Presburger arithmetic with stars, rational subsets of graph groups, and nested zero tests

Christoph Haase
University of Oxford
United Kingdom
Email: christoph.haase@cs.ox.ac.uk

Georg Zetzsche
Max Planck Institute for Software Systems (MPI-SWS)
Germany
Email: georg@mpi-sws.org

***Abstract*—We study the computational complexity of existential Presburger arithmetic with (possibly nested occurrences of) a Kleene-star operator. In addition to being a natural extension of Presburger arithmetic, our investigation is motivated by two other decision problems.**

The first problem is the rational subset membership problem in graph groups. A graph group is an infinite group specified by a finite undirected graph. While a characterisation of graph groups with a decidable rational subset membership problem was given by Lohrey and Steinberg [*J. Algebra*, 320(2) (2008)], it has been an open problem (i) whether the decidable fragment has elementary complexity and (ii) what is the complexity for each fixed graph group. The second problem is the reachability problem for integer vector addition systems with states and nested zero tests.

We prove that the satisfiability problem for existential Presburger arithmetic with stars is NEXP-complete and that all three problems are polynomially inter-reducible. Moreover, we consider for each problem a variant with a fixed parameter: We fix the star-height in the logic, a graph parameter for the membership problem, and the number of distinct zero-tests in the integer vector addition systems. We establish NP-completeness of all problems with fixed parameters.

In particular, this enables us to obtain a complete description of the complexity landscape of the rational subset membership problem for fixed graph groups: If the graph is a clique, the problem is NL-complete. If the graph is a disjoint union of cliques, it is P-complete. If it is a transitive forest (and not a union of cliques), the problem is NP-complete. Otherwise, the problem is undecidable.

I. Introduction

Presburger arithmetic is the first-order theory of the natural numbers with addition and order. Shown decidable by Presburger in 1929 [48], Presburger arithmetic has become a standard tool for showing decidability and complexity results in many areas of computer science such as automata theory, database theory, formal verification and knowledge representation; see also [27].

In many domains, both in theory and practice, the existential fragment of Presburger arithmetic is of particular interest for a variety of reasons. With an NP-complete satisfiability problem [8], this fragment is computationally rather lightweight. Furthermore, highly-optimised decision procedures for existential Presburger arithmetic have been developed and integrated into SMT-solvers such as CVC4 [2] and Z3 [12], which in practice enables solving a variety of problems via a reduction into existential Presburger arithmetic. Finally, existential Presburger arithmetic is expressively-complete: a seminal result due to Ginsburg and Spanier [23] established that the sets of integer vectors definable in Presburger arithmetic coincide with *semi-linear sets*, which arise as higher-dimensional generalisations of ultimately periodic sets. Since arbitrary semi-linear sets are definable in existential Presburger arithmetic, additional quantifiers do not lead to more expressiveness and only contribute succinctness.

Besides Presburger arithmetic, another classical representation for semi-linear sets is available in *rational expressions* over vectors [20], which consist of vectors, unions, (Minkowski) sums, and Kleene stars. Since each of these representations suits certain applications, it is not only natural, but also useful to consider representations that accommodate both rational expressions and existential Presburger arithmetic.

This is one of the reasons why we investigate the computational complexity of $\exists\mathsf{PA}^*$, the existential fragment of Presburger arithmetic enriched with a *Kleene-star operator* (subsequently star operator for brevity). Since existential Presburger arithmetic readily expresses (Minkowski) sums and unions, this extension encompasses the expressiveness of rational expressions. Given a formula $\phi(\boldsymbol{x})$, the star operator additionally allows for formulas of the form $\phi^*(\boldsymbol{x})$ such that $\phi^*(\boldsymbol{v})$ holds if there are $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{N}^d$ such that $\boldsymbol{v} = \boldsymbol{v}_1 + \cdots + \boldsymbol{v}_k$ and $\phi(\boldsymbol{v}_i)$ holds for all $1 \leq i \leq k$. Of course, the star-operator may occur in $\phi(\boldsymbol{x})$ itself; we refer to the maximum number of nested star operators occurring in a formula ϕ as the *star-height* of ϕ. Thus, existential Presburger arithmetic is the fragment of $\exists\mathsf{PA}^*$ of star-height 0. The fragment of $\exists\mathsf{PA}^*$ of star-height one was studied by Piskac and Kunčak [46]. They showed NP-completeness of the satisfiability problem, and furthermore natural applications of $\exists\mathsf{PA}^*$ to reasoning about multi-sets with cardinality constraints and a class of integer vector addition systems with states with semi-linear transition updates. One of our contributions is to show that $\exists\mathsf{PA}^*$ is NEXP-complete in general, and NP-complete for any arbitrary but fixed star-height. While being an interesting result in its own right, our main motivation for studying $\exists\mathsf{PA}^*$ emerges from two other problems, the

978-1-7281-3608-0/19/$31.00 ©2019 IEEE

computational complexity of the rational subset membership problem in graph groups and of reachability in integer vector addition systems with nested zero tests.

Rational subsets of graph groups

Graph groups (also known as *right-angled Artin groups* or *free partially commutative groups*) are infinite groups that are specified by an undirected simple graph. Here, each vertex represents a generator and an edge asserts that a pair of generators commute. This class of groups has received growing attention in the last decades, from computer science as well as from mathematics. In computer science, this is due to their close connection to Mazurkiewicz traces [16, 17, 18] and their prominent role in a general framework for infinite-state systems [9, 19, 56, 57, 58]. In mathematics, graph groups are currently an area of intense investigation because of their rich subgroup structure [53]: The class of *virtually special groups*, i.e. finite extensions of subgroups of graph groups, recently turned out to encompass an abundant array of groups, namely Coxeter groups [30], one-relator groups with torsion [54], fully residually free groups [54], and fundamental groups of hyperbolic 3-manifolds [33].

In a tradition initiated by Dehn [13] to consider groups together with their decision problems, there has been substantial interest in algorithmic questions for graph groups, such as the word problem [14, 55], solving equations [16, 17, 43], and membership in subgroups [36, 37] and submonoids [36, 41].

A decision problem that has been attracting attention in the last decades is the membership problem for *rational subsets*, i.e. those accepted by a finite automaton (e.g. [36, 41, 42] and [40] for a survey). This is because they naturally generalise subgroups and submonoids, and have been an important tool for solving equations [15] and other problems [3].

An important contribution in this context is a result of Lohrey and Steinberg [41]. It characterises those graphs for which the corresponding graph group has a decidable rational subset membership problem. However, the procedure they provide has non-elementary complexity and it has remained open until now whether there is an elementary one. The precise complexity has been left open both in the case (i) where the graph (from the decidable class) is part of the input and (ii) for each fixed group. We denote problem (ii) as $\mathsf{RatMP}^{\mathsf{tf}}$.

As we show, this problem is polynomially inter-reducible with satisfiability of $\exists\mathsf{PA}^*$. Therefore, our results imply that the problem is NEXP-complete if the graph (and hence the group) is part of the input. Moreover, the NP upper bound for fixed star height leads to a complete classification of the complexity for each fixed graph group: If the graph is a clique, the problem is NL-complete. If the graph is a disjoint union of at least two cliques, then it is P-complete. If the graph is a transitive forest (and not a disjoint union of cliques), the problem is NP-complete. In all other cases, Lohrey and Steinberg established undecidability [41].

Integer VASS with nested zero tests

Vector addition systems with states (VASS), equivalently known as Petri nets, are a fundamental model of computation. A VASS comprises a finite-state controller with a finite number of counters ranging over the non-negative integers. When taking a transition, counters can be incremented and decremented provided that resulting counter values are all non-negative. VASS find a plethora of applications, primarily for modelling and reasoning about concurrent systems, but also, for example, in formal language theory, logic, process calculi, see e.g. [51, Sec. 5]. While control-state reachability as well as configuration reachability are decidable for VASS, additionally allowing for testing counters for zero along transitions renders both problems undecidable [44]. A decidable extension of VASS with zero tests was given by Reinhardt [50]. He showed that reachability in VASS extended with nested zero tests (also called hierarchical zero tests) is decidable. Nested zero tests constraint arbitrary zero tests such that the k-th counter can only be tested for zero if at the same time the first $k-1$ counters are tested for zero. An alternative proof of Reinhardt's result was given by Bonnet [7].

Leaving aside decidability issues, one major obstacle in the automated analysis of VASS is the high computational complexity of decidable decision problems: control-state reachability is EXPSPACE-complete [39, 49], and a non-elementary lower bound for the configuration-reachability problem has recently been established [11]. To the best of the authors' knowledge, no dedicated complexity results have been established for VASS with nested zero tests. To overcome those high computational costs, relaxations of VASS and their extensions have been investigated with the goal of finding computationally more tractable models that can be used to over-approximate reachability sets of VASS and their extensions, for instance by allowing counters to range over the integers, the resulting model being commonly known as integer VASS ($\mathbb{Z}$-VASS) [29] or blind counter automata [25]. Configuration and *a fortiori* control-state reachability for $\mathbb{Z}$-VASS are only NP-complete [29]. $\mathbb{Z}$-VASS and related relaxations have successfully been used to enable the scalable analysis of concurrent programs, see e.g. [1, 5, 22]. Furthermore, over-approximations of VASS extended with, for instance, affine transformations [6], and ordered and unordered data [31, 32] have also been studied.

A further contribution of this paper is to show that configuration reachability in $\mathbb{Z}$-VASS extended with nested zero tests ($\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$) is inter-reducible with satisfiability in $\exists\mathsf{PA}^*$. As a consequence, we obtain NEXP-completeness of configuration reachability in $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$. Furthermore, our reduction preserves fixed-parameter properties, enabling us to show that configuration reachability is NP-complete when fixing the number of distinct zero-tests.

Due to space constraints, the proofs of some statements appear in an extended version of this article which can be obtained from the authors.

Main technical tools

We briefly comment on the main ideas and tools that our results rely on. For the NEXP and NP upper bounds for $\exists\mathsf{PA}^*$, just as in [46], we rely on a Carathéodory-type theorem [21] for decomposing semi-linear sets into semi-linear sets whose sets of period vectors have polynomial cardinality. We additionally exploit that this decomposition, together with results on the descriptional complexity of Boolean operations on semi-linear sets [10], also enables us to witness that a linear set is contained in the solutions of $\phi^*(\boldsymbol{x})$ by providing only a polynomial number of linear sets contained in the solutions of $\phi(\boldsymbol{x})$ as a certificate. For the NEXP lower bound, we reduce the succinct circuit satisfiability problem to satisfiability of $\exists\mathsf{PA}^*$. To this end, we show that one can define 2^{2^n} in an $\exists\mathsf{PA}^*$ formula of size polynomial in n. Moreover, we express a universal quantifier ranging over $[1, 2^n]$. This allows us to guess an evaluation of the succinct circuit as a number in $[1, 2^{2^n}]$ and express consistency with the gates using the quantifier.

In the translation of $\mathsf{RatMP}^{\mathsf{tf}}$ to $\exists\mathsf{PA}^*$, we avoid the non-elementary blow up as follows. The procedure of Lohrey and Steinberg [41] builds semi-linear sets by alternating two operations: (i) intersections and (ii) building context-free grammars (from semi-linear representations) and taking their Parikh images. Since Parikh images of context-free grammars may require exponential semi-linear representations [38], this results in non-elementary complexity. Here, we adapt a construction by Verma, Seidl, and Schwentick [52], which builds an existential Presburger formula for a given context-free grammar. Our modification uses the Kleene star to accommodate grammars that have infinite (but semi-linear) sets of productions. For the remaining reductions, we use ad-hoc constructions.

II. Preliminaries

A. General notation

We denote by $\|\cdot\|$ the L^∞-norm. Given an $m \times n$ integer matrix A, as usual $\|A\|_{1,\infty} = \max_{1 \le i \le m} \sum_{j=1}^n |a_{ij}|$. Throughout the paper, we interchangeably treat finite sets of vectors $Q \subseteq \mathbb{Z}^n$ as matrices (e.g. by lexicographically ordering the elements of Q), and vice versa.

B. Semi-linear sets

Let $\boldsymbol{b} \in \mathbb{N}^n$ be a *base vector* and $P = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_k\} \subseteq \mathbb{N}^n$ be a finite set of *period vectors*. The *linear set* $L(\boldsymbol{b}, P)$ *generated by* $\boldsymbol{b}$ *and* P is defined as $L(\boldsymbol{b}, P) = \boldsymbol{b} + \{\sum_{i=1}^k \lambda_i \cdot \boldsymbol{p}_i : \lambda_i \in \mathbb{N}, 1 \le i \le k\}$. We call $N \subseteq \mathbb{N}^n$ a *linear set* if there are $\boldsymbol{b}$ and P such that $N = L(\boldsymbol{b}, P)$. A *semi-linear set* is a finite union of linear sets. Given a linear set $N \subseteq \mathbb{N}^n$, we denote by $\|N\|$ the smallest s such that $N = L(\boldsymbol{b}, P)$ and $s = \max\{\|\boldsymbol{b}\|, \|\boldsymbol{p}\| : \boldsymbol{p} \in P\}$. For a semi-linear set $M \subseteq \mathbb{N}^n$, we denote by $\|M\|$ the smallest s such that $M = \bigcup_{i \in I} N_i$ for linear sets N_i and $\|N_i\| \le s$ for all $i \in I$.

For a linear set $N = L(\boldsymbol{b}, P) \subseteq \mathbb{N}^n$, we can a priori only derive $|P| \le (\|N\| + 1)^n$ as a bound on the cardinality of P. The following paraphrased result due to Eisenbrand and Shmonin shows that N is equivalent to a semi-linear set in which the cardinality of all sets of period vectors is small.

Proposition II.1 (Thm. 1 in [21])**.** *Let $N = L(\boldsymbol{b}, P) \subseteq \mathbb{N}^n$ be a linear set. Then $N = \bigcup_{i \in I} L(\boldsymbol{b}, P_i)$ such that $P_i \subseteq P$ and $|P_i| \le 2n \log(4n\|N\|)$ for all $i \in I$.*

In particular, Proposition II.1 enables us to assume that sets of period vectors of semi-linear sets have small cardinality.

Corollary II.2. *Let $M \subseteq \mathbb{N}^n$ be a semi-linear set. Then $M = M'$ such that $\|M\| = \|M'\|$, $M' = \bigcup_{i \in I} L(\boldsymbol{b}_i, P_i)$ and $|P_i| \le 2n \log(4n\|M\|)$ for all $i \in I$.*

Given a system of linear Diophantine inequalities $\mathcal{S}\colon A \cdot \boldsymbol{x} \ge \boldsymbol{c}$ for some $m \times n$ integer matrix A, denote by $[\![\mathcal{S}]\!] = \{\boldsymbol{x}^* \in \mathbb{N}^n : A \cdot \boldsymbol{x}^* \ge \boldsymbol{c}\}$ the set of non-negative solutions of $\mathcal{S}$. We will use the following bound on the semi-linear representation of the set of solutions of $\mathcal{S}$, which follows from [47] and [10].

Proposition II.3. *Let $\mathcal{S}\colon A \cdot \boldsymbol{x} \ge \boldsymbol{c}$ be a system of linear Diophantine inequalities. Then $M = [\![\mathcal{S}]\!] = \bigcup_{i \in I} L(\boldsymbol{b}_i, P)$, $\|M\| \le (\|A\|_{1,\infty} + \|\boldsymbol{c}\| + 2)^{m+n}$. Moreover, for every $L(\boldsymbol{b}, P) \subseteq [\![\mathcal{S}]\!]$, we have $A \cdot \boldsymbol{b} \ge \boldsymbol{c}$ and $A \cdot \boldsymbol{p} \ge \boldsymbol{0}$ for all $\boldsymbol{p} \in P$.*

Let $M \subseteq \mathbb{N}^d$, the *Kleene-star* of M is defined as

$$M^* := \bigcup_{k \ge 0} \left\{ \sum_{i=1}^k \boldsymbol{v}_i : \boldsymbol{v}_i \in M \right\}.$$

Here, the empty sum denotes $\boldsymbol{0}$. For linear sets, we write $L^*(\boldsymbol{b}, P)$ instead of $(L(\boldsymbol{b}, P))^*$.

C. Presburger arithmetic with stars

Let $\boldsymbol{x}, \boldsymbol{y}$ be tuples of first-order variables, and let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ be an n-tuple consisting of the first-order variables of $\boldsymbol{x}$ and $\boldsymbol{y}$. A formula of existential Presburger arithmetic of star height 0 is of the form

$$\phi(\boldsymbol{x}) = \exists \boldsymbol{y} : \psi(\boldsymbol{z}), \tag{1}$$

where $\psi(\boldsymbol{z})$ is a (possibly nested) conjunction and disjunction of linear Diophantine inequalities of the form $\boldsymbol{a} \cdot \boldsymbol{z} \ge c$ for $\boldsymbol{a}^\mathsf{T} \in \mathbb{Z}^n$ and $c \in \mathbb{Z}$. A formula of existential Presburger arithmetic of star height $k + 1$ is of the form of Eq. (1) and additionally allows for atomic formulas $\vartheta^*(\boldsymbol{z})$ with star height of ϑ at most k. *Existential Presburger arithmetic with stars (*$\exists\mathsf{PA}^*$*)* is the set of all formulas of existential Presburger arithmetic of star height k for any $k \ge 0$.

The semantics $[\![\phi]\!]$ of an $\exists\mathsf{PA}^*$ formula is given in terms of subsets of $\mathbb{N}^d$ by structural induction. By Proposition II.3, $[\![\boldsymbol{a} \cdot \boldsymbol{z} \ge c]\!]$ is a semi-linear set, for the remaining cases we define

- $[\![\phi(\boldsymbol{z}) \wedge \psi(\boldsymbol{z})]\!] = [\![\phi(\boldsymbol{z})]\!] \cap [\![\psi(\boldsymbol{z})]\!]$
- $[\![\phi(\boldsymbol{z}) \vee \psi(\boldsymbol{z})]\!] = [\![\phi(\boldsymbol{z})]\!] \cup [\![\psi(\boldsymbol{z})]\!]$
- $[\![\vartheta^*(\boldsymbol{z})]\!] = [\![\vartheta(\boldsymbol{z})]\!]^*$
- $[\![\exists \boldsymbol{y} : \psi(\boldsymbol{z})]\!] = \pi_{\boldsymbol{x}} [\![\psi(\boldsymbol{z})]\!]$, where $\pi_{\boldsymbol{x}}$ denotes the projection onto the variables $\boldsymbol{x}$.

It is not difficult to see, and will more formally be discussed in Section IV, that the sets definable in $\exists\mathsf{PA}^*$ are semi-linear sets. A formula $\phi(\boldsymbol{x})$ is *satisfiable* if $[\![\phi]\!] \ne \emptyset$. (Thus, if ϕ has no free variables, then satisfiability means $[\![\phi]\!] = \{\emptyset\}$).

Remark II.4. We do not allow negation to occur in our formulas in order to avoid complementing semi-linear sets.

Atomic formulas of the form $\boldsymbol{a} \cdot \boldsymbol{z} \geq c$ can, however, be negated since $\neg(\boldsymbol{a} \cdot \boldsymbol{z} \geq c) \equiv \boldsymbol{a} \cdot \boldsymbol{z} \leq c - 1$.

Moreover, by renaming variables if necessary, we may at any time with no loss of generality assume that no existential quantifiers occur in ϕ.

The *length* $|\phi|$ of a $\exists\mathsf{PA}^*$ formula ϕ is defined as the number of symbols required to write down ϕ, and $\|\phi\|$ denotes the absolute value of the largest constant occurring in ϕ. Without loss of generality we assume unary encoding of numbers, and that $|\phi| \geq 2$ for any ϕ.

D. Rational subsets of graph groups

A *graph* is a pair (A, I), where A is an alphabet and $I \subseteq \mathcal{P}_2(A)$. Here, $\mathcal{P}_k(S)$ denotes the set of k-element subsets of a set S. We will also use terminology from Mazurkiewicz traces [18], where I is called an *independence relation*. For an alphabet A, let $A^{-1} = \{a^{-1} : a \in A\}$ be the alphabet of formal inverses of A and $A^{\pm 1} = A \cup A^{-1}$. The *graph group defined by* (A, I), denoted $\mathbb{G}(A, I)$, is the group presented by $\langle a \in A : ab = ba \ (\{a, b\} \in I)\rangle$. We will work with a definition of $\mathbb{G}(A, I)$ as a monoid as follows. For $u, v \in (A^{\pm 1})^*$, let $u \to v$ if there are words $x, y \in (A^{\pm 1})^*$ and a pair $(r, s) \in \{(ab, ba) : \{a, b\} \in I)\} \cup \{(aa^{-1}, \varepsilon), (a^{-1}a, \varepsilon) : a \in A\}$ such that $u = xry$ and $v = xsy$. Then, let $\equiv_{A,I}$ be the symmetric, reflexive, transitive closure of $\to$. We can then define $\mathbb{G}(A, I) = (A \cup A^{-1})^* / \equiv_{A,I}$. In other words, $\mathbb{G}(A, I)$ consists of $\equiv_{A,I}$ congruence classes $[w]$ for $w \in (A^{\pm 1})^*$ that are multiplied by way of $[u][v] = [uv]$. Let $|A| = n$. Observe that if (A, I) is a *clique*, i.e. $I = \mathcal{P}_2(A)$, then $\mathbb{G}(A, I) \cong \mathbb{Z}^n$. If $I = \emptyset$, then $\mathbb{G}(A, I)$ is called the *free group (over n generators)* and is also denoted F_n.

For a subset $B \subseteq A$, let $I_B = I \cap \mathcal{P}_2(B)$ and $\pi_B \colon (A^{\pm 1})^* \to (B^{\pm 1})^*$ be the morphism with $\pi_B(b) = b$, $\pi_B(b^{-1}) = b^{-1}$ for $b \in B$ and $\pi_B(a) = \pi_B(a^{-1}) = \varepsilon$ for $a \notin B$. Then $u \equiv_{A,I} v$ implies $\pi_B(u) \equiv_{B,I_B} \pi_B(v)$, so that π_B induces a morphism $\mathbb{G}(A, I) \to \mathbb{G}(B, I_B)$, $[u] \mapsto [\pi_B(u)]$, also denoted π_B.

Let G be a group. An *automaton over* G is a tuple $\mathcal{A} = (Q, E, q_0, q_f)$, where Q is a finite set of *states*, $E \subseteq Q \times G \times Q$ is a finite set of *edges*, and $q_0, q_f \in Q$ are its *initial* and *final state*, respectively. A *configuration of* $\mathcal{A}$ is a pair $(p, g) \in Q \times G$, and we write $(p, g) \to_{\mathcal{A}} (p', g')$ if there is an edge $(p, h, p') \in E$ with $g' = gh$. Such an automaton describes a subset of G, namely $\mathsf{L}(\mathcal{A}) = \{g \in G : (q_0, 1) \to_{\mathcal{A}}^* (q_f, g)\}$, where $\to_{\mathcal{A}}^*$ denotes the reflexive transitive closure of $\to_{\mathcal{A}}$. The subsets of G of the form $\mathsf{L}(\mathcal{A})$ are called *rational subsets*.

In algorithms dealing with automata over groups, one usually considers *finitely generated* (short *f.g.*) groups, i.e. groups with a finite subset $\Sigma \subseteq G$ so that every element of G can be written as a product of members of Σ. In that case, the elements of G, and hence edge inscriptions, can be encoded by words over Σ. Since we will work with graph groups, we always assume that automata over $\mathbb{G}(A, I)$ are represented with the generating set $A^{\pm 1}$ (neither decidability nor complexity of the problems considered here depend on the chosen generating set). The *rational subset membership problem for* G is the following problem: Given an automaton $\mathcal{A}$ over G and an element $g \in G$, decide whether $g \in \mathsf{L}(\mathcal{A})$.

An important concept in the context of rational subsets of graph groups is that of transitive forests. By a *forest*, we mean a cycle-free graph where every connected component has a distinguished root vertex. A graph (A, I) is a *transitive forest* if it can be obtained from a forest by adding an edge between any two nodes that lie on a path between a root and a leaf. The rational subset membership problem for graph groups has been studied by Lohrey and Steinberg [41], who obtained a complete characterisation of those graph groups where the problem is decidable.

Theorem II.5 ([41])**.** *Let (A, I) be a graph. The rational subset membership problem is decidable for $\mathbb{G}(A, I)$ if and only if (A, I) is a transitive forest.*

E. Integer VASS with nested zero-tests

A *d-dimensional $\mathbb{Z}$-VASS with k nested zero-tests* ($\mathbb{Z}$-$\mathsf{VASS}_k^{\mathsf{nz}}$ for short) is a tuple $V = (Q, Z, E)$, where Q is a finite set of *states*, $Z \subseteq [0, d]$ is its set of *zero tests* with $|Z \setminus \{0\}| = k$, and $E \subseteq Q \times [-1, 1]^d \times Z \times Q$ is its set of *transitions*. A *configuration* is a pair $(q, \boldsymbol{v}) \in Q \times \mathbb{Z}^d$. We write $(q, \boldsymbol{v}) \to_V (q', \boldsymbol{v}')$ if there is an edge $(q, \boldsymbol{u}, \ell, q')$ such that (i) $\boldsymbol{v}' = \boldsymbol{v} + \boldsymbol{u}$ and (ii) $v_i = 0$ for every $i \in [1, \ell]$, where $\boldsymbol{v} = (v_1, \ldots, v_d)$.

Note that a $\mathbb{Z}$-$\mathsf{VASS}_k^{\mathsf{nz}}$ can have more than k counters. However, there are at most k elements $\ell \in [1, d]$ for which we can check whether all counters 1 up to ℓ are zero. Moreover, these tests can be performed arbitrarily often during a run. A $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ is a $\mathbb{Z}$-$\mathsf{VASS}_k^{\mathsf{nz}}$ for some $k \geq 0$.

III. Overview of main results

In this section, we present the main results of this work. Let us begin with the decision problems we study. The first is the *satisfiability problem for* $\exists\mathsf{PA}^*$:

Given: An $\exists\mathsf{PA}^*$ formula ϕ.
Question: Is ϕ satisfiable, i.e., is $[\![\phi]\!] \neq \emptyset$?

Slightly abusing notation, we denote this problem also as $\exists\mathsf{PA}^*$. If we restrict the input to $\exists\mathsf{PA}^*$ formulas of star-height k, then we denote the resulting decision problem by $\exists\mathsf{PA}_k^*$.

The second problem concerns rational subsets of graph groups. For the fixed-parameter version, we introduce the branching number of transitive forests. Note that every non-empty transitive forest is either (i) a disjoint union of connected transitive forests, or (ii) has a *universal vertex*, i.e. a vertex that is adjacent to all other vertices (take the root of the underlying tree). This induces a successive decomposition of the transitive forest into smaller transitive forests: For a disjoint union, take the disjoint connected transitive forests. If there is a universal vertex, remove that vertex to obtain a smaller transitive forest.

The decomposition is unique up to isomorphism: This is obvious in the case of a disjoint union. In the case of several universal vertices, note that all possible removals result in isomorphic graphs. This allows us to define the *branching number* $\beta(A, I)$ of a transitive forest (A, I): If $A = \emptyset$, then $\beta(A, I) = 0$. If (A, I) is a disjoint union of connected transitive forests $(A_1, I_1), \dots, (A_n, I_n)$, then $\beta(A, I) = \max\{\beta(A_i, I_i) : i \in [1, n]\} + (n-1)$. If (A, I) has a universal vertex u and removing u leaves (A', I'), then $\beta(A, I) = \beta(A', I')$.

The *rational subset membership problem for graph groups defined by transitive forests*, denoted $\mathsf{RatMP}^{\mathsf{tf}}$, is defined as follows:

Given: A transitive forest (A, I), an automaton $\mathcal{A}$ over $\mathbb{G}(A, I)$, and a word $w \in (A^{\pm 1})^*$.
Question: Does $[w] \in \mathsf{L}(\mathcal{A})$ hold?

If we restrict the input to transitive forests (A, I) with $\beta(A, I) \le k$, then the problem is denoted $\mathsf{RatMP}^{\mathsf{tf}}_k$.

Finally, we define the *reachability problem for* $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$.

Given: A $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ $\mathcal{A}$ and configurations $(q, \boldsymbol{v})$, $(r, \boldsymbol{w})$.
Question: Does $(q, \boldsymbol{v}) \to^*_{\mathcal{A}} (r, \boldsymbol{w})$ hold?

If the input is restricted to $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ with k nested zero-tests, we write $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}_k$. In our definition, the counter updates for $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ are given in unary. This is not a restriction: The reachability problem for the variant with binary updates is logspace reducible to our version.

The following is our first main result.

Theorem III.1. *The problem* $\exists\mathsf{PA}^*$ *is* NEXP*-complete, and* $\exists\mathsf{PA}^*_k$ *is* NP*-complete for every fixed* $k \in \mathbb{N}$.

We will prove Theorem III.1 in Section IV. Our second main result is that the problems $\exists\mathsf{PA}^*$, $\mathsf{RatMP}^{\mathsf{tf}}$ and $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ are polynomial-time inter-reducible.

Theorem III.2. *The three problems* $\exists\mathsf{PA}^*$, $\mathsf{RatMP}^{\mathsf{tf}}$ *and* $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ *are polynomially inter-reducible. Moreover, for each fixed* $k \in \mathbb{N}$, *the reductions translate among* $\exists\mathsf{PA}^*_k$, $\mathsf{RatMP}^{\mathsf{tf}}_k$, *and* $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}_k$.

Theorem III.2 will be shown in Sections V to VII. As obvious consequences, we have:

Corollary III.3. $\mathsf{RatMP}^{\mathsf{tf}}$ *and* $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ *are* NEXP*-complete.* $\mathsf{RatMP}^{\mathsf{tf}}_k$ *and* $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}_k$ *are* NP*-complete for all fixed* $k \in \mathbb{N}$.

In particular, this implies that if we restrict the input of $\mathsf{RatMP}^{\mathsf{tf}}$ to a fixed transitive forest (A, I), then the problem is always in NP. With this piece of the puzzle, we can show:

Theorem III.4. *Let* (A, I) *be a graph. Then the rational subset membership problem for* $\mathbb{G}(A, I)$ *is*

1) NL*-complete if* (A, I) *is a clique,*
2) P*-complete if* (A, I) *is a disjoint union of at least two cliques,*
3) NP*-complete if* (A, I) *is a transitive forest and not a disjoint union of cliques, and*
4) *undecidable if* (A, I) *is not a transitive forest.*

IV. The complexity of existential Presburger arithmetic with stars

This section proves NEXP-completeness of $\exists\mathsf{PA}^*$ respectively NP-completeness of $\exists\mathsf{PA}^*_k$ for every fixed $k \in \mathbb{N}$. We first establish some technical results on semi-linear sets before we provide a bound on the descriptional complexity of the semi-linear representation of the set of solutions of an $\exists\mathsf{PA}^*$ formula ϕ. This bound then gives rise to a decision procedure with the desired upper bounds. We close this section by establishing a matching NEXP lower bound for $\exists\mathsf{PA}^*$.

A. Properties of the Kleene star on semi-linear sets

As a preparatory step, we analyse the effects on the descriptional complexity of the application of the Kleene star on semi-linear sets. The facts derived below are paraphrased or indirectly stated in [46]; for that reason we only state the results and defer all proofs to the full version of this paper. The following lemma shows that semi-linear sets are closed under the star operator.

Lemma IV.1. Let $M = \bigcup_{j \in J} L(\boldsymbol{c}_j, Q_j)$ be a semi-linear set. Then $M^* = \bigcup_{K \subseteq J} L(\boldsymbol{b}_K, P_K)$, where

$$\boldsymbol{b}_K := \sum_{k \in K} \boldsymbol{c}_k \qquad C_K := \bigcup_{k \in K} \{\boldsymbol{c}_k\}$$
$$Q_K := \bigcup_{k \in K} Q_k \qquad P_K := C_K \cup Q_K.$$

For our purposes, this lemma is too weak: there is no bound on the cardinality of the P_K, and the magnitude of $\boldsymbol{b}_K$ depends on $|J|$, which *a priori* can only be bounded by $2^{(\|M\|+1)^n}$. Building upon Lemma IV.1, we can derive the following lemma whose statement and proof are adapted from [46, Thm. 2].

Lemma IV.2. Let $M = \bigcup_{j \in J} L(\boldsymbol{c}_j, Q_j) \subseteq \mathbb{N}^n$ be a semi-linear set, and let $c = 2n \log(4n\|M\|)$. Then $M^* = \bigcup_{i \in I} L(\boldsymbol{b}_i, P_i)$ such that for every $i \in I$, there is some $K \subseteq J$ with $|K| \le c$ such that

- $\boldsymbol{b}_i = \sum_{k \in K} \boldsymbol{c}_k$
- $P_i \subseteq \bigcup_{j \in J} \{\boldsymbol{c}_j\} \cup \bigcup_{k \in K} Q_k$ with $|P_i| \le c$, and
- $\|M^*\| \le c \cdot \|M\|$.

B. Properties of intersections of linear sets

We also need a bound on the descriptional complexity of intersecting k linear sets, which are provided by the following lemma. Its proof is analogous to the proof of Theorem 6 in [10] and deferred to the appendix.

Lemma IV.3. Let $N_j = L(\boldsymbol{c}_j, Q_j) \subseteq \mathbb{N}^n$ such that $\|N_j\| \le s$ and $|Q_j| \le m$, $j \in [k]$ for some k. Then $M := \bigcap_{1 \le j \le k} N_j$ is a semi-linear set $M = \bigcup_{i \in I} L(\boldsymbol{b}_i, P)$ such that

- $\|M\| \le (k \cdot m \cdot s)^{O(k \cdot n)}$.
- for every $\boldsymbol{b}_i$ and $1 \le j \le k$ there is some $\boldsymbol{d} \ge \boldsymbol{0}$ such that $\boldsymbol{b}_i = \boldsymbol{c}_j + Q_j \cdot \boldsymbol{d}$, and
- for every $1 \le j \le k$ there is some non-negative matrix R_j such that $P = Q_j \cdot R_j$.

C. Bounds on the semi-linear representation of $\exists\mathsf{PA}^*$ solutions

We are now fully prepared to give an estimation on the descriptional complexity of the semi-linear representation of the set of solutions of an $\exists\mathsf{PA}^*$ formula. To this end, we first prove a technical lemma that establishes bounds for conjunctive $\exists\mathsf{PA}^*$ formulas in which the maximum constants of the semi-linear representation of the top-level ϕ_i formulas is bounded.

Algorithm 1 Decision procedure for $\exists\mathsf{PA}^*$

```
1:  procedure Verify($(\boldsymbol{b}, P)$, $\phi$)
2:      if $\phi = \boldsymbol{a} \cdot \boldsymbol{x} \geq c$ then
3:          check $\boldsymbol{a} \cdot \boldsymbol{b} \geq c$ and $\boldsymbol{a} \cdot \boldsymbol{p} \geq 0$ for all $\boldsymbol{p} \in P$
4:      else if $\phi = \psi^*(\boldsymbol{x})$ then
5:          guess $(\boldsymbol{c}_i, Q_i)$, $i \in [m]$ and $K \subseteq [m]$
6:          Verify($(\boldsymbol{c}_i, Q_i), \psi$) for all $i \in [m]$
7:          check $\boldsymbol{b} = \sum_{k \in K} \boldsymbol{c}_k$
8:          check $P \subseteq \bigcup_{i \in [m]} \{\boldsymbol{c}_i\} \cup \bigcup_{k \in K} Q_k$
9:      else
10:         guess set of indices $I$ of top-level formulas of $\phi$
11:                 s.t. $\bigwedge_{i \in I} \phi_i$ is a clause of the DNF of $\phi$
12:         guess $(\boldsymbol{c}_i, Q_i)$ for all $i \in I$
13:         Verify($(\boldsymbol{c}_i, Q_i), \phi_i$) for all $i \in I$
14:         check $\boldsymbol{b} = \boldsymbol{c}_i + Q_i \cdot \boldsymbol{d}_i$ some $\boldsymbol{d}_i$ for all $i \in I$
15:         check $P = Q_i \cdot R_i$ for some $R_i$ for all $i \in I$
```

Lemma IV.4. *Let* $\phi(\boldsymbol{x}) = \exists \boldsymbol{y} : \bigwedge_{1 \leq i \leq j} \phi_i(\boldsymbol{x}, \boldsymbol{y}) \wedge \bigwedge_{j < i \leq k} \phi_i^*(\boldsymbol{x}, \boldsymbol{y})$ *be an* $\exists\mathsf{PA}^*$ *formula,* $L = [\![\phi]\!]$ *and* $M_i = [\![\phi_i]\!]$ *such that* $\|M_i\| \leq s$ *for all* $i \in [k]$*. Then* $\|L\| \leq s^{O(|\phi|^3)}$*.*

Proof. We first estimate $\|M_i^*\|$ for $j < i \leq k$. By Lemma IV.2, we have $\|M_i^*\| \leq c \cdot s$ with $c \leq 2|\phi| \log(4|\phi|s) \leq O(|\phi|^2 \cdot s)$. Hence $\|M_i^*\| \leq O(|\phi|^2 \cdot s^2)$. Likewise, we use Corollary II.2 to derive that the cardinality of the period vectors in the semi-linear representation of $\|M_i^*\|$ is bounded by $2|\phi| \log(4|\phi|^3 s^2) \leq O(|\phi|^2 \cdot s)$. Lemma IV.3 implies that for $M := \bigcap_{1 \leq i \leq j} M_i \cap \bigcap_{j < i \leq k} M_i^*$, we have

$$\|M\| \leq (|\phi| \cdot O(|\phi|^2 \cdot s) \cdot O(|\phi|^2 \cdot s^2))^{O(|\phi|^2)} \leq s^{O(|\phi|^3)}.$$

The statement follows since $\|L\| = \|M\|$. □

As an immediate consequence, we have:

Proposition IV.5. *For any* $\exists\mathsf{PA}^*$ *formula* ϕ *of star-height* k *and* $L = [\![\phi]\!]$*, we have* $\|L\| \leq \|\phi\|^{|\phi|^{O(k)}}$*.*

Proof. We show the statement by induction on k. Transform ϕ into disjunctive normal form. The statement follows for $k = 0$ from Proposition II.3. For the induction step, apply Lemma IV.4. □

D. A decision procedure for $\exists\mathsf{PA}^*$

We now derive one of the main results of this paper:

Proposition IV.6. $\exists\mathsf{PA}^*$ *is in* NEXP*, and* $\exists\mathsf{PA}^*_k$ *is in* NP *for any* $k \geq 0$*.*

From Proposition II.3, we know that the constants in the semi-linear representation of the set of solutions of an $\exists\mathsf{PA}^*$ formula $\phi(\boldsymbol{x})$ are doubly-exponentially bounded for arbitrary star height, and singly-exponentially bounded for fixed star height. However, checking satisfiability of ϕ is not possible by merely guessing some small $\boldsymbol{x}^* \in [\![\phi]\!]$. While we could easily verify whether $\boldsymbol{a} \cdot \boldsymbol{x}^* \geq c$, for subformulas of the form $\psi^*(\boldsymbol{x})$ of ϕ there is no obvious way to check whether $\boldsymbol{x}^* \in [\![\psi]\!]^*$.

Given an $\exists\mathsf{PA}^*$ formula ϕ with top-level subformulas ϕ_i, $i \in [k]$, let $T \subseteq 2^{[k]}$ be the set consisting of all subsets of indices of top-level formulas that give the clauses of the disjunctive normal form (DNF) of ϕ, i.e., the DNF of ϕ is $\bigvee_{I \in T} \bigwedge_{i \in I} \phi_i$. Obviously, $[\![\phi]\!] = \bigcup_{I \in T} \bigcap_{i \in I} [\![\phi_i]\!]$, and given $I \subseteq [k]$ one can check in polynomial time whether $I \in T$. To overcome the aforementioned problems, the non-deterministic recursive procedure Verify (Algorithm 1) deciding $\exists\mathsf{PA}^*$ has the following two properties: First, on input $(\boldsymbol{b}, P)$ and ϕ, if Verify$((\boldsymbol{b}, P), \phi)$ accepts then $L(\boldsymbol{b}, P) \subseteq \bigcap_{i \in I} [\![\phi_i]\!] \subseteq [\![\phi]\!]$ for some $I \in T$. Second, letting A consist of all tuples $(\boldsymbol{b}, P)$ such that Verify has an accepting run on input $(\boldsymbol{b}, P)$, we have $[\![\phi]\!] = \bigcup_{(\boldsymbol{b}, P) \in A} L(\boldsymbol{b}, P)$.

The recursion proceeds by structural induction on ϕ. In the base case, ϕ is a linear inequality $\boldsymbol{a} \cdot \boldsymbol{x} \geq c$ and according to Proposition II.3, it is sufficient in Line 3 to check whether $\boldsymbol{a} \cdot \boldsymbol{b} \geq c$ and $\boldsymbol{a} \cdot \boldsymbol{p} \geq 0$ for all $\boldsymbol{p} \in P$. In the recursion step, if $\phi = \psi^*(\boldsymbol{x})$, the algorithm guesses m generators of linear sets $M_i = L(\boldsymbol{c}_i, Q_i)$ in Line 5. The magnitude of each $L(\boldsymbol{c}_i, Q_i)$ can be bounded according to Proposition IV.5, and every $|Q_i|$ can also be bounded according to Corollary II.2. The algorithm then verifies in Line 6 whether $M_i \subseteq [\![\psi]\!]$ for every $i \in [m]$. If that is the case the algorithm determines $L(\boldsymbol{b}, P) \subseteq [\![\psi^*]\!]$ according to Lemma IV.2 via the previously guessed M_i. Finally, if ϕ is a Boolean combination of subformulas, Verify guesses subformulas ϕ_i that give a clause of the DNF of ϕ. Those ϕ_i could be some ψ^* or a linear inequality. To check whether $L(\boldsymbol{b}, P) \subseteq \bigcap_{i \in I} [\![\phi_i]\!]$, in Lines 10ff the algorithm recurses on each ϕ_i individually and checks the inclusion via Lemma IV.3.

Consequently, in order to decide whether ϕ of star height k is satisfiable, by Proposition IV.5 it suffices to guess some linear set $N = L(\boldsymbol{b}, P)$ such that $\|N\| \leq \|\phi\|^{|\phi|^{O(k)}}$ and invoke Verify$((\boldsymbol{b}, P), \phi)$. The depth of the recursion tree of Verify is bounded by $O(k)$, and due to Corollary II.2 and Proposition IV.5 in every call Verify guesses at most $|\phi|^{O(k)}$ linear sets of magnitude at most $\|\phi\|^{|\phi|^{O(k)}}$. Consequently, Verify is an NEXP procedure for arbitrary star height, and an NP procedure for fixed star height.

E. An NEXP *lower bound for* $\exists\mathsf{PA}^*$

Here, we show the NEXP lower bound for the satisfiability problem of $\exists\mathsf{PA}^*$ via a reduction from succinct circuit satisfiability (SC-SAT), which is known to be NEXP-complete [45].

Before we discuss the reduction, we illustrate a key idea by showing that $\exists\mathsf{PA}^*$ formulas may require solutions of doubly exponential magnitude. Suppose $\phi(x)$ is a Presburger formula that defines a single number, i.e., $\phi(x)$ is satisfied for exactly one $x \in \mathbb{N}$. Consider the formula

$$\phi' \equiv \exists x, y, y', z \colon \Big(\phi(x) \wedge y + z = 1 \wedge y' + z' = x \wedge (y = 1 \to y' = x) \wedge (z = 1 \to z' = x)\Big)^* \wedge y = 1 \wedge z = y'.$$

The formula under the star has two satisfying assignments: x always has the same value, and either $(y, y', z, z') = (1, x, 0, 0)$

or $(y, y', z, z') = (0, 0, 1, x)$. Since a satisfying assignment of ϕ' has to be a sum of such satisfying assignments, imposing $y = 1$ means the first case occurred exactly once, thus $y' = x$. Moreover, $z = y'$ means the second case occurred exactly $y' = x$ times, hence $z' = x^2$. Further, ϕ' only adds constant length (i.e. independent of ϕ) to ϕ. Thus, repeating this construction n times, starting with $\phi \equiv (x = 2)$, leads to an $\exists\mathsf{PA}^*$ formula of size linear in n that has 2^{2^n} as its only solution.

In order to encode an instance of SC-SAT, we will implement a universal quantifier in $\exists\mathsf{PA}^*$, stating that a certain assertion $\phi(x, i)$ holds for all $i \in [0, 2^n - 1]$. To this end, we will sum up assignments related to ϕ, but then we need finer control over which assignments occurred how many times. Moreover, this has to be achieved via conditions on the sums. Here, the following observation will be crucial.

Lemma IV.7. Let $a_0, \ldots, a_m \in \mathbb{N}$ and $b_0, \ldots, b_m \in \{0, 1\}$ with $\sum_{i=0}^{m} a_i = \sum_{i=0}^{m} b_i$. Then $\sum_{i=0}^{m} a_i \cdot 2^i = \sum_{i=0}^{m} b_i \cdot 2^i$ if and only if $a_i = b_i$ for every $i \in [0, m]$.

Note that Lemma IV.7 does not hold without the assumption $b_0, \ldots, b_m \in \{0, 1\}$: Without it, the lemma would say that a non-zero polynomial in $\mathbb{Z}[X]$ cannot have 1 and 2 as a root, but $(X - 1)(X - 2)$ is an obvious counter-example to that.

Let us now show how to encode SC-SAT by successively defining relevant $\exists\mathsf{PA}^*$ predicates. Below, m, n, k are fixed parameters for each formula, and the size of each formula will be polynomial in m, n, k:

1) We first define some auxiliary predicates expressible as $\exists\mathsf{PA}$ formulas; see also [28, Sec. 3.1] where the definition of such predicates is exposed in some more level of detail. In what follows, we abbreviate $\bigwedge_{i\in[0,n-1]} 0 \le y_i \le 1$ by $y_0, \ldots, y_{n-1} \in \{0, 1\}$, and $y = \langle y_0, \ldots, y_{n-1} \rangle$ abbreviates $y = \sum_{i=0}^{n-1} y_i \cdot 2^i \wedge y_0, \ldots, y_{n-1} \in \{0, 1\}$. Furthermore, we can define an $\exists\mathsf{PA}$ formula μ_n such that $\mu_n(u, v, w)$ holds whenever $u, v < 2^n$ and $w = u \cdot v$. The latter is equivalent to $u = \langle u_0, \ldots, u_{n-1} \rangle \wedge w = \sum_{i=0}^{n-1} u_i \cdot 2^i \cdot v$. Observe that we can express multiplication with u_i because $u_i \in \{0, 1\}$.

2) For given $y_1, \ldots, y_{n-1} \in \{0, 1\}$, our first $\exists\mathsf{PA}^*$ predicate $\textsc{BitPow}_n(x, y_0, \ldots, y_{n-1})$ is defined by induction on n and defines the predicate $x = 2^y$ with $y = \sum_{i=0}^{n-1} y_i$. For $n = 1$, $\textsc{BitPow}_1(x, y_0) \equiv (y_0 = 0 \to x = 1) \wedge (y_0 = 1 \to x = 2)$, and for $n > 1$ we define

$$\begin{aligned}
&\textsc{BitPow}_n(x, y_0, \ldots, y_{n-1}) \equiv y_0, \ldots, y_{n-1} \in \{0,1\} \wedge \\
&\quad \Big(\textsc{BitPow}_{n-1}(x', y'_1, \ldots, y'_{n-1}) \wedge z = 1 \wedge u + v = 1 \\
&\wedge u' + v' = x' \wedge (u = 1 \to u' = x') \wedge (v = 1 \to v' = x')\Big)^* \\
&\qquad \wedge \bigwedge_{i=1}^{n-1} y'_i = y_i \cdot z \wedge u = 1 \wedge v = u' \wedge x = 2^{y_0} \cdot v'.
\end{aligned}$$

Note that we can easily express $y'_i = y_i \cdot z$ and $x = 2^{y_0} \cdot v'$ in $\exists\mathsf{PA}$ as $y_i \in \{0, 1\}$ for all $i \in [0, n - 1]$. For the same reason, $y'_i = y_i \cdot z$ implies that in every satisfying assignment of the formula under the star, we have $y'_i = y_i$ for all $i \in [0, n - 1]$. The remaining reasoning is similar to the example for constructing 2^{2^n} above. Instead of producing x^2, we use the identity $\sum_{i=0}^{n-1} y_i \cdot 2^i = y_0 + 2 \cdot \sum_{i=1}^{n-1} y_i \cdot 2^{i-1}$.

Finally, we define $\textsc{Pow}_n(x, y)$ which defines the predicate $x = 2^y$ for all y below 2^n as $\textsc{Pow}_n(x, y) \equiv$

$$y = \langle y_0, \ldots, y_{n-1} \rangle \wedge \textsc{BitPow}_n(x, y_0, \ldots, y_{n-1}).$$

3) We now introduce a type of predicates that can be instantiated using other $\exists\mathsf{PA}^*$ formulas. It allows us to sum up a sequence of 2^n tuples, each of which satisfies some given predicate. Moreover, the given predicate has access to a loop variable i (which assumes the values $1, \ldots, 2^n$) and to parameters $y_1, \ldots, y_k < 2^n$. Suppose $\phi(i, x_1, \ldots, x_m, y_1, \ldots, y_k)$ is an $\exists\mathsf{PA}^*$ formula. Then, the predicate $\sum_{i<2^n}(x_1, \ldots, x_m)\colon\ \phi(i, x_1, \ldots, x_m, y_1, \ldots, y_k)$ states that $y_1, \ldots, y_k < 2^n$ and $x_j = \sum_{i=0}^{2^n-1} a_{i,j}$ for some numbers $a_{i,j}$ such that $\phi(i, a_{i,1}, \ldots, a_{i,m}, y_1, \ldots, y_k)$ for every $i \in [0, 2^n - 1]$. The variables before the colon designate which variables are to be added up (namely $x_1, \ldots, x_m$) and which are parameters that should be the same in every summand (namely $y_1, \ldots, y_k$):

$$\begin{aligned}
&\Big(\phi(i, x_1, \ldots, x_m, y'_1, \ldots, y'_m) \wedge \\
&\quad \bigwedge_{j\in[1,m]} y'_j = \langle y'_{j,1}, \ldots, y'_{j,n} \rangle \wedge u = 1 \wedge v = 2^i\Big)^* \wedge \\
&u = 2^n \wedge v = 2^{2^n} - 1 \wedge \bigwedge_{j\in[1,m]} y_j = \langle y_{j,1}, \ldots, y_{j,n} \rangle \wedge \\
&\qquad\qquad \bigwedge_{j\in[1,m],\ell\in[1,n]} y'_{j,\ell} = y_{j,\ell} \cdot u.
\end{aligned}$$

As in $\textsc{BitPow}_n$, we guarantee that in each satisfying assignment under the star, we have $y'_{j,\ell} = y_{j,\ell}$, which in turn means $y'_j = y_j$. This gives the formula under the star access to the parameters y_j. To see that i takes each value in $[0, 2^n - 1]$ exactly once, we use Lemma IV.7. Since we impose $v = 2^{2^n} - 1$ and $u = 2^n$ outside the star and $2^{2^n} - 1 = \sum_{i=0}^{2^n-1} 2^i$, the lemma yields that v has to assume each value 2^j for $j \in [0, 2^n - 1]$ exactly once. This implies the constraint on i.

4) We are now prepared to show how to access individual bits of large numbers. The predicate $\textsc{Bit}_n(x, y, z)$ evaluates to true whenever $x < 2^{2^n}$, $y < 2^n$, $z \in \{0, 1\}$, and the y-th bit in the binary expansion of x is z:

$$\begin{aligned}
&\textsc{Bit}_n(x, y, z) \equiv y < 2^n \wedge z \in \{0,1\} \wedge \\
&\quad \sum_{i<2^n} x\colon\ \big((x = 2^i \vee x = 0) \wedge \big(i = y \to x = z \cdot 2^i\big)\big)
\end{aligned}$$

Here, we assert that x has to be a sum of powers of 2 such that the y-th power has to occur if and only if $z = 1$.

5) The next predicate $\textsc{MultPow}_n(x, y, z)$ is true whenever $x = 2^y \cdot z$, $y < 2^n$, and $z < 2^{2^n}$: $\textsc{MultPow}_n(x, y, z) \equiv$

$$\begin{aligned}
&\sum_{i<2^{2n}} (x, z)\colon\ \big((i < y \to x = 0 \wedge z = 0) \wedge \\
&\qquad (i \ge y \to \exists b \in \{0,1\}\colon\ x = b \cdot 2^i \wedge z = b \cdot 2^{i-y})\big).
\end{aligned}$$

For $x = b \cdot 2^i$ and $z = b \cdot 2^{i-y}$, we use the fact that $b \in \{0,1\}$ and $\textsc{Pow}_{2n}$ to define 2^i and 2^{i-y}.

6) Let us now express that the binary expansion of $x < 2^{2^{m+n}}$ has a one at precisely those positions that are divisible by 2^n. Hence, $\textsc{Ones}_{n,m}(x)$ says that $x = \sum_{i=0}^{2^m-1} 2^{i \cdot 2^n}$:

$$\textsc{Ones}_{n,m}(x) \equiv \sum_{i<2^m} x\colon \big(x = 2^{i \cdot 2^n}\big)$$

For expressing $x = 2^{i \cdot 2^n}$, we use the formula μ_n defined above to express $i \cdot 2^n$ and the predicate $\textsc{Pow}_{2n}$.

7) We now define a predicate stating that the bit representation of a number x is a concatenation of 2^m copies of the binary representation $y < 2^{2^n}$. This is achieved by the $\exists\mathsf{PA}^*$ formula $\textsc{Repeat}_{n,m}(x,y)$ having satisfying assignments $y < 2^{2^n}$ and $x = \sum_{i=0}^{2^m-1} y \cdot 2^{i \cdot 2^n}$:

$$\begin{aligned}\textsc{Repeat}_{n,m}(x,y) \equiv \sum_{i<2^m} (x,y)\colon \Big(&\exists b \in \{0,1\} \\ &\exists z < 2^{2^{m+n}} : y = b \cdot 2^i \wedge \textsc{Ones}_{n,m}(z) \wedge x = b \cdot 2^i \cdot z\Big)\end{aligned}$$

Observe that each summand corresponds to one bit in the binary representation of y. Since we sum up shifts of numbers produced by $\textsc{Ones}_{n,m}$, that sum consists of 2^m copies of that representation. Note that we guarantee that the used bits are the ones of y because the sum of the $b \cdot 2^i$ has to be y outside of the sum. We can define $2^i \cdot z$ using $\textsc{MultPow}_n$ and then also multiply by b because $b \in \{0,1\}$.

8) We now express a *restricted* universal quantifier. As above, we define a predicate that is instantiated by another predicate ϕ. By $\forall^{n,m} i\colon \phi(x,i)$ we express that $x < 2^{2^n}$ and for every $i \in [0, 2^m - 1]$, we have $\phi(x,i)$. To this end, consider the following formula $\psi_{\phi,n,m}$:

$$\psi_{\phi,n,m}(y) \equiv \sum_{i<2^m} y\colon \ \Big(\exists z < 2^{2^n} : \phi(z,i) \wedge y = 2^{i \cdot 2^n} \cdot z\Big).$$

Observe that $\psi_{\phi,n,m}$ expresses that the binary representation of y is a concatenation of representations of numbers $x_0, \ldots, x_{2^m-1}$ such that for each $i \in [0, 2^m - 1]$, we have $\phi(x_i, i)$ and $x_i < 2^{2^n}$. Using $\textsc{Repeat}_{n,m}$, we make sure that $x_0 = \cdots = x_{2^m-1} = x$, and consequently define

$$\forall^{n,m} i\colon \phi(x,i) \equiv \exists y\colon \psi_{\phi,n,m}(y) \wedge \textsc{Repeat}_{n,m}(y,x).$$

Let us now provide a suitable definition of SC-SAT. A *circuit* (V, v_0, C, N) consists of a finite set V of *nodes*, a distinguished output node $v_0 \in V$, and two sets of hyper-edges: the *conjunctive edges* $C \subseteq V \times V \times V$ and the the *negation edges* $N \subseteq V \times V$. Edges are also called *gates*. We say that the circuit is *satisfiable* if there is an assignment $\eta\colon V \to \{0,1\}$ such that 1) $\eta(v_0) = 1$ and 2) for every $(u,v,w) \in C$, we have $\eta(w) = \min(\eta(u), \eta(v))$ and 3) for every $(u,v) \in N$, we have $\eta(v) = 1 - \eta(u)$.

A *succinct circuit representation* (n, α, ν) consists of a number n (given in unary) and Boolean formulas $\alpha(x_1, \ldots, x_n, y_1, \ldots, y_n, z_1, \ldots, z_n)$ and $\nu(x_1, \ldots, x_n, y_1, \ldots, y_n)$. The succinct circuit representation (n, α, ν) describes the circuit $C(n, \alpha, \nu) = (V, v_0, C, N)$, which is defined as follows. We have $V = \{0,1\}^n$, $v_0 = (0, \ldots, 0)$, and $C = \{(x,y,z) \in V \times V \times V : \alpha(x,y,z)\}$ and $N = \{(x,y) \in V \times V : \nu(x,y)\}$. Here, x, y, and z is short for $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$, and $z_1, \ldots, z_n$, respectively.

The *succinct circuit satisfiability* problem (SC-SAT) is defined as follows:

Given: A succinct circuit representation (n, α, ν).
Question: Is the circuit $C(n, \alpha, \nu)$ satisfiable?

We show that satisfiability of $\exists\mathsf{PA}^*$ is NEXP-hard by reducing SC-SAT, which is NEXP-complete, see e.g. [45]. Suppose (n, α, ν) is a given succinct circuit. First, we turn α and ν into formulas $\bar{\alpha}$ and $\bar{\nu}$ in $\exists\mathsf{PA}$ so that $\alpha(x_1, \ldots, x_n, y_1, \ldots, y_n, z_1, \ldots, z_n)$ if and only if $\bar{\alpha}(\sum_{i=1}^n x_i \cdot 2^{i-1}, \sum_{i=1}^n y_i \cdot 2^{i-1}, \sum_{i=1}^n z_i \cdot 2^{i-1})$ does *not* hold; analogously for $\bar{\nu}$. Then the following statement is expressible in an $\exists\mathsf{PA}^*$ formula of size polynomial in n:

$$\exists x < 2^{2^n} :$$

$$\begin{aligned}
\forall^{n,3n} i\colon \Big(&\exists u,v,w < 2^n : i = u + 2^n \cdot v + 2^{2n} \cdot w && (2)\\
&\wedge \textsc{Bit}_n(x,u,u') \wedge \textsc{Bit}_n(x,v,v') && (3)\\
&\wedge \textsc{Bit}_n(x,w,w') && (4)\\
&\wedge \big(\bar{\alpha}(u,v,w) \vee \textsc{Min}(w',u',v')\big)\Big) && (5)\\
\wedge\ \forall^{n,2n} i\colon \Big(&\exists u,v < 2^n : i = u + 2^n \cdot v && (6)\\
&\wedge \textsc{Bit}_n(x,u,u') \wedge \textsc{Bit}_n(x,v,v') && (7)\\
&\wedge \big(\bar{\nu}(u,v) \vee v' = 1 - u'\big)\Big) && (8)\\
\wedge\ \textsc{Bit}_n&(x,0,1) && (9)
\end{aligned}$$

Here, $\textsc{Min}(x,y,z)$ is an $\exists\mathsf{PA}$ formula defining $x = \min(y,z)$. Let us explain why the formula above is equivalent to satisfiability of $C(n, \alpha, \nu)$. We encode the assignment of the 2^n nodes in $C(n, \alpha, \nu)$ in the number $x < 2^{2^n}$. In Eqs. (2) to (5), we express that the assignment is consistent with all the conjunctive gates. Note that we cannot directly use a formula of the form $\forall^{n,n} i_1\colon \forall^{n,n} i_2\colon \forall^{n,n} i_3\colon \ldots$, because we would need to use the value of i_1 (and i_2) in the inner formula (but our quantifier $\forall^{n,m}$ allows only two free variables). Therefore, i ranges over $[0, 2^{3n} - 1]$ and is decomposed into $u, v, w \in [0, 2^n - 1]$ in Eq. (2). In Eqs. (3) and (4), we extract the bits at the nodes identified by u, v and w, and in Eq. (5), we require that if there is a conjunctive gate with input nodes u,v and output node w, then the node w carries the conjunction of nodes u and v. In Eqs. (6) to (8), we do the same for the negation gates. Finally, in Eq. (9), we state that the output node is indeed assigned value 1.

V. Translating $\exists\mathsf{PA}^*$ to $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$

We now outline a polynomial-time reduction from the $\exists\mathsf{PA}^*$ satisfiability problem to reachability in $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$. In fact, we prove a slightly stronger statement and show that sets of natural numbers definable in $\exists\mathsf{PA}^*$ are definable by $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$. Given $M \subseteq \mathbb{N}^d$, we say that M is $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$*-definable* if there is a $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ V in dimension $m + d$ with designated control states q, r such that $\boldsymbol{v} \in M$ if and only if $(q, \boldsymbol{0}) \xrightarrow{*} (r, (\boldsymbol{0}, \boldsymbol{v}))$. We moreover require that V only performs zero-tests on the first m counters.

Proposition V.1. *There is a polynomial-time reduction from $\exists\mathsf{PA}^*$ to $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$, i.e., for any $\exists\mathsf{PA}^*$ formula ϕ, $[\![\phi]\!]$ is definable by a $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ V. If ϕ has star height k then V has k nested zero tests.*

We show Proposition V.1 by structural induction on ϕ. The idea is inspired by the proof of the coNEXP lower bound for the inclusion problem for $\mathbb{Z}$-VASS [29, Lem. 12].

The induction base case, where $\phi \equiv \boldsymbol{a} \cdot \boldsymbol{x} \geq c$ is a linear inequality such that $\boldsymbol{a}^\intercal = (a_1, \ldots, a_d) \in \mathbb{Z}^d$, can easily be dealt with. For $i \in [d]$, define $\boldsymbol{a}_i = (a_i, \boldsymbol{e}_i) \in \mathbb{Z}^{d+1}$, where $\boldsymbol{e}_i$ denotes the i-th unit vector in dimension d. The desired $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ V non-deterministically adds $\boldsymbol{a}_i$ to the counters before non-deterministically decreasing the first counter by a value of at least c.

In order to deal with Boolean connectives, we first sketch how to obtain a gadget C in dimension $3d$ that checks for $i \in [d]$ whether the contents of the counters i and $d+i$ coincide, and at the same time copies the contents of counter i to counter $2d+i$. For $i \in [d]$, let $\boldsymbol{v}_i \in \mathbb{Z}^{3d}$ be all zero except for the i-th and $(d+i)$-th components which both equal -1, and the $(2d+i)$-th component which equals 1. The desired $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ C is then obtained by consecutively non-deterministically adding multiples of either $\boldsymbol{v}_i$ or $-\boldsymbol{v}_i$ for all $i \in [d]$.

For a conjunction $\phi \equiv \phi_1 \wedge \phi_2$, let V_1 and V_2 be $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ with at most k different zero tests defining $[\![\phi_1]\!]$ and $[\![\phi_2]\!]$, respectively. Let $\hat{V}_1 = (Q_1, Z_1, E_1)$ and $\hat{V}_2 = (Q_2, Z_2, E_2)$ be obtained from V_1 and V_2 such that $\hat{V}_1$ is equivalent to V_1, $\hat{V}_2$ is equivalent to V_2, $Z_1 = Z_2$, $|Z_1| = |Z_2| = k$, and counters not tested for zero in V_1 and V_2 do not overlap in $\hat{V}_1$ and $\hat{V}_2$. The $\mathbb{Z}$-VASS V defining $[\![\phi]\!]$ now sequentially composes $\hat{V}_1$, $\hat{V}_2$ and C. The correctness of the construction is easily seen. Observe that a disjunction $\phi \equiv \phi_1 \vee \phi_2$ can be dealt with analogously, by non-deterministically branching into $\hat{V}_1$ and $\hat{V}_2$, and by adapting the gadget C appropriately. Note that the construction ensures that any $\exists\mathsf{PA}^*$ formula of star height zero is definable by a $\mathbb{Z}$-VASS.

Finally, we provide a construction for formulas of the form ϕ^*. Let V_ϕ in dimension $m + d$ define $[\![\phi]\!]$, and let $\hat{V}_\phi$ be equivalent to V working in dimension $m + 2d$. Furthermore, for $i \in [d]$ let $\boldsymbol{w}_i \in \mathbb{Z}^{m+2d}$ be all zero except for the $(m+i)$-th component which equals -1, and the $(m+d+i)$-th component which equals 1. Then V depicted in Fig. 1 defines $[\![\phi^*]\!]$. Starting in q, V traverses $\hat{V}_\phi$ thereby pushing some $\boldsymbol{v} \in [\![\phi]\!]$ onto the counters $[m+1, m+d]$. Next, V adds the contents of the counters $[m+1, m+d]$ to the counters $[m+d+1, m+2d]$, similar to the gadget C above. When reaching r, V can repeat this process an arbitrary number of times (and also zero times as required by the definition of the star operator).

Observe that this construction ensures that the $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ $V = (Q, Z, E)$ defining an $\exists\mathsf{PA}^*$ formula ϕ has $|Z \setminus \{0\}| = k$ if and only if ϕ has star height k.

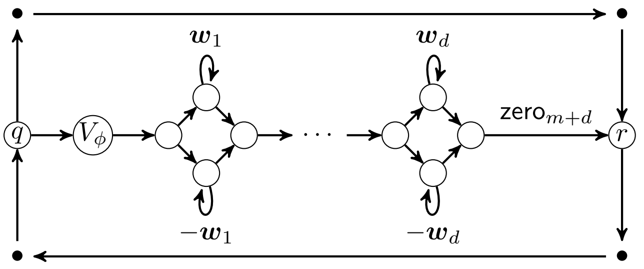

Fig. 1. $\mathbb{Z}$-VASS defining $[\![\phi]\!]^*$, where $[\![\phi]\!]$ is defined by V.

VI. Translating $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ to $\mathsf{RatMP}^{\mathsf{tf}}$

In this section, we show the following.

Proposition VI.1. *There is a polynomial reduction from $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}$ to $\mathsf{RatMP}^{\mathsf{tf}}$. Moreover, this reduction maps instances from $\mathbb{Z}$-$\mathsf{VASS}^{\mathsf{nz}}_k$ to $\mathsf{RatMP}^{\mathsf{tf}}_k$.*

We describe the construction, but defer the correctness proof to Appendix B. Let $V = (Q, Z, E)$ be a given d-dimensional $\mathbb{Z}$-VASS with k nested zero-tests and let $(q, \boldsymbol{v}), (q', \boldsymbol{v}')$ be configurations. We may assume that $\boldsymbol{v} = \boldsymbol{v}' = \boldsymbol{0}$. To define the graph group, let $A = \{a_1, \ldots, a_d\} \cup \{b_i : i \in Z \setminus \{0\}\}$ and $I = \{\{a_i, b_j\}, \{a_i, a_j\} \in \mathcal{P}_2(A) : i > j\}$. This is a transitive forest, because with $I' = \{\{a_{j+1}, b_j\}, \{a_{j+1}, a_j\} : j \in [1, d-1]\}$, the graph (A, I') is a forest that yields (A, I) (see Fig. 2). Moreover, we have $\beta(A, I) = k$: If $Z \setminus \{0\} = \{m_1, \ldots, m_k\}$, then we can build (A, I) by m_1 times adding a universal vertex, then a disjoint union with b_{m_1}, then m_2 times adding a universal vertex, etc. In total, we have taken k disjoint unions.

The idea is that since the a_i commute pairwise, they generate a group isomorphic to $\mathbb{Z}^d$, which realises the counters. Since b_{m_i} commutes with $a_{m_i+1}, \ldots, a_d$ but not with $a_1, \ldots, a_{m_i}$, each generator b_{m_i} serves to zero-test counters $a_1, \ldots, a_{m_i}$: It cannot be moved past a non-zero product of them.

From V, we construct the automaton $\mathcal{A}$ over $\mathbb{G}(A, I)$ as follows. It has states $Q' = Q \cup \{q_0\}$ and an edge (q_0, b_i, q_0) for each $i \in [1, d]$ and one edge (q_0, ε, q). Moreover, it has an edge $(p, b_\ell^{-1} a_1^{u_1} a_2^{u_2} \cdots a_d^{u_d}, p')$ for each $(p, (u_1, \ldots, u_d), \ell, p') \in E$. Finally, as the input word $w \in (A^{\pm})^*$, we choose $w = \varepsilon$.

VII. Translating $\mathsf{RatMP}^{\mathsf{tf}}$ to $\exists\mathsf{PA}^*$

In this section, we show the following:

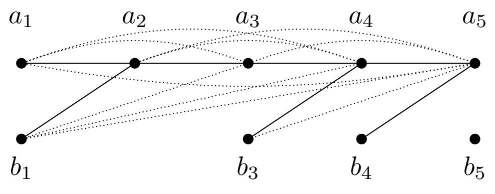

Fig. 2. Independence alphabet used in the proof of Proposition VI.1 in the case of $d = 5$ and $Z = \{0, 1, 3, 4, 5\}$. Solid edges are those in I', dotted edges are those only in I.

Proposition VII.1. *There is a polynomial-time reduction from* $\mathsf{RatMP}^{\mathsf{tf}}$ *to* $\exists\mathsf{PA}^*$. *Moreover, this reduction maps instances of* $\mathsf{RatMP}^{\mathsf{tf}}_k$ *to* $\exists\mathsf{PA}^*$ *formulas with star-height* $\leq k$.

The central notion in our translation is that of valence automata. What distinguishes them from automata over groups is that they have an input alphabet. This turns out to be convenient for our recursive algorithm.

Let G be a group. A *valence automaton over* G is a tuple $\mathcal{A} = (Q, \Sigma, E)$, where Q is a finite set of *states*, Σ is an alphabet, and $E \subseteq Q \times G \times \Sigma^* \times Q$ is a finite set of *edges*. A *configuration of* $\mathcal{A}$ is a triple $(q, g, w) \in Q \times G \times \Sigma^*$. The step relation is defined as follows. We have $(q, g, w) \to_{\mathcal{A}} (q', g', w')$ if there is an edge (q, h, u, q') such that $g' = gh$ and $w' = wu$. This induces a language for each pair of states:

$$\mathsf{L}(\mathcal{A}, p, q) = \{w \in \Sigma^* : (p, 1, \varepsilon) \to^*_{\mathcal{A}} (q, 1, w)\}.$$

The *non-emptiness problem* for a valence automaton asks, given a valence automaton $\mathcal{A}$ and states $p, q \in Q$, whether $\mathsf{L}(\mathcal{A}, p, q) \neq \emptyset$.

As observed in [36], one can reduce the rational subset membership problem for G to the non-emptiness problem of valence automata over G: Given an automaton $\mathcal{A}$ over G and $g \in G$, one can easily construct an automaton $\mathcal{A}_g$ over G with $\mathsf{L}(\mathcal{A}') = \mathsf{L}(\mathcal{A})g^{-1}$. Then, $1 \in \mathsf{L}(\mathcal{A}')$ if and only if $g \in \mathsf{L}(\mathcal{A})$. Moreover, we can turn $\mathcal{A}'$ into a valence automaton $\mathcal{A}''$ by adding a label, say, the letter a to every edge. Then, $\mathsf{L}(\mathcal{A}'', q_0, q_f) \neq \emptyset$ if and only if $g \in \mathsf{L}(\mathcal{A})$. We will therefore reduce non-emptiness of valence automata to $\exists\mathsf{PA}^*$.

Remark VII.2. Note that (unlike, e.g. [56]), we do not use initial and final states in valence automata. This is because we will construct an $\exists\mathsf{PA}^*$ formula that describes the Parikh image of a valence automaton for any choice of an initial and a final state. This will allow us to recurse into fewer subinstances in our translation.

Let Σ be an alphabet. The *Parikh map* $\Psi\colon \Sigma^* \to \mathbb{N}^\Sigma$ assigns each word $w \in \Sigma^*$ the vector $\Psi(w)$ with $\Psi(w)(a) = |w|_a$, where $|w|_a$ is the number of occurrences of a in w. The *Parikh image* of a language $L \subseteq \Sigma^*$ is $\Psi(L) = \{\Psi(w) : w \in L\}$, hence $\Psi(L) \subseteq \mathbb{N}^\Sigma$. We will also use a notion of Parikh images of valence automata, which encodes the Parikh image for every choice of initial and final state. For any two states $p, q \in Q$, let $\triangle^q_p$ be a fresh symbol and let $\Delta = \{\triangle^q_p : p, q \in Q\}$. The *Parikh image* of $\mathcal{A}$, denoted $\Psi(\mathcal{A})$, is defined as $\bigcup_{p,q\in Q} \triangle^q_p + \Psi(\mathsf{L}(\mathcal{A}, p, q))$. Hence, we have $\Psi(\mathcal{A}) \subseteq \mathbb{N}^{\Delta\cup\Sigma}$. Our reduction consists in the following.

Proposition VII.3. *Given a transitive forest* (A, I) *and a valence automaton* $\mathcal{A}$ *over* $\mathbb{G}(A, I)$, *one can construct in polynomial time an* $\exists\mathsf{PA}^*$ *formula* Φ *with* $[\![\Phi]\!] = \Psi(\mathcal{A})$ *and star height at most* $\beta(A, I)$.

Note that if this is established, we can reduce non-emptiness of valence automata over $\mathbb{G}(A, I)$ to $\exists\mathsf{PA}^*$: Given a valence automaton $\mathcal{A}$ over $\mathbb{G}(A, I)$, construct an $\exists\mathsf{PA}^*$ formula Φ with $[\![\Phi]\!] = \Psi(\mathcal{A})$. Then $\mathsf{L}(\mathcal{A}, q_0, q_f) \neq \emptyset$ iff the formula $\Phi \wedge \triangle^{q_f}_{q_0} > 0$ is satisfiable. The remainder of this section is devoted to proving Proposition VII.3. The construction is recursive with respect to the decomposition of transitive forests described in Section III. Hence, we need to treat three cases: (A, I) has a universal vertex, (A, I) is a disjoint union, or (A, I) is empty.

Clearly, we may assume that in $\mathcal{A}$, every edge is of the form $(p, [a], x, q)$ with $a \in A^{\pm 1} \cup \{\varepsilon\}$ and $x \in \Sigma$.

A. Universal vertex

We first consider the case that the independence alphabet (A, I) has a universal vertex. Suppose there is $u \in A$ such that $\{u, a\} \in I$ for every $a \in A$. Let $A' = A \setminus \{u\}$ and $I' = I \cap \mathcal{P}_2(A')$. Then the morphism $\mathbb{G}(A, I) \to \mathbb{G}(A', I') \times \mathbb{Z}$, $[w] \mapsto ([\pi_{A'}(w)], [\pi_{\{u\}}(w)])$, is an isomorphism. This means we may regard $\mathcal{A}$ as a valence automaton over $\mathbb{G}(A', I')$ with an additional integer counter. Therefore, we construct a formula for $\Psi(\mathcal{A})$ by first computing a formula for a valence automaton over $\mathbb{G}(A', I')$ and then imposing the condition that the additional counter be zero at the end.

We construct the valence automaton $\mathcal{A}' = (Q', \Sigma', E')$ over $\mathbb{G}(A', I')$. We modify Σ slightly so as to encode the effect of the additional counter: For each $x \in \Sigma$ and $e \in \{-1, 0, 1\}$, let x_e be a fresh symbol. Recall that all edges in $\mathcal{A}$ are of the form $(p, [a], x, q)$ with $a \in A^{\pm 1} \cup \{\varepsilon\}$ and $x \in \Sigma$. We obtain $\mathcal{A}'$ from $\mathcal{A}$ by replacing 1) each edge of the form $(p, [u^e], x, q)$, $e \in \{-1, 1\}$, with the edge $(p, 1, x_e, q)$ and 2) each edge of the form $(p, [a], x, q)$, $a \in A'^{\pm 1} \cup \{\varepsilon\}$, with the edge $(p, [a], x_0, q)$. The alphabet Σ' of course consists of the symbols x_e that occur in these edges. Then, $\mathcal{A}'$ is indeed over $\mathbb{G}(A', I')$.

Suppose Φ' is a formula for $\Psi(\mathcal{A}')$. Consider the formula

$$\Phi \equiv \exists\{x_e \in \Sigma' : x \in \Sigma\}\colon \Phi' \wedge \sum_{x_1 \in \Sigma'} x_1 = \sum_{x_{-1} \in \Sigma'} x_{-1} \wedge \bigwedge_{x\in\Sigma} x = x_{-1} + x_0 + x_1. \quad (10)$$

Slightly abusing notation, if in the sum $x_{-1} + x_0 + x_1$, one of the symbols is not in Σ', then it is meant not to occur in the sum. Then we clearly have $[\![\Phi]\!] = \Psi(\mathcal{A})$:

Step I. Suppose (A, I) has a universal vertex $u \in A$.

1) Construct $\mathcal{A}'$ and recursively compute Φ' for $\Psi(\mathcal{A}')$.
2) Define Φ as in Eq. (10) and return Φ.

B. Disjoint union

We now treat the case that (A, I) is a disjoint union of (A_0, I_0) and (A_1, I_1) (we do not assume that (A_0, I_0) and (A_1, I_1) are connected).

Modified automata. In order to build the formula for $\mathcal{A}$, we apply our translation recursively to two automata $\mathcal{A}_0$ and $\mathcal{A}_1$ over (A_0, I_0) and (A_1, I_1), respectively. For each $p, q \in Q$, let $\square^q_p$ be a fresh symbol and let $\square = \{\square^q_p : p, q \in Q\}$. The automaton $\mathcal{A}_i$ is obtained by removing in $\mathcal{A}$ all edges labelled with A_{1-i} and then adding, for each $p, q \in Q$, an edge $(p, 1, \square^q_p, q)$. Hence, $\square \cup \Sigma$ is the input alphabet for both.

Grammars. Let $\mathcal{C}$ be a language class. A *$\mathcal{C}$-grammar* is a tuple $G = (N, T, P)$, where N is a finite set of *nonterminals*, T is a finite set of *terminals*, and P is a finite set of pairs (B, L), where $B \in N$ and L is from $\mathcal{C}$. A pair (B, L) is also denoted $B \to L$. We write $u \Rightarrow_G v$ if there is a pair (B, L) and words $x, y, w \in (N \cup T)^*$ such that $u = xBy$, $v = xwy$, and $w \in L$. Whenever $B \to L \in P$ and $w \in L$, we say that the pair $B \to w$ is a *production* of G. If G has finitely many productions, we call it *context-free*. The *language generated by G from $B \in N$* is $\mathsf{L}(G, B) = \{w \in T^* : B \Rightarrow_G^* w\}$.

The grammar $G_{\mathcal{A}}$ is constructed as follows. It has $Q = \Box$ as its set of nonterminals and for every $p, q \in Q$, we have the productions $\Box_p^q \to \mathsf{L}(\mathcal{A}_i, p, q)$ for $i \in \{0, 1\}$.

Proposition VII.4. *For every $p, q \in Q$, we have $\mathsf{L}(G_{\mathcal{A}}, \Box_p^q) = \mathsf{L}(\mathcal{A}, p, q)$.*

Presburger formulas. Now that we have expressed the languages of $\mathcal{A}$ in terms of grammars, we can build upon methods for translating grammars into Presburger formulas.

The *Parikh image* of G is defined as $\Psi(G) = \bigcup_{B \in N} \hat{B} + \Psi(\mathsf{L}(G, B))$. Note that $\Psi(G) \subseteq \mathbb{N}^{\hat{N} \cup T}$. Our construction will build an $\exists\mathsf{PA}^*$ formula for $\Psi(G)$ from an $\exists\mathsf{PA}^*$ formula describing the right-hand sides in G as follows. The *production Parikh image* of G is defined as $\Psi^{\rightarrow}(G) = \bigcup_{B \to L \in P} \bar{B} + \Psi(L)$. Hence, we have $\Psi^{\rightarrow}(G) \subseteq \mathbb{N}^{\bar{N} \cup N \cup T}$.

To build an $\exists\mathsf{PA}^*$ formula for the Parikh image of G, we extend a construction by Verma, Seidl, and Schwentick [52], which yields an existential Presburger formula for a context-free grammar. We recall the latter in a slightly adapted fashion.

Suppose $G = (N, T, P)$ is context-free with productions $B_1 \to w_1, \ldots, B_\ell \to w_\ell$. For each symbol $x \in N \cup T$, the formula has a variable that counts how often that symbol is produced during a derivation. For each $B \in N$, it also has a variable $\bar{B}$ counting how often B is consumed. It has variables $p_1, \ldots, p_\ell$ that count how often each production is applied. The variables $\hat{B}$ for $B \in N$ signal which nonterminal is used as the start symbol. In the construction, we use two shorthands: First, $\delta_{B,C}$ stands for 1 if B and C are the same variable (i.e. not only carrying the same value) and 0 otherwise. Second, we use a ternary operator to construct the term $L?t_1{:}t_2$, where L is a literal and t_1 and t_2 are ordinary terms. The value of this term is t_1 if L is true and t_2 otherwise.

The formula has three conjuncts, λ, κ, and χ. The conjunct λ expresses that the number of times each nonterminal is produced and consumed is consistent with the number of applications of each production:

$$\begin{aligned} \lambda \equiv \exists p_1, \ldots, p_\ell\colon & \bigwedge_{B \in N} \bar{B} = \sum_{i \in J_B} p_i \\ & \wedge \bigwedge_{x \in N \cup T} x = \sum_{i \in [1,\ell]} \Psi(w_i)(x) \cdot p_i \\ & \wedge \bigwedge_{B,C \in N} X_{B,C} = \sum_{i \in J_B} \Psi(w_i)(C) \cdot p_i. \end{aligned} \tag{11}$$

Here, we use $J_B = \{i \in [1, \ell] : B_i = B\}$. Moreover, for each $B, C \in N$, λ stores in the variable $X_{B,C}$ whether there is a production with B on the left-hand side and C on the right-hand side. ($X_{B,C}$ even records the number of such occurrences.)

The formula κ selects exactly one of the nonterminals B as the start symbol, i.e. sets $\hat{B} = 1$ and $\hat{C} = 0$ for every $C \neq B$. Moreover, it requires each nonterminal to be consumed ($\bar{C}$) as many times as it is produced (C). Except for the start symbol, which is consumed once more:

$$\kappa \equiv \bigvee_{B \in N} \bigwedge_{C \in N} \hat{C} = \delta_{B,C} \wedge \bar{C} = C + \hat{C}$$

The formula χ states that every nonterminal is reachable from the start symbol: In the directed graph (N', E), whose vertices $N' \subseteq N$ are the used nonterminals, and with an edge $B \to C$ if $X_{B,C} > 0$, every vertex is reachable from the start symbol. To this end, we assign to every nonterminal B a distance d_B so that the start symbol has $d_B = 0$ and every used nonterminal B is directly reachable from some C with $d_B = d_C + 1$:

$$\begin{aligned} \chi \equiv \exists\{d_B : B \in N\}\colon \bigwedge_{B \in N} \Bigg(& \bar{B} = 0 \vee (\hat{B} = 1 \wedge d_B = 0) \\ & \vee \Bigg(\hat{B} = 0 \wedge \bigvee_{C \in N} X_{C,B} > 0 \wedge d_B = d_C + 1 \Bigg) \Bigg). \end{aligned} \tag{12}$$

The complete formula is then $\Phi \equiv \exists_{B,C \in N} X_{B,C} \exists_{B \in N} B\colon \lambda \wedge \kappa \wedge \chi$. The conditions in λ, κ, χ are clearly necessary and Verma, Seidl, and Schwentick [52] have shown sufficiency.

Theorem VII.5 ([52])**.** *If G is context-free as above, then $\Psi(G) = [\![\Phi]\!]$.*

The *key observation* in our reduction is that using the Kleene star, we can modify Φ to build an $\exists\mathsf{PA}^*$ formula when G is not necessarily context-free, and in particular for $\Psi(\mathcal{A})$. As a first step, note that λ is equivalent to the following:

$$\begin{aligned} \lambda_* \equiv \Bigg(\bigvee_{i \in [1,\ell]} \Bigg(& \bigwedge_{j \in [1,\ell]} \bar{B}_j = \delta_{i,j} \wedge \bigwedge_{x \in N \cup T} x = \Psi(w_i)(x) \Bigg) \\ & \wedge \bigwedge_{B,C \in N} X_{B,C} = (\bar{B} = 1) \mathbin{?} C : 0 \Bigg)^*. \end{aligned} \tag{13}$$

This is because each satisfying assignment of the formula under the star assigns variables as λ would, if only one production were applied. With the star, we generate all satisfying assignments of λ. Hence, with $\Phi_* \equiv \exists_{B,C \in N} X_{B,C} \exists_{B \in N} B\colon \lambda_* \wedge \kappa \wedge \chi$, we also have $[\![\Phi_*]\!] = \Psi(G)$. Since this formulation does not require a separate variable per production, it can be modified for infinite sets of productions. Suppose φ is a formula with $[\![\varphi]\!] = \Psi^{\rightarrow}(G)$ and consider the formula

$$\lambda_*^{\varphi} \equiv \Bigg(\varphi \wedge \bigwedge_{B,C \in N} X_{B,C} = (\bar{B} = 1 \mathbin{?} C : 0) \Bigg)^*$$

and let $\Phi_\varphi \equiv \exists_{B,C \in N} X_{B,C} \exists_{B \in N} B\colon \lambda_*^{\varphi} \wedge \kappa \wedge \chi$. Now the following is an easy consequence of Theorem VII.5.

Proposition VII.6. *If $[\![\varphi]\!] = \Psi^{\rightarrow}(G)$, then $[\![\Phi_\varphi]\!] = \Psi(G)$.*

Proof. Suppose $\mu \in [\![\Phi_\varphi]\!]$. Then, finitely many satisfying assignments of φ justify the satisfaction of Φ_φ. These satisfying assignments stem from finitely many productions in G. Hence, if we build the formula Φ'_* for the context-free grammar G' that consists of just these productions, then μ must satisfy Φ'_*. However, every derivation in G' is in particular possible in G, which means $\mu \in \Psi(G)$. For the converse, suppose $\mu \in \Psi(G)$. Then μ stems from a derivation in G, which can only involve a finite set of productions. Hence, $\mu \in \Psi(G')$ for the context-free grammar G' that contains only these productions. If we construct Φ'_* as above for G', then we have $\mu \in [\![\Phi'_*]\!]$. However, since then every satisfying assignment of the λ_* conjunct in Φ'_* also satisfies λ^φ_*, this implies $\mu \in [\![\Phi_\varphi]\!]$. □

We have thus built a formula for $\Psi(G_\mathcal{A})$ and hence $\Psi(\mathcal{A})$. If we have a formula Φ_i for $\Psi(\mathcal{A}_i)$, then renaming the free variables $\triangle^q_p$ into $\bar{\Box}^q_p$ yields formulas $\bar{\Phi}_i$ with $[\![\bar{\Phi}_0 \vee \bar{\Phi}_1]\!] = \Psi^{\rightarrow}(G_\mathcal{A})$. By Proposition VII.6, we can proceed as follows:

Step II. Let (A, I) be a disjoint union of (A_0, I_0) and (A_1, I_1).

1) For $i \in \{0, 1\}$, compute $\mathcal{A}_i$ and recursively compute $\exists\mathsf{PA}^*$ formula Φ_i for $\Psi(\mathcal{A}_i)$.
2) For $i \in \{0, 1\}$, obtain $\bar{\Phi}_i$ from Φ_i by renaming variables $\triangle^q_p$ to $\overline{\Box^q_p}$ and set $\varphi \equiv \bar{\Phi}_0 \vee \bar{\Phi}_1$.
3) Obtain $\Phi_\mathcal{A}$ from Φ_φ by renaming variables $\widehat{\Box^q_p}$ to $\triangle^q_p$.

C. Empty graph

If (A, I) is the empty graph, then valence automata over $\mathbb{G}(A, I)$ are just finite automata, for which we can build a Presburger formula using known methods. We construct the context-free grammar $G_\mathcal{A}$ with a productions $\Box^q_p \to x\Box^q_r$ for each edge $(p, x, 1, r)$ and $\Box^p_p \to \varepsilon$ for every $p \in Q$. Then, we construct the formula Φ for $G_\mathcal{A}$ as in Theorem VII.5.

Step III. Suppose (A, I) is empty.

1) Convert $\mathcal{A}$ into context-free grammar $G_\mathcal{A}$.
2) Compute Presburger formula for $G_\mathcal{A}$ as in Theorem VII.5.

A detailed complexity analysis of Steps I to III can be found in Appendix C-A. Since before invoking recursion, the automata are expanded in their alphabets (Step I) or set of edges (Step II), we have to show that these remain polynomial during the linear-depth recursion. We argue that the number n of states is never changed, the alphabet size is at most the number of edges, and we add at most n^2 edges in each step. Hence, every automaton occurring in the algorithm has at most $|A| \cdot n^2$ edges more than the initial one.

VIII. Remaining complexities for $\mathsf{RatMP}^{\mathsf{tf}}$

In this section, we prove Theorem III.4. The following Lemma is essentially due to Kambites, Silva, and Steinberg, who stated a version about decidability.

Lemma VIII.1 ([36]). *For f.g. groups G, the following problems are logspace inter-reducible: rational subset membership for G and non-emptiness for valence automata over G.*

We begin with the case that (A, I) is a clique. Then $\mathbb{G}(A, I) \cong \mathbb{Z}^{|A|}$. NL-hardness is trivially inherited from reachability in directed graphs. For membership in NL, we apply Lemma VIII.1 and observe that a valence automaton over $\mathbb{Z}^k$ is a blind k-counter automaton, which can be transformed into a 1-reversal-bounded $(k+1)$-counter automaton [34]. Since k is fixed here, reachability is known to be in NL [26].

Now consider the case that (A, I) is a disjoint union of (at least two) cliques. Since (A, I) is not a clique, it contains two non-adjacent vertices. Thus, F_2 is a subgroup of $\mathbb{G}(A, I)$. Hence the following version of the Chomsky-Schützenberger theorem [4, 35], together with Lemma VIII.1 tells us that the P-hard problem of deciding non-emptiness for context-free languages (CFL) [24] reduces to membership of rational subsets of $\mathbb{G}(A, I)$.

Lemma VIII.2 ([4, 35]). *Given a CFL L, one can construct in logspace a valence automaton over F_2 that accepts L.*

For a P algorithm, it suffices to decide non-emptiness of a valence automaton $\mathcal{A}$ over a disjoint union of cliques. Let k be the size of the largest clique. Observe that Proposition VII.4 tells us that given $\mathcal{A}$, we can construct a $\mathcal{C}$-grammar for $\mathcal{A}$, where $\mathcal{C}$ is the class of languages accepted by valence automata over $\mathbb{Z}^k$. As shown above, non-emptiness for $\mathcal{C}$ is in NL. Due to closure under regular intersection, we can even decide in NL whether a given language from $\mathcal{C}$ intersects a given regular language. Under these conditions, one can decide non-emptiness of $\mathcal{C}$-grammars in polynomial time with a simple saturation procedure as for context-free grammars [56, p. 25].

In light of Corollary III.3, the only remaining piece is NP-hardness of $\mathsf{RatMP}^{\mathsf{tf}}$ in case (A, I) is not a disjoint union of cliques. Observe that the latter is equivalent to saying that the adjacency relation is not transitive. In particular, (A, I) contains the graph •—•—• as an induced subgraph. This means $\mathbb{G}(A, I)$ has $F_2 \times \mathbb{Z}$ as a subgroup, so that it suffices to show NP-hardness of rational subset membership for $F_2 \times \mathbb{Z}$.

To this end, we reduce the subset sum problem to non-emptiness of valence automata over $F_2 \times \mathbb{Z}$. Given $x_1, \ldots, x_n, y \in \mathbb{N}$ in binary, one readily constructs a context-free grammar for the language $S = \{a^{x_1}, \varepsilon\} \cdot \{a^{x_2}, \varepsilon\} \cdots \{a^{x_n}, \varepsilon\} \cdot \{b^y\}$. With Lemma VIII.2 we build a valence automaton $\mathcal{A}$ over F_2 accepting S. From that, one easily obtains a valence automaton $\mathcal{B}$ over $F_2 \times \mathbb{Z}$ accepting $S' = \{w \in S : |w|_a = |w|_b\}$. Then the subset sum instance is positive if and only if $S' \neq \emptyset$. Thus, non-emptiness of valence automata over $F_2 \times \mathbb{Z}$ is NP-hard.

Acknowledgements

This research was initiated at the Gregynog 71717 workshop. We would like to thank the organisers Ranko Lazić and Patrick Totzke, as well as the EPSRC for their generous support. We are indebted to Markus Lohrey for fruitful discussions about graph groups, some of which provided key insights that found their way into the present paper.

References

[1] K. Athanasiou, P. Liu, and T. Wahl. “Unbounded-Thread Program Verification using Thread-State Equations”. In: *Automated Reasoning, IJCAR*. Vol. 9706. Lect. Notes Comp. Sci. Springer, 2016, pp. 516–531.

[2] C. Barrett, C. L. Conway, M. Deters, L. Hadarean, D. Jovanovi'c, T. King, A. Reynolds, and C. Tinelli. "CVC4". In: *Computer Aided Verification, CAV*. Vol. 6806. Lect. Notes Comp. Sci. Springer, 2011, pp. 171–177.

[3] L. Bartholdi and P. V. Silva. "Rational subsets of groups". In: *CoRR* abs/1012.1532 (2010). arXiv: 1012.1532.

[4] J. Berstel. *Transductions and Context-Free Languages*. Stuttgart: Teubner, 1979.

[5] M. Blondin, A. Finkel, C. Haase, and S. Haddad. "The Logical View on Continuous Petri Nets". In: *ACM Trans. Comput. Log.* 18.3 (2017), 24:1–24:28.

[6] M. Blondin, C. Haase, and F. Mazowiecki. "Affine Extensions of Integer Vector Addition Systems with States". In: *Concurrency Theory, CONCUR*. Vol. 118. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, 14:1–14:17.

[7] R. Bonnet. "Theory of Well-Structured Transition Systems and Extended Vector-Addition Systems". Thèse de doctorat. Laboratoire Spécification et Vérification, ENS Cachan, France, 2013.

[8] I. Borosh and L. B. Treybing. "Bounds on positive integral solutions of linear Diophantine equations". In: *Proc. Am. Math. Soc.* 55 (1976), pp. 299–304.

[9] P. Buckheister and G. Zetzsche. "Semilinearity and Context-Freeness of Languages Accepted by Valence Automata". In: *Mathematical Foundations of Computer Science, MFCS*. Vol. 8087. Lect. Notes Comp. Sci. Springer, 2013, pp. 231–242.

[10] D. Chistikov and C. Haase. "The Taming of the Semi-Linear Set". In: *Automata, Languages, and Programming, ICALP*. Vol. 55. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016, 128:1–128:13.

[11] W. Czerwinski, S. Lasota, R. Lazic, J. Leroux, and F. Mazowiecki. "The Reachability Problem for Petri Nets is Not Elementary". In: *Symposium on the Theory of Computing, STOC*. To appear. 2019.

[12] L. M. de Moura and N. Bjørner. "Z3: An Efficient SMT Solver". In: *Tools and Algorithms for the Construction and Analysis of Systems, TACAS*. Vol. 4963. Lect. Notes Comp. Sci. Springer, 2008, pp. 337–340.

[13] M. Dehn. "Über unendliche diskontinuierliche Gruppen". In: *Math. Ann.* 71 (1 1911). In German, pp. 116–144.

[14] V. Diekert. "Word Problems Over Traces which are Solvable in Linear Time". In: *Theor. Comput. Sci.* 74.1 (1990), pp. 3–18.

[15] V. Diekert, C. Gutierrez, and C. Hagenah. "The existential theory of equations with rational constraints in free groups is PSPACE - complete". In: *Inform. Comput.* 202.2 (2005), pp. 105–140.

[16] V. Diekert, A. Jeż, and M. Kufleitner. "Solutions of Word Equations Over Partially Commutative Structures". In: *Automata, Languages, and Programming, ICALP 2016*. Vol. 55. LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, 127:1–127:14.

[17] V. Diekert and A. Muscholl. "Solvability of Equations in Free Partially Commutative Groups Is decidable". In: *Automata, Languages and Programming, ICALP*. Springer Berlin Heidelberg, 2001, pp. 543–554.

[18] V. Diekert and G. Rozenberg, eds. *The Book of Traces*. World Scientific, 1995.

[19] E. D'Osualdo, R. Meyer, and G. Zetzsche. "First-order logic with reachability for infinite-state systems". In: *Logic in Computer Science, LICS*. IEEE, 2016, pp. 457–466.

[20] S. Eilenberg and M.-P. Schützenberger. "Rational sets in commutative monoids". In: *J. Algebra* 13.2 (1969), pp. 173–191.

[21] F. Eisenbrand and G. Shmonin. "Carathéodory bounds for integer cones". In: *Oper. Res. Lett.* 34.5 (2006), pp. 564–568.

[22] J. Esparza, R. Ledesma-Garza, R. Majumdar, P. Meyer, and F. Nikšić. "An SMT-Based Approach to Coverability Analysis". In: *Computer Aided Verification, CAV*. Vol. 8559. Lect. Notes Comp. Sci. Springer, 2014, pp. 603–619.

[23] S. Ginsburg and E. H. Spanier. "Bounded ALGOL-like languages". In: *T. Am. Math. Soc.* (1964), pp. 333–368.

[24] R. Greenlaw, H. J. Hoover, and W. L. Ruzzo. *Limits to Parallel Computation: P-Completeness Theory*. Oxford University Press, 1995.

[25] S. A. Greibach. "Remarks on blind and partially blind one-way multicounter machines". In: *Theor. Comput. Sci.* 7.3 (1978), pp. 311–324.

[26] E. M. Gurari and O. H. Ibarra. "The complexity of decision problems for finite-turn multicounter machines". In: *J. Comput. Syst. Sci.* 22.2 (1981), pp. 220–229.

[27] C. Haase. "A survival guide to Presburger arithmetic". In: *SIGLOG News* 5.3 (2018), pp. 67–82.

[28] C. Haase. "Subclasses of Presburger arithmetic and the weak EXP hierarchy". In: *Computer Science Logic and Logic in Computer Science, CSL-LICS*. ACM, 2014, 47:1–47:10.

[29] C. Haase and S. Halfon. "Integer Vector Addition Systems with States". In: *Reachability Problems, RP*. Vol. vol 8762. Lect. Notes Comp. Sci. Springer, 2014, pp. 112–124.

[30] F. Haglund and D. T. Wise. "Coxeter groups are virtually special". In: *Adv. Math.* 224.5 (2010), pp. 1890–1903.

[31] P. Hofman and S. Lasota. "Linear Equations with Ordered Data". In: *Concurrency Theory, CONCUR*. Vol. 118. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, 24:1–24:17.

[32] P. Hofman, J. Leroux, and P. Totzke. "Linear combinations of unordered data vectors". In: *Logic in Computer Science, LICS*. IEEE, 2017, pp. 1–11.

[33] I. Agol. "The virtual Haken conjecture. With an appendix by Agol, Daniel Groves, and Jason Manning". In: *Doc. Math.* 18 (2013), pp. 1045–1087.

[34] M. Jantzen and A. Kurganskyy. "Refining the hierarchy of blind multicounter languages and twist-closed trios". In: *Inform. Comput.* 185.2 (2003), pp. 159–181.

[35] M. Kambites. "Formal Languages and Groups as Memory". In: *Commun. Algebra* 37 (1 2009), pp. 193–208.

[36] M. Kambites, P. V. Silva, and B. Steinberg. "On the rational subset problem for groups". In: *J. Algebra* 309 (2007), pp. 622–639.

[37] I. Kapovich, R. Weidmann, and A. G. Myasnikov. "Foldings, Graphs of Groups and the Membership Problem". In: *Int. J. Algebr. Comput.* 15.1 (2005), pp. 95–128.

[38] E. Kopczyński and A. W. To. "Parikh Images of Grammars: Complexity and Applications". In: *Logic in Computer Science, LICS*. IEEE, 2010, pp. 80–89.

[39] R. J. Lipton. *The Reachability Problem Requires Exponential Space*. Tech. rep. 63. Department of Computer Science, Yale University, 1976.

[40] M. Lohrey. "The rational subset membership problem for groups: a survey". In: *Groups St Andrews 2013*. Vol. 422. Lond. Math. S. Cambridge University Press, 2016, pp. 368–389.

[41] M. Lohrey and B. Steinberg. "The submonoid and rational subset membership problems for graph groups". In: *J. Algebra* 320.2 (2008), pp. 728–755.

[42] M. Lohrey, B. Steinberg, and G. Zetzsche. "Rational subsets and submonoids of wreath products". In: *Inform. Comput.* 243 (2015), pp. 191–204.

[43] M. Lohrey and G. Zetzsche. "Knapsack in Graph Groups". In: *Theor. Comput. Syst.* 62 (2018), pp. 192–246.

[44] M. L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, 1967.

[45] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[46] R. Piskac and V. Kuncak. "Linear Arithmetic with Stars". In: *Computer Aided Verification, CAV*. Vol. 5123. Lect. Notes Comp. Sci. Springer, 2008, pp. 268–280.

[47] L. Pottier. "Minimal Solutions of Linear Diophantine Systems: Bounds and Algorithms". In: *Rewriting Techniques and Applications, RTA*. Vol. 488. Lect. Notes Comp. Sci. Springer, 1991, pp. 162–173.

[48] M. Presburger. "Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt". In: *Comptes Rendus du I congres de Mathematiciens des Pays Slaves*. 1929, pp. 92–101.

[49] C. Rackoff. "The Covering and Boundedness Problems for Vector Addition Systems". In: *Theor. Comput. Sci.* 6 (1978), pp. 223–231.

[50] K. Reinhardt. "Reachability in Petri Nets with Inhibitor Arcs". In: *Electr. Notes in Theor. Comput. Sci.* 223 (2008). Reachability Problems in Computational Models, RP, pp. 239–264.

[51] S. Schmitz. "The complexity of reachability in vector addition systems". In: *SIGLOG News* 3.1 (2016), pp. 4–21.

[52] K. N. Verma, H. Seidl, and T. Schwentick. "On the Complexity of Equational Horn Clauses". In: *Automated Deduction – CADE-20*. Vol. 3632. Lect. Notes Comp. Sci. 2005, pp. 337–352.

[53] D. Wise. *From Riches to Raags: 3-Manifolds, Right-Angled Artin Groups, and Cubical Geometry*. American Mathematical Society, 2012.

[54] D. T. Wise. "Research announcement: the structure of groups with a quasiconvex hierarchy". In: *Electr. Research Announ. Math. Sci.* 16 (2009), pp. 44–55.

[55] C. Wrathall. "Trace monoids with some invertible generators: two decision problems". In: *Discrete. Appl. Math.* 32.2 (1991), pp. 211–222.

[56] G. Zetzsche. “Monoids as Storage Mechanisms”. PhD thesis. Technische Universität Kaiserslautern, 2016.

[57] G. Zetzsche. “Silent Transitions in Automata with Storage”. In: *Automata, Languages and Programming, ICALP*. Vol. 7966. Lect. Notes Comp. Sci. Springer, 2013, pp. 434–445.

[58] G. Zetzsche. “The Emptiness Problem for Valence Automata or: Another Decidable Extension of Petri Nets”. In: *Reachability Problems, RP*. Vol. 9328. LNCS. Springer, 2015, pp. 166–178.

Appendix A
Missing proofs from Section IV

Lemma IV.1. Let $M = \bigcup_{j\in J} L(\boldsymbol{c}_j, Q_j)$ be a semi-linear set. Then $M^* = \bigcup_{K\subseteq J} L(\boldsymbol{b}_K, P_K)$, where

$$\boldsymbol{b}_K := \sum_{k\in K} \boldsymbol{c}_k \qquad C_K := \bigcup_{k\in K} \{\boldsymbol{c}_k\}$$
$$Q_K := \bigcup_{k\in K} Q_k \qquad P_K := C_K \cup Q_K.$$

Proof. First, observe that the following identities hold for any finite $P, Q \subseteq \mathbb{N}^d$:

$$\begin{aligned} L^*(\mathbf{0}, P) + L^*(\mathbf{0}, Q) &= L^*(\mathbf{0}, P \cup Q) \\ = L(\mathbf{0}, P \cup Q) &= L(\mathbf{0}, P) + L(\mathbf{0}, Q) \end{aligned} \tag{14}$$

Let $\boldsymbol{v} \in M^*$, by grouping and summing up vectors according to the linear set they originate from, we can write $\boldsymbol{v} = \boldsymbol{v}_1 + \cdots + \boldsymbol{v}_k$ such that $\boldsymbol{v}_i \in L^*(\boldsymbol{c}_{j_i}, Q_{j_i})$ for all $1 \le i \le k$ and indices $j_i \in J$. Observe that for a fixed i, we have

$$\boldsymbol{v}_i \in \boldsymbol{c}_{j_i} + L^*(\mathbf{0}, \{\boldsymbol{c}_{j_i}\} \cup Q_{j_i}) = c_{j_i} + L(\mathbf{0}, \{\boldsymbol{c}_{j_i}\} \cup Q_{j_i}),$$

where the last equality follows from Eq. (14). Hence

$$\begin{aligned} \boldsymbol{v} &\in \sum_{i=1}^{k} (\boldsymbol{c}_{j_i} + L(\mathbf{0}, \{\boldsymbol{c}_{j_i}\} \cup Q_{j_i})) \\ &= \sum_{i=1}^{k} \boldsymbol{c}_i + L\left(\mathbf{0}, C_K \cup Q_K\right) \\ &= L\left(\sum_{i=1}^{k} \boldsymbol{c}_{j_i}, C_K \cup Q_K\right) \\ &= L(\boldsymbol{b}_K, P_K), \end{aligned}$$

where $K := \{j_i : 1 \le i \le k\}$.

Conversely, suppose $\boldsymbol{v} \in L(\boldsymbol{b}_K, P_K)$ for some $K = \{j_1, \ldots, j_k\}$. Then there are $\lambda_1, \ldots, \lambda_k > 0$ and $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_k \ge \mathbf{0}$ such that

$$\boldsymbol{v} = \sum_{i=1}^{k} (\lambda_i \cdot \boldsymbol{c}_{j_i} + Q_{j_i} \cdot \boldsymbol{\gamma}_i) \in \sum_{i=1}^{k} \lambda_i \cdot L(\boldsymbol{c}_{j_i}, Q_{j_i}) \subseteq M^*. \quad \square$$

Lemma IV.2. Let $M = \bigcup_{j\in J} L(\boldsymbol{c}_j, Q_j) \subseteq \mathbb{N}^n$ be a semi-linear set, and let $c = 2n \log(4n\|M\|)$. Then $M^* = \bigcup_{i\in I} L(\boldsymbol{b}_i, P_i)$ such that for every $i \in I$, there is some $K \subseteq J$ with $|K| \le c$ such that

- $\boldsymbol{b}_i = \sum_{k\in K} \boldsymbol{c}_k$
- $P_i \subseteq \bigcup_{j\in J}\{\boldsymbol{c}_j\} \cup \bigcup_{k\in K} Q_k$ with $|P_i| \le c$, and
- $\|M^*\| \le c \cdot \|M\|$.

Proof. By Lemma IV.1, we have $M^* = \bigcup_{K\subseteq J} L(\boldsymbol{d}_K, R_K)$ such that $\boldsymbol{d}_K = \sum_{k\in K} \boldsymbol{c}_k$ and $R_K = C_K \cup Q_K$, where $C_K = \bigcup_{k\in K} \boldsymbol{c}_k$ and $Q_K = \bigcup_{k\in K}$. By Proposition II.1, for a fixed $K \subseteq J$ we have

$$L(\boldsymbol{d}_K, R_K) = \bigcup_{h\in H_K} L(\boldsymbol{d}_K, S_h)$$

such that $S_h \subseteq Q_K$ and $|S_h| \le c$ for all $h \in H_K$. For $h \in H_K$, let $K(S_h) = \{k \in K : S_h \cap Q_K \neq \emptyset\}$ and $\boldsymbol{b}_h = \sum_{k\in K(S_h)} \boldsymbol{c}_k$. We have $\boldsymbol{d}_K \in L(\boldsymbol{b}_h, C_K)$, and hence

$$L(\boldsymbol{d}_K, R_K) \subseteq \bigcup_{h\in H_K} L(\boldsymbol{b}_h, S_h \cup C_K) \subseteq M^*.$$

Applying Proposition II.1 once more to every $S_h \cup C_K$, the statement follows. $\square$

Lemma IV.3. Let $N_j = L(\boldsymbol{c}_j, Q_j) \subseteq \mathbb{N}^n$ such that $\|N_j\| \le s$ and $|Q_j| \le m$, $j \in [k]$ for some k. Then $M := \bigcap_{1\le j\le k} N_j$ is a semi-linear set $M = \bigcup_{i\in I} L(\boldsymbol{b}_i, P)$ such that

- $\|M\| \le (k \cdot m \cdot s)^{O(k\cdot n)}$.
- for every $\boldsymbol{b}_i$ and $1 \le j \le k$ there is some $\boldsymbol{d} \ge \mathbf{0}$ such that $\boldsymbol{b}_i = \boldsymbol{c}_j + Q_j \cdot \boldsymbol{d}$, and
- for every $1 \le j \le k$ there is some non-negative matrix R_j such that $P = Q_j \cdot R_j$.

Proof. Consider the following system of linear Diophantine equations $\mathcal{S}$:

$$\begin{gathered} \boldsymbol{c}_1 + Q_1 \cdot \boldsymbol{\lambda}_1 = \boldsymbol{c}_2 + Q_2 \cdot \boldsymbol{\lambda}_2 \\ \boldsymbol{c}_2 + Q_2 \cdot \boldsymbol{\lambda}_2 = \boldsymbol{c}_3 + Q_3 \cdot \boldsymbol{\lambda}_3 \\ \vdots \\ \boldsymbol{c}_{k-1} + Q_{k-1} \cdot \boldsymbol{\lambda}_{k-1} = \boldsymbol{c}_k + Q_k \cdot \boldsymbol{\lambda}_k \end{gathered}$$

The constraint matrix of $\mathcal{S}$ has $k \cdot n$ rows and at most $k \cdot m$ columns, and $\|\mathcal{S}\| \le 2s$. According to [10, Prop. 4], we have $[\![\mathcal{S}]\!] = L(E, S)$ such that $\|L(E, S)\| \le ((k \cdot m + 3) \cdot \|\mathcal{S}\| + 1)^{k\cdot n}$. For every $1 \le j \le k$, let D_j and R_j be obtained from the projection of E respectively S onto the components corresponding to $\boldsymbol{\lambda}_j$. Then as in the proof of [10, Thm. 6], for any $1 \le j \le k$ we have

$$\begin{aligned} M &= \boldsymbol{c}_j + \{Q_j \cdot \boldsymbol{\gamma} : \boldsymbol{\gamma} \in L(D_j, R_j)\} \\ &= L(\boldsymbol{c}_j + Q_j \cdot D_j, Q_j \cdot R_j). \end{aligned}$$

A quick inspection reveals

$$\begin{aligned} \|Q_j \cdot R_j\| &\le \|\boldsymbol{c}_j + Q_j \cdot D_j\| \\ &\le \|\boldsymbol{c}_j\| + ms((km + 3) \cdot \|\mathcal{S}\| + 1)^{k\cdot n} \\ &\le (k \cdot m \cdot s)^{O(k\cdot n)} \end{aligned}$$

$\square$

Lemma IV.7. Let $a_0, \ldots, a_m \in \mathbb{N}$ and $b_0, \ldots, b_m \in \{0, 1\}$ with $\sum_{i=0}^{m} a_i = \sum_{i=0}^{m} b_i$. Then $\sum_{i=0}^{m} a_i \cdot 2^i = \sum_{i=0}^{m} b_i \cdot 2^i$ if and only if $a_i = b_i$ for every $i \in [0, m]$.

Proof. Let X be a formal variable and consider the ring $\mathbb{Z}[X]$ of polynomials over the variable X with integer coefficients. In $\mathbb{Z}[X]$, consider the polynomials $p(X) = \sum_{i=0}^{m} a_i \cdot X^i$ and $q(X) = \sum_{i=0}^{m} b_i \cdot X^i$. We have to show that $p(1) = q(1)$

and $p(2) = q(2)$ implies $p(X) = q(X)$. If $a_i \geq 2$ for some $i \in [0, m]$, we modify $p_0(X) := p(X)$ to obtain $p_1(X) := p_0(X) - 2X^i + X^{i+1}$. Then, p_1 has coefficients ≥ 0 and verifies $p_1(2) = p_0(2)$ and $p_1(1) = p_0(1) - 1$. We repeat this process and obtain $p_{r+1}(X)$ from $p_r(X)$ as long as $p_r(X)$ has a coefficient ≥ 2. Since $p_{r+1}(1) = p_r(1) - 1$ and every $p_r(X)$ has coefficients ≥ 0, this process must terminate, say after s steps. Then $p_s(X)$ has coefficients in $\{0, 1\}$. It satisfies $p_s(1) = p(1) - s$ and $p_s(2) = p(2)$ and thus $p_s(2) = p(2) = q(2)$. Since binary representations are unique, this implies $p_s(X) = q(X)$. However, that also means $p(1) - s = p_s(1) = q(1) = p(1)$, hence $s = 0$ and $q(X) = p_s(X) = p(X)$. □

Appendix B
Missing proof from Section VI

Correctness proof for Proposition VI.1. We define the subset $A_\ell = \{a_1, \ldots, a_\ell\}$. Observe that since the $[a_i]$ commute pairwise, we can regard the morphism π_{A_ℓ} as $\pi_{A_\ell} \colon \mathbb{G}(A, I) \to \mathbb{Z}^\ell$ and shorten it to π_ℓ. We claim that for words $w \in \{b_1, \ldots, b_d\}^*$ and $y_i \in \{a_1^{\pm 1}, \ldots, a_d^{\pm 1}\}^*$, we have $[xy_1 b_{\ell_1}^{-1} y_2 b_{\ell_2}^{-1} \cdots y_n b_{\ell_n}] = 1$ if and only if $x = b_{\ell_n} \cdots b_{\ell_1}$, $[y_1 \cdots y_{n+1}] = 1$ and $\pi_{\ell_i}(y_1 \cdots y_i) = 0$ for every $i \in [1, n]$. Note that the claim immediately implies that $1 \in \mathsf{L}(\mathcal{A}, q_0, q')$ if and only if $(q, 0) \to_V^* (q', 0)$.

For the "if" direction, suppose $x = b_{\ell_n} \cdots b_{\ell_1}$, $[y_1 \cdots y_{n+1}] = 1$ and $\pi_{\ell_i}(y_1 \cdots y_i) = 0$ for every $i \in [1, n]$. We proceed by induction on n. If $n = 0$, then the claim is trivial. Let $n \geq 1$. Since the a_i commute pairwise, we can write $[y_1] = [zz']$ with some $z \in \{a_1^{\pm 1}, \ldots, a_{\ell_1}^{\pm 1}\}^*$ and $z' \in \{a_{\ell_1+1}^{\pm 1}, \ldots, a_d^{\pm 1}\}^*$. Moreover, $\pi_{\ell_1}(y_1) = 0$ implies $[z] = 1$. Since b_{ℓ_1} commutes with all the generators $a_{\ell_1+1}, \ldots, a_d$, we have

$$\begin{aligned}
&[xy_1 b_{\ell_1}^{-1} y_2 b_{\ell_2}^{-1} \cdots y_n b_{\ell_n}^{-1} y_{n+1}] \\
=&[xzz' b_{\ell_1}^{-1} y_2 b_{\ell_2}^{-1} \cdots y_n b_{\ell_n}^{-1} y_{n+1}] \\
=&[xz b_{\ell_1}^{-1} z' y_2 b_{\ell_2}^{-1} \cdots y_n b_{\ell_n}^{-1} y_{n+1}] \\
=&[b_{\ell_n} \cdots b_{\ell_1} z b_{\ell_1}^{-1} z' y_2 b_{\ell_2}^{-1} \cdots y_n b_{\ell_n}^{-1} y_{n+1}] \\
=&[b_{\ell_n} \cdots b_{\ell_2} z' y_2 b_{\ell_2}^{-1} \cdots y_n b_{\ell_n}^{-1} y_{n+1}].
\end{aligned}$$

Note that $[z] = 1$ implies $\pi_{\ell_i}(z' y_2 \cdots y_i) = \pi_{\ell_i}(zz' y_2 \cdots y_i) = \pi_{\ell_i}(y_1 \cdots y_i) = 0$ for every $i \in [1, n]$. Therefore, we may invoke induction to conclude that $[b_{\ell_n} \cdots b_{\ell_2} z' y_2 b_{\ell_2}^{-1} \cdots y_n b_{\ell_n}^{-1}] = 1$ holds and thus $[xy_1 b_{\ell_1}^{-1} y_2 b_{\ell_2}^{-1} \cdots y_n b_{\ell_n}^{-1}] = 1$.

For the "only if" direction, suppose the word $w = xy_1 b_{\ell_1}^{-1} y_2 b_{\ell_2}^{-1} \cdots y_n b_{\ell_n}^{-1}$ satisfies $[w] = 1$. We again proceed by induction and notice that the claim holds vacuously for $n = 0$. Now let $n \geq 1$. First, observe that $[w] = 1$ implies $[x b_{\ell_1}^{-1} b_{\ell_2}^{-1} \cdots b_{\ell_n}^{-1}] = \pi_B(w) = 1$ and thus $[x] = [b_{\ell_n} \cdots b_{\ell_1}]$. Since the b_i do not commute with each other, that means $x = b_{\ell_n} \cdots b_{\ell_1}$. We can find $z \in \{a_1^{\pm 1}, \ldots, a_{\ell_1}^{\pm 1}\}^*$ and $z' \in \{a_{\ell_1+1}^{\pm 1}, \ldots, a_d^{\pm 1}\}^*$ such that $[y_1] = [zz']$. Toward a contradiction, suppose $[z] \neq 1$. Then it is easy to see by induction on the number of applied rewriting steps that any word $w' \in (A^{\pm 1})^*$ with $[w'] = [w]$ has an infix $b_{\ell_1} y b_{\ell_1}^{-1}$ with $\pi_{\ell_1}(y) = [z] \neq 0$: We can certainly not apply any cancellation rule for a b_i; other cancellation rules might be applicable, but preserve this property; and since b_{ℓ_1} does not commute with $a_1, \ldots, a_{\ell_1}$, commutation rules preserve this property as well. Therefore, we would have $[w] \neq 1$. Thus, we have shown $[z] = 1$.

Observe that

$$\begin{aligned}
[w] &= [b_{\ell_n} \cdots b_{\ell_1} zz' b_{\ell_1}^{-1} y_2 b_{\ell_2}^{-1} \cdots y_n b_{\ell_n}^{-1} y_{n+1}] \\
&= [b_{\ell_n} \cdots b_{\ell_1} z b_{\ell_1}^{-1} z' y_2 b_{\ell_2}^{-1} \cdots y_n b_{\ell_n}^{-1} y_{n+1}] \\
&= [b_{\ell_n} \cdots b_{\ell_2} z' y_2 b_{\ell_2}^{-1} \cdots y_n b_{\ell_n}^{-1} y_{n+1}].
\end{aligned}$$

By induction, we may conclude that $\pi_{\ell_i}(z' y_2 \cdots y_i) = 1$ for every $i \in [2, n+1]$ and $[z' y_2 \cdots y_{n+1}] = 1$. Finally, observe that since $[z] = 1$, we have $\pi_{\ell_i}(y_1 \cdots y_i) = \pi_{\ell_i}(zz' y_2 \cdots y_i) = \pi_{\ell_i}(z' y_2 \cdots y_i) = 0$ and for $i \in [1, n]$ and $[y_1 \cdots y_{n+1}] = [zz' y_2 \cdots y_{n+1}] = 1$. □

Appendix C
Missing proofs from Section VII

In order to prove Proposition VII.4, we need a simple fact about free products of groups. Since $A = A_0 \uplus A_1$, we can uniquely decompose each $u \in (A^{\pm 1})^*$ into maximal factors from $(A_0^{\pm 1})^+ \cup (A_1^{\pm 1})^+$. In other words, we decompose $u = u_1 \cdots u_n$ with $n \geq 0$ such that $u_i \in (A_0^{\pm 1})^+ \cup (A_1^{\pm 1})^+$ for $i \in [1, n]$ and $u_i \in (A_s^{\pm 1})^+$ if and only if $u_{i+1} \in (A_{1-s}^{\pm 1})^+$ for $i \in [1, n-1]$ and $s \in \{0, 1\}$. This is the *block decomposition* of u.

Lemma C.1. *Let $u \in (A^{\pm 1})^*$ and let $u = u_1 \cdots u_n$ be its block decomposition. If $[u] = 1$, then $[u_i] = 1$ for some i.*

Proof. Suppose $[u_i] \neq 1$ for every $i \in [1, n]$. By induction on the number of rewriting steps, it is easy to see that every word u' with $[u] = [u']$ has a block decomposition $u' = u'_1 \cdots u'_m$ with $m = n$ and $[u'_i] = [u_i]$ for every $i \in [1, n]$. For cancellation rules, that is trivial. Moreover, commutation rules can only rearrange symbols within each block (recall that we assumed that there are no edges between A_0 and A_1). This implies that u' cannot be the empty word and hence $[u] \neq 1$. □

Proposition VII.4. *For every $p, q \in Q$, we have $\mathsf{L}(G_{\mathcal{A}}, \square_p^q) = \mathsf{L}(\mathcal{A}, p, q)$.*

Proof. The following notation for runs in valence automata will be convenient: If in a valence $\mathcal{A}$, we have $(p, 1, \varepsilon) \to_{\mathcal{A}}^* (q, [u], v)$, then we write $p \xrightarrow{u|v} q$.

We begin with the inclusion $\mathsf{L}(G_{\mathcal{A}}, \square_p^q) \subseteq \mathsf{L}(\mathcal{A}, p, q)$. We prove the inclusion by establishing the following claim: For every $v \in (N \cup \Sigma)^*$ with $\square_p^q \Rightarrow^k v$, we have $v_0 \square_{p_1}^{q_1} v_1 \cdots \square_{p_n}^{q_n} v_n$ for some $n \geq 0$ and there are runs

$$p \xrightarrow{u_0|v_0} p_1,\ q_1 \xrightarrow{u_1|v_1} p_2,\ \ldots,$$
$$q_{n-1} \xrightarrow{u_{n-1}|v_{n-1}} p_n,\ q_n \xrightarrow{u_n|v_n} q \tag{15}$$

in $\mathcal{A}$ such that $[u_0 \cdots u_n] = 1$. The claim implies that for every $v \in \mathsf{L}(G_{\mathcal{A}}, \square_p^q)$, there is a run $p \xrightarrow{u|v} q$ with $[u] = 1$, which means $v \in \mathsf{L}(\mathcal{A}, p, q)$. We prove the claim by induction on k.

If $k = 0$, then $v = \Box_p^q$ and the claim is trivial. Let $k \geq 1$ and $\Box_p^q \Rightarrow^{k-1} v' \Rightarrow v$. Then we can write $v' = v_0\Box_{p_1}^{q_1}v_1 \cdots \Box_{p_n}^{q_n}v_n$ for some $n \geq 0$ and have runs as in Eq. (15). This means v is obtained from v' by replacing a nonterminal, say $\Box_{p_i}^{q_i}$, by some word $v'' \in \mathsf{L}(\mathcal{A}_j, p_i, q_i)$ for some $j \in \{0,1\}$. By definition of $\mathcal{A}_j$, we have $v'' = v_0''\Box_{p_1'}^{q_1'}v_1'' \cdots \Box_{p_m'}^{q_m'}v_m''$ and runs

$$p_i \xrightarrow{u_0''|v_0''} p_1'',\ q_1'' \xrightarrow{u_1''|v_1''} p_2'',\ \ldots,\\ q_{m-1}'' \xrightarrow{u_{m-1}''|v_{m-1}''} p_m'',\ q_m'' \xrightarrow{u_m''|v_m''} q_i \quad (16)$$

such that $[u_0'' \cdots u_m''] = 1$. By inserting the runs in Eq. (16) into Eq. (15) between p_i and q_i, we obtain runs

$$p \xrightarrow{u_0|v_0} p_1,\ q_1 \xrightarrow{u_1|v_1} p_2,\ \ldots,\ q_{i-2} \xrightarrow{u_{i-2}|v_{i-2}} p_{i-1},\\ q_{i-1} \xrightarrow{u_{i-1}u_0''|v_{i-1}v_0''} p_1'',\ q_1'' \xrightarrow{u_1''|v_1''} p_2'',\ \ldots,\\ q_{m-1}'' \xrightarrow{u_{m-1}''|v_{m-1}''} p_m'',\ q_m'' \xrightarrow{u_m''u_i|v_m''v_i} p_{i+1},\\ q_{i+1} \xrightarrow{u_{i+1}|v_{i+1}} p_{i+1},\ \ldots\ ,\\ q_{n-1} \xrightarrow{u_{n-1}|v_{n-1}} p_n,\ q_n \xrightarrow{u_n|v_n} q. \quad (17)$$

Since

$$v = v_0\Box_{p_1}^{q_1}v_1 \cdots \Box_{p_{i-1}}^{q_{i-1}}v_{i-1}\left(v_0''\Box_{p_1''}^{q_1''}v_1'' \cdots \Box_{p_m''}^{q_m''}v_m''\right)\\ v_i\Box_{p_{i+1}}^{q_{i+1}} \cdots v_{n-1}\Box_{p_n}^{q_n}v_n, \quad (18)$$

this proves the claim for v.

Let us now prove the inclusion $\mathsf{L}(\mathcal{A}, p, q) \subseteq \mathsf{L}(G_\mathcal{A}, \Box_p^q)$. We claim that for every sequence of runs as in Eq. (15) in $\mathcal{A}$ with $[u_0 \cdots u_n] = 1$, we have $\Box_p^q \Rightarrow^* v_0\Box_{p_1}^{q_1}v_1 \cdots \Box_{p_n}^{q_n}v_n$. Note that this establishes the inclusion: For each $v \in \mathsf{L}(\mathcal{A}, p, q)$, there is a run $p \xrightarrow{u|v} q$ with $[u] = 1$. Thus, the claim implies $\Box_p^q \Rightarrow^* v$, which means $v \in \mathsf{L}(G_\mathcal{A}, \Box_p^q)$ because $v \in \Sigma^*$.

We proceed by induction on the number of blocks in $u = u_0 \cdots u_n$. According to Lemma C.1, the block decomposition of the word $u_0 \cdots u_n$ contains some block w with $[w] = 1$. Now w can occur within a single factor u_i or span several factors. We only consider the latter case, the former can be treated analogously.

Hence, we can find indices $i < j$ and a decomposition $u_i = u_i'u_i''$ and $u_j = u_j'u_j''$ such that $w = u_i''u_{i+1} \cdots u_{j-1}u_j'$. Since $\mathcal{A}$ reads at most one generator from $A^{\pm 1}$ in each edge, this decomposition induces a decomposition of the runs $q_i \xrightarrow{u_i|v_i} p_{i+1}$ and $q_j \xrightarrow{u_j|v_j} p_{j+1}$ into

$$q_i \xrightarrow{u_i'|v_i'} p' \xrightarrow{u_i''|v_i''} p_{i+1} \quad \text{and} \quad q_j \xrightarrow{u_j'|v_j'} q' \xrightarrow{u_j''|v_j''} p_{j+1}.$$

Consider the sequence of runs that is obtained from Eq. (15) by replacing $q_i \xrightarrow{u_i|v_i} p_{i+1}$ by $q_i \xrightarrow{u_i'|v_i'} p'$ and replacing $q_j \xrightarrow{u_j|v_j} p_{j+1}$ by $q' \xrightarrow{u_j''|v_j''} p_{j+1}$. Then the word $u_0 \cdots u_{i-1}u_i'u_j''u_{j+1} \cdots u_n$ has one fewer block than $u_0 \cdots u_n$. Hence, by induction, the word

$$v' = v_0\Box_{p_1}^{q_1}v_1 \cdots \Box_{p_{i-1}}^{q_{i-1}}v_{i-1}\Box_{p_i}^{q_i}v_i'\Box_{p'}^{q'}v_j''\Box_{p_{j+1}}^{q_{j+1}}v_{j+1} \cdots \Box_{p_n}^{q_n}v_n$$

verifies $\Box_p^q \Rightarrow^* v'$. On the other hand, since w was chosen to be a block, we have $u_i''u_{i+1} \cdots u_{j-1}u_j' = w \in (A_s^{\pm 1})^+$ for some $s \in \{0,1\}$ and therefore runs

$$p' \xrightarrow{u_i''|v_i''}_{\mathcal{A}_s} p_{i+1},\ q_{i+1} \xrightarrow{u_{i+1}|v_{i+1}}_{\mathcal{A}_s} p_{i+2},\ \ldots\ ,\\ q_{j-1} \xrightarrow{u_{j-1}|v_{j-1}}_{\mathcal{A}_s} p_j,\ q_j \xrightarrow{u_j'|v_j'}_{\mathcal{A}_s} q' \quad (19)$$

in $\mathcal{A}_s$. By construction of $\mathcal{A}_s$, this means we have $v'' = v_i''\Box_{p_{i+1}}^{q_{i+1}}v_{i+1} \cdots \Box_{p_{j-1}}^{q_{j-1}}v_{j-1}\Box_{p_j}^{q_j}v_j' \in \mathsf{L}(\mathcal{A}_s, p', q')$. Therefore, replacing the occurrence of $\Box_{p'}^{q'}$ in v' by v'' yields v, hence $\Box_p^q \Rightarrow^* v' \Rightarrow v$. □

A. Analysis

We now analyze our translation described in Steps I to III. On one hand, we show that the translation maps $\mathsf{RatMP}_k^{\mathsf{tf}}$ to formulas of star-height $\leq k$. On the other hand, we show that it runs in polynomial time.

Concerning the star-height, suppose we are given a transitive forest (A, I) with $\beta(A, I) = k$. Notice that our recursive algorithm introduces a star only in step Step II: Compared to the formulas of the automata $\mathcal{A}_0$ and $\mathcal{A}_1$, it raises the star height by one. According to the definition of the branching number, this directly implies that the resulting $\exists\mathsf{PA}^*$ formula has star-height k.

Let us now show that the construction runs in polynomial time. It is clear from the construction that computing the formulas takes time linear in the size of the formulas. Hence, it suffices to show that they are polynomial in size. Let $\Phi_\mathcal{A}$ denote the formula computed for each automaton $\mathcal{A}$. Now suppose the input automaton $\mathcal{A}$ has n states and m edges. An inspection of Steps I to III reveals that there is a polynomial $p(x, y)$ such that:

1) If (A, I) has a universal vertex, then $|\Phi_\mathcal{A}| \leq |\Phi_{\mathcal{A}'}| + p(n, m)$.
2) If (A, I) is a disjoint union, then $|\Phi_\mathcal{A}| \leq |\Phi_{\mathcal{A}_0}| + |\Phi_{\mathcal{A}_1}| + p(n, m)$.
3) If (A, I) is empty, then $|\Phi_\mathcal{A}| \leq p(n, m)$.

Note that the length of the formulas may depend on the size of the alphabet, but that is bounded by the number m of edges. Of course, we may assume that p has only positive coefficients.

Let us show by induction on $|A|$ that $|\Phi_\mathcal{A}| \leq |A|^2 p(n, m + |A|n^2)$ for every $\mathcal{A}$. If (A, I) is empty, then this is clear. If (A, I) has a universal vertex, then we construct $\mathcal{A}'$ over (A', I'), where $\mathcal{A}'$ has n states and m edges and $|A'| = |A| - 1$. Therefore:

$$\begin{aligned} |\Phi_\mathcal{A}| &\leq |\Phi_{\mathcal{A}'}| + p(|A|, n, m) \\ &\leq (|A| - 1)^2 p(n, m + |A|n^2) + p(n, m) \\ &\leq |A|^2 p(n, m + |A|n^2). \end{aligned}$$

Now suppose (A, I) is a disjoint union of (A_0, I_0) and (A_1, I_1). Then we construct $\mathcal{A}_0$ and $\mathcal{A}_1$. They each have n states and $m + n^2$ edges, because we add the edges labeled $\Box_p^q$ for each $p, q \in Q$. Moreover, we have $|A_0|, |A_1| < |A|$ and thus by induction

$$|\Phi_{\mathcal{A}_i}| \leq |A_i|^2 p(n, m + n^2 + |A_i|n^2) \leq |A_i|^2 p(n, m + |A|n^2).$$

That implies

$$\begin{aligned}|\Phi_{\mathcal{A}}| &\leq |\Phi_{\mathcal{A}_0}| + |\Phi_{\mathcal{A}_1}| + p(n,m)\\ &\leq (|A_0|^2 + |A_1|^2)p(n, m + |A|n^2) + p(n,m)\\ &\leq (|A_0|^2 + |A_1|^2 + 1)p(n, m + |A|n^2)\\ &\leq |A|^2 p(n, m + |A|n^2).\end{aligned}$$

Thus, the resulting formulas have polynomial size.