# Binary reachability of timed-register pushdown automata, and branching vector addition systems

LORENZO CLEMENTE*, Wydział Matematyki, Informatyki i Mechaniki, University of Warsaw, Poland
SŁAWOMIR LASOTA†, Wydział Matematyki, Informatyki i Mechaniki, University of Warsaw, Poland
RANKO LAZIĆ‡, Department of Computer Science, University of Warwick, United Kingdom
FILIP MAZOWIECKI§, LABRI, Université de Bordeaux, France

Timed-register pushdown automata constitute a very expressive class of automata, whose transitions may involve state, input, and top-of-stack timed-registers with unbounded differences. They strictly subsume pushdown timed automata of Bouajjani et al., dense-timed pushdown automata of Abdulla et al., and orbit-finite timed register pushdown automata of Clemente and Lasota. We give an effective logical characterisation of the reachability relation of timed-register pushdown automata. As a corollary, we obtain a doubly exponential time procedure for the non-emptiness problem. We show that the complexity reduces to singly exponential under the assumption of monotonic time. The proofs involve a novel model of one-dimensional integer branching vector addition systems with states. As a result interesting on its own, we show that reachability sets of the latter model are semilinear and computable in exponential time.

CCS Concepts: • **Theory of computation → Timed and hybrid models**; **Formal languages and automata theory**; *Automata over infinite objects*; *Logic*; *Logic and verification*; Grammars and context-free languages;

Additional Key Words and Phrases: Timed automata, pushdown automata, timed-register pushdown automata, branching vector addition systems

**ACM Reference Format:**
Lorenzo Clemente, Sławomir Lasota, Ranko Lazić, and Filip Mazowiecki. 2019. Binary reachability of timed-register pushdown automata, and branching vector addition systems. *ACM Trans. Comput. Logic* 1, 1 (April 2019), 31 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*Partially supported by Polish NCN grants 2016/21/B/ST6/01505 and 2017/26/D/ST6/00201.
†Partially supported by the European Research Council (ERC) project Lipa under the EU's Horizon 2020 research and innovation programme (grant agreement No. 683080).
‡Partially supported by the Engineering and Physical Sciences Research Council grant EP/M011801/1, and by the Leverhulme Trust Research Fellowship RF-2017-579.
§Partially supported by the Engineering and Physical Sciences Research Council grant EP/M011801/1 and the French National Research Agency (ANR) project "Investments for the future" Programme IdEx Bordeaux (ANR-10-IDEX-03-02).

Authors' addresses: Lorenzo Clemente, Wydział Matematyki, Informatyki i Mechaniki, University of Warsaw, Warsaw, Poland, clementelorenzo@gmail.com; Sławomir Lasota, Wydział Matematyki, Informatyki i Mechaniki, University of Warsaw, Warsaw, Poland, sl@mimuw.edu.pl; Ranko Lazić, Department of Computer Science, University of Warwick, Warwick, United Kingdom, R.S.Lazic@warwick.ac.uk; Filip Mazowiecki, LABRI, Université de Bordeaux, Bordeaux, France, f.mazowiecki@mimuw.edu.pl.


1529-3785/2019/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

*Background.* Timed automata [4] are one of the most studied and used models of reactive timed systems. Motivated by verification of programs with both procedural and timed features, several extensions of timed automata by a pushdown stack have been proposed, including pushdown timed automata (PDTA) [8], recursive timed automata (RTA) [5, 24], dense-timed pushdown automata (dtPDA) [1], and timed-register pushdown automata (trPDA) [10].

While PDTA simply add an untimed stack to a timed automaton, dtPDA are allegedly more powerful since they allow to store clocks on the stack evolving at the same rate as clocks in the finite control. Surprisingly, Clemente and Lasota showed that, as a consequence of the interplay of the stack discipline and the monotone elapsing of time, dtPDA are in fact not more expressive than PDTA, and the two models are strictly subsumed by *orbit-finite* trPDA [10], a strict subclass of trPDA. Moreover, subsumption still holds if orbit-finite trPDA are restricted to timeless stack, and in this case there is nothing to pay in terms of the complexity of non-emptiness, which is the central decision problem for model checking: it is ExpTime-complete for both PDTA, dtPDA, and orbit-finite trPDA with timeless stack; for orbit-finite trPDA, the best known upper bound rises to NExpTime (ibid.). The main question posed in the latter work is whether the heavy restriction of orbit finiteness, which demands that differences between state and top-of-stack registers be bounded, can be lifted while keeping non-emptiness decidable.

The proofs of the NExpTime and ExpTime upper bounds for orbit-finite and timeless-stack trPDA (respectively) [10] involved translations to systems of equations in which variables range over sets of integers, and available operations include addition, union, and intersection with the singleton set $\{0\}$. Similar systems have been studied in a variety of contexts, and extensions quickly lead to undecidability: e.g., already over the naturals, when arbitrary intersections are permitted, decidability is lost since this model subsumes unary conjunctive grammars [17].

*Contributions.* Our headline result answers positively the question raised by Clemente and Lasota [10]: we prove that non-emptiness remains decidable when the assumption of orbit-finiteness of trPDA is dropped. The resulting class of automata strictly subsumes all pushdown extensions of timed automata mentioned above (with the exception of RTA[1] [5, 6, 24]), and is the first one to allow timed stacks without bounding the differences of state and top-of-stack clocks[2]. For example, it is able to recognise the language of all *timed* palindromes over $\{a, b\}$ containing the same number of $a$'s and $b$'s.

The first half of the decidability proof is a multi-stage translation, in exponential time, from trPDA to one-dimensional branching vector addition systems with states over the integers ($\mathbb{Z}$-BVASS), where the latter's reachability sets encode the former's reachability relations. Branching vector addition systems with states have been studied extensively in recent years with motivations coming from computational linguistics, linear logic, and verification of recursively parallel programs amongst others; cf. Lazić and Schmitz [19] and references therein. The one-dimensional variant we work with allows negative counter values and encompasses two powerful features: subtraction and testing memberships in given semi-linear sets.

The second half of the proof proceeds by transforming $\mathbb{Z}$-BVASS to a normal form (this takes pseudopolynomial time if constants are encoded in unary, and polynomial time in unary), and then showing that, in exponential time, both their non-emptiness is decidable and their semi-linear reachability sets are computable. Several combinatorial arguments are involved here, as well as a

[1]The model of RTA differs significantly from the other models since the stack contains clock *values* which are constant with respect to the elapsing of time.

[2]Note that Clemente and Lasota denoted by trPDA an undecidable class in which many stack symbols can be popped and pushed in one step, like in prefix-rewriting. For simplicity, we use the same name for the new largest decidable subclass.

reduction to the reachability problem for one-dimensional BVASS with unary-encoded constants, which is PTime-complete [16].

Combining the two results, we obtain not only that the non-emptiness problem for trPDA is in 2-ExpTime, but also that a quantifier-free DNF formula that captures the trPDA's reachability relation is computable in doubly-exponential time. We additionally establish that one exponential can be saved just by assuming that transitions do not decrease the integer parts of timestamp registers: non-emptiness for these *monotonic* trPDA is decidable in ExpTime, and they suffice to model monotonic time devices such as PDTA and dtPDA.

There is an interesting connection between some aspects of this work and the analysis of dtPDA based on tree automata of [2]. It is shown there that runs of dtPDA can be represented as graphs of bounded split-width, and one can construct a finite tree automaton recognizing precisely those decompositions corresponding to timed runs of the dtPDA. Upon a closer inspection of our approach for trPDA (cf. the reduction to $\mathbb{Z}$-BVASS outlined below), it can be argued that we also perform a reduction to a kind of tree automaton, albeit not a finite one, but one *with an integer counter*. This extra counter is needed to keep track of possibly unbounded differences between register values for matching push/pop pairs. The fact that a finite tree automaton suffices when analyzing dtPDA follows from the previous semantic collapse result of dtPDA to the variant with timeless stack [10]. For the latter model, since the stack is timeless, there are no long push/pop timing dependencies and a finite tree automaton suffices.

*Full version.* This article is a new and full version of the preliminary conference paper [12], embodying a complete revision and a major extension. The main novelties in comparison with the former work are:

(1) We show an effective logical characterisation of the binary reachability relations of trPDA, instead of merely deciding non-emptiness.
(2) The central model of trPDA is more general in two ways: the logic of constraints is extended by equality modulo predicates, and orbit finiteness (equivalently, bounded span) is assumed only on states. Thus, input symbols, stack elements, and the transition relation are not assumed to be orbit finite. It was previously unclear whether the orbit finite restriction on stack elements could be dropped.
(3) The translation from trPDA to branching vector addition systems with states is entirely new (which is necessary in order to tackle the more general model) and more direct, thanks to establishing that the logic admits effective quantifier elimination.
(4) The integer one-dimensional branching vector addition systems with states are proved to have semi-linear reachability sets computable in exponential time, instead of just deciding non-emptiness in exponential time. This is a new result interesting on its own.
(5) We additionally show that for monotonic trPDA, we obtain better complexity bounds thanks to a direct translation to context-free grammars, instead of the more powerful branching vector additions systems.

Note that these results do not allow us to give a characterisation for the reachability relation of timed automata (neither for the reachability set of clock valuations), since the known translations from timed automata to orbit-finite timed-register automata preserve only non-emptiness, but not the reachability relation itself (essentially, because the former model uses *clocks* while the latter one uses *registers*). The problem of characterising the binary reachability relation in an expressive class of timed automata with a timed stack strictly generalising PDTA and dtPDA has been recently solved in [11].

## 2 PRELIMINARIES

We denote by $\mathbb{Q}$ the set of rational, by $\mathbb{Z}$ the set of integer, and by $\mathbb{N}$ the set of natural numbers. For a modulus $m \in \mathbb{N}$, let $\equiv_m$ be the congruence modulo $m$ in $\mathbb{Z}$. For two subsets $A, B$ of $\mathbb{Q}$, we denote by $A + B$ the set $\{a + b \mid a \in A, b \in B\}$, by $-A$ the set $\{-a \mid a \in A\}$, and by $A - B$ the set $A + (-B)$; for a constant $\lambda \in \mathbb{Q}$, by $\lambda \cdot A$ we denote $\{\lambda \cdot a \mid a \in A\}$. Moreover, with $A^*$ we denote the infinite union $A^* = \bigcup_{n \geq 0} A^n$, where $A^0 = \{0\}$ and $A^{n+1} = A^n + A$; for simplicity, we write $a^*$ instead of $\{a\}$.

The *span* of a vector $\vec{a} = (a_1, \ldots, a_k) \in \mathbb{Z}^k$ is $\textsc{span}(\vec{a}) := \max \left\{|a_i - a_j| \;\middle|\; 1 \leq i \leq k, 1 \leq j \leq k\right\}$; intuitively, it measures the maximum gap between any two components. A subset $A \subseteq \mathbb{Z}^k$ has *bounded span* if the set $\{\textsc{span}(\vec{a}) \mid \vec{a} \in A\}$ is finite. For a set of vectors $A \subseteq \mathbb{Z}^k$ and bound $K \in \mathbb{N}$, let the restriction of $A$ to vectors of span bounded by $K$ be $A_{\textsc{span} \leq K} = \{\vec{a} \in A \mid \textsc{span}(\vec{a}) \leq K\}$.

Let $\Sigma$ be a finite alphabet, and denote by $\Sigma^*$ the set of finite words over $\Sigma$. The *Parikh image* of a word $w \in \Sigma^*$ is the mapping $\pi_w : \mathbb{N}^\Sigma$ which, for every letter $a \in \Sigma$, returns its number of occurrences $\pi_w(a)$ in $w$; the Parikh image of a language $L \subseteq \Sigma^*$ extends naturally as $\pi(L) = \{\pi_w \mid w \in L\}$. If we fix a total ordering on the letters $\Sigma = \{a_1, \ldots, a_d\}$, Parikh images can equivalently be seen as subsets of $\mathbb{N}^d$.

In complexity estimations, we define the *magnitude* of a constant $k \in \mathbb{Z}$ as its absolute value $|k|$.

### 2.1 Hybrid linear sets

A *hybrid linear set* is a set of the form $A + B^*$, where $A \subseteq \mathbb{Z}^d$ is a finite set of *bases* and $B \subseteq \mathbb{Z}^d$ is a finite set of *periods.* A *linear set* is a hybrid set of the form $\{a\} + B^*$, also written as $a + B^*$ for simplicity. A *semilinear set* is a finite union of linear (equivalently, hybrid linear) sets. Whenever we *compute* or *construct* a semilinear set, we mean that we build a representation with bases and periods as above.

Let $M \in \mathbb{N}$ be a bound. A subset of $\mathbb{N}^d$ is a $M$-*bounded* hybrid linear set if it can be put in the form $A + B^*$ with $A, B \subseteq \{0, \ldots, M\}$; $M$-boundedness is defined in the obvious way for linear and semilinear sets. The following general property of hybrid linear sets in dimension one $d = 1$ justifies us to assume that semilinear sets in dimension one are of the form $S = L_1 \cup \cdots \cup L_n$ with $L_i$ just an arithmetic progression $L_i = a_i + b_i^*$, with no increase in complexity.

Lemma 2.1. *Any $M$-bounded hybrid linear set $S \subseteq \mathbb{N}$ can be put in the form*

$$F \cup (A + b^*), \qquad \text{with } F, A \subseteq \left\{0, \ldots, M + M^2\right\} \text{ and } b \leq M. \tag{1}$$

Proof. We start by proving the lemma in the special case of linear sets of the form $P^*$.

Claim 1. *A $M$-bounded linear set of the form $P^*$, with $P = \{p_1, \ldots, p_n\} \subseteq \{1, \ldots M\}$, can be put in the form $F \cup (a + b^*)$ with $F \subseteq \left\{0, \ldots, M^2\right\}$, $a \leq M^2$, and $0 < b \leq M$.*

Proof of the claim. Let $p_{\max} = \max(P)$, $p_\bullet = \gcd(P)$, and take base $a = p_{\max}^2 / p_\bullet$, period $b = p_\bullet$, and $F = \{k \in P^* \mid k < a\}$. We show that $P^* = F \cup (a + b^*)$. Assume $k \in P^*$. If $k < a$, then $k \in F$. If $k \geq a$, then $k - a \geq 0$ is divisible by $b$, and thus $k \in (a + b^*)$. For the other inclusion, consider the set $Q = 1/b \cdot P$. Since any number larger than $\max(Q)^2 = a/b$ is expressible as a linear combination of numbers in $Q$ ([3, 21]), $a/b + 1^* \subseteq Q^*$, and thus $a + b^* \subseteq P^*$. □

Let $S = Q + P^*$ be an $M$-bounded hybrid linear set. By the claim above, $P^* = F \cup (a + b^*)$, with $F \subseteq \left\{0, \ldots, M^2\right\}$, $a \leq M^2$, and $b \leq M$. Thus, $S = F' \cup (A + b^*)$ with $F' = Q + F \subseteq \left\{0, \ldots, M + M^2\right\}$ $A = Q + a \subseteq \left\{0, \ldots, M + M^2\right\}$, and $b \leq M$, as required. □

### 2.2 Presburger arithmetic

*Presburger arithmetic* is the first-order theory of the structure $(\mathbb{Z}, +, 0, 1, \leq, \equiv_m)$[3]. It is well-known that Presburger arithmetic admits effective elimination of quantifiers [23]. There is a close connection between semilinear sets, Presburger arithmetic, and Parikh images of context-free languages. Subsets of $\mathbb{N}^d$ definable in Presburger arithmetic coincide with the semilinear sets [15], which in turn coincide with the Parikh images of context-free languages [22]. By the following result, the latter are representable succinctly by a formula of existential Presburger arithmetic.

Lemma 2.2 (Theorem 4 in [26]). *The Parikh image of the language of a context-free grammar is described by an existential Presburger formula computable in linear time.*

For a linear set of the form $L = a + b^* \subseteq \mathbb{Z}$, let its *characteristic formula* $\psi_L$ s.t. $L = [\![\psi_L]\!]$ be $\psi_L(x) \equiv (x \equiv_b a)$, and for a semilinear set of the form $S = \bigcup_{i=1}^n L_i$ where $L_i = a_i + b_i^*$, let $\psi_S \equiv \bigvee_{i=1}^n \psi_{L_i}$.

## 3 HYBRID LOGIC AND QUANTIFIER ELIMINATION

We view dense time as a sequence of timestamps in $\mathbb{Q}$. It is technically convenient to reason separately about the integral and fractional part of timestamps. The integral part of timestamps is modelled by the *quantitative discrete time* structure[4] $(\mathbb{Z}, +1, \leq, \equiv_m)$, where $+1$ denotes the unary function that adds one to its argument, and $\equiv_m$ is the family of modulo congruences[5], where we assume that the modulus $m$ is encoded in binary. The total order between fractional values is captured by the *qualitative dense time* structure $(\mathbb{Q}, \leq)$. Combining discrete and dense time yields the following hybrid two-sorted structure (where $\leq^{\mathbb{H}} = \leq^{\mathbb{Z}} \uplus \leq^{\mathbb{Q}}$)

$$\mathbb{H} = (\mathbb{Z}, +1, \leq^{\mathbb{Z}}, \equiv_m) \uplus (\mathbb{Q}, \leq^{\mathbb{Q}}) = (\mathbb{Z} \uplus \mathbb{Q}, +1, \leq^{\mathbb{H}}, \equiv_m).$$

The domain of $\mathbb{H}$ is the disjoint union of $\mathbb{Z}$ and $\mathbb{Q}$ and its signature is the disjoint union of the respective signatures. When no confusion arises, we write $\leq$ instead of $\leq^{\mathbb{H}}$. We distinguish between *discrete variables* $x^{\mathbb{Z}}$ interpreted in $\mathbb{Z}$, and *dense variables* $x^{\mathbb{Q}}$ interpreted in $\mathbb{Q}$. *Discrete* $t^{\mathbb{Z}}$ and *dense terms* $t^{\mathbb{Q}}$ are built according to the following rules:

$$t^{\mathbb{Z}} ::= x^{\mathbb{Z}} \mid t^{\mathbb{Z}} + 1, \qquad\qquad t^{\mathbb{Q}} ::= x^{\mathbb{Q}}.$$

A *discrete atomic formula* is either of the form $t^{\mathbb{Z}} \leq u^{\mathbb{Z}}$ or $t^{\mathbb{Z}} \equiv_m u^{\mathbb{Z}}$ with $t^{\mathbb{Z}}, u^{\mathbb{Z}}$ discrete terms, and $m \in \mathbb{N}$. A *dense atomic formula* is of the form $x^{\mathbb{Q}} \leq y^{\mathbb{Q}}$ with $x^{\mathbb{Q}}, y^{\mathbb{Q}}$ two dense variables. As syntactic sugar, we also allow $\top$ as an atomic formula which is always satisfied. A formula of *hybrid logic* of dimension $(k, l)$ is a first-order formula $\varphi(\vec{x}^{\mathbb{Z}}, \vec{x}^{\mathbb{Q}})$, with $\vec{x}^{\mathbb{Z}} = (x_1^{\mathbb{Z}}, \ldots, x_k^{\mathbb{Z}})$ and $\vec{x}^{\mathbb{Q}} = (x_1^{\mathbb{Q}}, \ldots, x_l^{\mathbb{Q}})$, built from discrete and dense atomic formulas using variables $\vec{x}^{\mathbb{Z}}, \vec{x}^{\mathbb{Q}}$. Such a formula defines the set $[\![\varphi]\!] \subseteq \mathbb{Z}^k \times \mathbb{Q}^l$ of its satisfying valuations, and two formulas are *equivalent* if they define the same set. A subset of $\mathbb{Z}^k \times \mathbb{Q}^l$ is *definable* if it is defined by a formula of hybrid logic. The *satisfiability problem* for a given formula $\varphi$ amounts to decide whether $[\![\varphi]\!] \neq \emptyset$. We distinguish discrete (resp. dense) formulas which use only discrete (resp. dense) variables. As syntactic sugar, we allow integer constants in discrete formulas, which we assume to be encoded in binary. A *constraint* is a quantifier-free formula.

[3]Sometimes Presburger arithmetic is defined as the first-order theory of the more restricted structure $(\mathbb{N}, +, 0, 1)$, but since the predicates $\leq$ and $\equiv_m$ are first-order definable therein, the two logics are equi-expressive. Moreover, having $\equiv_m$ in the signature allows for quantifier elimination.

[4]For notational simplicity, we identify relational symbols such as "$\leq$" and their interpretation $\leq \subseteq \mathbb{Z} \times \mathbb{Z}$

[5]While the signature is infinite, each formula uses at most finitely many symbols from the signature.

### 3.1 Hybrid vs. quantitative dense time

*Quantitative dense time* is the structure $(\mathbb{Q}, +1^{\mathbb{Q}}, \leq^{\mathbb{Q}})$. This structure is rich enough to model dense time for timed automata [7] and timed pushdown automata [10]. We show that $(\mathbb{Q}, +1^{\mathbb{Q}}, \leq^{\mathbb{Q}})$ interprets in $\mathbb{H}$, which implies that the latter structure is at least as rich as the former, and in fact richer thanks to the modulo predicates $\equiv_m$. The domain of interpretation is the product $\mathbb{Z} \times \mathbb{Q}$. A rational number $x \in \mathbb{Q}$ is interpreted as the pair $(\lfloor x \rfloor, x - \lfloor x \rfloor) \in \mathbb{Z} \times \mathbb{Q}$, where $\lfloor x \rfloor$ is the integer part of $x$. The binary predicate $\leq^{\mathbb{Q}}$ and the unary function $+1^{\mathbb{Q}}$ are defined as follows:

$$(z, q) \leq^{\mathbb{Q}} (z', q') \ \equiv \ z <^{\mathbb{H}} z' \vee (z = z' \wedge q \leq^{\mathbb{H}} q'), \quad \text{and} \quad (z, q) + 1^{\mathbb{Q}} \ = \ (z + 1^{\mathbb{H}}, q).$$

### 3.2 Quantifier elimination

We say that a structure *admits effective quantifier elimination* if there is an algorithm that transforms every formula into an equivalent quantifier-free formula. The following is the main result of this section.

Theorem 3.1. *The structure $\mathbb{H}$ admits effective quantifier elimination.*

This result is a very useful tool that shows that, complexity considerations aside, it suffices to consider constraints instead of first-order logic formulas. Namely, this will be used in the definition of Timed register pushdown automata in Section 4, which will simplify the constructions afterwards.

Theorem 3.1 is proved by showing that both its two component structures $(\mathbb{Z}, \leq, \equiv_m, +1)$ and $(\mathbb{Q}, \leq)$ separately admit effective quantifier elimination (Lemmas 3.3 and 3.5 below). The following observation concludes the proof.

Lemma 3.2. *If two structures $\mathbb{A}, \mathbb{B}$ admit (effective) quantifier elimination, then the two-sorted structure $\mathbb{A} \uplus \mathbb{B}$ also admits (effective) quantifier elimination.*

Proof. A formula $\varphi$ of $\mathbb{A} \uplus \mathbb{B}$ can be written as $\varphi^{\mathbb{A}} \wedge \varphi^{\mathbb{B}}$, where $\varphi^{\mathbb{A}}$ is a formula of $\mathbb{A}$ and $\varphi^{\mathbb{B}}$ of $\mathbb{B}$. Thus, $\exists x^{\mathbb{A}} \cdot \varphi$ is equivalent to $(\exists x^{\mathbb{A}} \cdot \varphi^{\mathbb{A}}) \wedge \varphi^{\mathbb{B}}$. Since $\mathbb{A}$ admits quantifier elimination, there exists a quantifier-free formula $\psi^{\mathbb{A}}$ equivalent to $\exists x^{\mathbb{A}} \cdot \varphi^{\mathbb{A}}$, and thus $\psi^{\mathbb{A}} \wedge \varphi^{\mathbb{B}}$ is equivalent to $\varphi$. □

#### 3.2.1 *Quantifier elimination for discrete time.*

A discrete time constraint is effectively equivalent to a formula in disjunctive normal form (DNF), where atomic constraints are of the form $\alpha \leq x^{\mathbb{Z}} - y^{\mathbb{Z}} \leq \beta$ or $x^{\mathbb{Z}} - y^{\mathbb{Z}} \equiv_m k$, with $\alpha \in \mathbb{Z} \cup \{-\infty\}, \beta \in \mathbb{Z} \cup \{\infty\}$. Whenever we have a formula in DNF, we assume that its conjuncts are satisfiable. Consequently, a conjunctive discrete time constraint can be written as

$$\bigwedge_{i,j} \alpha_{ji} \leq x_j - x_i \leq \beta_{ji} \wedge x_j - x_i \equiv_m k_{ji},$$

where we assume w.l.o.g. that all modular constraints $\equiv_m$'s are over the same modulo $m$ (one can take as $m$ the least common multiplier of all moduli). Let $M \in \mathbb{N}$ be a bound. We say that a discrete time formula is *$M$-bounded* if the magnitude of all finite constants thereof is at most $M$. A conjunctive constraint of dimension $k$ needs to choose, for every pair of variables ($k^2$), an upper and a lower bound for their difference ($\leq (2M+1)^2$), and an equivalence class modulo $m$ ($\leq M$) This yields a crude estimation of at most $k^2 \cdot (2M+1)^2 \cdot M = O(k^2 M^3)$ inequivalent $M$-bounded conjunctive constraints of dimension $k$.

We show that quantitative discrete time admits effective quantifier elimination.

Lemma 3.3. *An $M$-bounded existential conjunctive formula of discrete time logic of dimension $k$ can be transformed in time $O(3^k M)$ into an equivalent $3^k M$-bounded constraint.*

Corollary 3.4. *The discrete time structure $(\mathbb{Z}, \leq, \equiv_m, +1)$ admits effective quantifier elimination.*

**Remark 1.** *Note that discrete time logic is a sublogic of Presburger arithmetic* $(\mathbb{Z}, +, 0, 1, \leq, \equiv_m)$*, which allows binary addition "+" (instead of just unary successor "+1") and constants* 0 *and* 1 *(instead of no constants). The lemma above does not follow from quantifier elimination of Presburger arithmetic, since it proves the stronger fact that for every formula of discrete time logic there exists an equivalent quantifier-free formula of discrete time logic itself.*

Proof of Lemma 3.3. Let $\varphi$ be a conjunctive formula of the form $\exists x \cdot \psi$, where (here and below, unless specified otherwise, indices $i, j$ range over $\{1, \ldots, k\}$)

$$\psi \equiv \bigwedge_i \alpha_i \leq x_i - x \leq \beta_i \wedge x_i - x \equiv_m k_i.$$

By solving it w.r.t. variable $x$, $\psi$ can be written in the equivalent form

$$\bigwedge_i x_i - \beta_i \leq x \leq x_i - \alpha_i \wedge x_i - x \equiv_m k_i.$$

Let $A = \{i \mid \alpha_i > -\infty\}$ and $B = \{i \mid \beta_i < \infty\}$. There are three cases to consider. For the first case, assume that $B \neq \emptyset$. If there exists a satisfying $x$, then there is one of the form $x_j - \beta_j + \delta$ with $\delta \in \{0, \ldots, m-1\}$, where $j$ maximises the lower bound $x_j - \beta_j$ (and thus $\beta_j < \infty$), yielding the following claim.

**Claim.** *The following quantifier-free formula is equivalent to* $\varphi$*:*

$$\widetilde{\varphi} \equiv \bigvee_{\delta \in \{0,\ldots,m-1\}} \bigvee_{j \in B} \bigwedge_i x_i - \beta_i \leq x_j - \beta_j + \delta \leq x_i - \alpha_i \wedge x_i - (x_j - \beta_j + \delta) \equiv_m k_i. \tag{2}$$

Proof of the claim. For the inclusion $[\![\widetilde{\varphi}]\!] \subseteq [\![\varphi]\!]$, let $(a_1, \ldots, a_n) \in [\![\widetilde{\varphi}]\!]$. There exist $\delta$ and $j$ as per (2), and thus taking $a := x_j - \beta_j + \delta$ yields $(a, a_1, \ldots, a_n) \in [\![\varphi]\!]$. For the other inclusion, let $(a_1, \ldots, a_n) \in [\![\varphi]\!]$. There exists $a \in \mathbb{Z}$ s.t. $(a, a_1, \ldots, a_n) \in [\![\psi]\!]$. Let $j$ be s.t. $a_j - \beta_j$ is maximised (hence $j \in B$), and define $\delta := a - (a_j - \beta_j) \mod m$. Clearly $\delta \geq 0$ since $a$ satisfies all the lower bounds $a \geq a_i - \beta_i$. Since $a$ satisfies all the upper bounds $a \leq a_i - \alpha_i$ and $a_j - \beta_j + \delta \leq a$, upper bounds are also satisfied. Finally, since $a_i - a \equiv_m k_i$ and $a \equiv_m a_j - \beta_j + \delta$, the modular constraints $a_i - (a_j - \beta_j + \delta) \equiv_m k_i$ are also satisfied. Thus, we have $(a_1, \ldots, a_n) \in [\![\widetilde{\varphi}]\!]$, as required. □

The constraint in (2) can be rewritten into the equivalent $3M$-bounded DNF constraint

$$\bigvee_{\delta \in \{0,\ldots,m-1\}} \bigvee_{j \in B} \bigwedge_i \beta_j - \delta - \beta_i \leq x_j - x_i \leq \beta_j - \delta - \alpha_i \wedge x_j - x_i \equiv_m \beta_j - \delta - k_i, \tag{3}$$

which concludes the first case.

For the second case, assume that $B = \emptyset$ but $A \neq \emptyset$. If there exists a satisfying $x$, then there is one of the form $x_j - \alpha_j - \delta$ for some $\delta \in \{0, \ldots, m-1\}$, where $j$ minimizes the upper bound $x_j - \alpha_j$ (and thus $\alpha_j > -\infty$). This[6] yields the following quantifier-free formula equivalent to $\varphi$:

$$\bigvee_{\delta \in \{0,\ldots,m-1\}} \bigvee_{j \in A} \bigwedge_i x_j - \alpha_j - \delta \leq x_i - \alpha_i \wedge x_i - (x_j - \alpha_j - \delta) \equiv_m k_i. \tag{4}$$

[6] Since lower bound constraints are trivial, in general there exists an arguably simpler witness for $x$ of the form $\delta$ for some $\delta \in \{0, \ldots, m-1\}$. This would yield a quantifier-free formula of the form

$$\bigvee_{\delta \in \{0,\ldots,m-1\}} \bigwedge_i \delta \leq x_i - \alpha_i \wedge x_i - \delta \equiv_m k_i,$$

which however would not be a formula of discrete time logic (which can speak only about differences $x_i - x_j$).

The formula above is shown to be equivalent to $\varphi$ with an argument analogous as in the previous case. The constraint in (4) can be rewritten into the equivalent $3M$-bounded DNF constraint

$$\bigvee_{\delta \in \{0,\ldots,m-1\}} \bigvee_{j \in A} \bigwedge_{i} x_j - x_i \leq \alpha_j + \delta - \alpha_i \wedge x_i - x_i \equiv_m \alpha_j + \delta - k_i. \tag{5}$$

Finally, for the last case, assume that $A = B = \emptyset$, and thus both upper and lower bound constraints are trivial. In this degenerate case, it suffices to find $x$ s.t. $\bigwedge_i x_i - x \equiv_m k_i$ is satisfied. By resolving the first such constraint, we obtain $x \equiv_m x_1 - k_1$. By replacing $x$ with $x_1 - k_1$ in all the other constraints, we obtain the following quantifier-free formula equivalent to $\varphi$.

$$\bigwedge_{i=2}^{k} x_i - (x_1 - k_1) \equiv_m k_i. \tag{6}$$

The constraint above can be rewritten into the equivalent $M$-bounded DNF constraint

$$\bigwedge_{i=2}^{k} x_1 - x_i \equiv_m k_1 - k_i. \tag{7}$$

In each case we obtain an equivalent $3M$-bounded DNF constraint. By repeating this argument, if $k$ variables are eliminated, we obtain an equivalent $3^k M$-bounded DNF constraint, as required. □

*3.2.2 Quantifier elimination for dense time.* The *orbit* of a vector $\vec{a} = (a_1 \ldots a_l) \in \mathbb{Q}^l$ is the set of those vectors $\vec{b} = (b_1 \ldots b_l) \in \mathbb{Q}^l$ s.t., for every $1 \leq i < j \leq l$, $a_i \leq a_j$ iff $b_i \leq b_j$. Intuitively, an orbit is uniquely defined by fixing a total preorder $\precsim$ on the set of coordinates $\{1, \ldots, l\}$ s.t. $i \precsim j$ iff $a_i \leq a_j$. For example, for $l = 4$ the two vectors $(0, 2.1, 2.1, 1)$ and $(7.3, 8, 8, 7.4)$ are in the same orbit as witnessed by the total preorder $1 \prec 4 \prec 2 \simeq 3$, but $(0, 2.1, 2.1, 2.1)$ is in another orbit since it corresponds to the different total preorder $1 \prec 2 \simeq 3 \simeq 4$. We write $\mathrm{orbits}(\mathbb{Q}^l) \subseteq 2^{\mathbb{Q}^l}$ for the set of orbits of $\mathbb{Q}^l$, which is finite and of size exponential in $l$. Two distinct orbits are disjoint and $\mathbb{Q}^l$ is partitioned into finitely many orbits. For an orbit $o \in \mathrm{orbits}(\mathbb{Q}^l)$, let its *characteristic formula* $\varphi_o$ be defined as

$$\varphi_o(x_1, \ldots, x_k) \equiv \bigwedge_{a_i \leq a_j} x_i \leq x_j,$$

where $(a_1, \ldots, a_l)$ is any representative in $o$ (by the definition of orbit, $\varphi_o$ does not depend on the choice of representative). Clearly, $[\![\varphi_o]\!] = o$, and the denotation $[\![\varphi]\!] \subseteq \mathbb{Q}^l$ of every formula of dense time $\varphi$ is a (necessarily finite) union of orbits [20].

Lemma 3.5. *For every formula of dense time logic $\varphi$ of dimension $l$ one can find in time exponential in $l$ an equivalent constraint in DNF.*

Proof. A constraint $\varphi$ of dimension $l$ can be transformed in DNF by enumerating all orbits $o \in \mathrm{orbits}(\mathbb{Q}^l)$ and checking whether $o \models \psi$, which can be done in time exponential in $l$. An existential formula of dimension $l$ of the form $\varphi \equiv \exists x \cdot \psi$, where $\psi \equiv \bigvee_i \psi_i$ is a constraint in DNF of dimension $l + 1$, is equivalent to the constraint in DNF $\widetilde{\varphi}$ obtained from $\psi$ by replacing all atomic formulas containing an occurrence of $x$ with the constant $\top$. □

Corollary 3.6 ([20]). *The dense time structure $(\mathbb{Q}, \leq)$ admits effective quantifier elimination.*

## 4 TIMED REGISTER PUSHDOWN AUTOMATA

We are interested in an extension of pushdown automata where control states and stack symbols are equipped with tuples of values from the hybrid time domain $\mathbb{H} = (\mathbb{Z}, +1, \leq, \equiv_m) \uplus (\mathbb{Q}, \leq)$ introduced in Sec. 3. Variables over $\mathbb{H}$ are also called *registers* in this context. We allow registers in the finite control (control registers), in the stack symbols (stack registers), and in the input symbols (input registers). Upon performing a transition, current and next control registers, as well as registers of the topmost stack symbol and input registers, are constrained with hybrid logic constraints. Thanks to the elimination of quantifiers result of Theorem 3.1, constraints are equi-expressive with first-order logic formulas and thus, complexity considerations aside, this is no restriction. Integer registers in the finite control are restricted to have bounded span (otherwise the model has undecidable nonemptiness). All other registers are not restricted to have bounded span. In particular, we allow possibly unbounded span between current and next control registers, registers on top of the stack, and in the input.

A *timed register pushdown automaton* (TRPDA) of *dimension* $(k, l) \in \mathbb{N} \times \mathbb{N}$ is a tuple

$$\mathcal{P} = \langle A, \Gamma, Q, I, F, K, (\mathsf{push}_\delta, \mathsf{pop}_\delta)_{\delta \in \Delta} \rangle$$

where $A$ is a finite input alphabet, $\Gamma$ is a finite stack alphabet, $Q$ is a finite set of control states, of which states in $I, F \subseteq Q$ are initial and final, respectively, $K \in \mathbb{N}$ is a universal bound on the span of integer control registers (encoded in binary), and $\Delta = Q \times A \times Q \times \Gamma$ is the set of transitions. For every transition $\delta = (p, a, q, \gamma) \in \Delta$, $\mathsf{push}_{paq\gamma}$ and $\mathsf{pop}_{paq\gamma}$ are constraints of dimension $(4k, 4l)$. A push constraint $\mathsf{push}_{paq\gamma}(\vec{x}_p, \vec{x}_a, \vec{x}'_q, \vec{x}_\gamma)$ has $4(k+l)$ free variables $\vec{x}_p, \vec{x}_a, \vec{x}'_q, \vec{x}_\gamma$ (each of size $k+l$), where $\vec{x}_p = (x^{\mathbb{Z}}_{p,1}, \dots, x^{\mathbb{Z}}_{p,k}, x^{\mathbb{Q}}_{p,1}, \dots, x^{\mathbb{Q}}_{p,l})$ represents integer and dense registers in the current control state $p$, $\vec{x}_a$ represents the timestamps associated with the input symbol $a$, $\vec{x}'_q$ represents the registers in the next control state $q$, and $\vec{x}_\gamma$ represents the registers associated with the stack symbol $\gamma$ (which in this case is pushed on the stack); similarly for $\mathsf{pop}_{paq\gamma}$. Since by Theorem 3.1 hybrid time domain admits effective quantifier elimination, considering arbitrary first-order formulas instead of constraints would not change the expressive power of the model. For complexity considerations, we assume that constraints are presented in DNF, that all modulo constraints $x - y \equiv_m k$ use the same modulus $m$, and that all integer constants are encoded in binary.

The semantics of a TRPDA $\mathcal{P}$ is given by the infinite-state pushdown automaton

$$\mathcal{P}' = \left\langle A', \Gamma', Q', I', F', \Delta_{\mathsf{push}}, \Delta_{\mathsf{pop}} \right\rangle, \text{ where}$$

- $A' = A \times \mathbb{Z}^k \times \mathbb{Q}^l$ is the infinite input alphabet,
- $\Gamma' = \Gamma \times \mathbb{Z}^k \times \mathbb{Q}^l$ is the infinite stack alphabet,
- $Q' = Q \times (\mathbb{Z}^k)_{\text{SPAN} \leq K} \times \mathbb{Q}^l$ is the infinite set of configurations, where the integer component has span bounded by $K$,
- $I' = I \times (\mathbb{Z}^k)_{\text{SPAN} \leq K} \times \mathbb{Q}^l \subseteq Q'$ and $F' = F \times (\mathbb{Z}^k)_{\text{SPAN} \leq K} \times \mathbb{Q}^l \subseteq Q'$ are the subsets of initial and final states, respectively, and
- $\Delta_{\mathsf{push}} \subseteq Q' \times A' \times Q' \times \Gamma'$ is defined as the union, over all $(p, a, q, \gamma) \in \Delta$, of relations of the form $\left\{((p, t), (a, u), (q, v), (\gamma, w)) \,\middle|\, (t, u, v, w) \in [\![\mathsf{push}_{paq\gamma}]\!]\right\}$; similarly for $\Delta_{\mathsf{pop}}$.

All classical notions for pushdown automata apply to $\mathcal{P}'$, and in particular the notion of run, accepting run, and recognised (timed) language $L(\mathcal{P}) \subseteq (A')^*$. For control states $p, q \in Q$ and vectors $\vec{u}, \vec{v} \in \mathbb{Z}^k \times \mathbb{Q}^l$, we write $\vec{u} \leadsto_{pq} \vec{v}$ if there exists a run from configuration $(p, \vec{u}) \in Q'$ to $(q, \vec{v}) \in Q'$ starting and ending with empty stack. Thus, $\leadsto_{pq}$ is a subset of $(\mathbb{Z}^k \times \mathbb{Q}^l) \times (\mathbb{Z}^k \times \mathbb{Q}^l)$, and we call the family of such relations $\left\{\leadsto_{pq}\right\}_{p,q \in Q}$ the *reachability relation* of $\mathcal{P}$.

### 4.1 Main result

The following is the most fundamental algorithmic problem in the analysis of infinite-state systems, such as TRPDA.

NON-EMPTINESS PROBLEM FOR TRPDA.
**Input:** A TRPDA $\mathcal{P}$.
**Output:** Do there exist an initial $(p, \vec{u}) \in I'$ and a final configuration $(q, \vec{v}) \in F'$ s.t. $\vec{u} \leadsto_{pq} \vec{v}$?

In this paper we solve a more general problem than non-emptiness: Instead of checking algorithmically whether $\vec{u} \leadsto_{pq} \vec{v}$ holds for some initial and final configurations, we effectively characterise as a constraint in hybrid logic *all pairs* of vectors $(\vec{u}, \vec{v})$ s.t. $\vec{u} \leadsto_{pq} \vec{v}$ holds. The following is our first major result.

THEOREM 4.1. *For any TRPDA $\mathcal{P}$ and control states $p, q$ thereof, one can compute in* 2-EXPTIME *a hybrid logic constraint $\psi_{pq}$ in DNF s.t. $[\![\psi_{pq}]\!] = \leadsto_{pq}$.*

Since the reachability relation is characterised in a decidable logic, the non-emptiness problem reduces to satisfiability and we obtain the following corollary, which is one of the main results of the original communication [12].

COROLLARY 4.2. *The non-emptiness problem for TRPDA is decidable in* 2-EXPTIME.

PROOF. Let $\mathcal{P}$ be a TRPDA and let $\{\psi_{pq}\}_{p,q \in Q}$ be a family of satisfiable constraints characterising the reachability relation of $\mathcal{P}$. Then $\mathcal{P}$ is non-empty if, and only if, $\bigvee_{p \in I, q \in F} \psi_{pq}$ is satisfiable. The latter condition is checked in linear time by direct inspection, since the $\psi_{pq}$'s are in DNF and contain only satisfiable conjuncts. □

The proof of Theorem 4.1 will be given in Section 6. It consists in reducing the computation of the TRPDA reachability relation to the reachability set of a suitably constructed *integer branching vector addition system*, which we introduce in Sec. 5. We conclude this section by describing known results for subclasses of TRPDA, their relationship with other models, and examples illustrating their expressive capabilities.

### 4.2 State of the art

*TRPDA vs. definable PDA.* The model of TRPDA is an instantiation of *definable* PDA [9], a generalisation of PDA along the lines of [? ] (cf. also the recent book on the subject [? ]). When the underlying data comes from an *oligomorphic*[7] structure, we provided a general construction showing decidability of the non-emptiness problem for definable PDA, and even a generic saturation procedure based on finite automata [9]. One could go a step forward and prove that the reachability relation for PDA over oligomorphic atoms is a set definable in first-order logic, thus providing an expressibility result along the lines of this paper. However, the structure of hybrid time $\mathbb{H} = (\mathbb{Z}, +1, \leq, \equiv_m) \uplus (\mathbb{Q}, \leq)$ that we consider in this paper is not oligomorphic: In fact, already discrete time $(\mathbb{Z}, +1, \leq)$ is not oligomorphic, for the simple reason that an automorphism of the structure $(\mathbb{Z}, +1, \leq)$ needs to preserve distances, and thus $\mathbb{Z}^2$ has infinitely many orbits (two pairs $(x, y)$ and $(x', y')$ are in the same orbit precisely when $x - y = x' - y'$). Consequently, the results of [9] do not apply to TRPDA, and new insights are needed.

In the rest of this section, we present examples of increasingly expressive subclasses of timed-register pushdown automata studied in previous works, culminating with the full model of TRPDA studied in this paper. Besides, we also consider timed-register context-free grammars, a model expressively incomparable with timed-register pushdown automata.

[7] A relational structure $\mathbb{A}$ is oligomorphic if the set of tuples $\mathbb{A}^k$ is orbit finite for every $k$.

*trPDA without stack.* In the definition of trPDA we require a fixed bound $K \in \mathbb{N}$ on the span of integer control registers. This requirement is necessary in order to ensure that non-emptiness is decidable. In fact, if the span is unbounded, then the non-emptiness problem is undecidable in discrete time and without stack, i.e., for definable finite automata over atoms $(\mathbb{Z}, +1, \leq)$. Moreover, $k = 3$ integer registers suffice to show undecidability. This is achieved by simulating a two counter machine with zero tests (i.e., a Minsky machine): Let $x, y, z$ be the three integer registers. The two counters are represented by $x - z$ and $y - z$, respectively. Increasing the first counter is simulated by $x' = x + 1$, decreasing by $x' = x - 1$, and zero test by $x = z$; similarly for the second counter. On the other hand, if we impose a bound on the span and no stack, then we obtain *orbit-finite timed-register automata* (we use the shorthand trNFA as this model is a timed-register counterpart of classical NFA), which have a PSpace-complete non-emptiness problem and generalise timed automata with uninitialised clocks [7].

*trPDA with timeless stack.* Going further, by adding a classical *timeless* stack to orbit-finite trNFA, we obtain trPDA with timeless stack, an expressive model with an ExpTime-complete non-emptiness problem [10, Theorem IV.8] already subsuming several other models from the literature, such as *pushdown timed automata* (PTA, timeless stack) [? ] with uninitialised clocks and *dense-timed PDA* (*timed* stack) [1] with uninitialised clocks. Due to the interplay of the monotonicity of time and the stack discipline, it was shown that the latter two models are semantically equivalent, in the sense that they recognise the same class of timed languages; moreover, the translation from dense-timed PDA to PTA is effective [10, Theorem II.1]. This is a somewhat unexpected result, since dense-timed PDA have a *timed stack*, and the fact that it can be untimed while preserving the recognised timed language is surprising.

*Timed-register context-free grammars (trCFG).* A generalisation of trNFA incomparable with trPDA with timeless stack is obtained by adding timing information to context-free grammars. A *timed-register context-free grammar* (trCFG) is obtained from trPDA (with timed stack) by requiring that there is only one control location, with no control registers, and with the possibility of pushing and popping many timed stack symbols at once. Non-emptiness of trCFG is ExpTime-complete [10, Theorem IV.3]. While the untiming of a trCFG language is still context-free [10, Lemma IV.2], trCFG recognise timed languages which cannot be recognised by trPDA with untimed stack, such as timed palindromes [10, Example IV.2].

In the example below, we demonstrate that trPDA with two (essentially untimed) control locations and timed stack also recognise timed palindromes, and thus are more expressive than trPDA with untimed stack.

*Example 4.3 (Timed palindromes).* Let the input alphabet $A = \{a, b\}$ contain two input symbols of dimension $(k, l) = (0, 1)$ (any other choice except $k = l = 0$ would do), and consider the language $L$ of *timed palindromes* of even length,

$$L = \left\{ ww^R \;\middle|\; w \in (A \times \mathbb{Q})^* \right\}.$$

Notice how palindromicity is required also in the timestamps, which makes it impossible for $L$ to be recognised by a trPDA with timeless stack. We construct trPDA $\mathcal{P}$ recognising $L$ with just two control states $Q = \{p, q\}$ of which $p$ is initial and $q$ is final, and stack alphabet $\Gamma = \left\{\bar{a}, \bar{b}\right\}$. In control state $p$, upon reading input $(c, t) \in A \times \mathbb{Q}$, the automaton pushes $(\bar{c}, t) \in \Gamma \times \mathbb{Q}$ on the stack, and it decides nondeterministically whether to stay in $p$, or move to control state $q$. From control state $q$, the automaton pops the topmost stack symbol $(\bar{c}, t)$ if it matches the current input symbol

$(c, t)$. This gives rise to transitions

$$\mathsf{push}_{pcr\bar{c}}(x_p, x_c, x'_r, x_{\bar{c}}) \equiv x_c = x_{\bar{c}} \qquad \text{where } r \in \{p, q\} \text{ and } c \in \{a, b\},$$
$$\mathsf{pop}_{qcq\bar{c}}(x_q, x_c, x'_q, x_{\bar{c}}) \equiv x_c = x_{\bar{c}} \qquad \text{where } c \in \{a, b\}\ .$$

Notice how the control registers of the form $x_r, x'_r$ with $r \in \{p, q\}$ are not mentioned in the constraints above, thus showing that the control is in fact timeless, and what only matters is that the stack is timed.

*Orbit-finite* TRPDA. A generalisation of TRPDA with timeless stack (and thus of TRNFA) incomparable with TRCFG is provided by *orbit-finite* TRPDA, which are obtained from TRPDA by requiring that the span between integer control and stack registers be bounded. Formally, for every $\delta = (p, a, q, \gamma) \in \Delta$ we require the following projection to be orbit-finite

$$\exists \vec{x}_a, \vec{x}_p \cdot \mathsf{push}_\delta(\vec{x}_p, \vec{x}_a, \vec{x}'_q, \vec{x}_\gamma) \qquad \exists \vec{x}_a, \vec{x}'_q \cdot \mathsf{pop}_\delta(\vec{x}_p, \vec{x}_a, \vec{x}'_q, \vec{x}_\gamma). \tag{†}$$

Orbit-finite TRPDA syntactically generalise TRPDA with timeless stack, because the latter satisfy the orbit-finite restriction (†) immediately since there are no stack variables $\bar{x}_\gamma$ and thanks to the bound on the span of integer control registers. While untimings of untimed-stack TRPDA and TRCFG languages are context-free, orbit-finite TRPDA can recognise timed languages with non-context-free untiming, as the following example demonstrates. This implies that this model strictly generalise TRPDA with untimed stack and is incomparable with TRCFG.

*Example 4.4 (Untimed palindromes with counting).* In this example we show that orbit-finite TRPDA can use the integer registers to check counting constraints on top of untimed palindromicity, and thus untimed non-context-free languages. Consider the untimed language $L$ of palindromes over $A = \{a, b\}$ containing the same number of $a$'s and $b$'s. We construct an orbit-finite TRPDA of dimension $(1, 0)$ recognising $L$ as follows. There are four control locations $Q = \{p, q, r, s\}$, of which $p$ is initial $I = \{p\}$ and $s$ is final $F = \{s\}$. The stack alphabet contains three symbols $\Gamma = \left\{\bar{a}, \bar{b}, \perp\right\}$, where the last one is used only at the beginning and at the end of the run. Along the lines of [10], in this example we assume symbols $\bar{a}$ and $\bar{b}$ to have dimension $(0, 0)$, i.e., to be untimed, thus deviating from the definition of TRPDA that does not allow different symbols to have different dimensions (simply because this would not increase the expressiveness of the model). In consequence, the orbit-finiteness restriction (†) is immediately satisfied in case of symbols $\bar{a}$ and $\bar{b}$, and is a non-trivial restriction only in case of $\perp$. The automaton initially guesses an integer $k \in \mathbb{Z}$, saves it in the control register and pushes it on the empty stack $(\perp, k)$; this will provide a reference value to be used at the end of the run:

$$\mathsf{push}_{p\varepsilon q\perp}(x_p, x'_q, x_\perp) \equiv x'_q = x_\perp.$$

(We use here an epsilon transition for simplicity, but it can easily be removed.) In the rest of the run, the automaton reads untimed input letters and checks palindromicity. Additionally, if an "$a$" is read, then the control register is increased, and if a "$b$" is read, then it is decreased:

$$\mathsf{push}_{qau\bar{a}}(x_q, x'_u, x_{\bar{a}}) \equiv x'_u = x_q + 1 \qquad \text{where } u \in \{q, r\},$$
$$\mathsf{push}_{qbu\bar{b}}(x_q, x'_u, x_{\bar{b}}) \equiv x'_u = x_q - 1 \qquad \text{where } u \in \{q, r\},$$
$$\mathsf{pop}_{rar\bar{a}}(x_r, x'_r, x_{\bar{a}}) \equiv x'_r = x_r + 1,$$
$$\mathsf{pop}_{rbr\bar{b}}(x_r, x'_r, x_{\bar{b}}) \equiv x'_r = x_r - 1.$$

(We assume for simplicity that the input is untimed, and thus there is no $x_c$ variable in the rules above.) Finally, when the bottom of the stack symbol $(\perp, k)$ is reached, the automaton checks that the control register $x_r$ equals the value $k$ at the bottom of the stack:

$$\mathsf{pop}_{r\varepsilon s\bot}(x_r, x'_s, x_\bot) \;\equiv\; x_r = x_\bot.$$

This shows that TRPDA recognise non-context-free languages.

The non-emptiness problem of orbit-finite TRPDA is in NEXPTIME and EXPTIME-hard [10, Theorem IV.5].

*TRPDA with orbit-finite stack alphabet.* Orbit-finite TRPDA have the essential limitation that push and pop operations require that the integer values on the top of the stack be close to those in the control. Consequently, orbit-finite TRPDA cannot recognise the language $M$ of *timed* palindromes with the same number of $a$'s and $b$'s (Example 4.4 showed how to recognise untimed such palindromes). The following example shows that lifting (†) allows to recognise such timed palindromes, and thus strictly increases the expressive power of TRPDA.

*Example 4.5 (Timed palindromes with counting).* We construct an TRPDA of dimension $(1, 0)$, and thus of orbit-finite stack, recognising $M$. The construction essentially combines Example 4.3 for timed palindromes (but no counting), and 4.4 for untimed palindromes with counting:

$$\begin{aligned}
\mathsf{push}_{p\varepsilon q\bot}(x_p, _, x'_q, x_\bot) \;&\equiv\; x'_q = x_p = x_\bot, \\
\mathsf{push}_{qau\bar{a}}(x_q, x_a, x'_u, x_{\bar{a}}) \;&\equiv\; x'_u = x_q + 1 \wedge x_{\bar{a}} = x_a && \text{where } u \in \{q, r\}, \\
\mathsf{push}_{qbu\bar{b}}(x_q, x_b, x'_u, x_{\bar{b}}) \;&\equiv\; x'_u = x_q - 1 \wedge x_{\bar{b}} = x_b && \text{where } u \in \{q, r\}, \\
\mathsf{pop}_{rar\bar{a}}(x_r, x_a, x'_r, x_{\bar{a}}) \;&\equiv\; x'_r = x_r + 1 \wedge x_{\bar{a}} = x_a, \\
\mathsf{pop}_{rbr\bar{b}}(x_r, x_b, x'_r, x_{\bar{b}}) \;&\equiv\; x'_r = x_r - 1 \wedge x_{\bar{b}} = x_b, \\
\mathsf{pop}_{r\varepsilon r\bot}(x_r, _, x'_r, x_\bot) \;&\equiv\; x'_r = x_r = x_\bot.
\end{aligned}$$

Notice that 1) we now have unbounded differences between control and stack clocks (the control clock $x_u$ is increased/decreased independently from the timestamp $x_{\bar{c}}$ on top of the stack) and 2) the TRPDA above has orbit-finite stack alphabet, since it has integer dimension $k = 1$.

Notwithstanding the increased expressive power gained by removing the orbit-finite restriction (†), decidability is preserved. We have shown in our previous communication that non-emptiness of TRPDA is in 2-EXPTIME and EXPTIME-hard [12, Theorem 1], under the somewhat technical assumption that the stack alphabet be orbit finite, i.e., there must exists a bound $K \in \mathbb{N}$ on the span of stack symbols $\Gamma' = \Gamma \times (\mathbb{Z}^k)_{\text{SPAN} \leq K} \times \mathbb{Q}^l$. (The orbit-finite restriction on the input alphabet is inessential for non-emptiness, since the input is existentially quantified and thanks to elimination of quantifiers of Theorem 3.1.) In this paper, we show that the same complexity applies even without the orbit-finite assumption on the stack alphabet, as announced earlier in Corollary 4.2.

## 5 INTEGER BRANCHING VECTOR ADDITION SYSTEMS

An *integer branching vector addition system* ($\mathbb{Z}$-BVASS) is a tuple $\mathcal{B} = (\mathrm{Var}, T)$, where Var is a set of nonterminal symbols and $T$ is a finite set of transitions of the form $X \longleftarrow t$, with $X \in \mathrm{Var}$ and $t$ an expression built according to the following abstract syntax:

$$t ::= S \mid X \mid t \cup t \mid t \cap S \mid t + t \mid t - t \mid -t,$$

with $S$ a semilinear subset of $\mathbb{Z}$ and $X \in \mathrm{Var}$. We say that $M$ is the moduli bound of $\mathcal{B}$ if it is the smallest number such that all semilinear sets used in $\mathcal{B}$ are $M$-bounded. A *valuation* $\mu : (2^{\mathbb{Z}})^{\mathrm{Var}}$ is a mapping that assings to every nonterminal $X$ a set of integers $\mu(X)$, which extends by structural induction to terms $t$. A *solution* is a valuation $\mu$ s.t. for every transition $X \longleftarrow t$ we have $\mu(X) \supseteq \mu(t)$.

Since transitions are monotone w.r.t. set inclusion, the least solution $\mu^*$ exists. Let the *reachability set* of nonterminal $X$ be its value in the least solution $[\![X]\!] = \mu^*(X)$.

*Example 5.1.* Semilinear subsets of $\mathbb{Z}$ encoded in binary can be expressed as reachability sets of $\mathbb{Z}$-BVASS of polynomial size using only the constant 1. An integer $k \in \mathbb{Z}$ encoded in binary can be expressed as the reachability set $[\![X_k]\!] = \{k\}$ of a nonterminal $X_k$ in the following $\mathbb{Z}$-BVASS with $\log k$ transitions

$$\begin{aligned} X_1 &\leftarrow \{1\} \\ X_0 &\leftarrow X_1 - X_1 \end{aligned} \qquad\qquad \left.\begin{aligned} X_{2k} &\leftarrow X_k + X_k \\ X_{2k+1} &\leftarrow X_{2k} + X_1 \\ X_{-k} &\leftarrow X_0 - X_k \end{aligned}\right\} \text{ for } k > 0$$

We can encode a linear set of the form $L = b + p^*$ as $[\![X_L]\!] = L$ with a transition $X_L \leftarrow X_b \cup (X_L + X_p)$. Finally, a semilinear set $S = L_1 \cup \cdots \cup L_k$ is encoded as $X_S \leftarrow X_{L_1} \cup \cdots \cup X_{L_k}$.

The following are the fundamental decision problems for $\mathbb{Z}$-BVASS.

Reachability problem for $\mathbb{Z}$-BVASS.
**Input:** A $\mathbb{Z}$-BVASS, a number $n$ encoded in binary, and a nonterminal $X$ thereof.
**Output:** Does $n \in [\![X]\!]$ hold?

Zero reachability problem for $\mathbb{Z}$-BVASS.
**Input:** A $\mathbb{Z}$-BVASS and a nonterminal $X$ thereof.
**Output:** Does $0 \in [\![X]\!]$ hold?

Non-emptiness problem for $\mathbb{Z}$-BVASS.
**Input:** A $\mathbb{Z}$-BVASS and a nonterminal $X$ thereof.
**Output:** Is $[\![X]\!]$ non-empty?

The three problems above are all PTime equivalent for $\mathbb{Z}$-BVASS. Reachability of $n \in [\![X]\!]$ reduces to zero reachability $0 \in [\![X']\!]$ for a new nonterminal $X'$ and transition $X' \leftarrow X - X_n$, where $X_n$ is defined in Example 5.1 above. Zero reachability $0 \in [\![X]\!]$ reduces to non-emptiness of $[\![X']\!]$ for a new nonterminal $X'$ and an additional transition $X' \leftarrow X \cap \{0\}$. Finally, non-emptiness of $[\![X]\!]$ reduces to zero reachability $0 \in [\![X']\!]$ for a new nonterminal $X'$ and transitions $X' \leftarrow X$, $X' \leftarrow X' + \{1\}$, and $X' \leftarrow X' - \{1\}$.

The use of intersection in $\mathbb{Z}$-BVASS is limited to the form $X_i \cap S$ where $S$ is a semilinear set. Unrestricted intersection of the form $X_i \cap X_j$ leads to undecidability of the non-emptiness problem. In fact, already over $\mathbb{N}$ unrestricted intersection enables the simulation of *unary conjunctive grammars*, which have an undecidable non-emptiness problem [17]: Given a unary conjunctive $\mathcal{G}$ grammar, one can build a $\mathbb{Z}$-BVASS $\mathcal{B}$ with unrestricted intersection by replacing every terminal in the grammar with the constant $\{1\}$, and concatenation "$\cdot$" with addition "$+$" . Then, $\mathcal{B}$ is non-empty iff $\mathcal{G}$ is non-empty.

The following is the second main result of this paper. The proof is postponed to Section 7.

Theorem 5.2. *Let $\mathcal{B}$ be a $\mathbb{Z}$-BVASS. Reachability sets of $\mathcal{B}$ are semilinear. They are computable in time exponential in the number of nonterminals and the moduli bound of $\mathcal{B}$.*

Corollary 5.3. *The non-emptiness, reachability, and zero-reachability problems for $\mathbb{Z}$-BVASS are in* 2-ExpTime *for moduli bound in binary and* ExpTime *for moduli bound in unary.*

Moreover, all the problems above are PSpace-hard, since $\mathbb{Z}$-BVASS can simulate $\mathbb{Z}$-BVASS with constants encoded in binary; cf. Theorem 5.6.

### 5.1 Intersection-free and singleton-intersection $\mathbb{Z}$-BVASS

A $\mathbb{Z}$-BVASS is *intersection-free* if no intersection is allowed, not even of the restricted form $X_i \cap S$:

$$t ::= S \mid X_j \mid t \cup t \mid t + t \mid t - t \mid -t.$$

THEOREM 5.4 ([10]). *The non-emptiness problem for intersection-free $\mathbb{Z}$-BVASS is in* PTIME*, and reachability sets thereof are semilinear and computable in* EXPTIME.

PROOF. Let $\mathcal{B}$ be a $\mathbb{Z}$-BVASS. The idea is to construct a context-free grammar $\mathcal{G}$ by replacing addition "+" with concatenation "·". First, we do some preprocessing on $\mathcal{B}$. Since there is no intersection in $\mathcal{B}$, we replace all semilinear constants $S$ with a corresponding nonterminal $X_S$, adding new transitions according to the construction of Example 5.1; in this way, the only constant used in $\mathcal{B}$ is $\{1\}$. For every nonterminal $X$, we add a new nonterminal $\widehat{X}$ (with the convention that $\widehat{\widehat{X}} = X$) s.t. for every rule $X \leftarrow t$ we have a new rule $\widehat{X} \leftarrow -t$; in this way, $[\![X]\!] = -[\![\widehat{X}]\!]$. We remove binary subtraction "−" with the equivalence $t_0 - t_1 = t_0 + (-t_1)$, and we push unary negation "−" inside, in order to appear only in front of constants and nonterminals, using the equivalences $-(t_0 \cup t_1) = (-t_0) \cup (-t_1)$ and $-(t_0 + t_1) = (-t_0) + (-t_1)$.

We are now ready to construct the grammar $\mathcal{G}$. The set of nonterminals is the same. There are two terminal symbols "+1" and "−1". A transition $X \leftarrow t$ of $\mathcal{B}$ generates a production $X \leftarrow F(t)$ of $\mathcal{G}$, where the translation function $F$ is defined by structural induction as

$$F(\{1\}) = +1 \qquad F(X) = X \qquad F(t_0 \cup t_1) = F(t_0) \cup F(t_1)$$

$$F(-\{1\}) = -1 \qquad F(-X) = \widehat{X} \qquad F(t_0 + t_1) = F(t_0) \cdot F(t_1).$$

Non-emptiness of $X$ in the $\mathbb{Z}$-BVASS is the same as non-emptiness of $X$ in the grammar, and the latter problem can be solved in PTIME. By Parikh's theorem [22], the Parikh image of the nonterminal $X$ is a semilinear set $S(X) \subseteq \mathbb{Z}^2$ constructible in EXPTIME, with the first component corresponding to terminal "+1" and the second to "−1". Since $[\![X]\!] = \{a - b \mid (a, b) \in S(X)\} \subseteq \mathbb{Z}$, $[\![X]\!]$ is semilinear and its presentation can be obtained from a presentation of $S$ in linear time. Thus, the reachability set $[\![X]\!]$ is a semilinear subset of $\mathbb{Z}$ constructible in EXPTIME, as required. □

For intersection-free $\mathbb{Z}$-BVASS, while reachability and zero-reachability are still PTIME equivalent problems, this is no longer the case for non-emptiness. In fact, zero-reachability is NP-hard already for intersection-free $\mathbb{Z}$-BVASS, and allowing intersection with the singleton constants $\{k\}$ (which for $k = 0$ is akin to a zero test in the jargon of counter machines) makes all three problems above NP-complete. A $\mathbb{Z}$-BVASS is *singleton-intersection* if intersections are allowed only of the form $t \cap \{k\}$ with $k \in \mathbb{Z}$ a constant encoded in binary:

$$t ::= S \mid X_j \mid t \cup t \mid t \cap \{k\} \mid t + t \mid t - t \mid -t.$$

THEOREM 5.5 ([10]). *Reachability and zero-rechability are* NP*-hard for intersection-free $\mathbb{Z}$-BVASS. Non-emptiness, reachabiltiy, and zero-rechability are* NP*-complete for singleton-intersection $\mathbb{Z}$-BVASS.*

### 5.2 $\mathbb{Z}$-BVASS v.s. $\mathbb{N}$-BVASS in dimension one

If we remove binary subtraction "−" and restrict our attention to non-negative solutions, then we obtain an equivalent presentation for *branching vector addition systems* ($\mathbb{N}$-BVASS) in dimension one [25], which can be defined according to the following abstract syntax (where $k \in \mathbb{Z}$):

$$t ::= X_j \mid t \cup t \mid (t + \{k\}) \cap \mathbb{N} \mid t + t.$$

While decidability of the reachability problem for $\mathbb{N}$-BVASS in higher dimension is a long-standing open problem, in dimension one decidability is easily established. Its exact complexity has recently been settled.

Theorem 5.6. *The reachability problem for $\mathbb{N}$-BVASS in dimension one is* PTime*-complete if constants are presented in unary [16], and* PSpace*-complete if in binary [14].*

Consequently, all decision problems for general $\mathbb{Z}$-BVASS are PSpace-hard.

## 6 FROM TRPDA TO $\mathbb{Z}$-BVASS

In this section we transform trPDA into a $\mathbb{Z}$-BVASS in such a way that the reachability relation of the former can be reconstructed from the reachability set of the latter. In the rest of this section, fix a trPDA $\mathcal{P} = \langle A, \Gamma, Q, I, F, K, (\mathsf{push}_\delta, \mathsf{pop}_\delta)_{\delta \in Q\times A\times Q\times\Gamma}\rangle$ of dimension $(k, l)$. First, we solve the case with discrete dimension $k = 1$, and in Sec. 6.2 we address the general case $k > 1$ by a reduction to the former.

### 6.1 Discrete dimension one

We prove Theorem 4.1 in the special case where configurations are of the form $Q \times \mathbb{Z} \times \mathbb{Q}^l$. For every pair $p, q$ of control states of the trPDA $\mathcal{P}$, and for each of the exponentially many (in $l$) orbits $o \in \mathsf{orbits}(\mathbb{Q}^{2l})$, there is a nonterminal $X_{pqo}$ in the $\mathbb{Z}$-BVASS $\mathcal{B}$. Intuitively, values reachable in $X_{pqo}$ represent the difference between the integer register of the ending control state $q$ and that of the starting control state $p$ along some run starting and ending with empty stack, when the rational values at $p$ and $q$ are related as specified by the orbit $o$.

Lemma 6.1. *For every trPDA $\mathcal{P}$ of dimension $(1, l)$ we can construct a $\mathbb{Z}$-BVASS $\mathcal{B}$ s.t. for control states $p, q$ of $\mathcal{P}$, orbit $o \in \mathsf{orbits}(\mathbb{Q}^{2l})$, integers $a, b \in \mathbb{Z}$, and rational vectors $\vec{a}, \vec{b} \in \mathbb{Q}^l$ s.t. $(\vec{a}, \vec{b}) \in o$,*

$$(a, \vec{a}) \rightsquigarrow_{pq} (b, \vec{b}) \quad \textit{iff} \quad (b - a) \in [\![X_{pqo}]\!].$$

*The number of nonterminals of $\mathcal{B}$ is exponential in $l$ and quadratic in $|Q|$, and the largest magnitude of integer constants in $\mathcal{B}$ is linear in that of $\mathcal{P}$.*

The construction of $\mathcal{B}$ is based on the following characterisation of the reachability relation of $\mathcal{P}$.

Lemma 6.2. *Let $p, q$ be control states of the trPDA $\mathcal{P}$. The relation $\rightsquigarrow_{pq}$ is the least relation satisfying the following three rules, for every $\vec{a}, \vec{b}, \vec{c}, \vec{d} \in \mathbb{Z} \times \mathbb{Q}^l$:*

$$\text{(base)} \qquad \frac{}{\vec{a} \rightsquigarrow_{pp} \vec{a}}$$

$$\text{(transitivity)} \qquad \frac{\vec{a} \rightsquigarrow_{pr} \vec{c} \quad \vec{c} \rightsquigarrow_{rq} \vec{b}}{\vec{a} \rightsquigarrow_{pq} \vec{b}}$$

$$\text{(push-pop)} \qquad \frac{\vec{c} \rightsquigarrow_{rs} \vec{d}}{\vec{a} \rightsquigarrow_{pq} \vec{b}} \quad \textit{if } (\vec{a}, \vec{c}, \vec{d}, \vec{b}) \in [\![\mathsf{push\text{-}pop}_{prsq}]\!], \textit{ where}$$

$$\mathsf{push\text{-}pop}_{prsq}(\vec{x}_p, \vec{x}_r, \vec{x}_s, \vec{x}_q) \equiv \bigvee_{a,b\in A,\gamma\in\Gamma} \exists \vec{x}_a, \vec{x}_b, \vec{x}_\gamma \cdot \mathsf{push}_{par\gamma}(\vec{x}_p, \vec{x}_a, \vec{x}_r, \vec{x}_\gamma) \wedge \mathsf{pop}_{sbq\gamma}(\vec{x}_s, \vec{x}_b, \vec{x}_q, \vec{x}_\gamma).$$

The proof of the lemma above is standard. We include a proof sketch for completeness.

Proof sketch. The reachability relation $\rightsquigarrow_{pq}$ satisfies the rules above by definition. In order to show that $\rightsquigarrow_{pq}$ is the *least* relation satisfying the rules above, one proceeds by induction on the height of derivation trees used to establish $\vec{a} \rightsquigarrow_{pq} \vec{b}$. Let $R_{pq}$ be any relation closed under the rules above. For the base case case, we have $\vec{a} \rightsquigarrow_{pp} \vec{a}$, and $(\vec{a}, \vec{a}) \in R_{pp}$ already holds by definition. There are two inductive cases. For the first case, assume $\vec{a} \rightsquigarrow_{pq} \vec{b}$ is established using the transitivity rule. There exist $\vec{c}$ and $r$ s.t. $\vec{a} \rightsquigarrow_{pr} \vec{c}$ and $\vec{c} \rightsquigarrow_{rq} \vec{b}$. By induction hypothesis, $(\vec{a}, \vec{c}) \in R_{pr}$ and $(\vec{c}, \vec{b}) \in R_{rq}$ hold, and thus by definition $(\vec{a}, \vec{b}) \in R_{pq}$, as required. The second case is analogous, using the definition of $\rightsquigarrow_{pq}$ and the inductive assumption. □

*Example 6.3 (Example 4.5 continued).* We illustrate the characterisation of the lemma above by applying it to the TRPDA of Example 4.5. We have a base case for each of the three control locations: $m \rightsquigarrow_{pp} m$, $m \rightsquigarrow_{qq} m$, and $m \rightsquigarrow_{rr} m$, for every $m \in \mathbb{Z}$. We skip the transitivity rules since the TRPDA is one-reversal bounded (runs consist of a sequence of pushes followed by a sequence of pops), and thus they do not allow one to deduce new reachability information. There are three push-pop rules, depending on whether we push on the stack $\bar{a}$, $\bar{b}$, or $\perp$:

$$(\bar{a})\ \frac{m+1 \rightsquigarrow_{ur} n}{m \rightsquigarrow_{qr} n+1}, \text{for } u \in \{q, r\} \qquad (\bar{b})\ \frac{m-1 \rightsquigarrow_{ur} n}{m \rightsquigarrow_{qr} n-1}, \text{for } u \in \{q, r\} \qquad (\perp)\ \frac{m \rightsquigarrow_{qr} m}{m \rightsquigarrow_{pr} m}$$

The last rule forces $m$ to be the same at the beginning and at the end of the run, since the corresponding conjunct of $\text{push-pop}_{pqrr}(x_p, x'_q, x_r, x'_r)$ is

$$\exists x_\perp \cdot \text{push}_{p\varepsilon q\perp}(x_p, \_, x'_q, x_\perp) \wedge \text{pop}_{r\varepsilon r\perp}(x_r, \_, x'_r, x_\perp) \equiv \exists x_\perp \cdot x'_q = x_p = x_\perp \wedge x'_r = x_r = x_\perp,$$

which is logically equivalent to $x'_q = x_p = x'_r = x_r$ by eliminating the existential quantifier. We can give the following explicit expression for the reachability relation as characterised by the rules above: $m \rightsquigarrow_{qr} n$ iff $2|n - m$, and $m \rightsquigarrow_{pr} m$. Indeed, $m \rightsquigarrow_{rr} m$ holds by the axiom, a single application rule $(\bar{a})$ with $u = r$ allows to derive $m \rightsquigarrow_{qr} m + 2$, and then further applications of the same rule with $u = q$ allow to reach $m \rightsquigarrow_{qr} m + 2k$ for every $k \geq 0$. The other rule $(\bar{b})$ allows one to decrease by even amounts.

The rules of the $\mathbb{Z}$-BVASS $\mathcal{B}$ are obtained following the characterisation of $\rightsquigarrow$ of the lemma above. For every control state $p$ and for every orbit $o \in \text{orbits}(\mathbb{Q}^{2l})$ s.t. $o \subseteq \left\{(\vec{b}, \vec{b}) \,\middle|\, \vec{b} \in \mathbb{Q}^l\right\}$, the $\mathbb{Z}$-BVASS $\mathcal{B}$ contains the transition

$$\text{(base)} \qquad X_{ppo} \leftarrow \{0\}\,.$$

For every control states $p, r, q$ and for every orbit $o \in \text{orbits}(\mathbb{Q}^{3l})$, the $\mathbb{Z}$-BVASS $\mathcal{B}$ contains the transition (where $o_{ij} \in \text{orbits}(\mathbb{Q}^{2l})$ is the projection to components $i, j \in \{1, 2, 3\}$ of the orbit $o$, defined as $o_{ij} = \left\{(\vec{a}_i, \vec{a}_j) \,\middle|\, (\vec{a}_1, \vec{a}_2, \vec{a}_3) \in o, \text{ with } \vec{a}_1, \vec{a}_2, \vec{a}_3 \in \mathbb{Q}^l\right\}$):

$$\text{(transitivity)} \qquad X_{pqo_{13}} \leftarrow X_{pro_{12}} + X_{rqo_{23}}\,.$$

Transitions simulating push-pop are more involved and are defined by a sequence of steps. In the sequel, fix arbitrary control states $p, r, s, q \in Q$.

*Step 0: Transformation in DNF.* We wish to transform $\text{push-pop}_{prsq}$ into a constraint in DNF. By assumption, $\text{push}_{par\gamma} \equiv \bigvee_i \varphi_i^{\mathbb{Z}} \wedge \varphi_i^{\mathbb{Q}}$ and $\text{pop}_{sbq\gamma} \equiv \bigvee_j \psi_j^{\mathbb{Z}} \wedge \psi_j^{\mathbb{Q}}$ are constraints in DNF, where $\varphi_i^{\mathbb{Z}}, \psi_j^{\mathbb{Z}}$ are constraints of discrete time and $\varphi_i^{\mathbb{Q}}, \psi_j^{\mathbb{Q}}$ of dense time. By distributing the connectives and by separating the discrete from the dense part, $\text{push-pop}_{prsq}$ is a disjunction of conjunctive constraints of the form $\exists \vec{x}_a, \vec{x}_b, \vec{x}_\gamma \cdot \varphi_i^{\mathbb{Z}} \wedge \varphi_i^{\mathbb{Q}} \wedge \psi_j^{\mathbb{Z}} \wedge \psi_j^{\mathbb{Q}}$. By separating the integer and rational sort, the latter formula can be rewritten equivalently as $\varphi^{\mathbb{Z}} \wedge \varphi^{\mathbb{Q}}$, where

$$\varphi^{\mathbb{Z}} \equiv \exists x_a^{\mathbb{Z}}, x_b^{\mathbb{Z}}, x_\gamma^{\mathbb{Z}} \cdot \varphi_i^{\mathbb{Z}} \wedge \psi_j^{\mathbb{Z}} \qquad \text{and} \qquad \varphi^{\mathbb{Q}} \equiv \exists \vec{x}_a^{\mathbb{Q}}, \vec{x}_b^{\mathbb{Q}}, \vec{x}_\gamma^{\mathbb{Q}} \cdot \varphi_i^{\mathbb{Q}} \wedge \psi_j^{\mathbb{Q}}.$$

By performing quantifier elimination as per Lemma 3.3, $\varphi^{\mathbb{Z}}$ is equivalent to a constraint $\widetilde{\varphi}^{\mathbb{Z}}$ in DNF constructible in exponential time (and thus of exponential size); similarly, thanks to Lemma 3.5 we obtain in exponential time a constraint $\widetilde{\varphi}^{\mathbb{Q}}$ in DNF equivalent to $\varphi^{\mathbb{Q}}$. Combining these constraints together, we have decomposed $\mathsf{push\text{-}pop}_{prsq}$ as an equivalent constraint in DNF constructible in exponential time. Let $\varphi$ be a conjunct of this DNF. It has the form

$$\varphi(\vec{x}_p, \vec{x}_r, \vec{x}_s, \vec{x}_q) \equiv \varphi^{\mathbb{Z}}(x_p^{\mathbb{Z}}, x_r^{\mathbb{Z}}, x_s^{\mathbb{Z}}, x_q^{\mathbb{Z}}) \wedge \varphi^{\mathbb{Q}}(\vec{x}_p^{\mathbb{Q}}, \vec{x}_r^{\mathbb{Q}}, \vec{x}_s^{\mathbb{Q}}, \vec{x}_q^{\mathbb{Q}}).$$

Let $o \subseteq \mathbb{Q}^{4l}$ be one of the finitely many orbits in $\mathrm{orbits}(\llbracket\varphi^{\mathbb{Q}}\rrbracket)$. The following discrete time formula $\psi(z, z')$ characterises $\llbracket\psi\rrbracket = \llbracket X_{rso_{23}}\rrbracket \times \llbracket X_{pqo_{14}}\rrbracket$ (from now on we concentrate on discrete time logic dropping the superscripts $\mathbb{Z}$ in variables for simplicity):

$$\psi(z, z') \equiv \exists x_p, x_q, x_r, x_s \cdot z = x_s - x_r \wedge z' = x_q - x_p \wedge \varphi^{\mathbb{Z}}(x_p, x_r, x_s, x_q). \tag{8}$$

*Example 6.4 (Example 6.3 continued).* In the case of our example TRPDA, we have

$$\begin{aligned}\mathsf{push\text{-}pop}_{qqrr}(x_q, x_q', x_r, x_r') \equiv\ & \exists x_a, x_a', x_{\bar{a}} \cdot \mathsf{push}_{qaq\bar{a}}(x_q, x_a, x_q', x_{\bar{a}}) \wedge \mathsf{pop}_{rar\bar{a}}(x_r, x_a', x_r', x_{\bar{a}}) \vee \\ & \exists x_b, x_b', x_{\bar{b}} \cdot \mathsf{push}_{qaq\bar{b}}(x_q, x_b, x_q', x_{\bar{b}}) \wedge \mathsf{pop}_{rar\bar{b}}(x_r, x_b', x_r', x_{\bar{b}}),\end{aligned}$$

which, by expanding the push and pop formulas, becomes

$$\begin{aligned}&\exists x_a, x_a', x_{\bar{a}} \cdot x_q' = x_q + 1 \wedge x_{\bar{a}} = x_a \wedge x_r' = x_r + 1 \wedge x_{\bar{a}} = x_a' \vee \\ &\exists x_b, x_b', x_{\bar{b}} \cdot x_q' = x_q - 1 \wedge x_{\bar{b}} = x_b \wedge x_r' = x_r - 1 \wedge x_{\bar{b}} = x_b'.\end{aligned}$$

(Notice that this TRPDA has dimension $(1, 0)$, and thus dense time formulas $\varphi^{\mathbb{Q}}$ are trivial and omitted.) We eliminate the quantifiers on the stack and input symbols and obtain the following DNF:

$$\mathsf{push\text{-}pop}_{qqrr}(x_q, x_q', x_r, x_r') \equiv (x_q' = x_q + 1 \wedge x_r' = x_r + 1) \vee (x_q' = x_q - 1 \wedge x_r' = x_r - 1). \tag{9}$$

The first conjunct above gives rise to the following existential formula

$$\psi(z, z') \equiv \exists x_q, x_q', x_r, x_r' \cdot z = x_r - x_q' \wedge z' = x_r' - x_q \wedge (x_q' = x_q + 1 \wedge x_r' = x_r + 1). \tag{10}$$

The formula $\psi$ in (8) is an existential Presburger arithmetic formula and does not allow us to immediately derive a set of $\mathbb{Z}$-BVASS rules $X_{pqo_{14}} \leftarrow (\cdots X_{rso_{23}} \cdots)$. Quantifier elimination for Presburger arithmetic yields an equivalent quantifier free formula $\widetilde{\psi}$ with atomic formulas of the general form $az + bz' \leq c$ and $az + bz' \equiv_m c$, with $a, b, c \in \mathbb{Z}$, which cannot be encoded into $\mathbb{Z}$-BVASS rules. In the following, we eliminate the quantifiers "manually", and observe that the resulting $\widetilde{\psi}$ has a special structure that we can exploit to derive the $\mathbb{Z}$-BVASS transitions. This is achieved in a number of steps.

*Step 1: Expansion.* The subformula $\varphi^{\mathbb{Z}}$ is a conjunction of atomic discrete time logic constraints of the forms $x_q - x_p \in [\alpha_{pq}, \beta_{pq}]$ with $\alpha_{pq}, \beta_{pq} \in \mathbb{Z} \cup \{-\infty, \infty\}$, and $x_q - x_p \equiv_m \gamma_{pq}$ with $\gamma_{pq} \in \mathbb{Z}$; similarly for the other indices. Thus, (8) expands to

$$\begin{aligned}\psi(z, z') &\equiv \exists x_p, x_q, x_r, x_s \cdot \psi', \\ \text{where } \psi' &\equiv z = x_s - x_r \wedge z' = x_q - x_p \wedge \\ &\qquad \alpha_{pq} \leq z' \leq \beta_{pq} \wedge z' \equiv_m \gamma_{pq} \wedge \\ &\qquad \alpha_{rs} \leq z \leq \beta_{rs} \wedge z \equiv_m \gamma_{rs} \wedge \\ &\qquad \alpha_{pr} \leq x_r - x_p \leq \beta_{pr} \wedge x_r - x_p \equiv_m \gamma_{pr} \wedge \\ &\qquad \alpha_{sq} \leq x_q - x_s \leq \beta_{sq} \wedge x_q - x_s \equiv_m \gamma_{sq} \wedge \\ &\qquad \alpha_{ps} \leq x_s - x_p \leq \beta_{ps} \wedge x_s - x_p \equiv_m \gamma_{ps} \wedge \\ &\qquad \alpha_{rq} \leq x_q - x_r \leq \beta_{rq} \wedge x_q - x_r \equiv_m \gamma_{rq}.\end{aligned} \tag{11}$$

*Step 2: Eliminate $x_s$ and $x_q$.* By using $z = x_s - x_r$ and $z' = x_q - x_p$, we can immediately eliminate $x_s$ and $x_q$, respectively. Let $\psi[x_s \mapsto z + x_r, x_q \mapsto z' + x_p]$ be obtained from $\psi$ by replacing $x_s$ with $z + x_r$, and $x_q$ by $z' + x_p$, and let $\psi_1$ be obtained from the former formula by eliminating the first two conjuncts $z = x_s - x_r \wedge z' = x_q - x_p$. Clearly $\psi_1$ is logically equivalent to $\psi$. By performing the substitution explicitly, we obtain

$$\begin{aligned}
\psi_1(z, z') \;\equiv\; & \exists x_p, x_r \cdot \psi_0 \wedge \\
& \alpha_{pr} \le x_r - x_p \le \beta_{pr} \;\wedge\; x_r - x_p \equiv_m \gamma_{pr} \wedge \\
& \alpha_{sq} \le z' + x_p - (z + x_r) \le \beta_{sq} \;\wedge\; z' + x_p - (z + x_r) \equiv_m \gamma_{sq} \wedge \\
& \alpha_{ps} \le z + x_r - x_p \le \beta_{ps} \;\wedge\; z + x_r - x_p \equiv_m \gamma_{ps} \wedge \\
& \alpha_{rq} \le z' + x_p - x_r \le \beta_{rq} \;\wedge\; z' + x_p - x_r \equiv_m \gamma_{rq}, \text{ with} \\
\psi_0(z, z') \;\equiv\; & \alpha_{pq} \le z' \le \beta_{pq} \;\wedge\; z' \equiv_m \gamma_{pq} \;\wedge\; \alpha_{rs} \le z \le \beta_{rs} \;\wedge\; z \equiv_m \gamma_{rs},
\end{aligned} \tag{12}$$

where we have singled out $\psi_0$ since it does not contain either $x_r$'s or $x_p$'s.

*Example 6.5 (Example 6.4 continued).* We eliminate the existential quantifier from variables $x_r$ and $x'_r$ in (10) and obtain

$$\psi_1(z, z') \;\equiv\; \exists x_q, x'_q \cdot x'_q - x_q = 1 \wedge z' - z = x'_q - x_q + 1. \tag{13}$$

*Step 3: Eliminate $x_r$ and $x_p$.* We observe that in $\psi_1$ the two variables $x_r$ and $x_p$ always appear together as a difference $x_r - x_p$, and thus we can eliminate the two existential quantifications jointly. We first rearrange the inequalities in $\psi_1$ to highlight $x_r - x_p$:

$$\begin{aligned}
\psi_1(z, z') \;\equiv\; \exists x_p, x_r \;\cdot\; & \psi_0 \wedge \\
\alpha_{pr} \le x_r - x_p \le \beta_{pr} \quad & \wedge\; x_r - x_p \equiv_m \gamma_{pr} \wedge \\
z' - z - \beta_{sq} \le x_r - x_p \le z' - z - \alpha_{sq} \quad & \wedge\; z' - z - (x_r - x_p) \equiv_m \gamma_{sq} \wedge \\
\alpha_{ps} - z \le x_r - x_p \le \beta_{ps} - z \quad & \wedge\; z + x_r - x_p \equiv_m \gamma_{ps} \wedge \\
z' - \beta_{rq} \le x_r - x_p \le z' - \alpha_{rq} \quad & \wedge\; z' - (x_r - x_p) \equiv_m \gamma_{rq}.
\end{aligned}$$

Following the quantifier elimination procedure used in the proof of Lemma 3.3, let $T$ be the set of *lower bound* terms, i.e., terms appearing on the left of inequalities in $\psi_1$ as written above:

$$T := \left\{\alpha_{pr},\; z' - z - \beta_{sq},\; \alpha_{ps} - z,\; z' - \beta_{rq}\right\}.$$

By guessing the largest lower bound $t \in T$, we write the following quantifier free formula $\psi_2$, equivalent to $\psi_1$

$$\begin{aligned}
\psi_2(z, z') \;\equiv\; & \psi_0 \wedge \bigvee_{\delta \in \{0, \ldots, m-1\}} \bigvee_{t \in T} \psi_{\delta, t}, \text{ with} \\
\psi_{\delta, t}(z, z') \;\equiv\; & \alpha_{pr} \le t + \delta \le \beta_{pr} \quad \wedge\, t + \delta \equiv_m \gamma_{pr} \wedge \\
& z' - z - \beta_{sq} \le t + \delta \le z' - z - \alpha_{sq} \quad \wedge\, z' - z - t - \delta \equiv_m \gamma_{sq} \wedge \\
& \alpha_{ps} - z \le t + \delta \le \beta_{ps} - z \quad \wedge\, z + t + \delta \equiv_m \gamma_{ps} \wedge \\
& z' - \beta_{rq} \le t + \delta \le z' - \alpha_{rq} \quad \wedge\, z' - t - \delta \equiv_m \gamma_{rq}.
\end{aligned} \tag{14}$$

*Step 4: Simplify $\psi_2$.* We simplify the formula $\bigvee_{t \in T} \psi_{\delta, t}$, and thus $\psi_2$, depending on the four possible values for $t$.

- *Case 1:* $t = \alpha_{pr}$. By replacing $t$ for its definition in $\psi_{\delta,t}$, we obtain

$$\begin{aligned}
\alpha_{pr} \leq \alpha_{pr} + \delta \leq \beta_{pr} & \quad \wedge\ \alpha_{pr} + \delta \equiv_m \gamma_{pr} \wedge \\
z' - z - \beta_{sq} \leq \alpha_{pr} + \delta \leq z' - z - \alpha_{sq} & \quad \wedge\ z' - z - \alpha_{pr} - \delta \equiv_m \gamma_{sq} \wedge \\
\alpha_{ps} - z \leq \alpha_{pr} + \delta \leq \beta_{ps} - z & \quad \wedge\ z + \alpha_{pr} + \delta \equiv_m \gamma_{ps} \wedge \\
z' - \beta_{rq} \leq \alpha_{pr} + \delta \leq z' - \alpha_{rq} & \quad \wedge\ z' - \alpha_{pr} - \delta \equiv_m \gamma_{rq}.
\end{aligned}$$

We now highlight $z, z'$ and obtain

$$\begin{aligned}
\widetilde{\psi}_1(z, z') \equiv\ & \alpha_{pr} + \delta \leq \beta_{pr} & \wedge\ \alpha_{pr} + \delta \equiv_m \gamma_{pr} \wedge \\
& \alpha_{sq} + \alpha_{pr} + \delta \leq z' - z \leq \beta_{sq} + \alpha_{pr} + \delta & \wedge\ z' - z \equiv_m \gamma_{sq} + \alpha_{pr} + \delta \wedge \\
& \alpha_{ps} - \alpha_{pr} - \delta \leq z \leq \beta_{ps} - \alpha_{pr} - \delta & \wedge\ z \equiv_m \gamma_{ps} - \alpha_{pr} - \delta \wedge \\
& \alpha_{pr} + \delta + \alpha_{rq} \leq z' \leq \alpha_{pr} + \delta + \beta_{rq} & \wedge\ z' \equiv_m \gamma_{rq} + \alpha_{pr} - \delta.
\end{aligned} \tag{15}$$

- *Case 2:* $t = z' - z - \beta_{sq}$. We proceed similarly as in the previous case, and obtain

$$\begin{aligned}
\widetilde{\psi}_2(z, z') \equiv\ & \alpha_{pr} + \beta_{sq} - \delta \leq z' - z \leq \beta_{pr} + \beta_{sq} - \delta & \wedge\ z' - z \equiv_m \gamma_{pr} + \beta_{sq} - \delta \wedge \\
& \alpha_{sq} + \delta \leq \beta_{sq} & \wedge\ \beta_{sq} - \delta \equiv_m \gamma_{sq} \wedge \\
& \alpha_{ps} + \beta_{sq} - \delta \leq z' \leq \beta_{ps} + \beta_{sq} - \delta & \wedge\ z' \equiv_m \gamma_{ps} + \beta_{sq} - \delta \wedge \\
& \alpha_{rq} - \beta_{sq} + \delta \leq z \leq \beta_{rq} - \beta_{sq} + \delta & \wedge\ z \equiv_m \gamma_{rq} + \beta_{sq} + \delta.
\end{aligned} \tag{16}$$

- *Case 3:* $t = \alpha_{ps} - z$. We proceed similarly as in the previous case, and obtain

$$\begin{aligned}
\widetilde{\psi}_3(z, z') \equiv\ & \alpha_{ps} + \delta - \beta_{pr} \leq z \leq \alpha_{ps} + \delta - \alpha_{pr} & \wedge\ z \equiv_m \alpha_{ps} + \delta - \gamma_{pr} \wedge \\
& \alpha_{ps} + \delta + \alpha_{sq} \leq z' \leq \alpha_{ps} + \delta + \beta_{sq} & \wedge\ z' \equiv_m \gamma_{sq} + \alpha_{ps} + \delta \wedge \\
& \alpha_{ps} + \delta \leq \beta_{ps} & \wedge\ \alpha_{ps} + \delta \equiv_m \gamma_{ps} \wedge \\
& \alpha_{ps} + \delta + \alpha_{rq} \leq z' + z \leq \alpha_{ps} + \delta + \beta_{rq} & \wedge\ z' + z \equiv_m \gamma_{rq} + \alpha_{ps} + \delta.
\end{aligned} \tag{17}$$

- *Case 4:* $t = z' - \beta_{rq}$. We proceed similarly as in the previous case, and obtain

$$\begin{aligned}
\widetilde{\psi}_4(z, z') \equiv\ & \alpha_{pr} + \beta_{rq} - \delta \leq z' \leq \beta_{pr} + \beta_{rq} - \delta & \wedge\ z' \equiv_m \gamma_{pr} + \beta_{rq} - \delta \wedge \\
& \beta_{rq} - \delta - \beta_{sq} \leq z \leq \beta_{rq} - \delta - \alpha_{sq} & \wedge\ z \equiv_m \beta_{rq} - \delta - \gamma_{sq} \wedge \\
& \alpha_{ps} + \beta_{rq} - \delta \leq z' + z \leq \beta_{ps} + \beta_{rq} - \delta & \wedge\ z + z' \equiv_m \gamma_{ps} + \beta_{rq} - \delta \wedge \\
& \alpha_{rq} + \delta \leq \beta_{rq} & \wedge\ \beta_{rq} - \delta \equiv_m \gamma_{rq},
\end{aligned} \tag{18}$$

*Step 5: Putting the formula in DNF.* Altogether, the original formula $\psi$ is equivalent to the constraint

$$\widetilde{\psi} \equiv \psi_0 \wedge \bigvee_{\delta \in \{0, \ldots, m-1\}} \widetilde{\psi}_1 \vee \widetilde{\psi}_2 \vee \widetilde{\psi}_3 \vee \widetilde{\psi}_4. \tag{19}$$

If $\psi$ is $M$-bounded, then the constraints $\widetilde{\psi}_1, \ldots, \widetilde{\psi}_4$ (of constant size) are $3M$-bounded. Due to the disjunction over exponentially many moduli $\delta$'s, the size of $\widetilde{\psi}$ is larger than the size of $\psi$ by a multiplicative exponential factor. By direct inspection, $\widetilde{\psi}$ can be written in DNF where atomic propositions are of the form $z \in I$, $z' \in I$, $z' + z \in I$, $z' - z \in I$ where $I$ is either an interval $I \subseteq \mathbb{Z} \cup \{\infty, -\infty\}$ or a arithmetic progression of the form $I = a + m^*$ with $a \in \mathbb{Z}$. Each conjunct contains either tests of the form $z' + z \in I$ or $z' - z \in I$, *but not both.* This is crucial in order to obtain $\mathbb{Z}$-BVASS transitions. We combine conjunctions of constraints of the same kind, i.e., $z \in I \wedge z \in J$ is the same as $z \in (I \cap J)$. Therefore, the DNF representation of $\widetilde{\psi}$ can be put in the form $\widetilde{\psi}^+ \vee \widetilde{\psi}^-$, where (in the formulas below, $h$ ranges over an appropriate index set for the DNF representation)

$$\widetilde{\psi}^+ \equiv \bigvee_h z \in I_h^+ \wedge z' \in J_h^+ \wedge (z' + z) \in K_h^+ \quad \text{and} \quad \widetilde{\psi}^- \equiv \bigvee_h z \in I_h^- \wedge z' \in J_h^- \wedge (z' - z) \in K_h^-.$$

*Step 6: Writing the $\mathbb{Z}$-BVASS transitions.* For every conjunct $z \in I_h^- \wedge z' \in J_h^- \wedge (z' - z) \in K_h^-$ of $\widetilde{\psi}^-$ we have a transition

$$(\text{push-pop})^- \qquad X_{pqo_{14}} \leftarrow (X_{rso_{23}} \cap I_h^- + K_h^-) \cap J_h^-,$$

and for every conjunct $z \in I_h^+ \wedge z' \in J_h^+ \wedge (z' + z) \in K_h^+$ of $\widetilde{\psi}^+$ we have a transition

$$(\text{push-pop})^+ \qquad X_{pqo_{14}} \leftarrow (-(X_{rso_{23}} \cap I_h^+) + K_h^+) \cap J_h^+.$$

To complete the definition of the $\mathbb{Z}$-BVASS transitions, we show how to succinctly encode semilinear constants $I_h^-, \dots, K_h^+$. Arithmetic progressions $I = a + m^*$ are already in the required form. A right-open interval $I = [\alpha, \infty)$ with $\alpha \in \mathbb{Z}$ is encoded by the linear set $I = \alpha + 1^*$, a left-open interval $I = (-\infty, \beta]$ with $\beta \in \mathbb{Z}$ by $I = \beta + (-1)^*$, and a finite non-empty interval $I = [\alpha, \beta]$, with $\alpha, \beta \in \mathbb{Z}$ and $\alpha \leq \beta$, by $I = [\alpha, \infty) \cap (-\infty, \beta] = (\alpha + 1^*) \cap (\beta + (-1)^*)$.

*Example 6.6 (Example 6.5 continued).* We eliminate the existential quantifier from $x_q$ and $x'_q$ in (13) and obtain $\psi_2(z, z') \equiv z' - z = 2$, for which no further simplification is necessary. Applying the same procedure to the second conjunct of (9) we obtain $\psi'_2(z, z') \equiv z' - z = -2$. This yields the two following $\mathbb{Z}$-BVASS transitions

$$X_{qr} \leftarrow X_{qr} + 2 \quad \text{and} \quad X_{qr} \leftarrow X_{qr} - 2.$$

This completes the construction of the $\mathbb{Z}$-BVASS $\mathcal{B}$ and the proof of Lemma 6.1. The $\mathbb{Z}$-BVASS $\mathcal{B}$ has a number of nonterminals exponential in $l$ and constants of magnitude bounded by $3M$, where $M$ is the bound for the magnitude of constants of $\mathcal{P}$, and thus linearly bounded, as required.

Proof of Theorem 4.1 for integer dimension $k = 1$. By Theorem 5.2, the $\mathbb{Z}$-BVASS reachability sets $[\![X_{pqo}]\!]$ are semilinear and computable in ExpTime in the number of nonterminals and modulus $m$. Since $\mathcal{B}$ has exponentially many nonterminals and the modulus $m$ is the same as in $\mathcal{P}$, the $[\![X_{pqo}]\!]$'s are computable in 2-ExpTime complexity. Let $\psi_{[\![X_{pqo}]\!]}$ be the characteristic DNF quantifier-free formula of $[\![X_{pqo}]\!]$, which is a formula of Presburger arithmetic. Let $\psi_{pqo}(x, y) \equiv \psi_{[\![X_{pqo}]\!]}(y - x)$ be the constraint in discrete time logic s.t. $[\![\psi_{pqo}]\!] = \left\{(a, b) \in \mathbb{Z} \times \mathbb{Z} \,\middle|\, b - a \in [\![X_{pqo}]\!]\right\}$. We reconstruct the reachability relation of $\mathcal{P}$ as the following constraint:

$$\psi_{pq}(x_p^{\mathbb{Z}}, \vec{x}_p^{\mathbb{Q}}, x_q^{\mathbb{Z}}, \vec{x}_q^{\mathbb{Q}}) \equiv \bigvee_{o \in \text{orbits}(\mathbb{Q}^{2l})} \psi_{pqo}(x_p^{\mathbb{Z}}, x_q^{\mathbb{Z}}) \wedge \varphi_o(\vec{x}_p^{\mathbb{Q}}, \vec{x}_q^{\mathbb{Q}}). \tag{20}$$

The constraint above is computable in 2-ExpTime and can be turned in DNF by distributivity within the same complexity. By the correctness of the construction of the $\mathbb{Z}$-BVASS $\mathcal{B}$ stated in Lemma 6.1, $[\![\psi_{pq}]\!] = \rightsquigarrow_{pq}$, as required. □

## 6.2 Discrete dimension greater than one

We now treat the general case of Theorem 4.1 where configurations are in $Q \times \mathbb{Z}^k_{\text{SPAN} \leq K} \times \mathbb{Q}^l$ with integer dimension $k > 1$. We construct a new trPDA $\mathcal{Q}$ of integer dimension $k = 1$ by encoding all integer control registers except the first one into the control state. This is possible because $\mathbb{Z}^k_{\text{SPAN} \leq K}$ has bounded span, and thus once the value of any register is fixed, there are only finitely many possibilities for the other registers. Let $\Lambda = \{0\} \times \{-K, \dots, 0, \dots, K\}^{\{2, \dots, k\}}$. For every control state $p$ in $\mathcal{P}$ and displacement $\vec{\varepsilon} \in \Lambda$, we have a state $(p, \vec{\varepsilon})$ in $\mathcal{Q}$, which is initial, resp. final, depending on whether $p$ is initial, resp. final, in $\mathcal{P}$. For every $p, q \in Q$, $a \in A$, $\gamma \in \Gamma$, and displacements $\vec{\varepsilon}, \vec{\delta} \in \Lambda$ we have the following push constraint in $\mathcal{Q}$

$$\mathsf{push}_{(p,\vec{\varepsilon})a(q,\vec{\delta})\gamma}((x_p^{\vec{\mathbb{Z}}}, \vec{x}_p^{\mathbb{Q}}), \vec{x}_a, (x_q^{\vec{\mathbb{Z}}}, \vec{x}_q^{\mathbb{Q}}), \vec{x}_\gamma) \equiv \mathsf{push}_{paq\gamma}((\vec{x}_p^{\mathbb{Z}} + \vec{\varepsilon}, \vec{x}_p^{\mathbb{Q}}), \vec{x}_a, (\vec{x}_q^{\mathbb{Z}} + \vec{\delta}, \vec{x}_q^{\mathbb{Q}}), \vec{x}_\gamma),$$

where $\mathsf{push}_{paq\gamma}$ is the corresponding push constraint of $\mathcal{P}$, $\vec{x}_p^{\mathbb{Z}}$ abbreviates $(x_p^{\mathbb{Z}}, \ldots, x_p^{\mathbb{Z}})$, and similarly for $\vec{x}_q^{\mathbb{Z}}$. Pop constraints $\mathsf{pop}_{(p,\vec{\varepsilon})a(q,\vec{\delta})\gamma}$ are definite similarly.

Proof of Theorem 4.1 for integer dimension $k > 1$. Let $\mathcal{Q}$ be the trPDA as constructed above. Since in $\mathcal{Q}$ the discrete part is one dimensional, by the previous section we can build a constraint $\psi_{(p,\vec{\varepsilon}),(q,\vec{\delta})}$ expressing its reachability relation $\rightsquigarrow_{(p,\vec{\varepsilon}),(q,\vec{\delta})} = [\![\psi_{(p,\vec{\varepsilon}),(q,\vec{\delta})}]\!]$. Notice that $\mathcal{Q}$ has exponentially more control states than $\mathcal{P}$, because the bound on the span $K$ is encoded in binary, and thus it may seem that it takes triply exponential time to build $\psi_{(p,\vec{\varepsilon}),(q,\vec{\delta})}$. However, by Lemma 6.1, the size of the $\mathbb{Z}$-BVASS that leads to the construction of $\psi_{(p,\vec{\varepsilon}),(q,\vec{\delta})}$ is quadratic w.r.t. the number of control states of $\mathcal{Q}$, and thus of combined singly exponential size. Consequently, $\psi_{(p,\vec{\varepsilon}),(q,\vec{\delta})}$ is still constructible in doubly exponential time. The following constraint $\psi_{pq}$ characterises the reachability relation $\rightsquigarrow_{pq} = [\![\psi_{pq}]\!]$ of $\mathcal{P}$:

$$\psi_{pq}(\vec{x}_p, \vec{x}_q) \equiv \bigvee_{\vec{\varepsilon},\vec{\delta}\in\Lambda} \psi_{(p,\vec{\varepsilon}),(q,\vec{\delta})}(x_{p,1}^{\mathbb{Z}}, \vec{x}_p^{\mathbb{Q}}, x_{q,1}^{\mathbb{Z}}, \vec{x}_q^{\mathbb{Q}}) \wedge \vec{x}_p^{\mathbb{Z}} = \vec{x}_{p,1}^{\mathbb{Z}} + \vec{\varepsilon} \wedge \vec{x}_q^{\mathbb{Z}} = \vec{x}_{q,1}^{\mathbb{Z}} + \vec{\delta},$$

where $\vec{x}_p = (\vec{x}_p^{\mathbb{Z}}, \vec{x}_p^{\mathbb{Q}})$, $\vec{x}_p^{\mathbb{Z}} = (x_{p,1}^{\mathbb{Z}}, \ldots, x_{p,k}^{\mathbb{Z}})$, $\vec{x}_{p,1}^{\mathbb{Z}} = (x_{p,1}^{\mathbb{Z}}, \ldots, x_{p,1}^{\mathbb{Z}})$, and similarly for $\vec{x}_q, \vec{x}_q^{\mathbb{Z}}, \vec{x}_{q,1}^{\mathbb{Z}}$. □

### 6.3 Reachability in monotonic trPDA

A trPDA is *monotonic* if, whenever $(\vec{u}^{\mathbb{Z}}, \vec{u}^{\mathbb{Q}}) \rightsquigarrow_{pq} (\vec{v}^{\mathbb{Z}}, \vec{v}^{\mathbb{Q}})$ with $\vec{u}^{\mathbb{Z}}, \vec{v}^{\mathbb{Z}} \in \mathbb{Z}^k$ and $\vec{u}^{\mathbb{Q}}, \vec{v}^{\mathbb{Q}} \in \mathbb{Q}^l$, then $\vec{u}^{\mathbb{Z}} \leq \vec{v}^{\mathbb{Z}}$, for every pair of control states $p, q$. In other words, integer registers are non-decreasing when going from one state to another. This is a significant restriction on the model which captures the idea of *monotonic time* (of integer timestamps). Additionally, it allows for substantial technical simplifications in the analysis and improved complexity bounds.

Theorem 6.7. *For a monotonic trPDA and control states $p, q$ thereof, one can compute in exponential time an existential formula of hybrid logic $\psi_{pq}(\vec{x}_p, \vec{x}_q)$ s.t. $[\![\psi_{pq}]\!] = \rightsquigarrow_{pq}$.*

As a corollary of the construction in the proof of the theorem above, we obtain the following improved upper-bound for the non-emptiness problem under the monotonicity assumption.

Corollary 6.8. *The non-emptiness problem of monotonic trPDA is decidable in* ExpTime.

In order to prove Theorem 6.7, we adapt the construction for the case of integer dimension $k = 1$ of Sec. 6.1 to monotone trPDA; the general case $k > 1$ is handled as in Sec. 6.2, and thus we omit it. Instead of constructing a $\mathbb{Z}$-BVASS, we construct a context-free grammar (CFG) $\mathcal{G}$ over a singleton alphabet $\Sigma = \{\checkmark\}$ containing a single symbol $\checkmark$ denoting the integral amount of time elapsed. The grammar $\mathcal{G}$ has exponentially many non-terminal symbols of the form $X_{pqo}$. By $[\![X_{pqo}]\!] \subseteq \mathbb{N}$ we denote the number of $\checkmark$'s (length) of those words accepted by $X_{pqo}$.

Lemma 6.9. *For every monotonic trPDA $\mathcal{P}$ we can construct a CFG $\mathcal{G}$ with an exponential blow-up in the number of control states s.t. for control states $p, q$ of $\mathcal{P}$, orbit $o$ in $\mathbb{Q}^{2l}$, integers $a, b \in \mathbb{Z}$ and rationals $\vec{a}, \vec{b} \in \mathbb{Q}^l$ s.t. $(\vec{a}, \vec{b}) \in o$,*

$$(a, \vec{a}) \rightsquigarrow_{pq} (b, \vec{b}) \quad \textit{iff} \quad b - a \geq 0 \wedge (b - a) \in [\![X_{pqo}]\!].$$

Since non-emptiness of a context-free grammar can be decided in PTIME, Lemma 6.9 immediately implies Corollary 6.8, and, together with Lemma 2.2, it implies Theorem 6.7. In the following we construct the grammar $\mathcal{G}$. The rules for the base case and the transitive case are the same as in Sec. 6, with some cosmetic changes to adapt them to CFG:

$$\begin{aligned} &\text{(base)} && X_{ppo} \leftarrow \varepsilon. \\ &\text{(transitivity)} && X_{pqo_{13}} \leftarrow X_{pro_{12}} \cdot X_{rqo_{23}} \end{aligned}$$

For the push-pop transitions, we follow step-by-step the transformation of Sec. 6.

*Step 0: Transformation in DNF.* By monotonicity, Eq. (8) is replaced by

$$\psi(z, z') \equiv \exists(x_p \le x_r \le x_s \le x_q) \cdot z = x_s - x_r \wedge z' = x_q - x_p \wedge \varphi^{\mathbb{Z}}(x_p, x_r, x_s, x_q). \tag{21}$$

*Step 1: Expansion.* Thanks to the monotonicity condition on variables $x_p \le x_r \le x_s \le x_q$, $\varphi^{\mathbb{Z}}$ is now a conjunction of atomic propositions either of the form $x_q - x_p \in [\alpha_{pq}, \beta_{pq}]$ with $\alpha_{pq} \le \beta_{pq}$, or $x_q - x_p \equiv_m \gamma_{pq}$, where now all constants $\alpha_{pq}, \gamma_{pq} \in \mathbb{N}$ and $\beta_{pq} \in \mathbb{N} \cup \{\infty\}$ are nonnegative; similarly for the other combinations of indices $p, r, s, q$. Thus, (11) becomes (where $\psi'$ is as in (11))

$$\psi(z, z') \equiv \exists(x_p \le x_r \le x_s \le x_q) \cdot \psi'. \tag{22}$$

*Step 2: Eliminate $x_s$ and $x_q$.* The formula $\psi_1(z, z')$ from (12) is unchanged except that the prefix of quantifiers is $\exists(x_p \le x_r)$.

*Step 3: Eliminate $x_r$ and $x_p$.* The formula $\psi_2(z, z')$ from (14) and the definition of $\psi_{\delta,t}$ therein are unchanged.

*Step 4: Simplify $\psi_2$.* Cases 1 and 2 are unchanged, and thus $\widetilde{\psi}_1$ is the same as from (15) and $\widetilde{\psi}_2$ from (16). For $\widetilde{\psi}_3, \widetilde{\psi}_4$ we perform the following modifications.

- *Case 3: $l = \alpha_{ps} - z$.* Since now $z, z' \ge 0$, formula $\widetilde{\psi}_3$ is modified by expanding the last constraint (17) on $z + z'$ as a finite disjunction on constraints on $z$ and $z'$ separately, using the fact that $\alpha \le z + z' \le \beta$ holds if, and only if, $\bigvee_{0 \le h \le \alpha} h \le z \wedge \alpha - h \le z'$ and $\bigvee_{0 \le h \le \beta} z \le h \wedge z' \le \beta - h$. For the modulo constraint, we have $z + z' \equiv_m \gamma$ iff $\bigvee_{0 \le h < m} z \equiv_m h \wedge z' \equiv_m \gamma - h$ (which holds without any assumption on $z, z'$). By instantiating $\alpha = \alpha_{ps} + \delta + \alpha_{rq}$, $\beta = \gamma_{rq} + \alpha_{ps} + \delta$, and $\gamma = \gamma_{rq} + \alpha_{ps} + \delta$, we obtain

$$\begin{aligned} \widetilde{\psi}_3(z, z') \equiv\ & \alpha_{ps} + \delta - \beta_{pr} \le z \le \alpha_{ps} + \delta - \alpha_{pr} && \wedge\ z \equiv_m \alpha_{ps} + \delta - \gamma_{pr}\ \wedge \\ & \alpha_{ps} + \delta + \alpha_{sq} \le z' \le \alpha_{ps} + \delta + \beta_{sq} && \wedge\ z' \equiv_m \gamma_{sq} + \alpha_{ps} + \delta\ \wedge \\ & \alpha_{ps} + \delta \le \beta_{ps} && \wedge\ \alpha_{ps} + \delta \equiv_m \gamma_{ps}\ \wedge \\ & \bigvee_{0 \le h \le \alpha_{ps} + \delta + \alpha_{rq}} h \le z' \ \wedge\ \alpha_{ps} + \delta + \alpha_{rq} - h \le z\ \wedge \\ & \bigvee_{0 \le h \le \alpha_{ps} + \delta + \beta_{rq}} z' \le h\ \wedge\ z \le \alpha_{ps} + \delta + \beta_{rq} - h\ \wedge \\ & \bigvee_{0 \le h < m} z' \equiv_m h\ \wedge\ z \equiv_m \gamma_{rq} + \alpha_{ps} + \delta - h. \end{aligned}$$

- *Case 4: $l = z' - \beta_{rq}$.* Similarly as in the previous case, we expand (18) as

$$\begin{aligned}\widetilde{\psi}_4(z,z') \;\equiv\;& \alpha_{pr} + \beta_{rq} - \delta \le z' \le \beta_{pr} + \beta_{rq} - \delta \;\wedge\; z' \equiv_m \gamma_{pr} + \beta_{rq} - \delta \;\wedge \\ & \beta_{rq} - \delta - \beta_{sq} \le z \le \beta_{rq} - \delta - \alpha_{sq} \;\wedge\; z \equiv_m \beta_{rq} - \delta - \gamma_{sq} \;\wedge \\ & \bigvee_{0 \le h \le \alpha_{ps} + \beta_{rq} - \delta} h \le z' \;\wedge\; \alpha_{ps} + \beta_{rq} - \delta - h \le z \;\wedge \\ & \bigvee_{0 \le h \le \beta_{ps} + \beta_{rq} - \delta} z' \le h \;\wedge\; z \le \beta_{ps} + \beta_{rq} - \delta - h \;\wedge \\ & \bigvee_{0 \le h < m} z' \equiv_m h \;\wedge\; z \equiv_m \gamma_{ps} + \beta_{rq} - \delta - h \;\wedge \\ & \alpha_{rq} + \delta \le \beta_{rq} \;\wedge\; \beta_{rq} - \delta \equiv_m \gamma_{rq}.\end{aligned}$$

*Step 5: Putting the formula in DNF.* We obtain a formula $\widetilde{\psi}$ in DNF as in (19), with the further restriction that now, thanks to the simplified form of $\widetilde{\psi}_3, \widetilde{\psi}_4$ above, we only have atomic constraints of the form $z \in I$, $z' \in I$, or $z' - z \in I$ with $I \subseteq \mathbb{N}$ either an interval or an arithmetic progression. Under the assumption of monotonic time, constraints of the form $z' + z \in I$ do not appear anymore. Consequently, we obtain the following DNF representation for $\widetilde{\psi}$

$$\widetilde{\psi}(z,z') \;\equiv\; \bigvee_h \; z \in I_h \wedge z' \in J_h \wedge (z' - z) \in K_h.$$

If $\psi$ is $M$-bounded, then $\psi$ is $3M$-bounded.

*Step 6: Writing the grammar productions.* The form above yields productions

$$\text{(push-pop)} \qquad X_{pqo_{14}} \leftarrow ((X_{rso_{23}} \cap \widetilde{I}_h) \cdot \widetilde{K}_h) \cap \widetilde{J}_h.$$

where $\widetilde{I}_h = \left\{ \checkmark^n \;\middle|\; n \in I_h \right\}$, $\widetilde{J}_h = \left\{ \checkmark^n \;\middle|\; n \in J_h \right\}$, and $\widetilde{K}_h = \left\{ \checkmark^n \;\middle|\; n \in K_h \right\}$. The intersections with the regular languages above can be eliminated by constructing a finite automaton $\mathcal{A}$ of size $O(M)$ (singly exponential since constants are encoded in binary) that counts the number of $\checkmark$'s up to threshold $3M$ and keeps track of its value modulo $m \le M$.

## 7 SEMILINEARITY OF $\mathbb{Z}$-BVASS REACHABILITY SETS

In this section we prove Theorem 5.2. To this end, we introduce a convenient normal form, show how to transform a $\mathbb{Z}$-BVASS to one in normal form (Sec. 7.1), and compute reachability sets for $\mathbb{Z}$-BVASS in normal form (Sec. 7.2). A $\mathbb{Z}$-BVASS is in *normal form* if variables $\mathrm{Var} = \{X_1\} \cup \mathrm{Var}_+ \cup \mathrm{Var}_-$ are partitioned into a singleton containing a distinguished *unit variable* $\{X_1\}$, *addition variables* $\mathrm{Var}_+$, and *subtraction variables* $\mathrm{Var}_-$; terms are of the following three kinds

$$t ::= \{1\} \;\mid (X + Y) \cap \mathbb{N} \mid (X - Y) \cap \mathbb{N};$$

there is precisely one transition $X_1 \leftarrow \{1\}$ with the unit variable $X_1$ on the l.h.s., for every addition $X \leftarrow (Y + Z) \cap \mathbb{N}$, $X \in \mathrm{Var}_+$, and for every subtraction $X \leftarrow (Y - Z) \cap \mathbb{N}$, $X \in \mathrm{Var}_-$. Note that reachability sets of $\mathbb{Z}$-BVASS in normal form contain only nonnegative integers $[\![X]\!] \subseteq \mathbb{N}$.

### 7.1 From $\mathbb{Z}$-BVASS to $\mathbb{Z}$-BVASS in normal form

Lemma 7.1. *For every $\mathbb{Z}$-BVASS $\mathcal{B}$, we can construct a $\mathbb{Z}$-BVASS in normal form, containing two variables $X^+, X^-$ for every variable $X$ in $\mathcal{B}$, s.t. $[\![X]\!] = [\![X^+]\!] \cup (-[\![X^-]\!])$. The construction takes time polynomial in the number of nonterminals and exponential in the binary encoding of constants of $\mathcal{B}$.*

From the lemma above, if $\varphi_X$ is a constraint encoding the reachability set $[\![\varphi_X]\!] = [\![X]\!]$, then $\varphi_X(x)$ can be taken to be $\varphi_X(x) \equiv \varphi_{X^+}(x) \vee \varphi_{X^-}(-x)$.

Proof. The construction consists of five steps.

*Step 1: Short terms.* By introducing new variables and transitions as necessary, we can readily assume that transitions are of the form $X \leftarrow t$, where $t$ is constructed according to the following grammar (where $S$ is a semilinear set):

$$t ::= S \mid (X + Y) \cap S \mid (X - Y) \cap S.$$

*Step 2: Linear constants.* Transitions $X \leftarrow S$ for a semilinear constant $S$ can be replaced with $X \leftarrow X_S + X_0$, where the new nonterminals $X_S$ s.t. $[\![X_S]\!] = S$ and $X_0$ with $[\![X_0]\!] = \{0\}$, and their associated transitions are built according to Example 5.1 (with a polynomial increase of the number of nonterminals and transitions). Consequently, the only transitions of the form $X \leftarrow S$ are now $X \leftarrow \{1\}$. For a semilinear set $S = L_1 \cup \cdots \cup L_n$, where the $L_i$'s are linear, we replace a transition $X \leftarrow (Y \pm Z) \cap S$ with transitions $X \leftarrow (Y \pm Z) \cap L_1, \ldots, X \leftarrow (Y \pm Z) \cap L_n$. This yields the fragment (where $L$ is a linear set)

$$t ::= \{1\} \mid (X + Y) \cap L \mid (X - Y) \cap L.$$

*Step 3: Intersection with* $\pm\mathbb{N}$. Thanks to Lemma 2.1, linear constants $L$ can be assumed to be of the simple form of arithmetic progressions $L = b + p^*$. Since $(X + Y) \cap (b + p^*)$ is the same as $((X + Y - b) \cap p^*) + b$, we can assume that terms $t$ occurring in transitions are already in the form

$$t ::= \{1\} \mid (X + Y) \cap p^* \mid (X - Y) \cap p^*.$$

If $p = 0$, then $(X+Y)\cap p^*$ is the same as $(X+Y)\cap\{0\}$, which can be expressed as $(X_0-((X+Y)\cap\mathbb{N}))\cap\mathbb{N}$; similarly for $(X - Y) \cap p^*$. Otherwise, assume that all periods are $> 0$, and let $p^\bullet$ be their least common multiplier. For each variable $X$, we introduce new variables, $X_0, \ldots, X_{p^\bullet-1}$ s.t. $[\![X_i]\!] = \left\{n \in [\![X]\!] \;\middle|\; n \equiv_{p^\bullet} i\right\}$, and thus $[\![X]\!] = \bigcup_{0 \le i < p^\bullet} [\![X_i]\!]$. For every transition $X \leftarrow (Y \pm Z) \cap p^*$, and remainders $i, j, k \in \{0, 1, \ldots, p^\bullet - 1\}$ s.t. $j \pm k \equiv_{p^\bullet} i$ and $j \pm k$ is divisible by $p$, there is a transition (where $\mathsf{sign}(p) = \frac{p}{|p|}$ is the sign of $p \neq 0$)

$$X_i \leftarrow (Y_j \pm Z_k) \cap \mathsf{sign}(p) \cdot \mathbb{N}.$$

Summarising, by introducing exponentially many nonterminals $X_i$'s (in the binary encoding of periods $p$'s), we obtain transitions of the form

$$t ::= \{1\} \mid (X + Y) \cap (\pm\mathbb{N}) \mid (X - Y) \cap (\pm\mathbb{N}).$$

*Step 4: Intersection with* $\mathbb{N}$. For each variable $X$ we introduce two non-negative variables, $X^+$ and $X^-$, which keep track of the positive and negative part of $X$, respectively, i.e., $[\![X^+]\!] = [\![X]\!] \cap \mathbb{N}$ and $[\![X^-]\!] = [\![-X]\!] \cap \mathbb{N}$. A transition $X \leftarrow (Y + Z) \cap \mathbb{N}$ generates transitions

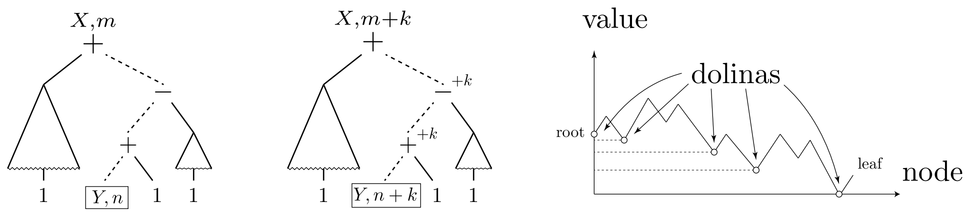
$$X^+ \leftarrow (Y^+ + Z^+) \cap \mathbb{N} \qquad X^+ \leftarrow (Y^+ - Z^-) \cap \mathbb{N} \qquad X^+ \leftarrow (Z^+ - Y^-) \cap \mathbb{N},$$

and similarly a transition $X \leftarrow (Y + Z) \cap (-\mathbb{N})$ generates transitions

$$X^- \leftarrow (Y^- + Z^-) \cap \mathbb{N} \qquad X^- \leftarrow (Y^- - Z^+) \cap \mathbb{N} \qquad X^- \leftarrow (Z^- - Y^+) \cap \mathbb{N}.$$

The case $X \leftarrow (Y - Z) \cap (\pm\mathbb{N})$ is analogous. We thus obtain only intersection with $\mathbb{N}$:

$$t ::= \{1\} \mid (X + Y) \cap \mathbb{N} \mid (X - Y) \cap \mathbb{N}.$$




(a) A context $s$ is a run with a hole.

(b) Increment $s + k$ of context $s$.

(c) Dolinas on one of the branches.

Fig. 1. Contexts and dolinas.

*Step 5: Normal form.* We replace every variable $X$ with an addition $X_+ \in \mathsf{Var}_+$ and a subtraction $X_- \in \mathsf{Var}_-$ copy thereof. There is a distinguished unit variable $X_1$ with transition $X_1 \longleftarrow \{1\}$, and an additional subtraction variable $X_0 \in \mathsf{Var}_-$ with transition $X_0 \longleftarrow (X_1 - X_1) \cap \mathbb{N}$. Every other unit transition $X \longleftarrow \{1\}$ with $X \neq X_1$, is replaced by $X_+ \longleftarrow (X_0 + X_1) \cap \mathbb{N}$. An addition transition $X \longleftarrow (Y + Z) \cap \mathbb{N}$ is replaced by $X_+ \longleftarrow (Y_+ + Z_+) \cap \mathbb{N}$ and a subtraction $X \longleftarrow (Y - Z) \cap \mathbb{N}$ by $X_- \longleftarrow (Y_+ - Z_+) \cap \mathbb{N}$. The values of subtraction variables can be transfered to addition ones with extra transitions $X_+ \longleftarrow (X_- + X_0) \cap \mathbb{N}$. □

### 7.2 Semilinearity of reachability sets of $\mathbb{Z}$-BVASS in normal form

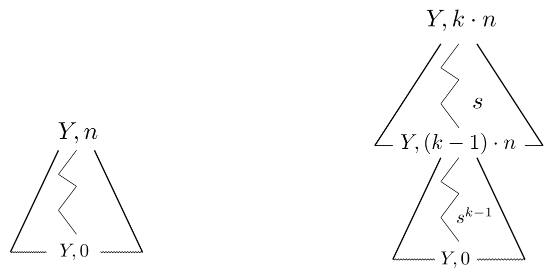
Fix a $\mathbb{Z}$-BVASS $\mathcal{B}$ in normal form with $|\mathsf{Var}| = K$ variables. Let Var be the set of variables. Thanks to the normal form, there is a unique variable $X_1$ with transition $X_1 \longleftarrow \{1\}$ of the first kind, and all other variables are partitioned into *addition variables* $X$ with transitions of the form $X \longleftarrow Y + Z$ and *subtraction variables* $X$ with transitions of the form $X \longleftarrow Y - Z$; for ease of notation, we do not write the intersection with $\mathbb{N}$, with the understanding that the value of a variable never gets negative. A *configuration* is a pair $(X, n)$ where $X$ is a variable and $n \in \mathbb{N}$. A *run* is a finite, rooted, binary, ordered tree labelled with configurations s.t.:

- Every internal node $u : (X, n)$ has a left child $u_l : (X_l, n_l)$ and a right child $u_r : (X_r, n_r)$. If $X$ is an addition variable, then there exists a rule $X \longleftarrow X_l + X_r$ and $n = n_l + n_r$. Otherwise, $X$ is a subtraction variable and there exists a rule $X \longleftarrow X_l - X_r$ and $n = n_l - n_r \geq 0$. In the latter case, $u_l$ is called the *minuend* and $u_r$ the *subtrahend* node.
- Every leaf is labeled by $(X_1, 1)$.

A $(X, m)$*-run* is a run whose root is labelled with $(X, m)$; sometimes we also speak of $X$-run, or $m$-run. The reachability set $[\![X]\!]$ thus equals the set of values $m$ s.t. there exists a $(X, m)$-run. A run is $M$*-bounded*, for a bound $M \in \mathbb{N}$, if all labels thereof are of the form $(X, m)$ with $m \leq M$.

A *branch* of a run is a path starting at the root and ending in a leaf. A *positive* branch is one that always turns left on subtraction nodes (i.e., it goes to the minuend subtree); a node is *positive* if it belongs to a positive branch. The *support* of a run is the set of variables $V \subseteq \mathsf{Var}$ that appear among positive nodes therein. Let $[\![X]\!]_V$ be the subset of the reachability set consisting of those values $m$ which can be reached by some $(X, m)$-run with support $V$; clearly, $[\![X]\!]_V \subseteq [\![X]\!]$ for every set of variables $V$, and $[\![X]\!] = \bigcup_{V \subseteq \mathsf{Var}} [\![X]\!]_V$.

A $(X, m)$*-context with hole* $(Y, n)$ is a $(X, m)$-run except that there exists precisely one positive leaf node, called *hole*, labelled with $(Y, n)$ instead of $(X_1, 1)$; all other rules regarding internal nodes apply; c.f. Fig. 1(a). For $s$ a $(X, m)$-context with hole $(Y, n)$, and $k \in \mathbb{Z}$, we denote by $s + k$ the $(X, m + k)$-context with hole $(Y, n + k)$ obtained from $s$ by increasing by $k$ the value of the hole and




(a) A pumpable $(Y, n)$-context $s$.

(b) Pumpable $(Y, k \cdot n)$-context $s^k$.

Fig. 2. Pumpable contexts.

all its ancestors, assuming that this operation is defined; c.f. Fig. 1(b). A $(X, m)$-context $s$ with hole $(Y, n)$ is *compatible with* a $(Z, k)$-run $t$ if $Y = Z$ and $s' = s + (k - n)$ is defined; when this holds, their *composition* $st$ is the $(X, m + (k - n))$-run obtained by replacing the hole in $s'$ with $t$. Composition for contexts is defined analogously.

For a tree $t$ and a node $u$ thereof, let $t_u$ denote the subtree of $t$ rooted at $u$. If $t$ is a $(X, m)$-run and $v : (Y, n)$ is a positive node thereof, then $t[v \mapsto \square]$ is the $(X, m)$-context with hole $(Y, n)$ obtained by replacing $t_v$ with a hole labelled by $(Y, n)$. For a run (or context) $t$ and a run $s$, together with a positive node $v$ thereof, we denote by $s[v \mapsto t] := s[v \mapsto \square]t$ the run (or context) obtained by replacing $s_v$ by $t$. A *dolina* is a positive node $u$ (in a run or in a context) whose value is strictly smaller than the value of any ancestor; c.f. Fig. 1(c). If a hole of a context $s$ is a dolina of value $m$, then $s - m$ is defined. A $(Y, n)$-context with hole $(Z, o)$ is *pumpable* if $Z = Y$ and $o = 0$; cf. Fig. 2(a). When $s$ is a pumpable context, let $s^0$ be the context consisting of just a $(Y, 0)$-hole, and let $s^k = ss^{k-1}$ for every $k \geq 1$; then, $s^k$ is a pumpable $(Y, k \cdot n)$-context; cf. Fig. 2(b).

The *dolina complexity* of a run is the maximum number of dolinas on the same branch. The following lemma shows that reachability sets are bounded semilinear; however, no method is provided yet as to compute a representation thereof.

Lemma 7.2. (1) *Every run of value $> 2^{K^2}$ has dolina complexity $> K^2$.*

(2) *Every run $t$ of dolina complexity $> K^2$ contains two $2^{K^2}$-bounded dolinas $u : Y, v : Y$ on the same branch s.t. $t_u, t_v$ have the same support.*

(3) *The reachability set $[\![X]\!]_V$ is a $2^{K^2}$-bounded hybrid linear set of the form $A + B^*$.*

Proof. (1) Construct a positive branch $\pi = v_1 \cdots v_k$ of values $n_1, \ldots, n_k$ starting from the root, which on addition nodes chooses the child of larger value. If $v_i$ is an addition node, a child $v_{i+1}$ is selected s.t. $n_{i+1} \geq \frac{1}{2} n_i$. Consider now the subsequence of $\pi$ consisting of all dolinas $v_{i_1} \cdots v_{i_m}$ (with $i_1 = 1$) of values $n_{i_1} > \cdots > n_{i_m}$. Since a dolina $v_{i_j}$ is necessarily a child of an addition node $v_{i_j - 1}$, $n_{i_j} \geq \frac{1}{2} n_{i_j - 1}$. Since there is no other dolina between $v_{i_{j-1}}$ and $v_{i_j}$, $n_{i_j - 1} \geq n_{i_{j-1}}$, and thus $n_{i_j} \geq \frac{1}{2} n_{i_{j-1}}$. Since the first dolina has value $n_{i_1} \geq 2^{K^2}$, there are at least $K^2$ dolinas on $\pi$.

(2) Assume that $t$ has dolina complexity $> K^2$. There exists a sequence $v_1 \cdots v_m$ of $m > K^2$ dolinas on the same positive branch. Since the last dolina $v_m$ has value $\leq 1$, the last $K^2$ dolinas have values $\leq 2^{K^2}$. Since $t_{v_i}$ is a subtree of $t_{v_{i-1}}$, the sequence of dolinas induces a decreasing chain of supports, and thus there are at most $K$ different supports. Finally, each dolina is labelled by a nonterminal $Y$, of which there are at most $K$. By the pigeonhole principle, there

are two $2^{K^2}$-bounded dolinas $v_i : Y, v_j : Y$ labelled by the same nonterminal $Y$, which are roots of subtrees $t_{v_i}$, resp., $t_{v_j}$, with the same support.

(3) Let $A$ be the set of those reachable $a \in [\![X]\!]_V$ witnessed by some $(X, a)$-run $t_a$ with support $V$ and $K^2$-bounded dolina complexity. By the first point, $a \leq 2^{K^2}$, and thus $A$ is $2^{K^2}$-bounded. For $a \in [\![X]\!]_V \setminus A$, $t_a$ necessarily has dolina complexity $> K^2$, and thus by the previous point contains two dolinas $u : (Y, m), v : (Y, n)$ of small values $n < m \leq 2^{K^2}$ on the same branch s.t. the subtrees $t_u, t_v$ have the same support. Let $b = m - n$, and thus $0 < b \leq 2^{K^2}$. We decompose $t_a$ into a run $t_{a-b}$ and context $t_b$. Let $t_b := t_u[v \mapsto \square] - n$ be the pumpable $b$-context obtained from the subtree $t_u$ rooted at the first dolina $u$ by making a hole at the second dolina $v$, and let $t_{a-b} := t[u \mapsto t_v]$ be the $(a - b)$-run obtained from the run $t$ by replacing the subtree $t_u$ rooted at first dolina $u$ with the subtree $t_v$ rooted at the second dolina $v$. The support of $t_b$ is included in $V$, and that of $t_{a-b}$ is exactly $V$; thus $(a - b) \in [\![X]\!]_V$. Let $B$ be the set of all $2^{K^2}$-bounded periods $b$'s obtained in this way. By iterating the reasoning above, every $a \in [\![X]\!]_V$ belongs to $A + B^*$. On the other hand, if $c = a + k_1 b_1 + \cdots + k_n b_n$ with $a \in A$ and $b_1, \ldots, b_n \in B$, then there exists an $a$-run $t_a$ of support $V$, and, for every $1 \leq i \leq n$, pumpable $(Y_i, b_i)$-contexts $t_{b_i}$ of supports included in $V$. Since $t_a$ has support $V$, for every variable $Y_i$ there exists a positive $Y_i$-node $u_i$ in $t_a$. We construct a $c$-run by inserting sufficiently many copies of the $t_{b_i}$'s in suitable nodes of $t_a$. Formally, for every $0 \leq i \leq n$, we construct a $c_i$-run $t_i$ of support $V$, where $c_i = a + k_1 b_1 + \cdots + k_i b_i$. Initially, $t_0$ is the $a$-run $t_a$ of support $V$. Assume $t_{i-1}$ is a $c_{i-1}$-run of support $V$. We define $t_i$ as $t_{i-1}[u_i \mapsto t_{b_i}^{k_i}]t_{u_i}$, which is thus a $c_i$-run of support $V$. Take $t_n$ as the sought $c$-run of support $V$. Consequently, $[\![X]\!]_V = A + B^*$, as required. □

The following lemma allows us to bound the value of every subtrahend. In the rest of this section, we will use the following constant

$$L = 2^{K^2} + 2^{3K^2}.$$

Lemma 7.3. *Let $n \in [\![X]\!]$. There exists an $(X, n)$-run s.t. all subtrahends are $L$-bounded.*

Proof. Let $t$ be an $(X, n)$-run with $X$ a subtraction variable, whose minuend subtree has label $(X_l, n_l)$ and subtrahend subtree has label $(X_r, n_r)$, and thus $n = n_l - n_r \geq 0$. Towards a contradiction, assume that the size of $t$ (in terms of number of nodes) is minimal amogst all $(X, n)$-runs, and let $n_r > L$ (and thus $n_l > L$). Let $V$ be the support of $t$, let $V_l \subseteq V$ be the support of its left subtree $t_l$, and similarly $V_r \subseteq V$ for the support of its right subtree $t_r$. By the last point of Lemma 7.2, $[\![X_l]\!]_{V_l}, [\![X_r]\!]_{V_r}$ are both $2^{K^2}$-bounded hybrid linear sets of the form, $A_l + B_l^*$, resp., $A_r + B_r^*$. The left value $n_l \in [\![X_l]\!]_{V_l}$ is of the form $n_l = a_l + k_{l1} b_{l1} + \cdots + k_{lm} b_{lm}$ with $a_l \in A_l$, $b_{l1}, \ldots, b_{lm} \in B_l$, and $k_{l1}, \ldots, k_{lm} \in \mathbb{N}$. The periods $b_{li}$'s are $2^{K^2}$-bounded and in particular $m \leq 2^{K^2}$. Since $n_l > L$, there exists a period $b_{li} \in B_l$ s.t. its multiplicity $k_{li}$ is $> 2^{K^2}$. Similarly, $n_r = a_r + k_{r1} b_{r1} + \cdots + k_{rm} b_{rm} \in [\![X_r]\!]_{V_r}$ for $a_r \in A_r$, $b_{r1}, \ldots, b_{rm} \in B_r$, and $k_{r1}, \ldots, k_{rm} \in \mathbb{N}$, and there exists $b_{rj} \in B_r$ with $k_{rj} > 2^{K^2}$. Since $b_{li}, b_{rj}$ are $2^{K^2}$-bounded, $k_{li} > b_{rj}$ and $k_{rj} > b_{li}$. Take the smaller value $n_l' = n_l - b_{rj} b_{li} \in [\![X_l]\!]_{V_l}$ obtained by removing $b_{rj}$ copies of period $b_{li}$, and similarly $n_r' = n_r - b_{li} b_{rj} \in [\![X_r]\!]_{V_r}$ by removing $b_{li}$ copies of period $b_{rj}$. Clearly, $n = n_l' - n_r'$. Moreover, by applying the construction in the proof of the last point of Lemma 7.2, the witnessing runs $t_l', t_r'$ for $n_l'$, resp., $n_r'$ can be constructed to be subtrees of $t_l$, resp., $t_r$. This yields a witness for $(X, n)$ of smaller size, which is a contradiction. □

Theorem 7.4. *Reachability for $\mathbb{Z}$-BVASS in normal form is decidable in* ExpTime.

Proof. We reduce to reachability for ordinary one-dimensional BVASS (i.e., without minus operations) with constants encoded in unary, which is solvable in PTime [16]. For $\mathbb{Z}$-BVASS, it suffices to decide whether $0 \in [\![X]\!]$: In order to decide $n \in [\![X]\!]$, we add a new nonterminal $\hat{X}$ with rule $\hat{X} \leftarrow X - X_n$, where $X_n$ is s.t. $[\![X_n]\!] = \{n\}$ and can be constructed according to the technique of Example 5.1, and we ask the equivalent question $0 \in [\![\hat{X}]\!]$.

Let $[\![X]\!]_{\leq L} := [\![X]\!] \cap \{0, \ldots, L\}$ be the $L$-bounded reachability set for nonterminal $X$. We define a sequence of $L$-bounded valuations $\mu_i : \mathsf{Var} \to 2^{\{0,\ldots,L\}}$ (which is a finite object) inductively as follows. Initially, $\mu_0(X) = \emptyset$ for every nonterminal $X$. Inductively, assume that $\mu_i$ is defined. Construct the following BVASS $\mathcal{B}_i$. Every addition rule $X \leftarrow Y + Z$ in the original $\mathbb{Z}$-BVASS produces an identical rule in $\mathcal{B}_i$. Every subtraction rule $r$ of the form $X \leftarrow Y - Z$ in the original $\mathbb{Z}$-BVASS produces a rule $r_z$ of the form $X \leftarrow Y - z$ in $\mathcal{B}_i$ for every $z \in \mu_i(Z)$. Then, $\mathcal{B}_i$ is of exponential size, and $\mu_{i+1}(X)$ is computed in exponential time as the set of those $n \in \{0, \ldots, L\}$ s.t. $(X, n)$ is reachable in $\mathcal{B}_i$ (which can be checked in exponential time).

Claim 2. *The sequence of approximants is non-decreasing and it converges at iteration L:*

$$\mu_0 \subseteq \mu_1 \subseteq \cdots \subseteq \mu_L = \mu_{L+1} = \cdots .$$

Clearly, $\mu_i(X) \subseteq [\![X]\!]_{\leq L}$ for every nonterminal $X$ since at every iteration we underapproximate the actual reachability set. By the next claim, the underapproximation is exact in the limit.

Claim 3. $[\![X]\!]_{\leq L} = \bigcup_i \mu_i(X)$.

Proof of Claim 3. We show that every $n \in [\![X]\!]_{\leq L}$ is witnessed as $n \in \mu_i(X)$ for some level $i \geq 0$. Let $t$ be a $(X, n)$-run. By Lemma 7.3 we assume that, in every subtraction node, the subtrahend child is $L$-bounded. Let the *height* of a node be the maximal number of subtraction nodes on any path from that node (included) to a leaf. We show the following stronger claim by complete induction on the height: For every $(X, n)$-run $t$ of height $i \geq 0$, $n \in \mu_{i+1}(X)$. Let $t$ be a $(X, n)$-run of height $i$. Let $t'$ be an arbitrary subtrahend $(Y, m)$-subrun of the first subtraction node encountered from the root of $t$. Then, $m \leq L$ and $t'$ has height $j < i$. By inductive assumption, $m \in \mu_{j+1}(Y)$, and hence $m \in \mu_i(Y)$. We build a $(X, n)$-run in $\mathcal{B}_i$ by replacing the rule $r$ of $\mathcal{B}$ used in the root of $t'$ by the rule $r_m$ of $\mathcal{B}_i$. This shows $n \in \mu_{i+1}(X)$, as required. □

Thanks to the two claims above, $[\![X]\!]_{\leq L} = \mu_L(X)$, and the latter set can be computed in exponential time. □

Corollary 7.5. *Let $V \subseteq \mathsf{Var}$ be a support and $X \in V$ a nonterminal. Checking reachability in $[\![X]\!]_V$ is in* ExpTime.

From the last point of Lemma 7.2 and Lemma 2.1, we immediately derive the following more restrictive form for reachability sets.

Corollary 7.6. *Let $V \subseteq \mathsf{Var}$ be a support. The reachability set $[\![X]\!]_V$ is an L-bounded semilinear set of the form $F \cup (A + b^*)$, where $F, A, \{b\} \subseteq \{0, \ldots, L\}$.*

Lemma 7.7. *Let S be an L-bounded semilinear linear set of the form $F \cup (A + b^*)$, and let R be an L-bounded linear set of the form $c + d^*$. Then, $R \subseteq S$ iff $\{c, c + d, \ldots, c + L \cdot d\} \subseteq S$.*

Proof. The "only if" direction is trivial. For the other direction, we will show a bound on the minimal element of $R \setminus S$. For any natural number $x$ if $x + b \notin (A + b^*)$, then also $x \notin (A + b^*)$. If $x + b \notin S$ and $x > L$, then also $x \notin S$. In particular, if $c + (k + b)d = (c + kd) + bd \notin S$ and $c + kd > L$, then also $c + kd \notin S$. Thus, the minimal $c + kd \notin S$ strictly above $L$ is at most $c + L \cdot d$. □

Corollary 7.8. *The reachability set $[\![X]\!]$ is an L-bounded semilinear set constructible in* ExpTime.

Proof. Since $[\![X]\!] = \bigcup_{V \subseteq \mathsf{Var}} [\![X]\!]_V$, it suffices to construct $L$-bounded semilinear representations for the $[\![X]\!]_V$'s. By Corollary 7.6, $[\![X]\!]_V$ is an $L$-bounded semilinear set of the form $F \cup (A + b^*)$, where $F, A, \{b\} \subseteq \{0, \ldots, L\}$. We enumerate all $L$-bounded linear sets of the form $c + d^*$ (clearly $[\![X]\!]_V$ can be expressed as a union of such sets). By Lemma 7.7, we can check whether $c + d^* \subseteq [\![X]\!]_V$ by performing $L$ reachability queries of the form $c + kd \in [\![X]\!]_V$ with $k \in \{0, \ldots, L\}$, each of which can be done in ExpTime by Corollary 7.5, and thus in ExpTime overall. □

Theorem 5.2 follows by transforming the $\mathbb{Z}$-BVASS into the normal form and from Corollary 7.8.

## 8 CONCLUSIONS

We have provided an effective characterisation of the trPDA reachability relation as a quantifier-free formula over the hybrid time domain $\mathbb{H} = (\mathbb{Z} \uplus \mathbb{Q}, +1, \leq^{\mathbb{H}}, \equiv_m)$ combining integer $\mathbb{Z}$ and fractional $\mathbb{Q}$ values. From a technical point of view, what is only required from the fractional values is to belong to a homogeneous structure, such as $(\mathbb{Q}, \leq)$ in our case. For example, we could consider fractional values belonging to more exotic homogeneous dense time domains, such as *cyclic order atoms* $(\mathbb{Q}, K)$ [11][8] or *betweenness atoms* $(\mathbb{Q}, B)$ [9][9]. All the non-trivial technical work goes in handling the discrete integer domain $(\mathbb{Z}, +1, \leq, \equiv_m)$, which is non-homogeneous, and thus requires specialized techniques.

Several directions for future work can be identified. While we provide a 2-ExpTime upper bound for deciding the trPDA non-emptiness problem, the only known lower bound is ExpTime, which holds already for the less expressive orbit-finite and grammar classes (cf. [10]). Moreover, in the special case of orbit-finite trPDA studied in [10], only a NExpTime upper-bound is known. Regarding $\mathbb{Z}$-BVASS,we have provided an ExpTime upper bound, while a PSpace lower bound can be immediately inferred by simulating bounded one-counter automata [13]. Moreover, there is a gap between our decidability result for $\mathbb{Z}$-BVASS in dimension one, and the known undecidability in dimension six [18].

## 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] S. Akshay, P. Gastin, and S. N. Krishna. Analyzing timed systems using tree automata. In *Proc. CONCUR'16*, pages 27:1–27:14, 2016.
[3] J. R. Alfonsín. *The Diophantine Frobenius Problem.* Oxford University Press, 2005.
[4] R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126:183–235, 1994.
[5] M. Benerecetti, S. Minopoli, and A. Peron. Analysis of timed recursive state machines. In *Proc. TIME'10*, pages 61–68. IEEE, sept. 2010.
[6] M. Benerecetti and A. Peron. Timed recursive state machines: Expressiveness and complexity. *Theoretical Computer Science*, 625:85–124, 2016.
[7] 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.
[8] A. Bouajjani, R. Echahed, and R. Robbana. Verification of context-free timed systems using linear hybrid observers. In *Proc. CAV'94*, volume 818 of *LNCS*, pages 118–131. Springer, 1994.
[9] L. Clemente and S. Lasota. Reachability analysis of first-order definable pushdown systems. In *Proc. CSL'15*, volume 41 of *LIPIcs*, pages 244–259. Dagstuhl, 2015.
[10] L. Clemente and S. Lasota. Timed pushdown automata revisited. In *Proc. LICS'15*, pages 738–749. IEEE, July 2015.
[11] L. Clemente and S. Lasota. Binary reachability of timed pushdown automata via quantifier elimination and cyclic order atoms. In *To appear in Proc. ICALP'18*, 2018.
[12] L. Clemente, S. Lasota, R. Lazić, and F. Mazowiecki. Timed pushdown automata and branching vector addition systems. In *Proc. LICS'17*, 2017.

[8] The ternary cyclic order relation $K \subseteq \mathbb{Q}^3$ is defined as $K(a, b, c) \equiv a < b < c \vee b < c < a \vee c < a < b$.
[9] The ternary betweenness relation $B \subseteq \mathbb{Q}^3$ is defined as $B(a, b, c) \equiv b < a < c \vee c < a < b$.

[13] J. Fearnley and M. Jurdziński. Reachability in two-clock timed automata is PSPACE-complete. *Inf. Comput.*, 243:26–36, 2015.

[14] D. Figueira, R. Lazic, J. Leroux, F. Mazowiecki, and G. Sutre. Polynomial-space completeness of reachability for succinct branching VASS in dimension one. In *In Proc. of ICALP'17*, 2017.

[15] S. Ginsburg and E. H. Spanier. Semigroups, presburger formulas, and languages. *Pacific J. Math.*, 16(2):285–296, 1966.

[16] S. Göller, C. Haase, R. Lazić, and P. Totzke. A polynomial-time algorithm for reachability in branching VASS in dimension one. In *Proc. ICALP'16*, volume 55 of *LIPIcs*. Dagstuhl, 2016.

[17] A. Jeż and A. Okhotin. Conjunctive grammars over a unary alphabet: Undecidability and unbounded growth. *Theory Comput. Syst.*, 46(1):27–58, 2010.

[18] R. Lazić. The reachability problem for branching vector addition systems requires doubly-exponential space. *Inf. Process. Lett.*, 110(17):740–745, 2010.

[19] R. Lazić and S. Schmitz. Non-elementary complexities for branching VASS, MELL, and extensions. In *Proc. CSL-LICS'14*. ACM, 2014.

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

[21] A. B. Matos. Periodic sets of integers. *Theor. Comput. Sci.*, 127(2):287–312, 1994.

[22] R. J. Parikh. On context-free languages. *J. ACM*, 13(4):570–581, Oct. 1966.

[23] 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.

[24] A. Trivedi and D. Wojtczak. Recursive timed automata. In *Proc. ATVA'10*, volume 6252 of *LNCS*, pages 306–324. Springer, 2010.

[25] K. N. Verma and J. Goubault-Larrecq. Karp-Miller Trees for a Branching Extension of VASS. *Discrete Mathematics and Theoretical Computer Science*, 7:217–230, 2005.

[26] K. N. Verma, H. Seidl, and T. Schwentick. On the complexity of equational Horn clauses. In *Proc. CADE'05*, pages 337–352, 2005.