloor$ represent the integer values of initial and final clocks, $\{\overline{x}\}, \{\overline{y}\}$ their fractional values, and $\overline{f}$ letter counts. The reachability relation $\{\rightsquigarrow_{\ell r}\}_{\ell, r \in L}$ is *expressed* by the family of formulas $\{\psi_{\ell r}\}_{\ell, r \in L}$ if the following holds: For every control locations $\ell, r \in L$, clock valuations $\mu, \nu \in \mathbb{Q}^X_{\geq 0}$ and $\pi \in \mathbb{N}^\Sigma$, $\mu \overset{\pi}{\rightsquigarrow}_{\ell r} \nu$ holds, if, and only if, $(\lfloor \mu \rfloor, \{\mu\}, \pi, \lfloor \nu \rfloor, \{\nu\}) \models \psi_{\ell r}$ holds.

Main results. As the main result of the paper we show that the reachability relation of TPDA and TA is expressible in linear arithmetic $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$.

▶ **Theorem 5.** *The reachability relation of a* TPDA *is expressed by a family of existential* $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$ *formulas computable in double exponential time. For* TA*, the complexity is exponential.*

This is a strengthening of analogous results for TA [11, 18] since our model, even without stack, is more expressive than classical TA due to fractional constraints. As a side effect of the proofs we get:

▶ **Theorem 6.** *Untimed* TPDA *languages* $L^{un}(\ell, r)$ *are effectively context-free.*

The following two sections are devoted to proving the two theorem above.

4 Fractional TPDA

A TPDA is *fractional* if it contains only fractional constraints. We show that computing the reachability relation reduces to the same problem for fractional TPDA. Our transformation is done in three steps, each one further restricting the set of allowed constraints.

A The TPDA is *push-copy*, that is, push operations can only copy global clocks into stack clocks. There is one stack clock z_x for each global clock x, and the only push constraint is

$$\psi_{\mathsf{copy}}(\overline{x}, \overline{z}_{\overline{x}}) \equiv \bigwedge_{x \in X} \lfloor z_x \rfloor = \lfloor x \rfloor \wedge \{z_x\} = \{x\}. \tag{3}$$

By pushing copies of global clocks into the stack, we can postpone checking all non-trivial stack constraints to the time of pop. This steps uses quantifier elimination. The blowup of the number of pop constraints and stack alphabet is exponential.

B The TPDA is *pop-integer-free*, that is, pop transitions do not contain integer constraints. The construction is similar to a construction from [8] and is presented in the technical report [10]. Removing pop integer constraints is crucial towards removing all integer clocks (modulo constraints will be removed by the next step). This step strongly relies on the fact that stack clocks are copies of global clocks, which allows one to remove integer pop constraints by reasoning about analogous constraints between global clocks at the time of push and their future values at the time of pop, thus bypassing the stack altogether. We introduce one global clock for each integer pop constraint, exponentially many locations in the number of clocks and pop constraints, and exponentially many stack symbols in the number of pop constraints. When combined with the previous step, altogether exponentially many new clocks are introduced, and doubly exponentially many locations/stack symbols. It is remarkable that pop integer constraints can be removed by translating them into finitely many transition constraints on global clocks.

C The TPDA is fractional. All integer clocks are removed. In order to recover their values (which are needed to express the reachability relation), a special symbol $\checkmark_x$ is produced when an integer clock elapses one time unit. This step introduces a further exponential blowup of control locations w.r.t. global clocks and polynomial in the maximal constant M. The overall complexity of control locations thus stays double exponential.

By **A**+**B**+**C** (in this order, since the latter properties are ensured assuming the previous ones), we get the following theorem.

▶ **Theorem 7.** *A* TPDA $\mathcal{P}$ *can be effectively transformed into a fractional* TPDA $\mathcal{Q}$ *s.t. a family of* $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$ *formulas* $\{\varphi_{\ell r}\}$ *expressing the reachability relation of* $\mathcal{P}$ *can effectively be computed from a family of* $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$ *formulas* $\{\varphi'_{\ell' r'}\}$ *expressing the reachability relation of* $\mathcal{Q}$*. The number of control locations and the size of the stack alphabet in* $\mathcal{Q}$ *have a double exponential blowup, and the number of clocks has an exponential blowup.*

If there is no stack, then we do not need the first two steps, and we can do directly **C**.

▶ **Corollary 8.** *The reachability relation of push-copy* TPDA/TA *effectively reduces to the reachability relation of fractional* TPDA/TA *with an exponential blowup in control locations.*

(A) The TPDA is push-copy

Let $K_\leq$ be the non-strict variant of the ternary cyclic order K from (1), defined as $K_\leq(a, b, c) \equiv K(a, b, c) \vee a = b \vee b = c$ for $a, b, c \in \mathbb{I}$. Let $\psi_{\mathsf{push}}(\overline{x}, \overline{z})$ be a push constraint, and let $\psi_{\mathsf{pop}}(\overline{x}', \overline{z}')$ be the corresponding pop constraint. Since stack clock z_0 is 0 when pushed on the stack, z'_0 is the total time elapsed between push and pop; let $\overline{z}'_0 = (z'_0, \ldots, z'_0)$ (the length of which depends on the context). Let $\overline{z}'_{\overline{x}}$ be a vector of stack variables representing the value of *global clocks* at the time of pop, provided they were not reset since the matching push. Since all clocks evolve at the same rate, for every global clock x and stack clock z, we have

$$x = z'_x - z'_0 \qquad \text{and} \qquad z = z' - z'_0. \tag{4}$$

If at the time of push, instead of pushing $\overline{z}$, we push on the stack a copy of global clocks $\overline{x}$, then at the time of pop it suffices to check that the following formula holds

$$\psi'_{\mathsf{pop}}(\overline{x}', \overline{z}'_{\overline{x}}) \;\equiv\; \exists \overline{z}' \geq \overline{0} \cdot \psi_{\mathsf{push}}(\overline{z}'_{\overline{x}} - \overline{z}'_0, \overline{z}' - \overline{z}'_0) \wedge \psi_{\mathsf{pop}}(\overline{x}', \overline{z}'). \tag{5}$$

Note that the assumption that $z_0 = 0$ at the time of push makes the existential quantification satisfiable by exactly one value of z'_0, namely the total time elapsed between push and pop. However, $\psi_{\mathsf{push}}(\overline{z}'_{\overline{x}} - \overline{z}'_0, \overline{z}' - \overline{z}'_0)$ is not a constraint anymore, since variables are replaced by differences of variables. We resolve this issue by showing that the latter is in fact equivalent to a clock constraint. Thanks to (4), for every clock x we have $\lfloor x \rfloor = \lfloor z'_x - z'_0 \rfloor, \{x\} = \{z'_x - z'_0\}$, and $\lfloor z \rfloor = \lfloor z' - z'_0 \rfloor, \{z\} = \{z' - z'_0\}$. Thus, a fractional constraint $\{y\} \leq \{z\}$ in ψ_{push} is equivalent to $\{z'_y - z'_0\} \leq \{z' - z'_0\}$, which is in turn equivalent to $C = K_{\leq}(\{z'_0\}, \{z'_y\}, \{z'\})$, which is definable from $\leq$. Moreover, $\lfloor y \rfloor - \lfloor z \rfloor = \lfloor z'_y - z'_0 \rfloor - \lfloor z' - z'_0 \rfloor = (z'_y - z'_0 - \{z'_y - z'_0\}) - (z' - z'_0 - \{z' - z'_0\}) = (z'_y - z') - \{z'_y - z'_0\} + \{z' - z'_0\} = (z'_y - z') - \{z'_y - z'\} + \mathbb{1}_{D?} = \lfloor z'_y - z' \rfloor + \mathbb{1}_{D?}$, with $D = C \wedge \{z'_y\} \neq \{z'\}$. (Notice that $\lfloor z'_0 \rfloor$ disappears in this process: This is not a coincidence, since diagonal integer/modular/fractional constraints are invariant under the elapse of an *integer* amount of time.) Thus by (2) we obtain a constraint $\psi'_{\mathsf{push}}(\overline{z}'_{\overline{x}}, \overline{z}')$ logically equivalent to $\psi_{\mathsf{push}}(\overline{z}'_{\overline{x}} - \overline{z}'_0, \overline{z}' - \overline{z}'_0)$, and, by separating the fractional and integer constraints (cf. Remark 3), $\psi'_{\mathsf{pop}}(\overline{x}', \overline{z}'_{\overline{x}}) \equiv \exists \lfloor \overline{z}' \rfloor, \{\overline{z}'\} \cdot \psi'_{\mathsf{push}}(\lfloor \overline{z}'_{\overline{x}} \rfloor, \{\overline{z}'_{\overline{x}}\}, \lfloor \overline{z}' \rfloor, \{\overline{z}'\}) \wedge \psi_{\mathsf{pop}}(\lfloor \overline{x}' \rfloor, \{\overline{x}'\}, \lfloor \overline{z}' \rfloor, \{\overline{z}'\})$. By Corollary 3, we can perform quantifier elimination and we obtain a logically equivalent clock constraint of exponential size (in DNF) $\xi_{\psi_{\mathsf{push}}, \psi_{\mathsf{pop}}}(\lfloor \overline{x}' \rfloor, \{\overline{x}'\}, \lfloor \overline{z}'_{\overline{x}} \rfloor, \{\overline{z}'_{\overline{x}}\})$, where the subscript indicates that this formula depends on the pair $(\psi_{\mathsf{push}}, \psi_{\mathsf{pop}})$ of push and pop constraints. The construction of $\mathcal{P}'$ consists in checking $\xi_{\psi_{\mathsf{push}}, \psi_{\mathsf{pop}}}$ in place of ψ_{pop}, assuming that the push constraint was ψ_{push}. The latter is replaced by ψ_{copy}. Control states are the same in the two automata; we can break down the $\xi_{\psi_{\mathsf{push}}, \psi_{\mathsf{pop}}}$ in DNF and record each conjunct in the stack, yielding a new stack alphabet of exponential size.

▶ **Lemma 9.** *Let $\{\leadsto_{\ell r}\}_{\ell, r \in L}$, $\{\leadsto'_{\ell r}\}_{\ell, r \in L}$ be the reachability relations of $\mathcal{P}$, resp., $\mathcal{P}'$. Then, $\leadsto_{\ell r} = \leadsto'_{\ell r}$ for every $\ell, r \in L$, and $\mathcal{P}'$ has stack alphabet exponential in the size of $\mathcal{P}$.*

(C) The TPDA is fractional

Assume that the TPDA $\mathcal{P}$ is both push-copy (**A**) and pop-integer-free (**B**). We remove diagonal integer $\lfloor y \rfloor - \lfloor x \rfloor \sim k$ and modulo $\lfloor y \rfloor - \lfloor x \rfloor \equiv_m k$ constraints on global clocks x, y as in TA [2]. In the rest of the section, transition and stack constraints of $\mathcal{P}$ are of the form

$$\begin{array}{llllll}
\text{(trans.)} & \lfloor x \rfloor \leq k, & \lfloor x \rfloor \equiv_m k, & \{x\} = 0, & \{x\} \leq \{y\}, & (6) \\
\text{(push)} & \lfloor z_x \rfloor = \lfloor x \rfloor, & & & \{z_x\} = \{x\}, & (7) \\
\text{(pop)} & & \lfloor y \rfloor - \lfloor z_x \rfloor \equiv_m k, & \{z_x\} = 0, & \{y\} \leq \{z_x\}, & (8) \\
 & & \lfloor z_y \rfloor - \lfloor z_x \rfloor \equiv_m k, & & \{z_y\} \leq \{z_x\}. &
\end{array}$$

Unary abstraction. We replace the integer value of clocks by their *unary abstraction*: Valuations $\mu, \nu \in \mathbb{Q}_{\geq 0}^X$ are *M-unary equivalent*, written $\mu \approx_M \nu$, if, for every clock $x \in X$, $\lfloor \mu(x) \rfloor \equiv_M \lfloor \nu(x) \rfloor$ and $\lfloor \mu(x) \rfloor \leq M \Leftrightarrow \lfloor \nu(x) \rfloor \leq M$. Let Λ_M be the (finite) set of M-unary equivalence classes of clock valuations. For $\lambda \in \Lambda_M$ we abuse notation and write $\lambda(x)$ to indicate $\mu(x)$ for some $\mu \in \lambda$, where the choice of representative μ does not matter. We write $\lambda[Y \mapsto 0]$ for the equivalence class of $\nu[Y \mapsto 0]$ and we write $\lambda[x \mapsto x + 1]$ for the equivalence class of $\nu[x \mapsto \nu(x) + 1]$, for some $\nu \in \lambda$ (whose choice is irrelevant). Let $\varphi_\lambda(\overline{x}) \equiv \bigwedge_{x \in X} \lfloor x \rfloor \equiv_M \lambda(x) \wedge (\lfloor x \rfloor < M \Leftrightarrow \lambda(x) < M)$ say that clocks belong to λ. For φ containing transition constraints of the form (6), $\varphi|_\lambda$ is φ where every integer $\lfloor x \rfloor \leq k$ or modulo constraint $\lfloor x \rfloor \equiv_M k$ is uniquely resolved to be **true** or **false** by replacing every occurrence of $\lfloor x \rfloor$ with $\lambda(x)$. Similarly, for ψ a pop constraint of the form (8), $\psi|_{\lambda_{\mathsf{push}}, \lambda_{\mathsf{pop}}}$ is obtained by resolving modulo constraints $\lfloor y \rfloor - \lfloor z_x \rfloor \equiv_M k$ and $\lfloor z_y \rfloor - \lfloor z_x \rfloor \equiv_M k$ to be **true** or **false** by replacing every occurrence of $\lfloor y \rfloor$ by its abstraction at the time of pop $\lambda_{\mathsf{pop}}(y)$, and every occurrence of $\lfloor z_x \rfloor$ by $\lambda_{\mathsf{push}}(x) + \Delta(\lambda_{\mathsf{push}}, \lambda_{\mathsf{pop}})$, i.e., the initial value of clock x plus the total integer time elapsed until the pop, defined as $\Delta(\lambda_{\mathsf{push}}, \lambda_{\mathsf{pop}}) = \lambda_{\mathsf{pop}}(x_0) - \lambda_{\mathsf{push}}(x_0) - \mathbb{1}_{\{z_0\} > \{x_0\}?}$, i.e., we take the difference of x_0 (which is never reset) between push and pop, possibly corrected by "-1" if the last time unit only partially elapsed; the substitution for $\lfloor z_y \rfloor$ is analogous. Fractional constraints are unchanged.

Sketch of the construction. Given a push-copy and pop-integer-free TPDA $\mathcal{P}$, we build a fractional TPDA $\mathcal{Q}$ over the extended alphabet $\Sigma' = \Sigma \cup \{\checkmark_x \mid x \in X\}$ as follows. We eliminate integer $\lfloor x \rfloor \leq k$ and modulo constraints $\lfloor x \rfloor \equiv_M k$ by storing in the control the M-unary abstraction λ. To reconstruct the reachability relation of $\mathcal{P}$, we store the set of clocks Y which will not be reset anymore in the future. Thus, control locations L' of $\mathcal{Q}$ are of the form $\langle \ell, \lambda, Y \rangle$. In order to properly update the M-unary abstraction λ, the automaton checks how much time elapses by looking at the fractional values of clocks. When λ is updated to $\lambda[x \mapsto x + 1]$, a symbol $\checkmark_x$ is optionally produced if $x \in Y$ was guessed not to be reset anymore in the future. A test transition $\langle \ell, \varphi, r \rangle$ is simulated by $\langle \langle \ell, \lambda, Y \rangle, \varphi|_\lambda, \langle r, \lambda, Y \rangle \rangle$. A push-copy transition $\langle \ell, \mathsf{push}(\alpha : \psi_{\mathsf{copy}}), r \rangle$ is simulated by $\langle \langle \ell, \lambda, Y \rangle, \mathsf{push}(\langle \alpha, \lambda \rangle : \bigwedge_{x \in X} \{z_0\} = 0 \wedge \{z_x\} = \{x\}), \langle r, \lambda, Y \rangle \rangle$ copying only the fractional parts and the unary class of global clocks. A pop-integer-free transition $\langle \ell, \mathsf{pop}(\alpha : \psi), r \rangle$ is simulated by $\langle \langle \ell, \lambda_{\mathsf{pop}}, Y \rangle, \mathsf{pop}(\langle \alpha, \lambda_{\mathsf{push}} \rangle : \psi|_{\lambda_{\mathsf{push}}, \lambda_{\mathsf{pop}}}), \langle r, \lambda_{\mathsf{pop}}, Y \rangle \rangle$. The reachability formula $\varphi_{\ell r}$ for $\mathcal{P}$ can be expressed by guessing the initial and final abstractions λ, μ, and the set of clocks Y which is never reset in the run. For clocks $x \in Y$, we must observe precisely $\lfloor x' \rfloor - \lfloor x \rfloor$ ticks $\checkmark_x$, and for the others, $\lfloor x' \rfloor$, where x is the initial and x' the final value. Let $g_x^Y = \lfloor x' \rfloor - \lfloor x \rfloor$ if $x \in Y$, and $\lfloor x' \rfloor$ otherwise.

▶ **Lemma 10.** *Let $\{\psi_{\ell' r'}(\{\overline{x}\}, (\overline{f}, \overline{g}), \{\overline{x}'\})\}_{\ell', r' \in L'}$ express the reachability relation of the fractional $\mathcal{Q}$ where $\{\overline{x}\}, \{\overline{x}'\}$ are the fractional values of clocks (we ignore integer values), $\overline{f}$ is the Parikh image of the original input letters from Σ, and $\overline{g}$ of the new input letters $\checkmark_x$'s. The reachability relation of $\mathcal{P}$ is expressed by $\varphi_{\ell r}(\lfloor \overline{x} \rfloor, \{\overline{x}\}, \overline{f}, \lfloor \overline{x}' \rfloor, \{\overline{x}'\}) \equiv \bigvee_{\lambda, Y, \mu} \varphi_\lambda(\lfloor \overline{x} \rfloor) \wedge \psi_{\langle \ell, \lambda, Y \rangle \langle r, \mu, X \rangle}(\{\overline{x}\}, (\overline{f}, \overline{g}^Y), \{\overline{x}'\})$.*

5 From fractional TPDA to register PDA

The aim of this section is to prove the following result which, together with Theorem 7, completes the proof of our main result Theorem 5.

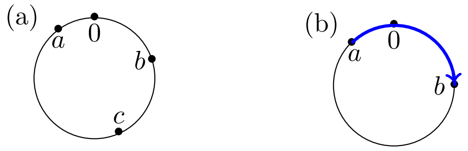

Figure 1 (a) Relation K. (b) The cyclic difference $b \ominus a$.

▶ **Theorem 11.** *The fractional reachability relation of a fractional* TPDA $\mathcal{P}$ *is expressed by existential* $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$ *formulas, computable in time exponential in the number of clocks and polynomial in the number of control locations and stack alphabet.*

Cyclic atoms. We model fractional clock values by the *cyclic atoms* structure $(\mathbb{I}, K)$ with universe $\mathbb{I} = \mathbb{Q} \cap [0, 1)$, where K is the ternary cyclic order (1). Since K is invariant under cyclic shift, it is convenient to think of elements of $\mathbb{I}$ as placed clockwise on a circle of unit perimeter; cf. Fig. 1(a). An *automorphism* is a bijection α that preserves and reflects K, i.e., $K(a, b, c)$ iff $K(\alpha(a), \alpha(b), \alpha(c))$; automorphisms are extended to tuples $\mathbb{I}^n$ point-wise. Cyclic atoms are homogeneous [16] and thus $\mathbb{I}^n$ splits into exponentially many *orbits* $\mathrm{Orb}(\mathbb{I}^n)$, where $u, v \in \mathbb{I}^n$ are in the same orbit if some automorphism maps u to v. An orbit is an equivalence class of indistinguishable tuples, similarly as regions for clock valuations, but in a different logical structure: For instance $(0.2, 0.3, 0.7)$, $(0.7, 0.2, 0.3)$, and $(0.8, 0.2, 0.3)$ belong to the same orbit, while $(0.2, 0.3, 0.3)$ belongs to a different orbit.

Register PDA. We extend classical pushdown automata with additional $\mathbb{I}$-valued *registers*, both in the finite control (i.e., global registers) and in the stack. Registers can be compared by quantifier-free formulas with equality and K, called *K-constraints*. For simplicity, we assume that there are the same number of global and stack registers. A *register pushdown automaton* (RPDA) is a tuple $\mathcal{Q} = \langle \Sigma, \Gamma, L, X, Z, \Delta \rangle$ where Σ is a finite input alphabet, Γ is a finite stack alphabet, L is a finite set of control locations, X is a finite set of *global registers*, Z is a finite set of *stack registers*, and the last item Δ is a set of transition rules $\langle \ell, \mathsf{op}, r \rangle$ with $\ell, r \in L$ control locations, where op is either: 1) an input letter $a \in \Sigma_\varepsilon$, 2) a $2k$-ary K-constraint $\psi(\overline{x}, \overline{x}')$ relating pre- and post-values of global registers, 3) a push operation $\mathsf{push}(\alpha : \psi(\overline{x}, \overline{z}))$ with $\alpha \in \Gamma$ a stack symbol to be pushed on the stack under the $2k$-ary K-constraint ψ relating global $\overline{x}$ and stack $\overline{z}$ registers, or 4) a pop operation $\mathsf{pop}(\alpha : \psi(\overline{x}, \overline{z}))$, similarly as push. We consider RPDA as symbolic representations of classical PDA with infinite sets of control states $\widetilde{L} = L \times \mathbb{I}^X$ and infinite stack alphabet $\widetilde{\Gamma} = \Gamma \times \mathbb{I}^Z$. A *configuration* is thus a tuple $\langle \ell, \mu, w \rangle \in L \times \mathbb{I}^X \times \widetilde{\Gamma}^*$ where ℓ is a control location, μ is a valuation of the global registers, and w is the current content of the stack. Let $\langle \ell, \mu, u \rangle, \langle r, \nu, v \rangle$ be two configurations. For every input symbol $a \in \Sigma_\varepsilon$ we have a transition $\langle \ell, \mu, u \rangle \xrightarrow{a} \langle r, \nu, v \rangle$ whenever there exists a rule $\langle \ell, \mathsf{op}, r \rangle \in \Delta$ s.t. one of the following holds: 1) $\mathsf{op} = a \in \Sigma_\varepsilon$, $\mu = \nu$, $u = v$, or 2) $\mathsf{op} = \varphi$, $a = \varepsilon$, $(\mu, \nu) \models \varphi$, $u = v$, or 3) $\mathsf{op} = \mathsf{push}(\gamma : \psi)$, $a = \varepsilon$, $\mu = \nu$, $v = u \cdot \langle \gamma, \mu_1 \rangle$ if $\mu_1 \in \mathbb{I}^Z$ satisfies $(\mu, \mu_1) \models \psi$, or 4) $\mathsf{op} = \mathsf{pop}(\gamma : \psi)$, $a = \varepsilon$, $\mu = \nu$, $u = v \cdot \langle \gamma, \mu_1 \rangle$ if $\mu_1 \in \mathbb{I}^Z$ satisfies $(\mu, \mu_1) \models \psi$.

Reachability relation. The reachability relations $\mu \overset{w}{\rightsquigarrow}_{\ell r} \nu$ and $\mu \overset{f}{\rightsquigarrow}_{\ell r} \nu$ are defined as for TPDA by extending one-step transitions $\langle \ell, \mu, u \rangle \xrightarrow{a} \langle r, \nu, v \rangle$ to words $w \in \Sigma^*$ and their Parikh images $f = \mathrm{PI}_w \in \mathbb{N}^\Sigma$. Thus, $\mu \overset{f}{\rightsquigarrow}_{\ell r} \nu$ is a subset of $\mathbb{I}^X \times \mathbb{N}^\Sigma \times \mathbb{I}^X$, which is furthermore invariant under orbits. In the following let X' be a copy of global clocks. An initial valuation μ belongs to $\mathbb{I}^X$, a final valuation ν to $\mathbb{I}^{X'}$, and the joint valuation (μ, ν) belongs to $\mathbb{I}^{X \times X'}$. The following two lemmas hold for RPDA with homogeneous atoms; cf. [7], or Sec. 9 in [4].

▶ **Lemma 12.** *If $(\mu, \nu), (\mu', \nu')$ belong to the same orbit of $\mathbb{I}^{X \times X'}$, then $\mu \overset{f}{\rightsquigarrow}_{\ell r} \nu$ iff $\mu' \overset{f}{\rightsquigarrow}_{\ell r} \nu'$.*

▶ **Lemma 13.** *Given a RPDA $\mathcal{Q}$ one can construct a context-free grammar G of exponential size with nonterminals of the form $X_{\ell r o}$, for control locations ℓ, r and an orbit $o \in \mathrm{Orb}(\mathbb{I}^{X \times X'})$, recognising the language $L(X_{\ell r o}) = \left\{ \pi_\Sigma(w) \in \Sigma^* \;\middle|\; \exists (\mu, \nu) \in o \cdot \mu \overset{w}{\rightsquigarrow}_{\ell r} \nu \right\}$, where $\pi_\Sigma(w)$ is w without the ε's. Consequently, RPDA recognise context-free languages.*

▶ **Lemma 14** (Theorem 4 of [22]). *The Parikh image of $L(X_{\ell r o})$ is expressed by an existential Presburger formula $\varphi^{\mathbb{Z}}_{\ell r o}$ computable in time linear in the size of the grammar.*

▶ **Corollary 15.** *Let $\varphi^{\mathbb{I}}_o$ be the characteristic K-constraint of the orbit $o \in \mathrm{Orb}(\mathbb{I}^{X \times X'})$. The reachability relation $\rightsquigarrow_{\ell r}$ of an RPDA $\mathcal{Q}$ is expressed by $\varphi_{\ell r}(\overline{x}, \overline{f}, \overline{x}') \equiv \bigvee_{o \in \mathrm{Orb}(\mathbb{I}^{X \times X'})} \varphi^{\mathbb{Z}}_{\ell r o}(\overline{f}) \wedge \varphi^{\mathbb{I}}_o(\overline{x}, \overline{x}')$. The size of $\varphi_{\ell r}$ is exponential in the size of $\mathcal{Q}$.*

Proof of Theorem 11. Define *cyclic sum* and *difference* of $a, b \in \mathbb{Q}$ to be $a \oplus b = \{a + b\}$, resp., $a \ominus b := \{a - b\}$. For a set of clocks X, let $X_{x_0} = X \cup \{x_0\}$ be its extension with an extra clock $x_0 \notin X$ which is never reset, and let $\hat{X}_{x_0} = \{\hat{x} \mid x \in X_{x_0}\}$ be a corresponding set of registers. The special register $\hat{x}_0$ stores the (fractional part of the) current timestamp, and register $\hat{x}$ stores the (fractional part of the) timestamp of the last reset of x. In this way we can recover the fractional value of x as the cyclic difference $\{x\} = \hat{x}_0 \ominus \hat{x}$. Let (cf. Fig. 1(b))

$$\varphi_\ominus(\overline{x}, \overline{\hat{x}}) \equiv \bigwedge_{x \in X} \{x\} = \hat{x}_0 \ominus \hat{x}. \tag{9}$$

Resetting clocks in $Y \subseteq X$ is simulated by $\varphi_{\mathsf{reset}(Y)} \equiv \hat{x}'_0 = \hat{x}_0 \wedge \bigwedge_{x \in Y} \hat{x}' = \hat{x}_0 \wedge \bigwedge_{x \in X \setminus Y} \hat{x}' = \hat{x}$ and time elapse by $\varphi_{\mathsf{elapse}} \equiv \bigwedge_{x \in X} \hat{x}' = \hat{x}$. The equality $\hat{x}'_0 = \hat{x}_0$ in $\varphi_{\mathsf{reset}(Y)}$ says that time does not elapse, and the absence of constraints on $\hat{x}_0, \hat{x}'_0$ in $\varphi_{\mathsf{elapse}}$ allows for an arbitrary elapse of time. A clock constraint φ is converted into a K-constraint $\hat{\varphi}$ by replacing $\{x\} = 0$ with $\hat{x} = \hat{x}_0$ and $\{x\} \leq \{y\}$ by $K_{\leq}(\hat{y}, \hat{x}, \hat{x}_0)$, for $x, y \in X \cup Z$. For a TPDA $\mathcal{P} = \langle \Sigma, \Gamma, L, X, Z, \Delta \rangle$, we define the following RPDA $\mathcal{Q} = \left\langle \Sigma, \Gamma, L, \hat{X}_{x_0}, \hat{Z}, \hat{\Delta} \right\rangle$. The input rules are preserved. A reset rule $\langle \ell, \mathsf{reset}(Y), r \rangle \in \Delta$, is simulated by $\langle \ell, \varphi_{\mathsf{reset}(Y)}, r \rangle \in \hat{\Delta}$, a time elapse rule $\langle \ell, \mathsf{elapse}, r \rangle \in \Delta$ is simulated by $\langle \ell, \varphi_{\mathsf{elapse}}, r \rangle \in \hat{\Delta}$, a push rule $\langle \ell, \mathsf{push}(\gamma : \varphi), r \rangle \in \Delta$ is simulated by $\langle \ell, \mathsf{push}(\gamma : \hat{\varphi}), r \rangle \in \hat{\Delta}$, and similarly for pop rules. By Corollary 15, let $\varphi_{\ell r}(\overline{\hat{x}}, \overline{f}, \overline{\hat{x}'})$ express the reachability relation of $\mathcal{Q}$, and define $\xi^{\mathbb{I}}_o(\overline{x}, \overline{x}') \equiv \exists \overline{\hat{x}}, \overline{\hat{x}'} \cdot \varphi^{\mathbb{I}}_o(\overline{\hat{x}}, \overline{\hat{x}'}) \wedge \varphi_\ominus(\overline{x}, \overline{\hat{x}}) \wedge \varphi_\ominus(\overline{x}', \overline{\hat{x}'})$. The reachability relation of $\mathcal{P}$ is recovered as

$$\psi_{\ell r}(\overline{x}, \overline{f}, \overline{x}') \equiv \bigvee \{\varphi^{\mathbb{Z}}_{\ell r o}(\overline{f}) \wedge \xi^{\mathbb{I}}_o(\overline{x}, \overline{x}') | o \in \mathrm{Orb}(\mathbb{I}^{X \times X'})\}. \tag{10}$$

Intuitively, we guess the value for registers $\overline{\hat{x}}, \overline{\hat{x}'}$ and we check that they correctly describe the fractional values of global clocks as prescribed by $\varphi_\ominus$. We now remove the quantifiers from $\xi^{\mathbb{I}}_o$ to uncover the structure of fractional value comparisons. Introduce a new variable $\delta = \hat{x}_0 \ominus \hat{x}'_0$, and perform the following substitutions in $\varphi^{\mathbb{I}}_o$ (c.f. the definition of $\varphi_\ominus$ in (9)): $\hat{x} \mapsto \hat{x}_0 \ominus \{x\}$, $\hat{x}' \mapsto (\hat{x}_0 \ominus \delta) \ominus \{x'\}$, and $\hat{x}'_0 \mapsto \hat{x}_0 \ominus \delta$. By writing $(\hat{x}_0 \ominus \delta) \ominus \{x'\}$ as $\hat{x}_0 \ominus (\delta \oplus \{x'\})$, we have only atomic constraints of the forms $K(\hat{x}_0 \ominus u, \hat{x}_0 \ominus v, \hat{x}_0 \ominus t)$ and $\hat{x}_0 \ominus u = \hat{x}_0 \ominus v$, where terms u, v, t are of one of the forms 0, $\{x\}$, $\delta \oplus \{x'\}$, δ. These constraints are equivalent, respectively, to $K(t, v, u)$ and $u = v$. By expanding the definition of K (cf. (1)), we obtain only constraints of the form $u \precsim v$ with $\precsim \in \{<, \leq\}$. Since δ appears at most once on either side, it can either be eliminated if it appears on both u, v, or otherwise exactly one of u, v is of the form δ or $\delta \oplus \{x'\}$, and the other of the form 0 or $\{x\}$. By moving $\{x'\}$ on the other side of the inequality in constraints containing $\delta \oplus \{x'\}$, $\xi_o^{\mathbb{I}}$ is equivalent to $\bigwedge_i s_i \precsim t_i \;\wedge\; \exists 0 \leq \delta < 1 \cdot \bigwedge_j u_j \precsim \delta \wedge \bigwedge_k \delta \precsim v_k$, where the terms s_i, t_i, u_j, v_k's are of the form 0, $\{x\}$, or $\{x\} \ominus \{y'\}$. We can now eliminate the quantification on δ and get a constraint of the form $\bigwedge_h s_h \precsim t_h$. Finally, by expanding $b \ominus a$ as $b - a + 1$ if $b < a$ and $b - a$ otherwise (since $a, b \in \mathbb{I}$) we have $\xi_o^{\mathbb{I}}(\overline{x}, \overline{x}') \equiv \bigwedge_h s'_h \precsim t'_h$, where the s'_h, t'_h's are of one of the forms: 0, $\{x\}$, $\{x\} - \{y'\}$, or $\{x\} - \{y'\} + 1$. ◀

References

1. P. A. Abdulla, M. F. Atig, and J. Stenman. Dense-timed pushdown automata. In *Proc. LICS'12*, pages 35–44. IEEE, 2012. doi:10.1109/LICS.2012.15.
2. Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126:183–235, 1994.
3. M. Benerecetti, S. Minopoli, and A. Peron. Analysis of timed recursive state machines. In *Proc. TIME'10*, pages 61–68. IEEE, sept. 2010. doi:10.1109/TIME.2010.10.
4. Mikołaj Bojańczyk. Slightly infinite sets. URL: https://www.mimuw.edu.pl/~bojan/paper/atom-book.
5. Ahmed Bouajjani, Rachid Echahed, and Riadh Robbana. On the automatic verification of systems with continuous variables and unbounded discrete data structures. In *Proc. Hybrid Systems '94*, volume 999 of *LNCS*, pages 64–85. Springer, 1995.
6. Lorenzo Clemente. Decidability of timed communicating automata. *ArXiv e-prints*, 04 2018. arXiv:1804.07815.
7. Lorenzo Clemente and Slawomir Lasota. Reachability analysis of first-order definable pushdown systems. In *Proc. of CSL'15*, volume 41 of *LIPIcs*, pages 244–259. Dagstuhl, 2015.
8. Lorenzo Clemente and Slawomir Lasota. Timed pushdown automata revisited. In *Proc. LICS'15*, pages 738–749. IEEE, July 2015.
9. Lorenzo Clemente, Sławomir Lasota, Ranko Lazić, and Filip Mazowiecki. Timed pushdown automata and branching vector addition systems. In *Proc. of LICS'17*, 2017.
10. Lorenzo Clemente and Sławomir Lasota. Binary reachability of timed pushdown automata via quantifier elimination and cyclic order atoms. *ArXiv e-prints*, 04 2018. arXiv:1804.10772.
11. Hubert Comon and Yan Jurski. Timed automata and the theory of real numbers. In *Proc. of CONCUR'99*, CONCUR '99, pages 242–257, London, UK, UK, 1999. Springer-Verlag.
12. Zhe Dang. Pushdown timed automata: a binary reachability characterization and safety verification. *Theor. Comput. Sci.*, 302(1–3):93–121, 2003. doi:10.1016/S0304-3975(02)00743-0.
13. C. Dima. Computing reachability relations in timed automata. In *In Proc. of LICS'02*, pages 177–186, 2002.
14. Jeanne Ferrante and Charles Rackoff. A decision procedure for the first order theory of real addition with order. *SIAM Journal on Computing*, 4(1):69–76, 1975.
15. Pavel Krčál and Radek Pelánek. On sampled semantics of timed systems. In Sundar Sarukkai and Sandeep Sen, editors, *In Proc. of FSTTCS'05*, volume 3821 of *LNCS*, pages 310–321. Springer, 2005.
16. Dugald Macpherson. A survey of homogeneous structures. *Discrete Mathematics*, 311(15):1599–1634, 2011.
17. Mojżesz Presburger. Über der vollständigkeit eines gewissen systems der arithmetik ganzer zahlen, in welchen die addition als einzige operation hervortritt. *Comptes Rendus Premier Congrès des Mathématicienes des Pays Slaves*, 395:92–101, 1930.

18 K. Quaas, M. Shirmohammadi, and J. Worrell. Revisiting reachability in timed automata. In *Proc. of LICS'17*, pages 1–12, June 2017. doi:10.1109/LICS.2017.8005098.

19 Eduardo D. Sontag. Real addition and the polynomial hierarchy. *Information Processing Letters*, 20(3):115–120, 1985.

20 Ashutosh Trivedi and Dominik Wojtczak. Recursive timed automata. In *Proc. ATVA'10*, volume 6252 of *LNCS*, pages 306–324. Springer, 2010.

21 Yuya Uezato and Yasuhiko Minamide. Synchronized recursive timed automata. In *Proc. of LPAR'15*, 2015.

22 Kumar Neeraj Verma, Helmut Seidl, and Thomas Schwentick. On the complexity of equational Horn clauses. In *Proc. CADE-20, 2005*, pages 337–352, 2005. doi:10.1007/11532231_25.

23 Volker Weispfenning. The complexity of linear problems in fields. *Journal of Symbolic Computation*, 5(1):3–27, 1988.