Binary reachability of timed pushdown automata via quantifier elimination

Lorenzo Clemente[1] **and Sławomir Lasota**[2]

1 University of Warsaw
2 University of Warsaw

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 linear arithmetic, using quantifier elimination as a key tool. This subsumes analogous characterisations previously known for timed automata and pushdown timed automata with untimed stack.

1998 ACM Subject Classification F.1.1 Models of Computation, F.4.1 Mathematical Logic

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

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.

1 Introduction

Timed automata (TA) [2], one of the most studied models of reactive timed systems, are classical automata equipped with clocks which can be reset and compared by inequality constraints. In this paper, we investigate timed automata extended with a stack. An early model of *pushdown timed automata* (PDTA) extending TA with a (classical, untimed) stack has been considered by Bouajjani *et al.* [6]. More recently, *dense-timed pushdown automata* (dtPDA) have been proposed by Abdulla *et al.* [1] as an extension of PDTA. 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, its age is tested for membership in an interval. Despite the apparent richness of the model, it turned out that, as a consequence of the interplay between heavy syntactic restrictions on constraints and the monotonicity of time, dtPDA semantically collapse to untimed stack [9], in the sense that stack timed constraints can be removed while preserving the timed language recognised.

This motivates a quest for an extension of PDA with a truly timed stack. One direction was proposed in [9, 11] by introducing the model of *timed register pushdown automata* (inspired by the earlier generalisation of TA [5]), which are more expressive than dtPDA but also depart from the clock semantics in favour of the register semantics (a.k.a. reset-point semantics [15]). Another direction, which is the one that we take here, is to enrich the kind of constraints allowed to compare clocks. It has been observed in [22] that adding *fractional inequality constraints* prevents the stack from being untimed, and thus strictly enriches the expressive power of the model. In fact, one could add as well *modular constraints*, allowing modulo tests on the integer part of clocks. (This contrasts to TA, where adding modular constraints does not increase the expressive power [7] assuming epsilon transitions [3].)

Contributions. We study *timed pushdown automata* (TPDA) as an expressive model combining time and recursion, where we allow arbitrary rational-valued clocks on the stack. At the time of push/pop stack clocks can be compared with each other, and with global

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Daniel Marx, and Don Sannella;
Article No. ; pp. :1–:21
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

clocks, using diagonal and non-diagonal integer, modulo, and fractional constraints, thus significantly lifting the syntactic restrictions of dtPDA. The main contribution of the paper is an effective characterisation of the *binary reachability relation* of TPDA, i.e., the set of pairs (μ, ν) of initial and final clock valuations s.t. from μ we can reach ν (with empty stack at the beginning and at the end), in the existential fragment of *linear arithmetic* $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$, a two-sorted logic combining Presburger arithmetic $(\mathbb{Z}, \leqslant, (\equiv_m)_{m\in\mathbb{N}}, +, 0)$ and linear rational arithmetic $(\mathbb{Q}, \leqslant, +, 0)$. This generalises an analogous result for TA [12, 15, 17, 20], and PDTA in discrete [14] and dense time [13]; interestingly, the same fragment of linear arithmetic suffices to capture reachability in TPDA as in the case of TA or PDTA.

As a byproduct of our constructions, we characterise the richer *ternary reachability relation* which relates together triples (μ, π, ν) where additionally π counts the number of occurrences of input letters in the run (i.e., the so-called Parikh image). In our knowledge, this ternary relation was not previously considered. As an application, ternary reachability can model the Parikh image of initial and final stack contents: In order to go from stack u to stack v (topmost symbol on the right), the automaton can initially read u from the input and push it on the stack, and then, at the end, can read v^R from the input and pop v from the stack; thus, stack counts reduce to input counts.

We proceed by two consecutive translations, each of them allowing for recovering the original reachability relation. The difficulty of the analysis of dense time is that it combines discrete and continuous features. We split dense time into its integer and fractional components. The integer component is removed by reducing it to the recognised (untimed) language. The fractional component is addressed by reduction to cyclic order, which is easier to analyse. More precisely, first we transform a TPDA into a *fractional* TPDA, which uses only fractional constrains. 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 novelty. 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 it is convenient to consider the more general ternary reachability relation. Secondly, a fractional TPDA is transformed into a register PDA over *cyclic order atoms* $(\mathbb{Q} \cap [0, 1), K)$ [8]: Registers store the fractional parts of absolute times of last clock resets and they are compared by the ternary *cyclic order* $K(x, y, z)$, which is invariant under time elapse. Interestingly, the analysis of register PDA is substantially easier than a direct analysis of fractional TPDA, due to (again) quantifier elimination for cyclic order atoms.

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 PDTA and, a posteriori, dtPDA) the complexity drops to singly exponential, which meets with the previously known complexity for TA [20]. For PDTA, no complexity was previously given in [13], and thus the result is new. For TPDA, no characterisation was previously known.

Since linear arithmetic $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$ is a decidable logic, binary reachability generalises the reachability/nonemptiness problem, which asks, for two given control states, whether we can reach one from the other. Since the existential fragment of linear arithmetic is decidable in NP (because so is existential linear rational arithmetic [21] and existential Presburger arithmetic [24]), we can solve the emptiness problem by reduction to satisfiability, and we derive that emptiness of TPDA can be solved in 2NEXP, which is also a new result.

Finally, while TPDA recognise more timed languages than PDTA, since our constructions preserve the *untimed* language, we deduce that untimed TPDA languages are context-free.

Notations. Let $\mathbb{Q}$, $\mathbb{Q}_{\geqslant 0}$, $\mathbb{Z}$, and $\mathbb{N}$ denote the rationals, the non-negative rationals, the integers, and the natural numbers. 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. $k \leqslant a$, 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

$$\mathcal{A} = \mathcal{A}_\mathbb{Z} \uplus \mathcal{A}_\mathbb{Q}, \quad \text{where} \quad \mathcal{A}_\mathbb{Z} = (\mathbb{Z}, \leqslant, (\equiv_m)_{m \in \mathbb{N}}, +, (k)_{k \in \mathbb{Z}}), \quad \mathcal{A}_\mathbb{Q} = (\mathbb{Q}, \leqslant, +, (k)_{k \in \mathbb{Q}}).$$

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*). For complexity considerations, 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 $\mathcal{A}_\mathbb{Z}$ [19] and $\mathcal{A}_\mathbb{Q}$ [16] 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). The proofs for this section can be found in Sec. A.1.

▸ **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 a sub-logic $\mathcal{L}^c_{\mathbb{N},\mathbb{I}}$ where the integer sort is restricted to $\mathbb{N}$, full addition "+" in the integer sort is replaced by the unary successor operation "+1", the rational sort is restricted to $\mathbb{I} := \mathbb{Q}_{\geqslant 0} \cap [0, 1)$ with no addition, and the only allowed constant is 0. Formally, by $\mathcal{L}^c_{\mathbb{N},\mathbb{I}}$ we mean the first-order language of the two-sorted structure

$$\mathcal{A}^c = \mathcal{A}^c_\mathbb{N} \uplus \mathcal{A}^c_\mathbb{I}, \quad \text{where} \quad \mathcal{A}^c_\mathbb{N} = (\mathbb{N}, \leqslant, (\equiv_m)_{m \in \mathbb{N}}, +1, 0), \quad \mathcal{A}^c_\mathbb{I} = (\mathbb{I}, \leqslant, 0).$$

(As syntactic sugar we allow to use addition of arbitrary, even negative, integer constants in integer formulas, e.g. $x - 4 \leqslant y + 2$.) As before we have the restrictions $\mathcal{L}^c_\mathbb{N}$ and $\mathcal{L}^c_\mathbb{I}$ to the respective sorts. The sub-logics admit effective elimination of quantifiers.

▸ **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.*

$\mathcal{L}^c_\mathbb{N}$ is a fragment of Presburger arithmetic $\mathcal{L}_\mathbb{Z}$. Hence, one can apply the quantifier elimination for $\mathcal{L}_\mathbb{Z}$ to get a quantifier-free $\mathcal{L}_\mathbb{Z}$ formula, but our result is stronger since we prove the existence of an equivalent quantifier-free $\mathcal{L}^c_\mathbb{N}$ formula.

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

3 Timed pushdown automata

Clock constraints. We consider constraints which can separately speak about the integer $\lfloor x \rfloor$ and fractional value $\{x\}$ of a clock x. For the integer part we allow diagonal $\lfloor x \rfloor - \lfloor y \rfloor \sim k$

and modulo $\lfloor x \rfloor - \lfloor y \rfloor \equiv_m k$ constraints, as well as their non-diagonal counterparts $\lfloor x \rfloor \sim k$ and $\lfloor x \rfloor \equiv_m k$. For the fractional part we allow order constraints $\{x\} \leqslant \{y\}$ as well as testing for zero $\{x\} = 0$. Let X be a finite set of clocks. A *clock constraint* over X is a boolean combination of *atomic clock constraints* of one of the forms

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

where $x, y \in X$, $m \in \mathbb{N}$ and $k \in \mathbb{Z}$. Since we allow arbitrary boolean combinations, we allow as syntactic sugar also the atomic clock constraint **true**, which is always true, and variants with any $\sim \in \{\leqslant, <, \geqslant, >\}$ in place of $\leqslant$. A *clock valuation* is a mapping $\mu \in \mathbb{Q}_{\geqslant 0}^X$ 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 1 (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 2 (Comparing with classical clock constraints). Integer and fractional constraints subsume classical ones. For clocks x, y, since $x = \lfloor x \rfloor + \{x\}$ (and similarly for y), $x - y \leqslant k$ for an integer k is equivalent to $(\lfloor x \rfloor - \lfloor y \rfloor \leqslant k \wedge \{x\} \leqslant \{y\}) \vee \lfloor x \rfloor - \lfloor y \rfloor \leqslant k - 1$, and similarly for $x \leqslant k$. On the other hand, fractional constraints are strictly more expressive than classical constraints since $\{x\} = 0$ is not expressible as a classical clock constraint.

▸ Remark 3 ($\lfloor x \rfloor - \lfloor y \rfloor$ 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 $\lfloor x \rfloor - \lfloor y \rfloor \leqslant k$ speaking of the *difference of the integer parts*, we would choose $\lfloor x - y \rfloor \equiv_m k$ and $\lfloor x - y \rfloor \leqslant k$ speaking of the *integer part of the difference*, since the two are inter-expressible:

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

Timed pushdown automata. 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:

- *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. An automaton has *untimed stack* if the only stack constraint is **true**. Without push/pop operations, we obtain a nondeterministic timed automaton (TA).

▸ Remark 4 (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 used by the automaton, and that all modular constraints use the same modulus M.

▸ Remark 5 (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 6 (Comparison with dtPDA [1]). The dtPDA model [1] allows only one stack clock $Z = \{z\}$ and stack constraints of the form $z \sim k$. As shown in [9], 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 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 $w \in (\Gamma \times \mathbb{Q}_{\geqslant 0}^{Z})^*$ 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 $\delta \in \mathbb{Q}_{\geqslant 0}$, 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 $\langle \ell, \mu, w \rangle \in L \times \mathbb{Q}_{\geqslant 0}^{X} \times (\Gamma \times \mathbb{Q}_{\geqslant 0}^{Z})^*$ where ℓ is a control location, μ is a clock valuation over the global clocks X, and w is a timed stack. Let $\langle \ell, \mu, u \rangle, \langle \reflectbox{ℓ}, \nu, v \rangle$ be two configurations. For every input symbol or time increment $a \in (\Sigma_\varepsilon \cup \mathbb{Q}_{\geqslant 0})$ we have a transition

$$\langle \ell, \mu, u \rangle \xrightarrow{a} \langle \reflectbox{ℓ}, \nu, v \rangle$$

whenever there exists a rule $\langle \ell, \mathsf{op}, \reflectbox{ℓ} \rangle \in \Delta$ s.t. one of the following holds:

(elapse) $\mathsf{op} = \mathsf{elapse}$, $a \in \mathbb{Q}_{\geqslant 0}$, $\nu = \mu + a$, $v = u + a$.

(input) $\mathsf{op} = a \in \Sigma_\varepsilon$.

(test) $\mathsf{op} = \varphi$, $a = \varepsilon$, $\mu \models \varphi$, $\nu = \mu$, $u = v$.

(reset) $\mathsf{op} = \mathsf{reset}(Y)$, $a = \varepsilon$, $\nu = \mu[Y \mapsto 0]$, $v = u$.

(push) $\mathsf{op} = \mathsf{push}(\gamma \models \psi)$, $a = \varepsilon$, $\mu = \nu$, $v = u \cdot \langle \gamma, \mu_1 \rangle$ provided that the stack valuation $\mu_1 \in \mathbb{Q}_{\geqslant 0}^{Z}$ satisfies $(\mu, \mu_1) \models \psi$, where $(\mu, \mu_1) \in \mathbb{Q}_{\geqslant 0}^{X \cup Z}$ is the unique clock valuation that agrees with μ on X and with μ_1 on Z.

(pop) $\mathsf{op} = \mathsf{pop}(\gamma \models \psi)$, $a = \varepsilon$, $\mu = \nu$, $u = v \cdot \langle \gamma, \mu_1 \rangle$ provided that the stack valuation $\mu_1 \in \mathbb{Q}_{\geqslant 0}^{Z}$ satisfies $(\mu, \mu_1) \models \psi$.

A *timed word* is a sequence $w = \delta_1 a_1 \cdots \delta_n a_n \in (\mathbb{Q}_{\geqslant 0} \Sigma_\varepsilon)^*$ of alternating time elapses and input symbols, and the one-step relation above on configurations is extended naturally on timed words as $\langle \ell, \mu, u \rangle \xrightarrow{w} \langle \reflectbox{ℓ}, \nu, v \rangle$. The *timed language* from location ℓ to $\reflectbox{$\ell$}$ is the set of timed words $L(\ell, \reflectbox{$\ell$}) := \left\{ \pi_\varepsilon(w) \in (\mathbb{Q}_{\geqslant 0} \Sigma)^* \;\middle|\; \langle \ell, \mu_0, \varepsilon \rangle \xrightarrow{w} \langle \reflectbox{ℓ}, \mu_0, \varepsilon \rangle \right\}$ 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* $L^{\mathrm{un}}(\ell, \reflectbox{ℓ})$ is obtained by removing the time elapses from $L(\ell, \reflectbox{$\ell$})$.

▸ **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:

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

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

$$\begin{array}{ll} \langle \ell, \alpha; \mathsf{push}(\alpha : \{z\} = 0), \ell\rangle & \langle \ell, \varepsilon, \imath\rangle \\ \langle \imath, \alpha; \mathsf{pop}(\alpha : \{z\} = 0), \imath\rangle & \langle \ell, \mathsf{elapse}, \ell\rangle, \langle \imath, \mathsf{elapse}, \imath\rangle \end{array}$$

We have $L = L(\ell, \imath)$. Since L cannot be recognised by TPDA with untimed stack, this shows that 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, $\imath$, clock valuations $\mu, \nu \in \mathbb{Q}^X_{\geqslant 0}$, and a timed word $w \in (\mathbb{Q}_{\geqslant 0}\Sigma_\varepsilon)^*$, we write $\mu \overset{w}{\rightsquigarrow}_{\ell\imath} \nu$ if $\langle \ell, \mu, \varepsilon\rangle \xrightarrow{w} \langle \imath, \nu, \varepsilon\rangle$. We overload the notation and, for $\pi \in \mathbb{N}^\Sigma$, we write $\mu \overset{\pi}{\rightsquigarrow}_{\ell\imath} \nu$ if there exists a timed word w s.t. $\mu \overset{w}{\rightsquigarrow}_{\ell\imath} \nu$ and $\pi = \mathrm{PI}_w$. Thus, $\rightsquigarrow_{\ell\imath}$ is a subset of $\mathbb{Q}^X_{\geqslant 0} \times \mathbb{N}^\Sigma \times \mathbb{Q}^X_{\geqslant 0}$.

Let $\left\{\psi_{\ell\imath}(\lfloor \overline{x}\rfloor, \{\overline{x}\}, \overline{f}, \lfloor \overline{y}\rfloor, \{\overline{y}\})\right\}_{\ell,\imath\in L}$ be a family $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$ 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 is *expressed* by $\{\psi_{\ell\imath}\}$ if: $\mu \overset{\pi}{\rightsquigarrow}_{\ell\imath} \nu$ iff $(\lfloor\mu\rfloor, \{\mu\}, \pi, \lfloor\nu\rfloor, \{\nu\}) \models \varphi_{\ell\imath}$.

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 $\rightsquigarrow_{\ell\imath}$ of a* TPDA *is expressed by a family of existential $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$ formulas $\psi_{\ell\imath}$ computable in double exponential time. For* TA*, the complexity is singly exponential.*

For TA this is a strengthening of [12, 20] as our model, even without stack, is more expressive than classical TA due to fractional constraints. As a side effect of the proof we get:

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

4 Fractional TPDA

We show that computing the TPDA reachability relation reduces to computing the reachability relation of a very restricted form of TPDA called *fractional* TPDA. Our transformation is done in three steps, by further restricting the set of allowed constraints at every step:

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

$$\psi_{\text{copy}}(\overline{x}, z_{\overline{x}}) \equiv \bigwedge_{x\in X} \lfloor z_x\rfloor = \lfloor x\rfloor \wedge \{z_x\} = \{x\}\,. \tag{2}$$

We use quantifier elimination and introduce exponentially many new pop constraints.

B The TPDA is *pop-integer-free*, that is, pop transitions do not contain integer constraints. This step introduces one global clock for each integer pop constraint. Moreover, the number of locations is exponential in the number of clocks and pop constraints, and the number of stack symbols is exponential in the number of pop constraints. Thus, when combined with the previous step, exponentially many new clocks are introduced and doubly exponentially many locations/stack symbols. This is similar to a construction from [9] and is presented in Sec. A.4.

C The TPDA is *fractional*, that is, only fractional constraints are allowed. This step introduces an exponential blowup of control locations w.r.t. global clocks and in the size of the binary encoding of 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 effectively be transformed into a fractional* TPDA $\mathcal{Q}$ *s.t. a family of* $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$ *formulas* $\{\varphi_{\ell\ell'}\}$ *describing the reachability relation of* $\mathcal{P}$ *can effectively be computed from a family of* $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$ *formulas* $\{\varphi'_{\ell'\ell''}\}$ *describing 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, 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 a singly exponential blowup in control locations.*

(A) The TPDA is push-copy

For $a, b, c \in \mathbb{I}$ define the strict *cyclic order* $K_<(a,b,c)$, and its non-strict variant $K_\leqslant(a,b,c)$:

$$\begin{aligned} K_<(a,b,c) &\equiv a < b < c \;\vee\; b < c < a \;\vee\; c < a < b \\ K_\leqslant(a,b,c) &\equiv K(a,b,c) \;\vee\; a = b \;\vee\; b = c. \end{aligned} \tag{3}$$

These relations will be useful in this section as well as in Sect. 5 to speak conveniently about the relation of fractional parts of clocks. 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. For the correctness of the following calculations, assume that there exists a special stack clock z_0 which is always initialized to 0 when pushed on the stack. In this way, 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 (thus $z'_{x_0} = z'_0$). 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

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

Note that the assumption that $z_0 = 0$ at every 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 $\{y\} \leqslant \{z\}$ in ψ_{push} is equivalent to $\{z'_y - z'_0\} \leqslant \{z' - z'_0\}$, which is in turn equivalent to $C = K_{\leqslant}(\{z'_0\}, \{z'_y\}, \{z'\})$, which is defined using only $\leqslant$ (c.f. (3)). Furthermore, we have

$$\{z'_y - z'_0\} - \{z' - z'_0\} = \{z'_y - z'\} - \mathbb{1}_{C?}, \tag{6}$$

which allows us to expand the quantity $\lfloor y \rfloor - \lfloor z \rfloor$ appearing in ψ_{push} integer and modular constraints as

$$\begin{aligned}\lfloor z'_y - z'_0 \rfloor - \lfloor z' - z'_0 \rfloor &= (z'_y - z'_0 - \{z'_y - z'_0\}) - (z' - z'_0 - \{z' - z'_0\}) = \\ &= (z'_y - z') - \{z'_y - z'_0\} - \{z' - z'_0\} = \text{(by (6))} \\ &= (z'_y - z') - \{z'_y - z'\} + \mathbb{1}_{C?} = \lfloor z'_y - z' \rfloor + \mathbb{1}_{C?}.\end{aligned}$$

(Notice that $\lfloor z'_0 \rfloor$ disappears in this process: This is not a coincidence, since diagonal constraints are invariant under the elapse of an *integer* amount of time.) Thus by (1) 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 1) (5) is equivalent to

$$\exists \lfloor \overline{z}' \rfloor, \{\overline{z}'\} \cdot \psi'_{\mathsf{push}}(\lfloor \overline{z}'_{\overline{x}} \rfloor, \{\overline{z}'_{\overline{x}}\}, \lfloor \overline{z}' \rfloor, \{\overline{z}'\}) \wedge \psi_{\mathsf{pop}}(\lfloor \overline{x}' \rfloor, \{\overline{x}'\}, \lfloor \overline{z}' \rfloor, \{\overline{z}'\}).$$

By Corollary 3 we can perform quantifier elimination and we obtain a logically equivalent clock constraint of exponential size (in DNF)

$$\xi_{\psi_{\mathsf{push}}, \psi_{\mathsf{pop}}}(\lfloor \overline{x}' \rfloor, \{\overline{x}'\}, \lfloor \overline{z}'_{\overline{x}} \rfloor, \{\overline{z}'_{\overline{x}}\}), \tag{7}$$

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}, assumed 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. The formal construction and its proof of correctness are presented in Sec. A.3.

▸ **Lemma 9.** *Let $\leadsto_{\ell \ell}$ and $\leadsto'_{\ell \ell}$ be the reachability relations of $\mathcal{P}$ and $\mathcal{P}'$, respectively. Then, $\leadsto_{\ell \ell} = \leadsto'_{\ell \ell}$, 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**). Diagonal integer $\lfloor y \rfloor - \lfloor x \rfloor \sim k$ and modulo $\lfloor y \rfloor - \lfloor x \rfloor \equiv_m k$ constraints can be converted to their non-diagonal counterparts by a standard procedure like in classical timed automata [2], by introducing a new global clock for each such constraint. Thus, in the rest of the section, transition and stack constraints of $\mathcal{P}$ are of the form

$$\begin{array}{llllll} \text{(trans.)} & \lfloor x \rfloor \leqslant k, & \lfloor x \rfloor \equiv_m k, & \{x\} = 0, & \{x\} \leqslant \{y\}, & (8) \\ \text{(push)} & \lfloor z_x \rfloor = \lfloor x \rfloor, & & & \{z_x\} = \{x\}, & (9) \\ \text{(pop)} & & \lfloor y \rfloor - \lfloor z_x \rfloor \equiv_m k, & \{z_x\} = 0, & \{y\} \leqslant \{z_x\}, & (10) \\ & & \lfloor z_y \rfloor - \lfloor z_x \rfloor \equiv_m k, & & \{z_y\} \leqslant \{z_x\}. & \end{array}$$

Recall that we assume that all modular constraints use the same modulus M. Assume additionally that all constants in integer constraints are strictly below M. Let x_0 be a global clock that is never reset, and let z_0 be a stack clock which has always fractional value 0 at the time of push. The main idea is to replace the integer value of clocks by the following *unary abstraction*.

Unary abstraction. Two clock valuations $\mu, \nu \in \mathbb{Q}_{\geqslant 0}^X$ 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 $\lfloor\mu(x)\rfloor \leqslant M \Leftrightarrow \lfloor\nu(x)\rfloor \leqslant M$. 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 $\nu[x \mapsto \nu(x)+1]$; in both cases the choice of $\nu \in \lambda$ is irrelevant. Let $\varphi_\lambda(\overline{x})$ be the following constraint expressing that clocks belong to λ: $\bigwedge_{x \in X} \lfloor x\rfloor \equiv_M \lambda(x) \wedge (\lfloor x\rfloor < M \Leftrightarrow \lambda(x) < M)$. For φ containing transition constraints of the form (8), $\varphi|_\lambda$ is φ where every integer $\lfloor x\rfloor \leqslant k$ 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 (10), $\psi|_{\lambda_{\mathsf{push}}, \lambda_{\mathsf{pop}}}$ 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 (cf. Sec. A.5 for the full construction). We eliminate integer $\lfloor x\rfloor \leqslant k$ 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 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 $\langle \ell, \varphi, \imath\rangle$ is simulated by $\langle\langle \ell, \lambda, Y\rangle, \varphi|_\lambda, \langle \imath, \lambda, Y\rangle\rangle$. A push-copy transition $\langle \ell, \mathsf{push}(\alpha : \psi_{\mathrm{copy}}), \imath\rangle$ is simulated by a push transition copying only the fractional parts and the unary class of global clocks:

$$\langle\langle \ell, \lambda, Y\rangle, \mathsf{push}(\langle \alpha, \lambda\rangle : \bigwedge_{x \in X} \{z_0\} = 0 \wedge \{z_x\} = \{x\}), \langle \imath, \lambda, Y\rangle\rangle.$$

A pop-integer-free transition $\langle \ell, \mathsf{pop}(\alpha : \psi), \imath\rangle$ is simulated by a fractional pop transition

$$\langle\langle \ell, \lambda_{\mathsf{pop}}, Y\rangle, \mathsf{pop}(\langle \alpha, \lambda_{\mathsf{push}}\rangle : \psi|_{\lambda_{\mathsf{push}}, \lambda_{\mathsf{pop}}}), \langle \imath, \lambda_{\mathsf{pop}}, Y\rangle\rangle.$$

The reachability formula $\varphi_{\ell\imath}$ 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.

By *fractional* reachability we mean the reachability relation where the integer parts of initial and final values of clocks are ignored and only fractional parts are taken into account.

▸ **Lemma 10.** *Let the fractional reachability relation of $\mathcal{Q}$ be expressed by $\psi_{\ell'\imath'}(\{\overline{x}\}, (\overline{f}, \overline{g}), \{\overline{x}'\})$ using only the fractional values of clocks $\{\overline{x}\}, \{\overline{x}'\}$, the Parikh image $\overline{f}$ of the original input letters from Σ, and the Parikh image $\overline{g}$ of the new input letters $\checkmark_x$. The reachability relation of $\mathcal{P}$ can be expressed by*

$$\varphi_{\ell\imath}(\lfloor\overline{x}\rfloor, \{\overline{x}\}, \overline{f}, \lfloor\overline{x}'\rfloor, \{\overline{x}'\}) \equiv \bigvee_{\lambda, Y, \mu} \varphi_\lambda(\lfloor\overline{x}\rfloor) \wedge \psi_{\langle \ell, \lambda, Y\rangle\langle \imath, \mu, X\rangle}(\{\overline{x}\}, (\overline{f}, \overline{g}^Y), \{\overline{x}'\}).$$

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.

▸ **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 structure. We model fractional clock values by the structure $\mathcal{C} = (\mathbb{I}, K_<, =)$ with universe $\mathbb{I} = \mathbb{Q} \cap [0, 1)$, where $K_<$ is the strict ternary cyclic order defined in (3). Since $K_<$ is invariant under cyclic shift, it is convenient to think of elements of $\mathcal{C}$ as placed clockwise on a circle of perimeter 1; cf. Fig. 1(a). An *automorphism* of $\mathcal{C}$ is a bijection α that preserves and reflects $K_<$, i.e., $K_<(a, b, c)$ iff $K_<(\alpha(a), \alpha(b), \alpha(c))$.
The structure $\mathcal{C}$ is homogeneous [18] and thus for every n the set $\mathbb{I}^n$ of n-tuples of elements of $\mathcal{C}$ splits into exponentially many *orbits*, where $u, v \in \mathbb{I}^n$ are in the same orbit if some automorphism maps u to v; let $\mathrm{Orb}(\mathbb{I}^n)$ denote the set of orbits. An orbit is an equivalence class of tuples which are indistinguishable from the point of view of $\mathcal{C}$; orbits are analogous to 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.

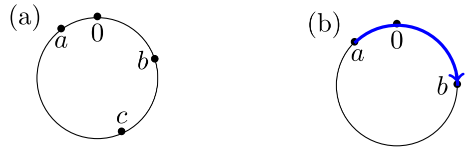

Figure 1 (a) $K_<(a, b, c)$ is invariant under cyclic shift. (b) The cyclic difference $b \ominus a$, interpreted as the length of the clockwise-oriented arc from a to b.

$\mathcal{C}$-PDA. We extend classical PDA with additional *registers* holding values from $\mathcal{C}$, both in the finite control (i.e., global registers) and in the stack. Registers can be compared by quantifier-free formulas over $\mathcal{C}$ (called *$\mathcal{C}$-constraints*). For simplicity, we assume that there are the same number of global and stack registers. A *$\mathcal{C}$-PDA* 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 $\langle \ell, \mathsf{op}, \ell' \rangle$ with $\ell, \ell' \in L$ control locations, where op determines the type of transition:

- *input* $\mathsf{op} = a \in \Sigma_\varepsilon$ an input letter,
- *global transition* $\mathsf{op} = \psi(\overline{x}, \overline{x}')$ an $2k$-ary $\mathcal{C}$-constraint relating pre- and post-values of global registers,
- *push* $\mathsf{op} = \mathsf{push}(\alpha : \psi(\overline{x}, \overline{z}))$ with $\alpha \in \Gamma$ a stack symbol to be pushed on the stack under the $2k$-ary $\mathcal{C}$-constraint ψ relating global $\overline{x}$ and stack $\overline{z}$ registers, or
- *pop* $\mathsf{op} = \mathsf{pop}(\alpha : \psi(\overline{x}, \overline{z}))$, similarly as push.

We consider $\mathcal{C}$-PDA as symbolic representations of classical PDA wit 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 a stack. Let $\langle \ell, \mu, u \rangle, \langle \ell', \nu, v \rangle$ be two configurations. For every input symbol $a \in \Sigma_\varepsilon$ we have a transition

$$\langle \ell, \mu, u \rangle \xrightarrow{a} \langle \ell', \nu, v \rangle$$

whenever there exists a rule $\langle \ell, \mathsf{op}, \ell' \rangle \in \Delta$ s.t. one of the following holds:

(input) $\mathsf{op} = a \in \Sigma_\varepsilon$, $\mu = \nu$, $u = v$.
(global) $\mathsf{op} = \varphi$, $a = \varepsilon$, $(\mu, \nu) \models \varphi$, $u = v$.
(push) $\mathsf{op} = \mathsf{push}(\gamma \models \psi)$, $a = \varepsilon$, $\mu = \nu$, $v = u \cdot \langle \gamma, \mu_1 \rangle$ if $\mu_1 \in \mathbb{I}^Z$ satisfies $(\mu, \mu_1) \models \psi$.
(pop) $\mathsf{op} = \mathsf{pop}(\gamma \models \psi)$, $a = \varepsilon$, $\mu = \nu$, $u = v \cdot \langle \gamma, \mu_1 \rangle$ if $\mu_1 \in \mathbb{I}^Z$ satisfies $(\mu, \mu_1) \models \psi$.

Reachability relation. The reachability relations $\mu \overset{w}{\rightsquigarrow}_{\ell \rotatebox{90}{\ensuremath{\ell}}} \nu$ and $\mu \overset{f}{\rightsquigarrow}_{\ell \rotatebox{90}{\ensuremath{\ell}}} \nu$ are defined as for TPDA by extending one-step transitions $\langle \ell, \mu, u \rangle \xrightarrow{a} \langle \rotatebox{90}{\ensuremath{\ell}}, \nu, v \rangle$ to words $w \in \Sigma^*$ and their Parikh images $f = \mathrm{PI}_w \in \mathbb{N}^\Sigma$. Thus, $\mu \overset{f}{\rightsquigarrow}_{\ell \rotatebox{90}{\ensuremath{\ell}}} \nu$ 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 $\mathbb{I}^{X \times X'}$.

▸ **Lemma 12.** *If $(\mu, \nu), (\mu', \nu')$ belong to the same orbit of $\mathbb{I}^{X \times X'}$, then $\mu \overset{f}{\rightsquigarrow}_{\ell \rotatebox{90}{\ensuremath{\ell}}} \nu$ iff $\mu' \overset{f}{\rightsquigarrow}_{\ell \rotatebox{90}{\ensuremath{\ell}}} \nu'$.*

The lemma is a special case of a more general result that holds for PDA with atoms, or FO-definable PDA (the reachability relation of an equivariant PDA is equivariant, cf. [8] or Sec. 9 in [4] for details). Likewise the next lemma:

▸ **Lemma 13.** *Given a $\mathcal{C}$-PDA $\mathcal{Q}$ one can construct a context-free grammar G of exponential size with nonterminals of the form $X_{\ell \rotatebox{90}{\ensuremath{\ell}} o}$, for control locations ℓ, $\rotatebox{90}{\ensuremath{\ell}}$ and an orbit $o \in \mathrm{Orb}(\mathbb{I}^{X \times X'})$, recognising the language*

$$L(X_{\ell \rotatebox{90}{\ensuremath{\ell}} o}) = \left\{ \pi_\Sigma(w) \in \Sigma^* \;\middle|\; \exists (\mu, \nu) \in o \cdot \mu \overset{w}{\rightsquigarrow}_{\ell \rotatebox{90}{\ensuremath{\ell}}} \nu \right\}, \tag{11}$$

where $\pi_\Sigma(w)$ is w without the ε's. Consequently, $\mathcal{C}$-PDAs recognise context-free languages.

▸ **Lemma 14** (Theorem 4 of [23]). *The Parikh image of $L(X_{\ell \rotatebox{90}{\ensuremath{\ell}} o})$ is described by an existential Presburger formula $\varphi^{\mathbb{Z}}_{\ell \rotatebox{90}{\ensuremath{\ell}} o}$ computable in time linear in the size of the grammar.*

Denote by $\varphi^{\mathbb{I}}_o$ the characteristic quantifier-free $\mathcal{C}$-formula of an orbit $o \in \mathrm{Orb}(\mathbb{I}^{X \times X'})$ which specifies the cyclic order and the equality type of $2|X|$ elements. Relying on Lemma 12, we deduce that the reachability relation is definable in a two sorted logic: $\mathcal{L}_{\mathbb{Z}}$ on the integer sort, and $\mathcal{C}$-constraints on the other sort.

▸ **Corollary 15.** *The reachability relation $\rightsquigarrow_{\ell \rotatebox{90}{\ensuremath{\ell}}}$ in a $\mathcal{C}$-PDA $\mathcal{Q}$ is characterised by the formula*

$$\varphi_{\ell \rotatebox{90}{\ensuremath{\ell}}}(\overline{x}, \overline{f}, \overline{x}') \equiv \bigvee_{o \in \mathrm{Orb}(\mathbb{I}^{X \times X'})} \varphi^{\mathbb{Z}}_{\ell \rotatebox{90}{\ensuremath{\ell}} o}(\overline{f}) \wedge \varphi^{\mathbb{I}}_o(\overline{x}, \overline{x}'). \tag{12}$$

The size of $\varphi_{\ell \rotatebox{90}{\ensuremath{\ell}}}$ 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))

$$\varphi_\ominus(\overline{x}, \overline{\hat{x}}) \equiv \bigwedge_{x \in X} \{x\} = \hat{x}_0 \ominus \hat{x}. \tag{13}$$

Resetting clocks in $Y \subseteq X$ is simulated by $\varphi_{\mathsf{reset}(Y)}$ and time elapse by $\varphi_{\mathsf{elapse}}$:

$$\varphi_{\mathsf{reset}(Y)} \equiv \hat{x}'_0 = \hat{x}_0 \wedge \bigwedge_{x \in Y} \hat{x}' = \hat{x}_0 \wedge \bigwedge_{x \in X \setminus Y} \hat{x}' = \hat{x} \qquad \text{and} \qquad \varphi_{\mathsf{elapse}} \equiv \bigwedge_{x \in X} \hat{x}' = \hat{x}.$$

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 register constraint $\hat{\varphi}$ by replacing $\{x\} = 0$ with $\hat{x} = \hat{x}_0$ and $\{x\} \leqslant \{y\}$ by $K(\hat{y}, \hat{x}, \hat{x}_0)$, for $x, y \in X \cup Z$. Let $\mathcal{P}$ be the TPDA $\langle \Sigma, \Gamma, L, X, Z, \Delta \rangle$. We define a simulating $\mathcal{C}$-PDA $\mathcal{Q} = \left\langle \Sigma, \Gamma, L, \hat{X}_{x_0}, \hat{Z}, \hat{\Delta} \right\rangle$ over the same input, stack alphabet, and control locations as $\mathcal{P}$. The input rules are preserved. A reset rule $\langle \ell, \mathsf{reset}(Y), \ell' \rangle \in \Delta$, is simulated by the global rule $\langle \ell, \varphi_{\mathsf{reset}(Y)}, \ell' \rangle \in \hat{\Delta}$, a time elapse rule $\langle \ell, \mathsf{elapse}, \ell' \rangle \in \Delta$ is simulated by the global rule $\langle \ell, \varphi_{\mathsf{elapse}}, \ell' \rangle \in \hat{\Delta}$, a push rule $\langle \ell, \mathsf{push}(\gamma : \varphi), \ell' \rangle \in \Delta$ is simulated by the push rule $\langle \ell, \mathsf{push}(\gamma : \hat{\varphi}), \ell' \rangle \in \hat{\Delta}$, and similarly for pop rules.

By Corollary 15, let $\varphi_{\ell \ell'}(\overline{\hat{x}}, \overline{f}, \overline{\hat{x}'})$ as in (12) express the reachability relation of $\mathcal{Q}$. The reachability relation $\psi_{\ell \ell'}$ of $\mathcal{P}$ is recovered as (the correctness is shown in Sec. A.6.1)

$$\begin{aligned} \psi_{\ell \ell'}(\overline{x}, \overline{f}, \overline{x}') \equiv & \bigvee_{o \in \mathrm{Orb}(\mathbb{I}^{X \times X'})} \varphi^{\mathbb{Z}}_{\ell \ell' o}(\overline{f}) \wedge \xi^{\mathbb{I}}_o(\overline{x}, \overline{x}'), \\ & \text{where } \xi^{\mathbb{I}}_o(\overline{x}, \overline{x}') \equiv \exists \overline{\hat{x}}, \overline{\hat{x}'} \cdot \varphi^{\mathbb{I}}_o(\overline{\hat{x}}, \overline{\hat{x}'}) \wedge \varphi_{\ominus}(\overline{x}, \overline{\hat{x}}) \wedge \varphi_{\ominus}(\overline{x}', \overline{\hat{x}'}). \end{aligned} \tag{14}$$

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 (13)):

$$\hat{x} \mapsto \hat{x}_0 \ominus \{x\} \qquad \hat{x}' \mapsto (\hat{x}_0 \ominus \delta) \ominus \{x'\} \qquad \hat{x}_0' \mapsto \hat{x}_0 \ominus \delta.$$

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. (3)), we obtain only constraints of the form $u \precsim v$ with $\precsim \in \{<, \leqslant\}$. 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 $\delta \oplus \{x'\}$, we obtain the following formula equivalent to $\xi^{\mathbb{I}}_o$

$$\bigwedge_i s_i \precsim t_i \ \wedge \ \exists 0 \leqslant \delta < 1 \cdot \bigwedge_j u_j \precsim \delta \wedge \bigwedge_k \delta \precsim v_k \tag{15}$$

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 $\xi^{\mathbb{I}}_o(\overline{x}, \overline{x}') \equiv \bigwedge_h s_h' \precsim t_h'$, where the s_h', t_h''s are of one of the forms: 0, $\{x\}$, $\{x\} - \{y'\}$, or $\{x\} - \{y'\} + 1$. ◂

6 Conclusions

We have shown that the ternary (and *a posteriori* binary) reachability relation of TPDA is expressible in a fragment of linear arithmetic, which is a hybrid logic combining $\mathbb{N}$ and $\mathbb{I}$-valued variables. This same fragment already expresses the reachability relation of TA, i.e., without a stack [20], and thus our result shows that it suffices for the more general TPDA. This result should be compared to the TA *clock difference relations* of [17], which are similar fragment of (non-hybrid) linear rational arithmetic, where clocks are not split into integer and fractional part, yielding $\mathbb{Q}$-valued variables instead.

References

1 P. A. Abdulla, M. F. Atig, and J. Stenman. Dense-timed pushdown automata. In *Proc. LICS'12*, pages 35–44. IEEE, 2012.

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

3 B. Bérard, A. Petit, V. Diekert, and P. Gastin. Characterization of the expressive power of silent transitions in timed automata. *Fundam. Inf.*, 36(2–3):145–182, 1998.

4 M. Bojańczyk. Slightly infinite sets. A draft of a book available at `https://www.mimuw.edu.pl/~bojan/paper/atom-book`.

5 M. Bojańczyk and S. Lasota. A machine-independent characterization of timed languages. In *Proc. ICALP'12*, volume 7392 of *LNCS*, pages 92–103. Springer, 2012.

6 A. Bouajjani, R. Echahed, and R. 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.

7 C. Choffrut and M. Goldwurm. Timed automata with periodic clock constraints. *J. Autom. Lang. Comb.*, 5(4):371–403, 2000.

8 L. Clemente and S. Lasota. Reachability analysis of first-order definable pushdown systems. In *Proc. of CSL'15*, volume 41 of *LIPIcs*, pages 244–259. Dagstuhl, 2015.

9 L. Clemente and S. Lasota. Timed pushdown automata revisited. In *Proc. LICS'15*, pages 738–749. IEEE, July 2015.

10 L. Clemente and S. Lasota. Timed pushdown automata revisited. Technical report, University of Warsaw, 2015.

11 L. Clemente, S. Lasota, R. Lazić, and F. Mazowiecki. Timed pushdown automata and branching vector addition systems. In *Proc. of LICS'17*, 2017.

12 H. Comon and Y. 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.

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

14 Z. Dang, O. H. Ibarra, T. Bultan, R. A. Kemmerer, and J. Su. Binary reachability analysis of discrete pushdown timed automata. In E. A. Emerson and A. P. Sistla, editors, *Proc. of CAV 2000*, pages 69–84, Berlin, Heidelberg, 2000.

15 C. Dima. Computing reachability relations in timed automata. In *In Proc. of LICS'02*, pages 177–186, 2002.

16 J. Ferrante and C. Rackoff. A decision procedure for the first order theory of real addition with order. *SIAM Journal on Computing*, 4(1):69–76, 1975.

17 P. Krčál and R. Pelánek. On sampled semantics of timed systems. In S. Sarukkai and S. Sen, editors, *In Proc. of FSTTCS'05*, volume 3821 of *LNCS*, pages 310–321. Springer, 2005.

18 D. Macpherson. A survey of homogeneous structures. *Discrete Mathematics*, 311(15):1599–1634, 2011.

19 M. 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.

20 K. Quaas, M. Shirmohammadi, and J. Worrell. Revisiting reachability in timed automata. In *Proc. of LICS'17*, pages 1–12, June 2017.

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

22 Y. Uezato and Y. Minamide. Synchronized recursive timed automata. In *Proc. of LPAR'15*, 2015.

23 K. N. Verma, H. Seidl, and T. Schwentick. On the complexity of equational Horn clauses. In *Proc. CADE-20, 2005*, pages 337–352, 2005.
24 V. Weispfenning. The complexity of linear problems in fields. *Journal of Symbolic Computation*, 5(1):3–27, 1988.

A Appendix

A.1 Quantifier elimination

▶ **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.*

Proof. It suffices to consider a conjunctive formula of the form $\varphi \equiv \exists y \cdot \varphi_1 \wedge \varphi_2$ where φ_1 is a quantifier-free $\mathcal{A}_1$-formula and φ_2 is a quantifier-free $\mathcal{A}_2$-formula. W.l.o.g. suppose y is quantified over $\mathcal{A}_1$. Since y is a variable of the first sort, it does not appear free in φ_2, and thus $\varphi \equiv (\exists y \cdot \varphi_1) \wedge \varphi_2$. By assumption that $\mathcal{A}_1$ admits quantifier elimination, $\exists y \cdot \varphi_1$ is equivalent to a quantifier free formula $\widetilde{\varphi}_1$, and thus the original formula φ is equivalent to $\widetilde{\varphi}_1 \wedge \varphi_2$. It is easy to see that the complexities combine as claimed. ◀

Let $[\![\varphi]\!]$ be the set of valuations satisfying φ.

▶ **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.*

We prove this by splitting it in two claims.

▶ **Lemma 16.** *The structure $\mathcal{A}^{c}_{\mathbb{N}}$ admits effective elimination of quantifiers. The complexity is singly exponential for conjunctive formulas.*

Proof. We assume that all modulo statements are over the same modulus m. It suffices to consider a conjunctive formula of the form

$$\exists y \cdot \varphi \;\equiv\; \exists y \cdot \bigwedge_i x_i + \alpha_i \leqslant y \leqslant x_i + \beta_i \;\wedge\; y \equiv_m x_i + \gamma_i, \tag{16}$$

where, for every i, $\alpha_i, \beta_i \in \mathbb{Z} \cup \{-\infty, +\infty\}$ with $\alpha_i \leqslant \beta_i$, $\gamma_i \in \{0, \dots, m-1\}$, where for uniformity of notation we assume $x_0 = 0, \alpha_0 \geqslant 0$ in order to model non-diagonal constraints on y. If not all α_i's are equal to $-\infty$, then a satisfying y will be of the form $x_j + \alpha_j + \delta$ with $\delta \in \{0, \dots, m-1\}$ where j maximises $x_j + \alpha_j$. We claim that the following quantifier free formula $\widetilde{\varphi}$ is equivalent to (16):

$$\bigvee_{\delta \in \{0,\dots,m-1\}} \bigvee_j \bigwedge_i x_i + \alpha_i \leqslant x_j + \alpha_j + \delta \leqslant x_i + \beta_i \;\wedge\; x_j + \alpha_j + \delta \equiv_m x_i + \gamma_i. \tag{17}$$

For the complexity claim, $\widetilde{\varphi}$ is exponentially bigger than (16) when constants are encoded in binary. For the inclusion $[\![\widetilde{\varphi}]\!] \subseteq [\![\exists y \cdot \varphi]\!]$, let $(a_1, \dots, a_n) \in [\![\widetilde{\varphi}]\!]$. There exist δ and j as per (17), and thus taking $a_0 := a_j + \alpha_j + \delta$ yields $(a_0, a_1, \dots, a_n) \in [\![\exists y \cdot \varphi]\!]$. For the other inclusion, let $(a_0, a_1, \dots, a_n) \in [\![\varphi]\!]$. Let $j \neq 0$ be s.t. $a_j + \alpha_j$ is maximised, and define $\delta := a_0 - (a_j + \alpha_j) \mod m$. Clearly $\delta \geqslant 0$ since a_0 satisfies all the lower bounds $a_i + \alpha_i$. Since a_0 satisfies all the upper bounds $a_i + \beta_i$ and $a_j + \alpha_j + \delta \leqslant a_0$, upper bounds are also satisfied. Finally, since $a_0 \equiv_m a_i + \gamma_i$ and $a_0 \equiv_m a_j + \alpha_j + \delta$, we have that also the modular constraints $a_j + \alpha_j + \delta \equiv_m a_i + \gamma_i$ are satisfied. Thus, we have $(a_1, \dots, a_n) \in [\![\widetilde{\varphi}]\!]$, as required.

If all α_i's are equal to $-\infty$, then there are no lower bound constraints and only modulo constraints remain, hence and a satisfying y (if it exists) can be taken in the interval $\{0, \ldots, m-1\}$, yielding

$$\bigvee_{\delta \in \{0,\ldots,m-1\}} \bigwedge_i \delta \leqslant x_i + \beta_i \ \wedge \ \delta \equiv_m x_i + \gamma_i.$$

The same complexity holds. The formula above is shown equivalent to (16) by a reasoning as in the previous paragraph. ◂

▸ **Lemma 17.** *The structure $\mathcal{A}^{\mathrm{c}}_{\mathbb{I}}$ admits effective elimination of quantifiers. The complexity is quadratic for conjunctive formulas.*

Proof. It suffices to consider a conjunctive formula of the form $\varphi \equiv \exists y \cdot \bigwedge_k \varphi_k$ where φ_k are atomic rational formulas. If any φ_k is the constraint $y = 0$, then we obtain $\widetilde{\varphi}$ by replacing y with 0 everywhere. Otherwise, φ is of the form

$$\exists y \cdot \bigwedge_{i \in I} x_i \leqslant y \wedge \bigwedge_{j \in J} y \leqslant x_j,$$

and we can eliminate y by writing the equivalent constraint $\widetilde{\varphi}$

$$\bigwedge_{i \in I} \bigwedge_{j \in J} x_i \leqslant x_j.$$

The size of $\widetilde{\varphi}$ is quadratic in the size of φ. ◂

A.2 Characterisation of the reachability relation

The following characterisation is used in the proof of Lemma 9.

▸ **Lemma 18.** *The relation $\leadsto_{\ell \wr}$ is the least relation satisfying the following rules, for valuations $\mu, \nu, \mu', \nu' : \mathbb{Q}^X$ and words $w, u, v \in \Sigma^*$:*

(input) $$\frac{}{\mu \overset{a}{\leadsto}_{\ell \wr} \mu} \qquad \textit{if } \exists \langle \ell, a, \wr \rangle \in \Delta$$

(test) $$\frac{}{\mu \overset{\varepsilon}{\leadsto}_{\ell \wr} \mu} \qquad \textit{if } \exists \langle \ell, \varphi, \wr \rangle \in \Delta \cdot \mu \models \varphi$$

(reset) $$\frac{}{\mu \overset{\varepsilon}{\leadsto}_{\ell \wr} \mu[Y \mapsto 0]} \qquad \textit{if } \exists \langle \ell, \mathsf{reset}(Y), \wr \rangle \in \Delta$$

(elapse) $$\frac{}{\mu \overset{\varepsilon}{\leadsto}_{\ell \wr} \nu} \qquad \textit{if } \exists \langle \ell, \mathsf{elapse}, \wr \rangle \in \Delta, \delta > 0 \cdot \nu = \{\mu + \delta\}$$

(push-pop) $$\frac{\mu \overset{w}{\leadsto}_{\ell' \wr'} \nu}{\mu \overset{w}{\leadsto}_{\ell \wr} \nu} \qquad \textit{if } (18)$$

(transitivity) $$\frac{\mu \overset{u}{\leadsto}_{\ell \ell'} \mu' \quad \mu' \overset{v}{\leadsto}_{\ell' \wr} \nu}{\mu \overset{uv}{\leadsto}_{\ell \wr} \nu}$$

$$\bigvee_{\substack{\langle \ell, \mathsf{push}(\gamma:\psi_{\mathsf{push}}), \ell' \rangle, \\ \langle \wr', \mathsf{pop}(\gamma:\psi_{\mathsf{pop}}), \wr \rangle \in \Delta}} \exists \mu_Z \in \mathbb{Q}^Z_{\geqslant 0}, \exists \delta \in \mathbb{Q}_{\geqslant 0} \cdot \begin{cases} (\mu, \mu_Z) \models \psi_{\mathsf{push}}(\overline{x}, \overline{z}) \ \wedge \\ (\nu, \mu_Z + \delta) \models \psi_{\mathsf{pop}}(\overline{x}, \overline{z}). \end{cases} \tag{18}$$

A.3 Missing details for (A) push-copy

Let Ξ be the set of all $\xi_{\psi_{\mathsf{push}},\psi_{\mathsf{pop}}}$'s. Let the original TPDA be $\mathcal{P} = (\Sigma, \Gamma, L, X, Z, \Delta)$, let Ψ_{push} be the set of all push constraints ψ_{push} of $\mathcal{P}$, and let Ψ_{pop} be the set of all pop constraints ψ_{pop} of $\mathcal{P}$. We construct an equivalent TPDA $\mathcal{P}' = (\Sigma, \Gamma', L, X, Z', \Delta')$ which only pushes on the stack copies of stack clocks. Let $\Gamma' = \Gamma \times \Xi$, $Z' = \{z_x \mid x \in X\}$, and transitions in Δ' are determined as follows.

Every input, test, time elapse, and clock reset transitions in $\mathcal{P}$ generate identical transitions in $\mathcal{P}'$. For every push transition $\langle \ell, \mathsf{push}(\alpha : \psi_{\mathsf{push}}), \not\ell\rangle$ in $\mathcal{P}$, we have a push transition in $\mathcal{P}'$ of the form

$$\langle \ell, \mathsf{push}(\big\langle \alpha, \xi_{\psi_{\mathsf{push}},\psi_{\mathsf{pop}}}\big\rangle : \psi_{\mathrm{copy}} \wedge z_0 = 0), \not\ell\rangle$$

($z_0 = 0$ is compatible with push-copy by adding a new clock x_0 which is 0 at the time of push and using $z_0 = x_0$; we avoid this for simplicity) for every guessed pop constraint $\psi_{\mathsf{pop}} \in \Psi_{\mathsf{pop}}$ of $\mathcal{P}$ and corresponding new pop constraint $\xi_{\psi_{\mathsf{push}},\psi_{\mathsf{pop}}} \in \Xi$ from (7), and where ψ_{copy} is as in (2). Finally, for every pop transition $\langle \ell, \mathsf{pop}(\alpha : \psi_{\mathsf{pop}}), \not\ell\rangle$ in $\mathcal{P}$ and for every potential push constraint $\psi_{\mathsf{push}} \in \Psi_{\mathsf{push}}$, we have a pop transition in $\mathcal{P}'$

$$\langle \ell, \mathsf{pop}(\big\langle \alpha, \xi_{\psi_{\mathsf{push}},\psi_{\mathsf{pop}}}\big\rangle : \xi_{\psi_{\mathsf{push}},\psi_{\mathsf{pop}}}), \not\ell\rangle$$

which checks that the pop constraint ψ_{pop} was indeed correctly guessed.

This translation preserves the reachability relation.

▶ **Lemma 9.** *Let $\leadsto_{\ell \not\ell}$ and $\leadsto'_{\ell \not\ell}$ be the reachability relations of $\mathcal{P}$ and $\mathcal{P}'$, respectively. Then, $\leadsto_{\ell \not\ell} = \leadsto'_{\ell \not\ell}$, and $\mathcal{P}'$ has stack alphabet exponential in the size of $\mathcal{P}$.*

Proof. We prove

$$\mu \overset{w}{\leadsto}_{\ell \not\ell} \nu \quad \Leftrightarrow \quad \mu \overset{w}{\leadsto}{}'_{\ell \not\ell} \nu$$

by induction on the length of derivations, following the characterisation of Lemma 18. Let $\mu \overset{w}{\leadsto}_{\ell \not\ell} \nu$ (the other direction is proved analogously). Since all transitions are the same except push and pop transitions, it suffices to prove it for matching pairs of push-pop transitions. By (18), there exist transitions $\langle \ell, \mathsf{push}(\gamma : \psi_{\mathsf{push}}), \ell'\rangle, \langle \not\ell', \mathsf{pop}(\gamma : \psi_{\mathsf{pop}}), \not\ell\rangle \in \Delta$, a stack clock valuation $\mu_Z \in \mathbb{Q}_{\geqslant 0}^Z$, and a time elapse $\delta \in \mathbb{Q}_{\geqslant 0}$ s.t. $(\mu, \mu_Z) \models \psi_{\mathsf{push}}(\overline{x}, \overline{z})$, $(\nu, \mu_Z + \delta) \models \psi_{\mathsf{pop}}(\overline{x}', \overline{z}')$, and $\mu \overset{w}{\leadsto}_{\ell' \not\ell'} \nu$ in $\mathcal{P}$. By inductive hypothesis, $\mu \overset{w}{\leadsto}{}'_{\ell' \not\ell'} \nu$ in $\mathcal{P}'$. By construction, $\mathcal{P}'$ has matching transitions $\langle \ell, \mathsf{push}(\big\langle \gamma, \xi_{\psi_{\mathsf{push}},\psi_{\mathsf{pop}}}\big\rangle : \psi_{\mathrm{copy}}), \ell'\rangle$ and $\langle \not\ell', \mathsf{pop}(\big\langle \gamma, \xi_{\psi_{\mathsf{push}},\psi_{\mathsf{pop}}}\big\rangle : \xi_{\psi_{\mathsf{push}},\psi_{\mathsf{pop}}}), \not\ell\rangle$. Clearly, $(\mu, \mu) \models \psi_{\mathrm{copy}}(\overline{x}, \overline{z}_{\overline{x}})$, where z_x is the stack clock copying the value of clock x at the time of push. Since stack clock z_0 was initially 0, we have that its value at the end is exactly δ. We show that

$$(\overline{x}' : \nu, \overline{z}'_{\overline{x}} : \mu + \delta) \models \xi_{\psi_{\mathsf{push}},\psi_{\mathsf{pop}}}(\overline{x}', \overline{z}'_{\overline{x}}),$$

thus showing $\mu \overset{w}{\leadsto}{}'_{\ell' \not\ell'} \nu$ in $\mathcal{P}'$ by (18). By its definition, $\xi_{\psi_{\mathsf{push}},\psi_{\mathsf{pop}}}(\overline{x}', \overline{z}'_{\overline{x}})$ is equivalent to $\psi'_{\mathsf{pop}}(\overline{x}', \overline{z}'_{\overline{x}})$ from (5). Take $\mu_Z + \delta$ as the valuation for $\overline{z}'$, and we have

$$(\overline{x}' : \nu, \overline{z}' : \mu_Z + \delta, \overline{z}'_{\overline{x}} : \mu + \delta) \models \psi_{\mathsf{push}}(\overline{z}'_{\overline{x}} - \overline{\delta}, \overline{z}' - \overline{\delta}) \wedge \psi_{\mathsf{pop}}(\overline{x}', \overline{z}')$$

because $(\overline{x}' : \nu, \overline{z}' : \mu_Z + \delta) \models \psi_{\mathsf{pop}}(\overline{x}', \overline{z}')$ and $(\overline{z}_{\overline{x}} : \mu, \overline{z} : \mu_Z) \models \psi_{\mathsf{push}}(z_{\overline{x}}, \overline{z})$. ◀

A.4 (B) The TPDA is pop-integer-free

The aim of this section is to remove integer constraints from pop transitions. Thanks to (**A**), we assume that the TPDA is push-copy. Since diagonal integer constraints can simulate non-diagonal ones, we can further assume that pop transitions do not contain non-diagonal integer constraints (i.e., of the form $\lfloor z \rfloor \leqslant k$), and thus we only need to eliminate the diagonal ones.

Let $\mathcal{P}$ be a push-copy TPDA. By Remark 2, we replace integer pop constraints of the form $\lfloor x \rfloor - \lfloor z_y \rfloor \leqslant k$, $\lfloor z_x \rfloor - \lfloor z_y \rfloor \leqslant k$ by *classical* $x - z_y \leqslant k$, resp., $z_x - z_y \leqslant k$, and fractional constraints. This has the advantage that classical diagonal constraints are invariant under time elapse, which will simplify the construction below. Pop constraints of the form $z_y - z_x \sim k$ can easily be eliminated since, thanks to push-copy, they can be checked at the time of push as the *transition* constraint $y - x \sim k$. Thus, we concentrate on pop constraints

$$\psi_{\mathsf{pop}} \ \equiv \ \psi_1^{\mathrm{c}} \wedge \cdots \wedge \psi_m^{\mathrm{c}} \wedge \psi^{\mathrm{nc}} \tag{19}$$

where the ψ_i^{c}'s are classical diagonal constraints of the form $y - z_x \sim k$, with $\sim \in \{<, \leqslant, \geqslant, >\}$, and ψ^{nc} contains only non-classical (i.e., modular and fractional) constraints. Let be $\mathcal{C}$ the set of all ψ_i^{c}'s. Constraints $y - z_x \sim k$ are eliminated by introducing linearly many new global clocks (one for each atomic clock constraint) satisfying suitable conditions at the time of push. Thus, in the new automaton pop constraints are only of the form ψ^{nc}, i.e., modulo and fractional, as required. The construction is similar to [9]. Control states of the new automaton $\mathcal{P}'$ are of the form $\langle \ell, T, \Phi^-, \Phi^+ \rangle$, where T is a set of clocks and Φ^-, Φ^+ are sets of atomic constraints. Thus, from a complexity standpoint, the number of control locations of $\mathcal{P}'$ is exponential in the number of clocks and constraints, and the size of the stack alphabet is exponential in the number of constraints.

▸ **Lemma 19.** *Let the reachability relation of $\mathcal{P}'$ be expressed by the formula $\varphi_{\ell' \reflectbox{ℓ}'}$. The reachability relation of $\mathcal{P}$ is expressed by $\bigvee \left\{ \varphi_{\langle \ell, T, \varnothing, \varnothing \rangle \langle \reflectbox{ℓ}, \varnothing, \Phi^-, \Phi^+ \rangle} \;\middle|\; T \subseteq X, \Phi^-, \Phi^+ \subseteq \mathcal{C} \right\}$.*

Proof. Let $\mathcal{P}$ be a push-copy TPDA $(\Sigma, \Gamma, L, X, Z, \Delta)$. Let $\mathcal{C}^- / \mathcal{C}^+$ be the set of all lower/upper bound classical pop constraints of the form $y - z_x \geqslant k, y - z_x > k$, or, resp., $y - z_x \leqslant k, y - z_x < k$, and let $\mathcal{C} = \mathcal{C}^- \cup \mathcal{C}^+$. We construct a TPDA $\mathcal{P}' = (\Sigma, \Gamma', L', X', Z, \Delta')$ with the same set of stack clocks as $\mathcal{P}$, and with global clocks being those of $\mathcal{P}$, plus a copy of each global clock for each lower/upper bound constraint: $X' := X \cup \{x_\psi \mid \psi \in \mathcal{C}\}$. A control location of $\mathcal{P}'$ is of the form $(\ell, T, \Phi^-, \Phi^+) \in L'$, where

- ℓ is a control location of $\mathcal{P}$,
- $T \subseteq X$ is a set of clocks of $\mathcal{P}$ which cannot be reset till the next push (this is used to guess and check last resets before a push), and
- $\Phi^- \subseteq \mathcal{C}^-, \Phi^+ \subseteq \mathcal{C}^+$ are the currently *active* lower/upper bound constraints.

The new stack alphabet Γ' consists of tuples of the form $\langle \alpha, \Phi^-, \Phi^+ \rangle$ with $\alpha \in \Gamma$ a stack symbol of $\mathcal{P}$ and Φ^-, Φ^+ as above.

Let $\langle \ell, \mathsf{op}, \reflectbox{ℓ} \rangle$ be a transition in $\mathcal{P}$. If it is either an input $\mathsf{op} = a \in \Sigma_\varepsilon$, test $\mathsf{op} = \varphi$, or time elapse $\mathsf{op} = \mathsf{elapse}$ transition, then it generates corresponding transitions in $\mathcal{P}$ of the form $\langle (\ell, T, \Phi^-, \Phi^+), \mathsf{op}, (\reflectbox{ℓ}, T, \Phi^-, \Phi^+) \rangle$ for every choice of T, Φ^-, Φ^+. A reset transition $\mathsf{op} = \mathsf{reset}(Y)$ generates several reset transitions of the form

$$\langle (\ell, T, \Phi^-, \Phi^+), \mathsf{reset}(Y \cup Y'), (\reflectbox{ℓ}, T \cup U, \Phi^- \cup \Psi^-, \Phi^+ \cup \Psi^+) \rangle$$

whenever

1. $Y \cap T = \varnothing$ (no forbidden clock is reset),

2. $U \subseteq Y$ is a subset of reset clocks which are guessed to be reset for the last time till the next push,
3. $\Psi^- \subseteq \bigcup_{x \in U} \mathcal{C}_x^- \backslash \Phi^-$ is a new set of lower bound constraints involving newly reset clocks in U, similarly
4. $\Psi^+ \subseteq \bigcup_{x \in U} \mathcal{C}_x^+ \backslash \Phi^+$ likewise for the upper bound constraints, and finally
5. $Y' \subseteq \{x_\psi \mid \psi \in \mathcal{C}\}$ contains all clocks relating to *new* active lower bound constraints, and all clocks relating to (new or not) active upper bound constraints w.r.t. clocks Y reset in this transition:

$$\begin{aligned} Y' &= \left\{x_\varphi \mid \varphi \in \Psi^- \text{ or } \varphi \in \Phi_Y^+ \cup \Psi^+\right\}, \text{ where} \\ \Phi_Y^+ &= \left\{(y - z_x \lesssim k) \in \Phi^+ \mid x \in Y\right\}. \end{aligned}$$

A push transition $\mathsf{op} = \mathsf{push}(\alpha : \psi_{\text{copy}})$ (where ψ_{copy} is defined in (2)), generates a transition in $\mathcal{P}'$ of the form

$$\langle(\ell, T, \Phi^-, \Phi^+), \mathsf{push}(\left\langle\alpha, \Phi^-, \Phi^+\right\rangle : \psi_{\text{copy}}), (r, T', \Phi^-, \Phi^+)\rangle$$

only if $T = X$, i.e., all clocks were correctly guessed to be reset for the last time till this push, and for every set of clocks $T' \subseteq X$ which are guessed not to be reset till the *next* push. Moreover, we push on the stack the current set of guessed constraints Φ^-, Φ^+. Finally, a pop transition $\mathsf{op} = \mathsf{pop}(\alpha \models \psi_{\mathsf{pop}})$ of $\mathcal{P}$ with ψ_{pop} as in (19), generates in $\mathcal{P}'$ a test followed by a pop transition of the form (omitting the intermediate state)

$$\langle(\ell, T, \Phi^-, \Phi^+),\ \widetilde{\psi};\ \mathsf{pop}(\left\langle\alpha, \hat{\Phi}^-, \hat{\Phi}^+\right\rangle : \psi^{\text{nc}}),\ (r, T, \hat{\Phi}^-, \hat{\Phi}^+)\rangle$$

for every $T \subseteq X$, $\hat{\Phi}^- \subseteq \mathcal{C}^-$, $\hat{\Phi}^+ \subseteq \mathcal{C}^+$, whenever $\Phi^- \cup \Phi^+ = \{\psi_1^{\text{c}} \wedge \cdots \wedge \psi_m^{\text{c}}\}$, i.e., the guess of upper and lower bounds was indeed correct, and where $\widetilde{\psi}$ is defined as $\widetilde{\psi} \equiv \bigwedge \left\{y - x_{\psi_i^{\text{c}}} \sim k \mid \psi_i^{\text{c}} \in \Phi^- \cup \Phi^+, \psi_i^{\text{c}} \equiv y - z_x \sim k\right\}$. We have removed pop integer constraints ψ_i^{c}'s by introducing classical constraints in $\widetilde{\psi}$, and the latter can be converted into integer and fractional constraints according to Remark 2. Notice that the stack non-classical constraint ψ^{nc} is preserved from $\mathcal{P}$ to $\mathcal{P}'$. Thus, we obtain a pop-integer-free TPDA, as required.

The number of control locations of $\mathcal{P}'$ is $|L'| = |L| \cdot 2^{|X|} \cdot 2^{|\mathcal{C}| \cdot 2}$, the number of stack symbols of $\mathcal{P}'$ is $|\Gamma'| = |\Gamma| \cdot |\mathcal{C}|^2$, and the number of clocks of $\mathcal{P}'$ is $|X'| = |X| + |\mathcal{C}|$. Thus, $\mathcal{P}'$ has number of control locations and stack symbol exponential in the size of $\mathcal{P}$, and number of clocks linear in the size of $\mathcal{P}$.

The construction can be proved correct by the same argument for stack classical constraints as in [10], except that now non-classical stack constraints (not considered in [10]) are kept unchanged. ◂

A.5 Missing details for (C) fractional

Recall the structure of fractional values $\mathcal{A}_{\mathbb{I}}^c = (\mathbb{I}, \leqslant, 0)$. An *automorphism* of $\mathcal{A}_{\mathbb{I}}^c$ is a bijection α s.t. $\alpha(0) = 0$ and $a \leqslant b$ iff $\alpha(a) \leqslant \alpha(b)$; in other words, 0 is fixed, but otherwise distances can be stretched or compressed monotonically. The set $\mathbb{I}^X$ of (fractional parts of) clock valuations splits into finitely many *orbits*, where $u, v \in \mathbb{I}^n$ are in the same orbit if some automorphism of $\mathcal{A}_{\mathbb{I}}^c$ maps u to v. Note that an orbit o is determined by the order of elements, their equality type, and their equalities with 0; hence the number of orbits is exponential in $|X|$. For an orbit o, let its *characteristic formula* be the following quantifier-free $(\mathbb{I}, \leqslant, 0)$ formula

$$\varphi_o(\overline{x}) \equiv \bigwedge_{\tilde{a}_i = 0} x_i = 0 \wedge \bigwedge_{\tilde{a}_i \leqslant \tilde{a}_j} x_i \leqslant x_j, \tag{20}$$

where $(\tilde{a}_1, \ldots, \tilde{a}_n)$ is any fixed element of o (by the definition of orbit, φ_o does not depend on the choice of representative).

Let $\mathcal{P} = (\Sigma, \Gamma, L, X, Z, \Delta)$ be a push-copy and pop-integer-free TPDA. We build a fractional TPDA $\mathcal{P}' = (\Sigma', \Gamma', L', X, Z, \Delta')$ where Σ' equals Σ extended with an extra symbol $\checkmark_x \notin \Sigma$ for every clock x of $\mathcal{P}$, $\Gamma' = \Gamma \times \Lambda_M$ extends Γ by recording the M-unary equivalence class of clocks which are pushed on the stack, and $L' = L \times \Lambda_M \times 2^X \cup L_\bullet$, where $Y_1 \in 2^X$ is the set of clocks which are *not* allowed to be reset any more in the future, and $L_\bullet$ contains some extra control locations used in the simulation. Every transition $\langle \ell, \mathsf{op}, \ell' \rangle \in \Delta$ generates one or more transitions in Δ' according to op. If $\mathsf{op} = a \in \Sigma_\varepsilon$ is an input transition, then Δ' contains a corresponding input transition $\langle \langle \ell, \lambda, Y_1 \rangle, a, \langle \ell', \lambda, Y_1 \rangle \rangle$, for every choice of λ, Y_1. If $\mathsf{op} = \varphi$ is a test transition, then Δ' contains a corresponding test transition

$$\langle \langle \ell, \lambda, Y_1 \rangle, \varphi|_\lambda, \langle \ell', \lambda, Y_1 \rangle \rangle,$$

where $\varphi|_\lambda$ contains only fractional constraints. If $\mathsf{op} = \mathsf{reset}(Y)$ is a reset transition, then Δ' contains a reset transition

$$\langle \langle \ell, \lambda, Y_1 \rangle, \mathsf{reset}(Y), \langle \ell', \lambda[Y \mapsto 0], Y_1 \cup Y_2 \rangle \rangle$$

provided that $Y \subseteq X \backslash Y_1$ (no forbidden clocks are reset), and where $Y_2 \subseteq Y$ are declared to be reset now for the last time. If $\mathsf{op} = \mathsf{elapse}$ is a time elapse transition, then we have the following 4 groups of transitions:

1. First, we silently go to control location $\langle \ell, \lambda, Y_1, 1 \rangle$ to start the simulation:

   $\langle \langle \ell, \lambda, Y_1 \rangle, \varepsilon, \langle \ell, \lambda, Y_1, 1 \rangle \rangle$.

2. We test that the current orbit of fractional values is o, we let time elapse, and then we test that the new orbit is o'. We can reconstruct the set of clocks $Y_{o,o'}$ which have just overflown and for which we need to update their unary abstraction as $Y_{o,o'} = \{x \in X \mid o(x) > 0 \text{ and } o'(x) = 0\}$. This yields the following sequence of transitions, where we omit the intermediate states for conciseness:

   $\langle \langle \ell, \lambda, Y_1, 1 \rangle, (\varphi_o; \mathsf{elapse}; \varphi_{o'}), \langle \ell, \lambda, Y_1, Y_{o,o'}, 2 \rangle \rangle$.

3. For each clock that needs to be updated in $Y_{o,o'}$, we increment its unary abstraction one by one, and we optionally emit a tick if this clock was guessed not to be reset anymore in the future:

   $\langle \langle \ell, \lambda, Y_1, Y_2, 2 \rangle, \checkmark_x^?, \langle \ell, \lambda[x \mapsto x + 1], Y_1, Y_2 \backslash \{x\}, 2 \rangle \rangle$,

   where $\checkmark_x^?$ equals $\checkmark_x$ if $x \in Y_2 \cap Y_1$, and ε if $x \in Y_2 \backslash Y_1$.
4. When the unary class of all overflown clocks has been updated, we either return to the beginning of the simulation (in order to simulate longer elapses of time), or we quit:

   $\langle \langle \ell, \lambda, Y_1, \varnothing, 2 \rangle, \varepsilon, \langle \ell, \lambda, Y_1, 1 \rangle \rangle, \quad \langle \langle \ell, \lambda, Y_1, \varnothing, 2 \rangle, \varepsilon, \langle \ell', \lambda, Y_1 \rangle \rangle$.

If $\mathsf{op} = \mathsf{push}(\alpha : \psi_{\mathrm{copy}})$ is a push-copy transition, then Δ' contains a push transition copying only the fractional parts and the unary class of global clocks:

$$\langle \langle \ell, \lambda, Y_1 \rangle, \mathsf{push}(\langle \alpha, \lambda \rangle : \bigwedge_{x \in X} \{z_0\} = 0 \wedge \{z_x\} = \{x\}), \langle \ell', \lambda, Y_1 \rangle \rangle.$$

If $\mathsf{op} = \mathsf{pop}(\alpha : \psi)$ is a pop-integer-free transition, then Δ' contains a fractional pop transition of the form

$$\langle \langle \ell, \lambda_{\mathsf{pop}}, Y_1 \rangle, \mathsf{pop}(\langle \alpha, \lambda_{\mathsf{push}} \rangle : \psi|_{\lambda_{\mathsf{push}}, \lambda_{\mathsf{pop}}}), \langle \ell', \lambda_{\mathsf{pop}}, Y_1 \rangle \rangle.$$

We eliminated all occurrences of $\lfloor x \rfloor$ both from transition and push/pop stack constraints. Thus, transition and stack constraints of $\mathcal{P}'$ are only fractional.

Reconstruction of the reachability relation. We reconstruct the reachability relation of $\mathcal{P}$ from that of $\mathcal{P}'$ as follows. The reachability relation $\leadsto_{\ell \wr}$ of $\mathcal{P}$ is expressed as the $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$ formula

$$
\begin{aligned}
\varphi_{\ell \wr}(\lfloor \overline{x} \rfloor, \{\overline{x}\}, \overline{f}, \lfloor \overline{x}' \rfloor, \{\overline{x}'\}) \;&\equiv\; \bigvee_{\lambda, Y, \mu} \exists \overline{g} \cdot \varphi_\lambda(\lfloor \overline{x} \rfloor) \wedge \varphi_{\mathrm{step}} \wedge \varphi_{\mathrm{end}}, \text{ where} \\
\varphi_{\mathrm{step}} \;&\equiv\; \psi_{\langle \ell, \lambda, Y \rangle \langle \wr, \mu, X \rangle}(\{\overline{x}\}, (\overline{f}, \overline{g}), \{\overline{x}'\}) \\
\varphi_{\mathrm{end}} \;&\equiv\; \bigwedge_{x \in Y} \lfloor x' \rfloor = \lfloor x \rfloor + g_x \wedge \bigwedge_{x \notin Y} \lfloor x' \rfloor = g_x.
\end{aligned}
$$

- The formula φ_λ ensures that the initial integer value of clocks has the same unary class as prescribed by λ.
- The formula φ_{step} invokes the fractional reachability relation of $\mathcal{P}'$ where g_x counts the number of marks $\checkmark_x$ since clock x was last reset.
- The formula φ_{end} uniquely determines the final integer values $\lfloor x' \rfloor$ of all clocks of $\mathcal{P}$: For those clocks $x \notin Y$ which are ever reset during the run, the final value of its integer part $\lfloor x' \rfloor$ equals the integer time g_x that elapsed since the last reset; for those clocks $x \in Y$ which are not reset during the run, $\lfloor x' \rfloor$ equals their initial value plus the time elapsed since the beginning.

We can eliminate the existential quantification on $\overline{g}$ from the formula above by noticing that φ_{end} uniquely determines $\overline{g}$ as a function of $\lfloor \overline{x} \rfloor, \lfloor \overline{x}' \rfloor$ and Y, thus obtaining the equivalent $\mathcal{L}_{\mathbb{Z},\mathbb{Q}}$ formula in the following lemma.

▸ **Lemma 10.** *Let the fractional reachability relation of $\mathcal{Q}$ be expressed by $\psi_{\ell' \wr'}(\{\overline{x}\}, (\overline{f}, \overline{g}), \{\overline{x}'\})$ using only the fractional values of clocks $\{\overline{x}\}, \{\overline{x}'\}$, the Parikh image $\overline{f}$ of the original input letters from Σ, and the Parikh image $\overline{g}$ of the new input letters $\checkmark_x$. The reachability relation of $\mathcal{P}$ can be expressed by*

$$
\varphi_{\ell \wr}(\lfloor \overline{x} \rfloor, \{\overline{x}\}, \overline{f}, \lfloor \overline{x}' \rfloor, \{\overline{x}'\}) \equiv \bigvee_{\lambda, Y, \mu} \varphi_\lambda(\lfloor \overline{x} \rfloor) \wedge \psi_{\langle \ell, \lambda, Y \rangle \langle \wr, \mu, X \rangle}(\{\overline{x}\}, (\overline{f}, \overline{g}^Y), \{\overline{x}'\}).
$$

In the statement above, $\overline{g}^Y$ is defined as follows:

$$
g_x^Y \equiv \begin{cases} \lfloor x' \rfloor - \lfloor x \rfloor & \text{if } x \in Y \\ \lfloor x' \rfloor & \text{otherwise.} \end{cases}
$$

A.6 Missing proofs from Sec. 5

▸ **Lemma 13.** *Given a $\mathcal{C}$-PDA $\mathcal{Q}$ one can construct a context-free grammar G of exponential size with nonterminals of the form $X_{\ell \wr o}$, for control locations $\ell, \wr$ and an orbit $o \in \mathrm{Orb}(\mathbb{I}^{X \times X'})$, recognising the language*

$$
L(X_{\ell \wr o}) = \left\{ \pi_\Sigma(w) \in \Sigma^* \;\middle|\; \exists (\mu, \nu) \in o \cdot \mu \overset{w}{\leadsto}_{\ell \wr} \nu \right\}, \tag{11}
$$

where $\pi_\Sigma(w)$ is w without the ε's. Consequently, $\mathcal{C}$-PDAs recognise context-free languages.

Proof. This is a special case of the following general fact: An equivariant orbit-finite PDA over homogeneous atoms can be transformed into an equivariant orbit-finite context-free grammar (see [4, 8]). For concreteness, we provide the productions of the grammar. For $o \in \mathrm{Orb}(\mathbb{I}^{X \times X'})$ we write o_1 (resp. o_2) for the projections of o on the first (resp. last) k

coordinates. For every input transition $\langle \ell, a, \rotatebox{90}{ℓ} \rangle$ and o s.t. $o_1 = o_2$ we have in the grammar a production

$$\text{(input)} \qquad X_{\ell \rotatebox{90}{ℓ} o} \leftarrow a.$$

For every global transition rule $\langle \ell, \varphi, \rotatebox{90}{ℓ} \rangle$ and o s.t. $o \models \varphi$ we have a production

$$\text{(global)} \qquad X_{\ell \rotatebox{90}{ℓ} o} \leftarrow \varepsilon.$$

For an orbit $o \in \mathrm{Orb}(\mathbb{I}^{X_1 \times X_2 \times X_3})$ and $i, j \in \{1, 2, 3\}$, denote by $o_{ij} \in \mathrm{Orb}(\mathbb{I}^{X_i \times X_j})$ the projection of o to (k-ary) components i, j. For every orbit $o \in \mathrm{Orb}(\mathbb{I}^{X \times X' \times X''})$ we have a production

$$\text{(transitivity)} \qquad X_{\ell \rotatebox{90}{ℓ} o_{13}} \leftarrow X_{\ell \ell' o_{12}} \cdot X_{\ell' \rotatebox{90}{ℓ} o_{23}}.$$

Finally, for every pair of transitions $\langle \ell, \mathsf{push}(\gamma \models \varphi), \ell' \rangle, \langle \rotatebox{90}{ℓ'}, \mathsf{pop}(\gamma \models \psi), \rotatebox{90}{ℓ} \rangle \in \Delta$ and orbit $o \in \mathrm{Orb}(\mathbb{I}^{X \times X' \times Z})$ s.t. $o_{13} \models \varphi$ and $o_{23} \models \psi$, we have a production

$$\text{(push-pop)} \qquad X_{\ell \rotatebox{90}{ℓ} o_{12}} \leftarrow X_{\ell' \rotatebox{90}{ℓ'} o_{12}}.$$

◀

A.6.1 Correctness of the construction

We argue that $\mathcal{Q}$ and $\mathcal{P}$ faithfully simulate each other by providing a variant of strong bisimulation between their configurations. A configuration $\langle \ell, \mu, u \rangle$ of $\mathcal{P}$ *is consistent with* a configuration $\langle \rotatebox{90}{$\ell$}, \nu, v \rangle$ of $\mathcal{Q}$, if

- they have the same control locations $\ell = \rotatebox{90}{$\ell$}$,
- every global clock x and the corresponding register $\hat{x}$ satisfy $\{\mu(x)\} = \nu(\hat{x}_0) \ominus \nu(\hat{x})$,
- $u = (\gamma_1, \mu_1) \cdots (\gamma_n, \mu_n)$, $v = (\gamma_1, \nu_1) \cdots (\gamma_n, \nu_n)$ and, for every $1 \leqslant i \leqslant n$, stack clock z and corresponding register $\hat{z}$, we have $\{\mu_i(z)\} = \nu(\hat{x}_0) \ominus \nu_i(\hat{z})$.

The consistency is not one-to-one, for two reasons: on the side of $\mathcal{P}$ the integer parts of clocks are irrelevant and hence can be arbitrary; and on the side of $\mathcal{Q}$ the configuration is unique only up to cyclic shift.

A configuration $\langle \rotatebox{90}{$\ell$}, \nu, v \rangle$ (of $\mathcal{P}$ or $\mathcal{Q}$) is an *a-successor* of $\langle \ell, \mu, u \rangle$ if $\langle \ell, \mu, u \rangle \xrightarrow{a} \langle \rotatebox{90}{ℓ}, \nu, v \rangle$ (in $\mathcal{P}$ or $\mathcal{Q}$, resp.); in $\mathcal{P}$, additionally, if $a \in \mathbb{Q}_{\geqslant 0}$, then we call $\langle \rotatebox{90}{$\ell$}, \nu, v \rangle$ an ε-successor of $\langle \ell, \mu, u \rangle$. By inspection of the construction of $\mathcal{Q}$ we deduce:

▸ Claim 1. Every configuration of $\mathcal{P}$ (resp. $\mathcal{Q}$) is consistent with some configuration of $\mathcal{Q}$ (resp. $\mathcal{P}$). Moreover, for every pair of consistent configurations of $\mathcal{P}$ and $\mathcal{Q}$, respectively, and $a \in \Sigma_\varepsilon$, every a-successor of one of the configurations is consistent with exactly one a-successor of the other one.

Thus, once a pair of consistent configurations is fixed, the a-successors in $\mathcal{P}$ and $\mathcal{Q}$ are in a one-to-one correspondence. For the correctness of (14) in Sec. 5 observe that a configuration $\langle \ell, \mu, \varepsilon \rangle$ of $\mathcal{P}$ and a configuration $\langle \ell, \nu, \varepsilon \rangle$ of $\mathcal{Q}$ are consistent if, and only if, $(\mu, \nu) \models \varphi_\ominus$.