Orbit-finite linear programming

ARKA GHOSH, PIOTR HOFMAN, and SŁAWOMIR LASOTA, University of Warsaw, Poland

An infinite set is orbit-finite if, up to permutations of atoms, it has only finitely many elements. We study a generalisation of linear programming where constraints are expressed by an orbit-finite system of linear inequalities. As our principal contribution we provide a decision procedure for checking if such a system has a real solution, and for computing the minimal/maximal value of a linear objective function over the solution set. We also show undecidability of these problems in case when only integer solutions are considered. Therefore orbit-finite linear programming is decidable, while orbit-finite integer linear programming is not.

CCS Concepts: • **Theory of computation → Integer programming**; **Linear programming**.

Additional Key Words and Phrases: Orbit-finite linear programming, linear programming, integer linear programming, sets with atoms, orbit-finite sets.

ACM Reference Format:
Arka Ghosh, Piotr Hofman, and Sławomir Lasota. 2023. Orbit-finite linear programming. In . ACM, New York, NY, USA, 39 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Applications of (integer) linear programming, and linear algebra in general, are ubiquitous in computer science (see e.g. [13, 14, 31]), including recent and potential future applications to analysis of data-enriched models [9, 18–20]. Whenever finite (integer) linear programs arise in analysis of finite models of computation, orbit-finite (integer) linear programs arise naturally in data-enriched versions of these models. For example, in decision problems for data Petri nets, such as reachability [27] or continuous reachability [18]; or in process mining [34]. Similar approach seems applicable also to structural properties or termination time of data Petri nets; and to learning of probabilistic automata with registers.

This paper is a continuation of the study of *orbit-finite* systems of linear equations [16], i.e., systems which are infinite but finite up to permutations. In this setting one fixes a countably infinite set $\mathbb{A}$, whose elements are called *atoms* (or data values) [5, 30], assuming that atoms can only be accessed in a very limited way, namely can only be tested for equality. Starting from atoms one builds a hierarchy of sets which are *orbit-finite*: they are infinite, but finite up to permutations of atoms. Along these lines, we study orbit-finite sets of linear inequalities, over an orbit-finite set of unknowns.

The main result of [16] is a decision procedure to check if a given orbit-finite system of equations is solvable. This result is general and applies to solvability over a wide range of commutative rings, in particular to real and integer solvability. In this paper we do a next step and extend the setting from equations to inequalities. Our goal is algorithmic solvability of orbit-finite systems of inequalities, but also optimisation of linear objective functions over solution sets of such systems. We call this problem *orbit-finite (integer) linear programming* (depending on whether the considered solutions are real or integer).

Manuscript submitted to ACM

Example 1. For illustration, consider the set $\mathbb{A}$ as unknowns, and the infinite system of constraints given by an infinite matrix whose rows and columns are indexed by $\mathbb{A}$:

$$\begin{bmatrix} 0 & 1 & 1 & \cdots \\ 1 & 0 & 1 & \cdots \\ 1 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \cdot \mathbf{x} \quad \geq \quad \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix} \tag{1}$$

Alternatively, one can write the infinite set of non-strict inequalities over unknowns $\alpha \in \mathbb{A}$, indexed by atoms $\beta \in \mathbb{A}$:

$$\sum_{\alpha \in \mathbb{A} \setminus \{\beta\}} \alpha \ \geq \ 1 \quad (\beta \in \mathbb{A}). \tag{2}$$

Any permutation $\pi : \mathbb{A} \to \mathbb{A}$ induces a permutation of the inequalities by sending

$$\sum_{\alpha \in \mathbb{A} \setminus \{\beta\}} \alpha \ \geq \ 1 \qquad \stackrel{\pi}{\longmapsto} \qquad \sum_{\alpha \in \mathbb{A} \setminus \{\pi(\beta)\}} \alpha \ \geq \ 1,$$

but the whole system (2) is invariant under permutations of atoms. Furthermore, up to permutations of atoms the system consists of just one equation – it is one *orbit*; in the sequel we consider orbit-finite systems (finite unions of orbits). Likewise, the matrix $\mathbb{A} \times \mathbb{A} \to \mathbb{R}$ in (1) consists, up to permutation of atoms, of just two entries. Indeed, its domain $\mathbb{A} \times \mathbb{A}$ is a union of two orbits: $\{ (\alpha, \beta) \mid \alpha = \beta \}$ and $\{ (\alpha, \beta) \mid \alpha \neq \beta \}$, and the matrix is constant inside each orbit. It is therefore invariant under permutations of atoms.

The system (2) is solvable. For example, given $n > 1$ atoms $S = \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{A}$, the vector $\mathbf{x}_n : \mathbb{A} \to \mathbb{R}$ defined by:

$$\mathbf{x}_n(\alpha) = \frac{1}{n-1} \ \text{ if } \ \alpha \in S, \qquad \mathbf{x}_n(\alpha) = 0 \ \text{ if } \ \alpha \notin S,$$

is a solution since the left-hand side of (2) sums up to 1 if $\beta \in S$, and to $\frac{n}{n-1}$ if $\beta \notin S$. ◁

Orbit-finite linear programming, i.e., optimisation of a linear objective function subject to an orbit-finite system of inequality constraints, faces phenomena not present in the classical setting. For instance, as illustrated in the next example, the objective function may not achieve its optimum over solutions of non-strict inequalities.

Example 2. Suppose that we aim at *minimization*, with respect to the constraints (2), of the value of the objective function:

$$S(\mathbf{x}) \ = \ 2 \cdot \sum_{\alpha \in \mathbb{A}} \mathbf{x}(\alpha). \tag{3}$$

The function is invariant under permutations of atoms, and its value is always greater than 2. Indeed, for every solution $\mathbf{x} : \mathbb{A} \to \mathbb{R}$ there is necessarily some $\beta \in \mathbb{A}$ such that $\mathbf{x}(\beta) > 0$, and hence

$$S(\mathbf{x}) \ > \ 2 \cdot \sum_{\alpha \in \mathbb{A} \setminus \{\beta\}} \mathbf{x}(\alpha) \ \geq \ 2. \tag{4}$$

What is the minimal value of the objective function? For solutions $\mathbf{x}_n$ defined in Example 1, the value $S(\mathbf{x}_n) = \frac{2n}{n-1}$ may be arbitrarily close to 2 but, according to (4), S never achieves 2. Surprisingly, this is in contrast with classical linear programming where, whenever constraints are specified by non-strict inequalities and are solvable, a linear objective function always achieves its minimum (or is unbounded from below). ◁

Inequalities and unknowns can be indexed by more than one atom, as illustrated in the next example.

Example 3. Let $\mathbb{A}^{(2)} = \left\{ \alpha\beta \in \mathbb{A}^2 \;\middle|\; \alpha \neq \beta \right\}$ be the set of pairs of distinct atoms. Consider a system whose inequalities and unknowns are indexed by $\mathbb{A} \uplus \{*\}$ and $\mathbb{A} \uplus \mathbb{A}^{(2)}$, respectively. Intuitively, unknowns correspond to vertices α and edges $\alpha\beta$ of an infinite directed clique. Let the system contain an inequality

$$\sum_{\alpha \in \mathbb{A}} \alpha \;\geq\; 1 \tag{5}$$

enforcing the sum of values assigned to all vertices to be at least 1, and the following inequalities

$$\sum_{\beta \in \mathbb{A} \setminus \{\alpha\}} \alpha\beta \;-\; \alpha \;-\; \sum_{\beta \in \mathbb{A} \setminus \{\alpha\}} \beta\alpha \;\geq\; 0 \qquad (\alpha \in \mathbb{A}) \tag{6}$$

enforcing, for each vertex $\alpha \in \mathbb{A}$, the sum of values assigned to all outgoing edges to be larger or equal to the sum of values assigned to all incoming edges, plus the value assigned to the vertex α. In matrix form (0 entries of the matrix are omitted):

$$\begin{array}{c} \\ \mathbb{A} \\ \\ \{*\} \end{array} \overset{\qquad \mathbb{A} \qquad\qquad\qquad \mathbb{A}^{(2)}}{\left[\begin{array}{ccc|c} -1 & & & \\ & -1 & & \mathbf{A} \\ & & \ddots & \\ \hline 1 & 1 & \cdots & \end{array}\right]} \cdot \mathbf{x} \;\geq\; \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline 1 \end{array}\right] \tag{7}$$

where $\mathbf{A}$ is the oriented incidence matrix, namely for every distinct atoms $\alpha, \beta \in \mathbb{A}$,

$$\mathbf{A}(\alpha, \alpha\beta) = 1 \qquad \mathbf{A}(\alpha, \beta\alpha) = -1,$$

and all other entries of $\mathbf{A}$ are 0. As in previous examples, the system is invariant under permutations of atoms. Solutions of the system correspond to directed graphs, whose vertices and edges are labeled in accordance with constraints (5) and (6). We return to this system in Example 5 below, and in Sections 6 and 8. ◁

The systems appearing in the above examples are invariant under all permutations of atoms. As usual when working in *sets with atoms* [5] (also known as *nominal sets* [30]), we allow systems which, for some finite subset $S \subseteq_{\text{fin}} \mathbb{A}$ (called a *support*), are only invariant under permutations that fix S. In the standard terminology of orbit-finite sets [5], we allow for *finitely supported* systems. Likewise, we allow for finitely supported objective functions, and seek for finitely supported solutions. Equivalently, using terminology of sets with atoms, we allow for *orbit-finite* systems and objective functions, and seek for orbit-finite solutions. Note that solutions appearing in Examples 1 and 2 are *finitary*, i.e. assign non-zero to only finitely many unknowns, and are therefore finitely supported. Each finite system is finitely supported, and thus orbit-finite linear programming is a generalisation of the classical one.

Contribution. As our main contribution, we provide decision procedures for orbit-finite linear programming, both for the decision problem of solvability of systems of inequalities, and for the optimisation problem.

The core ingredient of our approach is to reduce solvability (resp. optimisation) of an orbit-finite system of inequalities to the analogous question on a finite system which is *polynomially parametrised*, i.e., where coefficients are univariate polynomials in an integer variable n. The parameter n corresponds, intuitively, to the number of atoms appearing in (a support of) a solution. In this parametrised setting we ask for solvability for *some* $n \in \mathbb{N}$, or for optimisation when ranging over *all* $n \in \mathbb{N}$. We can compute an answer by encoding the problem into first-order real arithmetic [2, 32], which yields decidability. We provide also an efficient PTime algorithm for a relevant subclass of instances, resorting to polynomially many calls of classical linear programming.

Example 4. For instance, the system (1) is transformed to the following two inequalities with one unknown x, which are polynomially (actually, linearly) parametrised in a parameter n (the details are exposed in Example 44 in Section 7.3):

$$\begin{bmatrix} n-1 \\ n \end{bmatrix} \cdot x \;\geq\; \begin{bmatrix} n \\ n \end{bmatrix} \tag{8}$$

The objective function S (3), on the other hand, is transformed to a non-parametrised linear map $x \mapsto 2 \cdot x$. For every $n > 1$, the system (8) is solvable, the minimal solution is $x = \frac{n}{n-1}$, and the minimum of the objective function is $\frac{2n}{n-1}$. Ranging over all $n \in \mathbb{N}$, the minimum can get arbitrarily close to 2, but never reaches 2. ◁

Our reduction to a polynomially-parametrised linear program involves a size blowup which is exponential in *atom dimension* (i.e., the maximal number of atoms appearing in an index of inequality or unknown of a system) but polynomial in the number of its orbits. In consequence, orbit-finite linear programming is solvable in ExpTime, and in PTime when atom dimension is fixed. In the setting of orbit-finite sets this means that the problem is feasible [11]. Therefore, in the latter case the complexity is not worse than in case of classical linear programming.

One of cornerstones of the reduction is an observation that every system of inequalities that admits a finitary solution, admits also a solution which is invariant under all permutations of its support, i.e., of atoms it uses.

Example 5. As an illustration, let's prove that the system in Example 3 has no finitary solution. Indeed, if there existed a finite labeled directed graph satisfying constraints (5) and (6), by the above observation there would also exist a finite labeled directed clique, where labels of all vertices are pairwise equal, and labels of all edges are pairwise equal as well. In particular each vertex would carry the same value, necessarily positive due to constraint (5), and all edges incoming to a vertex would carry the same value as all outgoing edges. These requirements are clearly contradictory with constraint (6). In conclusion, the system has no solutions. ◁

As our second main result we prove undecidability of orbit-finite *integer* linear programming, already for the decision problem of solvability. While the classical linear programming and integer linear programming are on the opposite sides of the feasibility border, in case of orbit-finite systems the two problems are on the opposite sides of the decidability border. One of key reasons behind undecidability of integer linear programming is that it can express existence of a finite path. Specifically, if integer solutions are sought, the system of inequalities of the form

$$0 \;\leq\; \sum_{\alpha \in \mathbb{A}\setminus\{\beta\}} \beta\alpha \;\leq\; 1 \qquad (\beta \in \mathbb{A})$$

allow us to say that every node β has either no successors, or just one successor. This clearly fails if real solutions are sought.

This article is an improved and extended version of the conference submission [17]. The major improvement is lowering the complexity of orbit-finite linear programming from 2-ExpTime to ExpTime, and from ExpTime to PTime for fixed atom dimension. This is achieved by identifying a suitable subclass of polynomially-parametrised linear programs which we show to be solvable in PTime, and a reduction from orbit-finite linear programming to this subclass.

Related research. This paper belongs to a wider research program that aims at lifting different aspects of theory of computation from finite to orbit-finite sets (essentially equivalent to first-order definable sets) [4, 6–8, 10–12, 22–25].

Our findings generalise, or are closely related to, some earlier results on systems of linear equations. Systems studied in [20] have row indexes of atom dimension 1. In a more general but still restricted case studied in [21], all row indexes are assumed to have the same atom dimension. Furthermore, columns of a matrix are assumed to be finitary in [20, 21]. Both the papers investigate (nonnegative) integer solvability and are subsumed by [16] (in terms of decidability, but not in terms of complexity), a starting point for our investigations. Orbit-finite systems of equations are a special case of our present setting, as long as the solution domain is reals or integers. However, the setting of [16] is not restricted to reals or integers, and allows for an arbitrary commutative ring as a solution domain (under some mild effectiveness assumption). This larger generality makes the results of [16] and our results incomparable. Consequently, our methods are different than those of [16].

The work [19] goes beyond [20] and investigates linear equations, in atom dimension 1, over ordered atoms. Nonnegative integer solvability is decidable and equivalent to VAS reachability (and hence Ackermann-complete [15, 26, 28]).

Systems in another related work [23] are over a finite field, contain only finite equations, and are studied as a special case of orbit-finite constraint satisfaction problems. Furthermore, solutions sought are not restricted to be finitely-supported.

Orbit-finitely generated vector spaces were recently investigated in [9] and [16]. The former paper shows that every chain of vector subspaces which are invariant under permutations of atoms eventually stabilises, and apply this observation to prove decidability of zero-ness for orbit-finite *weighted* automata. The two papers jointly show that dual of an orbit-finitely generated vector space has an orbit-finite base. In [35] the authors study cones in such spaces which are invariant under permutations of atoms, and extend accordingly theorems of Carathéodory and Minkowski-Weyl.

Finally, our technique discussed in Example 5, and developed formally in Section 6, seems to be reminiscent of (but independent from) the techniques in the recent work [1].

Outline. After preliminaries on orbit-finite sets in Section 2, in Section 3 we introduce the setting of orbit-finite linear inequalities and in Section 4 we state our results. The rest of the paper is devoted to proofs. In Sections 6 and 5 we develop tools that are later used in decision procedures for linear programming in Sections 7 and 8. Finally, Section 9 contains the proof of undecidability of integer linear programming. We conclude in Section 10. Some routine or lengthy arguments are moved to Appendix.

2 PRELIMINARIES ON ORBIT-FINITE SETS

Our definitions rely on basic notions and results of the theory of *sets with atoms* [5], also known as nominal sets [7, 30]. We only work with *equality atoms* which have no additional structure except for equality.

Sets with atoms. We fix a countably infinite set $\mathbb{A}$ whose elements we call *atoms*. Greek letters $\alpha, \beta, \gamma, \ldots$ are reserved to range over atoms. The universe of sets with atoms is defined formally by a suitably adapted cumulative hierarchy of sets, by transfinite induction: the only set of *rank* 0 is the empty set; and for a cardinal i, a set of rank i may contain, as elements, sets of rank smaller than i as well as atoms. In particular, nonempty subsets $X \subseteq \mathbb{A}$ have rank 1.

The group Aut of all permutations of $\mathbb{A}$, called in this paper *atom automorphisms*, acts on sets with atoms by consistently renaming all atoms in a given set. Formally, by another transfinite induction, for $\pi \in$ Aut we define $\pi(X) = \{ \pi(x) \mid x \in X \}$. Via standard set-theoretic encodings of pairs or finite sequences we obtain, in particular, the pointwise action on pairs $\pi(xy) = \pi(x)\pi(y)$, and likewise on finite sequences. Relations and functions from X to Y are considered as subsets of $X \times Y$.

We restrict to sets with atoms X that only depend on finitely many atoms, in the following sense. For $T \subseteq \mathbb{A}$, let $\text{Aut}_T = \{ \pi \in \text{Aut} \mid \pi(\alpha) = \alpha \text{ for every } \alpha \in T \}$ be the set of atom automorphisms that *fix* T[1]; they are called T-automorphisms. A finite set $T \subseteq_{\text{fin}} \mathbb{A}$ (we use the symbol $\subseteq_{\text{fin}}$ for finite subsets) is a *support* of X if for all $\pi \in \text{Aut}_T$ it holds $\pi(X) = X$. We also say: T *supports* X, or X is T*-supported.* Thus a set is T-supported if and only if it is invariant under all $\pi \in \text{Aut}_T$. As an important special case, a function $f : X \to Y$, understood as its diagram $\{ (x, f(x)) \mid x \in X \}$, is T-supported if $f(\pi(x)) = \pi(f(x))$ for every argument x and $\pi \in \text{Aut}_T$. In particular, whenever f is T-supported, its domain X is necessarily T-supported too. A T-supported set is also T'-supported, assuming $T \subseteq T'$.

A set x is *finitely supported* if it has some finite support; in this case x always has the least (inclusion-wise) support, denoted $\text{supp}(x)$, called *the support* of x (cf. [5, Sect. 6]). Thus x is T-supported if and only if $\text{supp}(x) \subseteq T$. Sets supported by $\emptyset$ (i.e., invariant under all atom automorphisms) we call *equivariant.*

Example 6. Given $\alpha, \beta \in \mathbb{A}$, the support of the set $\mathbb{A} \setminus \{\alpha, \beta\}$ is $\{\alpha, \beta\}$. The set $\mathbb{A}^2$ and the projection function $\pi_1 : \mathbb{A}^2 \to \mathbb{A} : (\alpha, \beta) \mapsto \alpha$ are both equivariant; and the support of a tuple $\langle \alpha_1, \ldots, \alpha_n \rangle \in \mathbb{A}^n$, encoded as a set in a standard way, is the set of atoms $\{\alpha_1, \ldots, \alpha_n\}$ appearing in it. ◁

From now on, we shall only consider sets that are hereditarily finitely supported, i.e., ones that have a finite support, whose every element has some finite support, and so on.

Orbit-finite sets. Let $T \subseteq_{\text{fin}} \mathbb{A}$. Two atoms or sets x, y are *in the same* T*-orbit* if $\pi(x) = y$ for some $\pi \in \text{Aut}_T$. This equivalence relation splits atoms and sets into equivalence classes, which we call T*-orbits*; $\emptyset$-orbits we call equivariant orbits, or simply *orbits.* By the very definition, every T-orbit U is T-supported: $\text{supp}(U) \subseteq T$.[2]

T-supported sets are exactly unions of (necessarily disjoint) T-orbits. Finite unions of T-orbits, for any $T \subseteq_{\text{fin}} \mathbb{A}$, are called *orbit-finite* sets. Orbit-finiteness is stable under orbit-refinement: if $T \subseteq T' \subseteq_{\text{fin}} \mathbb{A}$, a finite union of T-orbits is also a finite union of T'-orbits (but the number of orbits may increase, cf. [5, Theorem 3.16]).

Example 7. Examples of orbit-finite sets are:

- the set of atoms $\mathbb{A}$ (1 orbit);
- $\mathbb{A} \setminus \{\alpha\}$ for some $\alpha \in \mathbb{A}$ (1 $\{\alpha\}$-orbit);
- pairs of atoms $\mathbb{A}^2$ (2 orbits: diagonal $\{ \alpha\alpha \mid \alpha \in \mathbb{A} \}$ and off-diagonal $\mathbb{A}^{(2)} = \left\{ \alpha\beta \in \mathbb{A}^2 \;\middle|\; \alpha \neq \beta \right\}$);
- n-tuples of atoms $\mathbb{A}^n$ for $n \in \mathbb{N}$; each orbit $U \subseteq \mathbb{A}^n$ contains all n-tuples of the same equality type, where by the *equality type* of an n-tuple $a_1 \ldots a_n \in \mathbb{A}^n$ we mean the set $\left\{ (i, j) \;\middle|\; a_i = a_j \right\}$;
- non-repeating n-tuples of atoms $\mathbb{A}^{(n)} = \left\{ \alpha_1 \ldots \alpha_n \in \mathbb{A}^n \;\middle|\; \alpha_i \neq \alpha_j \text{ for } i \neq j \right\}$ (1 orbit);
- n-sets of atoms $\binom{\mathbb{A}}{n} = \{ X \subseteq \mathbb{A} \mid |X| = n \}$ (1 orbit).

All of them are equivariant, except $\mathbb{A} \setminus \{\alpha\}$. On the other hand, the set $\mathcal{P}_{\text{fin}}(\mathbb{A})$ of all finite subsets of atoms is orbit-infinite as cardinality is an invariant of each orbit. ◁

We now state few properties to be used in the sequel. For $T \subseteq_{\text{fin}} \mathbb{A}$, each T-orbit $U \subseteq \mathbb{A}^{(n)}$ is determined by fixing pairwise distinct atoms from T on a subset $I \subseteq \{1, \ldots, n\}$ of positions, while allowing arbitrary atoms from $\mathbb{A} \setminus T$ on remaining positions $\{1, \ldots, n\} \setminus I$:

[1] Aut_T is often called the *pointwise stabilizer* of T.
[2] The inclusion may be strict, for singleton T-orbits O. For instance, the singleton $\{\alpha\} \subseteq \mathbb{A}$ is a $\{\alpha\}$-orbit, but also a $\{\alpha, \beta\}$-orbit for $\beta \neq \alpha$.

Lemma 8. *Let $T \subseteq_{fin} \mathbb{A}$. T-orbits $U \subseteq \mathbb{A}^{(n)}$ are exactly sets of the form*

$$\left\{ a \in \mathbb{A}^{(n)} \;\middle|\; \Pi_{n,I}(a) = u, \;\; \Pi_{n,\{1,\ldots,n\}\setminus I}(a) \in (\mathbb{A} \setminus T)^{(n-\ell)} \right\}, \tag{9}$$

where $I \subseteq \{1, \ldots, n\}$, $|I| = \ell$, and $u \in T^{(\ell)}$. The projection $\Pi_{n,I} : \mathbb{A}^{(n)} \to \mathbb{A}^{(\ell)}$ is defined in the expected way.

Indeed, the set (9) is invariant under all T-automorphisms, and each two of its elements are related by some T-automorphism. The orbit (9) is in T-supported bijection with $(\mathbb{A} \setminus T)^{(n-\ell)}$. This is a special case of a general property of every T-orbit, not necessarily included in $\mathbb{A}^{(n)}$. The following lemma is proved exactly as [5, Theorem 6.3] and provides finite representations of T-orbits:

Lemma 9. *Let $T \subseteq_{fin} \mathbb{A}$. Every T-orbit admits a T-supported bijection to a set of the form $(\mathbb{A} \setminus T)^{(n)}/_G$, for some $n \in \mathbb{N}$ and some subgroup G of S_n.*

Recall that each orbit $U \subseteq \mathbb{A}^n$ contains all n-tuples of the same equality type. In particular, each orbit included in $\mathbb{A}^{(n)} \times \mathbb{A}^{(m)} \subseteq \mathbb{A}^{n+m}$ is induced by a partial injection ι from $\{1, \ldots, n\}$ to $\{1, \ldots, m\}$:

Lemma 10. *Orbits $U \subseteq \mathbb{A}^{(n)} \times \mathbb{A}^{(m)}$ are exactly sets of the form $\left\{ (a, b) \in \mathbb{A}^{(n)} \times \mathbb{A}^{(m)} \;\middle|\; \forall i, j : a(i) = b(j) \iff \iota(i) = j \right\}$, where ι is a partial injection from $\{1, \ldots, n\}$ to $\{1, \ldots, m\}$.*

Atom automorphisms preserve the size of the support: $|\mathrm{supp}(X)| = |\mathrm{supp}(\pi(X))|$ for every set X and $\pi \in \textsc{Aut}$. We define *atom dimension* of an orbit as the size of the support of its elements. For instance, atom dimension of $\mathbb{A}^{(n)}$ is n.

3 ORBIT-FINITE (INTEGER) LINEAR PROGRAMMING

We introduce now the setting of linear inequalities we work with, and formulate our main results. We are working in vector spaces over the real[3] field $\mathbb{R}$, where vectors are indexed by a fixed orbit-finite set B, i.e., are functions $\mathbf{v} : B \to \mathbb{R}$. Observe that such a function $\mathbf{v}$, understood as its diagram $\{ (b, \mathbf{v}(b)) \mid b \in B \}$, is orbit-finite exactly when it is finitely supported (according to definitions in Section 2).

Definition 11. By a *vector* over B we mean any orbit-finite (i.e., finitely-supported) function $\mathbf{v}$ from B to $\mathbb{R}$, written $\mathbf{v} : B \to_{\mathrm{fs}} \mathbb{R}$ (vectors are written using boldface). Vectors with integer range, $\mathbf{v} : B \to_{\mathrm{fs}} \mathbb{Z}$, we call *integer* vectors.

The set of all vectors over B we denote by $\textsc{Lin}(B) = B \to_{\mathrm{fs}} \mathbb{R}$. It is a vector space, with pointwise addition and scalar multiplication: for $\mathbf{v}, \mathbf{v}' \in \textsc{Lin}(B)$, $b \in B$ and $q \in \mathbb{R}$, we have $(\mathbf{v} + \mathbf{v}')(b) = \mathbf{v}(b) + \mathbf{v}'(b)$ and $(q \cdot \mathbf{v})(b) = q \cdot \mathbf{v}(b)$. These operation preserve the property of being finitely-supported, e.g., $\mathrm{supp}(\mathbf{v} + \mathbf{v}') \subseteq \mathrm{supp}(\mathbf{v}) \cup \mathrm{supp}(\mathbf{v}')$. We define the *domain* of a vector $\mathbf{v} \in \textsc{Lin}(B)$ as $\mathrm{dom}(\mathbf{v}) = \{ b \in B \mid \mathbf{v}(b) \neq 0 \}$. A vector $\mathbf{v}$ over B is *finitary*, written $\mathbf{v} : B \to_{\mathrm{fin}} \mathbb{R}$, if $\mathrm{dom}(\mathbf{v})$ is finite, i.e., $\mathbf{v}(b) = 0$ for almost all $b \in B$.

Example 12. Let $B = \mathbb{A}^{(2)}$. Let $\alpha, \beta \in \mathbb{A}$ be two fixed atoms. The function $\mathbf{v} : B \to \mathbb{R}$ defined, for $\chi, \gamma \in \mathbb{A} \setminus \{\alpha, \beta\}$, by

$$\mathbf{v}(\alpha\chi) = \mathbf{v}(\chi\alpha) = -1 \qquad \mathbf{v}(\alpha\beta) = \mathbf{v}(\beta\alpha) = 3$$
$$\mathbf{v}(\beta\chi) = \mathbf{v}(\chi\beta) = -2 \qquad \mathbf{v}(\chi\gamma) = 0$$

is an $\{\alpha, \beta\}$-supported integer vector over B. It is not finitary, as $\mathrm{dom}(\mathbf{v}) = \left\{ \delta\sigma \in \mathbb{A}^{(2)} \;\middle|\; \{\delta, \sigma\} \cap \{\alpha, \beta\} \neq \emptyset \right\}$ is infinite. Finitary $\{\alpha, \beta\}$-supported vectors over B assign 0 to all elements of B except for $\alpha\beta$ and $\beta\alpha$. ◁

[3] All the results of the paper still hold if reals $\mathbb{R}$ are replaced by rationals $\mathbb{Q}$ in all the subsequent definitions and results.

A finitary vector $\mathbf{v}$ with domain $\mathrm{dom}(\mathbf{v}) = \{b_1, \ldots, b_k\}$ such that $\mathbf{v}(b_1) = q_1, \ldots, \mathbf{v}(b_k) = q_k$, may be identified with a formal linear combination of elements of B:

$$\mathbf{v} \;=\; q_1 \cdot b_1 + \ldots + q_k \cdot b_k. \tag{10}$$

The subspace of $\textsc{Lin}(B)$ consisting of all finitary vectors we denote by $\textsc{Fin-Lin}(B) = B \to_{\mathrm{fin}} \mathbb{R}$. For finite B of size $|B| = n$, $\textsc{Lin}(B) = \textsc{Fin-Lin}(B)$ is isomorphic to $\mathbb{R}^n$.

For a subset $X \subseteq B$, we denote by $\mathbf{1}_X \in \textsc{Lin}(B)$ the characteristic function of X, i.e., the vector that maps each element of X to 1 and all elements of $B \setminus X$ to 0:

$$\mathbf{1}_X : b \mapsto \begin{cases} 1 & \text{if } b \in X \\ 0 & \text{otherwise.} \end{cases}$$

We write $\mathbf{1}_b$ instead of $\mathbf{1}_{\{b\}}$, and $\mathbf{1}$ instead of $\mathbf{1}_B$.

Lemma 13. *Let $T \subseteq_{fin} \mathbb{A}$ and $\mathbf{v} \in \textsc{Lin}(B)$ such that* $\mathrm{supp}(\mathbf{v}) \subseteq T$. *Then*

(i) $\mathbf{v}$ *is constant, when restricted to every T-orbit $U \subseteq B$;*
(ii) $\mathbf{v}$ *is a linear combination of characteristic vectors $\mathbf{1}_U$ of T-orbits $U \subseteq B$.*

Proof. The first part follows immediately as T supports $\mathbf{v}$. As required in the second part, we have:

$$\mathbf{v} = \sum_U \mathbf{v}(b_U) \cdot \mathbf{1}_U, \tag{11}$$

where U ranges over finitely many T-orbits $U \subseteq B$, and $b_U \in U$ are arbitrarily chosen representatives of T-orbits. □

Notation 14. In the sequel, whenever we know that a vector $\mathbf{v} : B \to_{\mathrm{fs}} \mathbb{R}$ is constant over a T-orbit $U \subseteq B$, we may write $\dot{\mathbf{v}}(U)$ instead of $\mathbf{v}(b)$, where $b \in U$. In particular, when $\mathbf{v}$ is equivariant, we have the *orbit-value* vector

$$\dot{\mathbf{v}} : \textsc{orbits}\,(B) \to \mathbb{R},$$

where $\textsc{orbits}\,(B)$ stands for the set of all equivariant orbits U included in B.

We note that the inner product of vectors $\mathbf{x}, \mathbf{y} \in \textsc{Lin}(B)$,

$$\mathbf{x} \cdot \mathbf{y} \;=\; \sum_{b \in B} \mathbf{x}(b)\, \mathbf{y}(b),$$

is not always well-defined. We consider the right-hand side sum as well-defined when there are only finitely many $b \in B$ for which both $\mathbf{x}(b)$ and $\mathbf{y}(b)$ are non-zero (equivalently, the intersection $\mathrm{dom}(\mathbf{x}) \cap \mathrm{dom}(\mathbf{y})$ is finite).[4]

Orbit-finite systems of linear inequalities. Fix an orbit-finite set C (it can be thought of as the set of unknowns). By a linear inequality over C we mean a pair $e = (\mathbf{a}, t)$ where $\mathbf{a} : C \to_{\mathrm{fs}} \mathbb{Z}$ is an integer vector of left-hand side coefficients and $t \in \mathbb{Z}$ is a right-hand side target value[5]. An $\mathbb{R}$-*solution* (real solution) of e is any vector $\mathbf{x} : C \to_{\mathrm{fs}} \mathbb{R}$ such that the inner product $\mathbf{a} \cdot \mathbf{x}$ is well-defined and

$$\mathbf{a} \cdot \mathbf{x} \geq t;$$

[4] In particular, $\mathbf{x} \cdot \mathbf{y}$ is always well-defined when one of $\mathbf{x}$, $\mathbf{y}$ is finitary.
[5] Rational coefficients and target are easily scaled up to integers.

$\mathbf{x}$ is an $\mathbb{Z}$-*solution* (integer solution) if $\mathbf{x} : C \to_{\text{fs}} \mathbb{Z}$. We may also consider constrained solutions, e.g., *finitary* ones. A linear *equality* $\mathbf{a} \cdot \mathbf{x} = t$ may be encoded by two opposite inequalities:

$$\mathbf{a} \cdot \mathbf{x} \geq t \qquad -\mathbf{a} \cdot \mathbf{x} \geq -t.$$

In this paper we investigate sets of inequalities indexed by an orbit-finite set. Formally, an orbit-finite system of linear inequalities (over C) is the pair $(\mathbf{A}, \mathbf{t})$, where $\mathbf{A} : B \times C \to_{\text{fs}} \mathbb{Z}$ is an integer *matrix* with row index B and column index C, and $\mathbf{t} : B \to_{\text{fs}} \mathbb{Z}$ is an integer *target* vector:

$$\begin{array}{c} \\ \vdots \\ b \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccc} \cdots & c & \cdots \end{array} \\ \left[\begin{array}{ccc} & \vdots & \\ \cdots & \mathbf{A}(b,c) & \cdots \\ & \vdots & \end{array}\right] \end{array} \qquad \begin{array}{c} \\ \left[\begin{array}{c} \vdots \\ \mathbf{t}(b) \\ \vdots \end{array}\right] \end{array}$$

For $b \in B$ we denote by $\mathbf{A}(b, \_) \in \textsc{Lin}(C)$ the corresponding (row) vector. A solution of a system $(\mathbf{A}, \mathbf{t})$ is any vector $\mathbf{x} \in \textsc{Lin}(C)$ which is a solution of all inequalities $(\mathbf{A}(b, \_), \mathbf{t}(b))$, $b \in B$. Equivalently, $\mathbf{x}$ is a solution if $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{t}$, where $\geq$ is the pointwise order on vectors, and the (partial) operation of multiplication of a matrix $\mathbf{A}$ by a vector $\mathbf{x}$ is defined in an expected way:

$$(\mathbf{A} \cdot \mathbf{x})(b) = \mathbf{A}(b, \_) \cdot \mathbf{x}$$

for every $b \in B$. The result $\mathbf{A} \cdot \mathbf{x} \in \textsc{Lin}(B)$ is well-defined if $\mathbf{A}(b, \_) \cdot \mathbf{x}$ is well-defined for all $b \in B$.

By the following examples, restricting to equivariant, finitary or integer solutions only has impact on solvability:

Example 15. Let columns be indexed by $C = \mathbb{A}$, and consider the system consisting of just one infinitary inequality $(\mathbf{1}_{\mathbb{A}}, 1)$ (B is thus a singleton). Identifying column indexes $\alpha \in \mathbb{A}$ with unknowns, the inequality may be written as:

$$\sum_{\alpha \in \mathbb{A}} \alpha \;\geq\; 1.$$

The inequality has an integer (finitary) solution, i.e., $\mathbf{x} = \mathbf{1}_\alpha$ for any $\alpha \in \mathbb{A}$, but no equivariant one. Indeed, equivariant vectors $\mathbf{x} : \mathbb{A} \to_{\text{fs}} \mathbb{R}$ are necessarily constant ones $\mathbf{x} = r \cdot \mathbf{1}_{\mathbb{A}}$ (cf. Lemma 13), and then the inner product

$$\mathbf{1}_{\mathbb{A}} \cdot \mathbf{x} = \sum_{\alpha \in \mathbb{A}} \mathbf{x}(\alpha) = \sum_{\alpha \in \mathbb{A}} r$$

is well-defined only if $r = 0$, i.e. $\mathbf{x}(\alpha) = 0$ for all $\alpha \in \mathbb{A}$. ◁

Example 16. Let columns be indexed by $C = \mathbb{A}^{(2)}$ and rows by $B = \binom{\mathbb{A}}{2}$. Consider the system containing, for every $\{\alpha, \beta\} \in B$, the inequality $(\mathbf{1}_{\alpha\beta} + \mathbf{1}_{\beta\alpha}, 1)$. Using the formal-sum notation as in (10) it may be written as $(\alpha\beta + \beta\alpha, 1)$ or, identifying column indexes $\alpha\beta \in C$ with unknowns, as:

$$\alpha\beta + \beta\alpha \;\geq\; 1 \qquad\qquad (\alpha, \beta \in \mathbb{A}, \alpha \neq \beta).$$

All the equations are thus finitary, and the target is $\mathbf{t} = \mathbf{1}_B$. The constant vector $\mathbf{x} = \frac{1}{2} : \alpha\beta \mapsto \frac{1}{2}$ is a solution, even if we extend the system with symmetric inequalities

$$\alpha\beta + \beta\alpha \;\leq\; 1 \qquad\qquad (\alpha, \beta \in \mathbb{A}, \alpha \neq \beta).$$

The extended system has no finitary solution. It has no integer solution either. Indeed, since we restrict to finitely supported solutions only, any such solution $\mathbf{x}$ necessarily satisfies, for every distinct atoms $\alpha, \beta \in \mathbb{A} \setminus \text{supp}(\mathbf{x})$, the equality $\mathbf{x}(\alpha\beta) = \mathbf{x}(\beta\alpha)$, which is incompatible with $\mathbf{x}(\alpha\beta) + \mathbf{x}(\beta\alpha) = 1$. ◁

4 RESULTS

Solvability problems. We investigate decision problems of solvability of orbit-finite systems of inequalities over the ring of reals or integers. Consequently, we use $\mathbb{F}$ to stand either for $\mathbb{R}$ or $\mathbb{Z}$. We identify a couple of variants. In the first one we ask about existence of a finitely-supported solution:

Ineq-Solv($\mathbb{F}$):
Input: *An orbit-finite system of linear inequalities.*
Question: *Does it have an $\mathbb{F}$-solution?*

We recall that solutions are finitely-supported (equivalently, orbit-finite), by definition. A closely related variant is solvability of equalities, under the restriction to nonnegative solutions only:

Nonneg-Eq-Solv($\mathbb{F}$):
Input: *An orbit-finite system of linear equations.*
Question: *Does it have a nonnegative $\mathbb{F}$-solution?*

Furthermore, both the problems have "finitary" versions, where one seeks for finitary solutions only, denoted as Fin-Ineq-Solv($\mathbb{F}$) and Fin-Nonneg-Eq-Solv($\mathbb{F}$), respectively.

Three out of the four problems are inter-reducible, and hence equi-decidable, both for $\mathbb{F} = \mathbb{R}$ and $\mathbb{F} = \mathbb{Z}$:

Theorem 17. *Let $\mathbb{F} \in \{\mathbb{R}, \mathbb{Z}\}$. The problems* Ineq-Solv($\mathbb{F}$), Fin-Ineq-Solv($\mathbb{F}$) *and* Nonneg-Eq-Solv($\mathbb{F}$) *are inter-reducible. All reductions are in PTime, except the one from* Ineq-Solv($\mathbb{F}$) *to* Fin-Ineq-Solv($\mathbb{F}$) *which is in ExpTime, but in PTime for fixed atom dimension.*

(The proof is in Section A.1.) The three problems listed in Theorem 17 deserve a shared name *orbit-finite linear programming* (in case of $\mathbb{F} = \mathbb{R}$) and *orbit-finite integer linear programming* (in case of $\mathbb{F} = \mathbb{Z}$). Figure 1 shows the reductions of Theorem 17 using dashed arrows.

As our two main results we prove that the linear programming is decidable, while the integer one is not. Furthermore, the complexity of the decision procedure is exponential in atom dimension of the input system, but polynomial in the number of orbits. This yields ExpTime complexity in general, and PTime complexity for any fixed atom dimension of input.

Theorem 18. Fin-Ineq-Solv($\mathbb{R}$) *is decidable in ExpTime. For fixed atom dimension, it is decidable in PTime.*

Theorem 19. Fin-Ineq-Solv($\mathbb{Z}$) *is undecidable.*

(The proofs occupy Sections 7 and 9, respectively.) Additionally, we settle the status of the last variant, Fin-Nonneg-Eq-Solv($\mathbb{F}$). The problem is decidable for $\mathbb{F} = \mathbb{R}$, as it reduces to both Fin-Ineq-Solv($\mathbb{F}$) and Nonneg-Eq-Solv($\mathbb{F}$) via reductions analogous to those of Theorem 17. We derive decidability also in case $\mathbb{F} = \mathbb{Z}$:

Theorem 20. Fin-Nonneg-Eq-Solv($\mathbb{F}$) *is decidable, for $\mathbb{F} \in \{\mathbb{R}, \mathbb{Z}\}$.*

(The proof is in Section A.2.) In consequence of Theorems 19 and 20, in case $\mathbb{F} = \mathbb{Z}$ the two arrows outgoing from Fin-Nonneg-Eq-Solv($\mathbb{Z}$) in Figure 1 can not be completed by the reverse arrows.

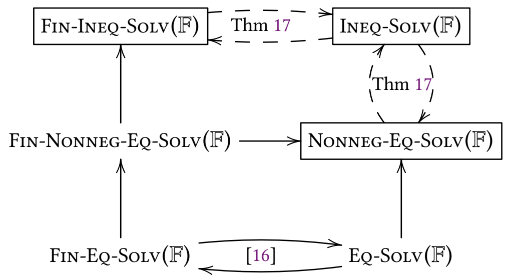


Fig. 1. Diagram of reductions between solvability problems.

Linear equations vs inequalities. Solvability of orbit-finite systems of *equations* (Eq-Solv($\mathbb{F}$)) easily reduces to Ineq-Solv($\mathbb{F}$), by replacing each equation with two opposite inequalities, but also to Nonneg-Eq-Solv($\mathbb{F}$), by replacing each unknown with a difference of two unknowns. Likewise does the variant Fin-Eq-Solv($\mathbb{F}$), where one only seeks for finitary solutions.

Theorem 21 ([16] Thms 4.4 and 6.1). Eq-Solv($\mathbb{F}$) *and* Fin-Eq-Solv($\mathbb{F}$) *are inter-reducible and decidable*[6].

In summary, for each choice of $\mathbb{F}$ one may distinguish three different decision problems: solving of systems of linear equations (two bottom nodes in Figure 1), solving of system of linear inequalities (three upper nodes in Figure 1), and the intermediate problem Fin-Nonneg-Eq-Solv($\mathbb{F}$).

Optimisation problems. We consider $\mathbb{F} = \mathbb{R}$, due to the undecidability of Theorem 19. All variants of linear programming mentioned above have corresponding *maximisation* problems. In each variant the input contains, except for a system $(\mathbf{A}, \mathbf{t})$, an integer vector $\mathbf{s} : C \to_{\text{fs}} \mathbb{Z}$ that represents a (partial) linear *objective* function $S : \text{Lin}(C) \to_{\text{fs}} \mathbb{R}$, defined by

$$S(\mathbf{x}) = \mathbf{s} \cdot \mathbf{x}.$$

The maximisation problem asks to compute the supremum of the objective function over all (finitary, nonnegative) solutions of $(\mathbf{A}, \mathbf{t})$. A symmetrical *minimisation* problem is easily transformed to a maximisation one by replacing $\mathbf{s}$ with $-\mathbf{s}$. This yields three optimisation problems Ineq-Max($\mathbb{R}$), Fin-Ineq-Max($\mathbb{R}$) and Nonneg-Eq-Max($\mathbb{R}$) which are, as before, inter-reducible:

Theorem 22. *The problems* Ineq-Max($\mathbb{R}$), Fin-Ineq-Max($\mathbb{R}$) *and* Nonneg-Eq-Max($\mathbb{R}$) *are inter-reducible, with the same complexity as in Theorem 17.*

(The proof is in Section A.1.) As our last main result we strengthen Theorem 18 to the optimisation setting:

Theorem 23. Fin-Ineq-Max($\mathbb{R}$) *is computable in ExpTime. For fixed atom dimension, it is computable in PTime.*

(The proof is in Section 8.) Hence, for every fixed atom dimension, orbit-finite linear programming is not more costly than the classical finite linear programming.

[6] The results of [16] apply to systems of equations where coefficients and solutions are from any fixed commutative and effective ring $\mathbb{F}$. This includes integers $\mathbb{Z}$ or rationals $\mathbb{Q}$ (and hence applies also to real solutions).

Representation of input. There are several possible ways of representing input $(\mathbf{A}, \mathbf{t}, \mathbf{s})$ to our algorithms. One possibility is to rely on the equivalence between (hereditary) orbit-finite sets and *definable* sets [5, Sect. 4]. We choose another standard possibility, as specified in items (1)–(3) below. First, the representation includes:

(1) a common support $T \subseteq_{\text{fin}} \mathbb{A}$ of $\mathbf{A}$, $\mathbf{t}$ and $\mathbf{s}$.

Second, knowing that B and C are disjoint unions of T-orbits, and relying on Lemma 9, the representation includes also:

(2) a list of all T-orbits included in B and C, each one represented by some tuple $a \in \mathbb{A}^{(n)}$ and $G \leq S_n$; and a list of T-orbits included in $B \times C$, each one represented by some its element.

Finally, relying on Lemma 13, we assume that the representation includes also:

(3) a list of integer values $\dot{\mathbf{t}}(U)$, $\dot{\mathbf{s}}(U)$, and $\dot{\mathbf{A}}(U)$, respectively, for all T-orbits U included in B, C, and $B \times C$, respectively (we apply Notation 14). Integers are represented in binary.

Strict inequalities. In this paper we consider system of *non-strict* inequalities, for the sake of presentation. The decision procedures of Theorems 18 and 23, work equally well if both *non-strict and strict* inequalities are allowed. Reductions between Fin-Ineq-Solv($\mathbb{F}$) and Ineq-Solv($\mathbb{F}$) work as well, but not the reductions from (Fin-)Nonneg-Eq-Solv($\mathbb{F}$) to (Fin-)Ineq-Solv($\mathbb{F}$) as we can not simulate equalities with strict inequalities.

Proviso. When investigating different systems of inequalities in the following sections, we implicitly consider their *real* solutions, unless specified otherwise.

5 POLYNOMIALLY-PARAMETRISED INEQUALITIES

We now introduce a core problem that will serve as a target of reductions in the proofs of Theorems 18 and 23 in Sections 7 and 8. Consider a finite inequality $\mathcal{E}$ of the form:

$$p_1(n) \cdot x_1 + \ldots + p_k(n) \cdot x_k \;\geq\; q(n), \tag{12}$$

where $p_1, \ldots, p_n$ and q are univariate polynomials with integer coefficients, and $x_1, \ldots, x_n$ are unknowns. The special unknown n plays a role of a nonnegative integer parameter, and that is why we call such an inequality *polynomially-parametrised*. For every fixed value $n \in \mathbb{N}$, by evaluating all polynomials in n we get an ordinary inequality $\mathcal{E}(n)$ with integer coefficients. Also, if n does not appear in $\mathcal{E}$, i.e., all polynomials are constants, $\mathcal{E}$ is an ordinary inequality.

In the sequel we study solvability of a finite system P of such inequalities (12) with the same unknowns $x_1, \ldots, x_k$. Again, by evaluating all polynomials in n we get an ordinary system $P(n)$. We use the matrix form $P(n) = (\mathbf{A}(n), \mathbf{t}(n))$ when convenient. A fundamental problem is to check if for some value $n \in \mathbb{N}$, the system $P(n)$ has a real solution:

Poly-Ineq-Solv:
Input: *A finite system of polynomially-parametrised inequalities P.*
Question: *Does $P(n)$ have a real solution for some $n \in \mathbb{N}$?*

Theorem 24. Poly-Ineq-Solv *is decidable.*

(The proof is in Section A.3.) In the sequel we will not use the decision procedure of Theorem 24, but rather the algorithm of Theorem 25 stated below, since our later applications only use *monotonic* instances of Poly-Ineq-Solv.

5.1 Monotonic polynomially-parametrised inequalities

A system P is *monotonic* if there is some $n_0 \in \mathbb{N}$ such that every solution of $P(n)$, for an integer $n \geq n_0$, is also a solution of $P(n+1)$. Note that monotonicity vacuously holds (with any value of n_0) if n does not appear in P, i.e., when P is an ordinary (non-parametrised) system. When P is monotonic, a solution of $P(n)$ for $n \geq n_0$ is also a solution of $P(n')$ for all integers $n' \geq n$. Therefore, instead of Poly-Ineq-Solv we prefer to use in the sequel the following core problem, where we do not assume monotonicity but seek for a solution of $P(n)$ for *almost all* (all sufficiently large) values of the parameter $n \in \mathbb{N}$:

Almost-all-Poly-Ineq-Solv:
Input: *A finite system of polynomially-parametrised inequalities P.*
Question: *Is there $n_0 \in \mathbb{N}$ and a vector $(x_1, \ldots, x_k)$ which is a solution of $P(n)$ for every integer $n \geq n_0$?*

From now on, a vector $(x_1, \ldots, x_k)$ which is a solution of an inequality $\mathcal{E}(n)$ (resp. a system $P(n)$) for almost all $n \in \mathbb{N}$ we call *almost-all-solution* of $\mathcal{E}$ (resp. P). The rest of this section is devoted to designing a PTime algorithm for Almost-all-Poly-Ineq-Solv:

Theorem 25. Almost-all-Poly-Ineq-Solv *is in PTime.*

Proof. Consider a polynomially-parametrised inequality $\mathcal{E}$ of the form:

$$p_1(n) \cdot x_1 + \ldots + p_k(n) \cdot x_k \;\geq\; q(n). \tag{13}$$

Let d be the maximal degree of polynomials $p_1, \ldots, p_k, q$ appearing in $\mathcal{E}$. We call d *the degree* of $\mathcal{E}$, and denote it also as $\deg \mathcal{E}$. Let $a_1, \ldots, a_k, b$ be (integer) coefficients of the monomial n^d in $p_1, \ldots, p_k, q$, respectively. Therefore

$$p_1(n) = a_1 \cdot n^d + p'_1(n) \qquad \ldots \qquad p_k(n) = a_k \cdot n^d + p'_k(n) \qquad\qquad q(n) \quad = b \cdot n^d + q'(n) \tag{14}$$

for some polynomials $p'_1, \ldots, p'_k, q$ of degree strictly smaller than d. The ordinary inequality with integer coefficients

$$a_1 \cdot x_1 + \ldots + a_k \cdot x_k \geq b, \tag{15}$$

we call *the head inequality* of $\mathcal{E}$, and denote by $\textsc{hd}(\mathcal{E})$. Furthermore, the polynomially-parametrised inequality

$$p'_1(n) \cdot x_1 + \ldots + p'_k(n) \cdot x_k \;\geq\; q'(n), \tag{16}$$

obtained by removing all appearances of the monomial n^d, we call the *tail* of $\mathcal{E}$, and denote it by $\textsc{tl}(\mathcal{E})$. We also consider below the strict strengthening of the head inequality (15), denoted as $\textsc{hd}_>(\mathcal{E})$, and the equality, denoted as $\textsc{hd}_=(\mathcal{E})$:

$$a_1 \cdot x_1 + \ldots + a_k \cdot x_k > b, \qquad\qquad a_1 \cdot x_1 + \ldots + a_k \cdot x_k = b. \tag{17}$$

As $\mathcal{E}$ is equal to the sum of its head $\textsc{hd}(\mathcal{E})$ multiplied by n^d, and its tail $\textsc{tl}(\mathcal{E})$, we immediately deduce:

Claim 26. *For every $n \in \mathbb{N}$, every solution of $\textsc{hd}_=(\mathcal{E})$ is either a solution of both $\mathcal{E}(n)$ and $\textsc{tl}(\mathcal{E})(n)$, or of none of them.*

We now provide under- and over-approximations of the solution set of $\mathcal{E}$ (in Claims 28 and 27).

Claim 27. *Every almost-all-solution of $\mathcal{E}$ is also a solution of $\textsc{hd}(\mathcal{E})$.*

Proof. Consider an inequality $\mathcal{E}$ (13) and its almost-all-solution $\mathbf{x} = (x_1, \ldots, x_k)$. Let $d = \deg \mathcal{E}$. We thus have

$$\frac{p_1(n)}{n^d} \cdot x_1 + \ldots + \frac{p_k(n)}{n^d} \cdot x_k \;\geq\; \frac{q(n)}{n^d}$$

for all sufficiently large $n \in \mathbb{N}$. Using the decomposition (14), we rewrite the above inequality to

$$\left(a_1 + \frac{p_1'(n)}{n^d}\right) \cdot x_1 + \ldots + \left(a_k + \frac{p_k'(n)}{n^d}\right) \cdot x_k \ \geq \ b + \frac{q'(n)}{n^d}.$$

As the degrees of all polynomials $p_1', \ldots, p_k', q'$ are smaller than d, all the fractions tend to 0 when n tends to ∞, and we may deduce

$$a_1 \cdot x_1 + \ldots + a_k \cdot x_k \ \geq \ b,$$

i.e., $\mathbf{x}$ is a solution of $\text{HD}(\mathcal{E})$, as required. □

Claim 28. *Every solution of* $\text{HD}_>(\mathcal{E})$ *is also an almost-all-solution of* $\mathcal{E}$.

Proof. Let $d = \deg \mathcal{E}$. Consider any vector $\mathbf{x} = (x_1, \ldots, x_k)$ satisfying the strict inequality $\text{HD}_>(\mathcal{E})$ (in (17) on the left). Therefore for any polynomials $p_1', \ldots, p_k', q'$ of degree strictly smaller than d, the inequality

$$\left(a_1 + \frac{p_1'(n)}{n^d}\right) \cdot x_1 + \ldots + \left(a_k + \frac{p_k'(n)}{n^d}\right) \cdot x_k > b + \frac{q'(n)}{n^d}$$

is satisfied for all sufficiently large $n \in \mathbb{N}$. Applying the above inequality to polynomials appearing in (14), we obtain:

$$\frac{p_1(n)}{n^d} \cdot x_1 + \ldots + \frac{p_k(n)}{n^d} \cdot x_k \ > \ \frac{q(n)}{n^d}$$

for all sufficiently large $n \in \mathbb{N}$. We multiply both sides by n^d in order to derive that $\mathbf{x}$ is a solution of $\mathcal{E}(n)$ for all sufficiently large $n \in \mathbb{N}$, as required. □

Consider an instance P of Poly-Ineq-Solv, i.e., a finite system of polynomially-parametrised inequalities of the form (13). Let $\text{HD}(P) := \{ \text{HD}(\mathcal{E}) \mid \mathcal{E} \in P \}$ be the system of head inequalities (note that degrees of different inequalities in P may differ), and let $\text{HD}_>(P) := \{ \text{HD}_>(\mathcal{E}) \mid \mathcal{E} \in P \}$. Using Claims 28 and 27 we derive:

Claim 29. *Every almost-all-solution of P is also a solution of* $\text{HD}(P)$.

Claim 30. *Every solution of* $\text{HD}_>(P)$ *is also an almost-all-solution of P.*

For time estimation, as the size measure $|\mathcal{E}|$ of an inequality $\mathcal{E}$ we take the total number of monomials appearing in $\mathcal{E}$. In particular, $|\mathcal{E}| > |\text{TL}(\mathcal{E})|$. The size of a system P is the sum of sizes of all its inequalities. For two systems P', P'' of inequalities, we denote their union by $P' \cup P''$ (clearly, union of systems corresponds to conjunction of constraints). We write $P \cup \mathcal{E}$ instead of $P \cup \{\mathcal{E}\}$. By $P \setminus \mathcal{E}$ we denote the system obtained from P by removing an inequality $\mathcal{E}$.

The algorithm. A decision procedure for Almost-all-Poly-Ineq-Solv iteratively transforms an instance of the form $P \cup \Gamma$, where P is a system of polynomially-parametrised inequalities, and Γ is a system of ordinary (non-parametrised) equalities over the same unknowns. Initially, Γ is empty. We define a transformation step that given such an instance $P \cup \Gamma$, either confirms its solvability (existence of an almost-all-solution), or confirms its non-solvability (non-existence of an almost-all-solution), or outputs an instance $P' \cup \Gamma'$ which has the same almost-all-solutions as $P \cup \Gamma$, and such that $|P'| < |P|$. Almost-all-Poly-Ineq-Solv is solved by iterating the transformation step until it confirms either solvability or non-solvability. Termination after a polynomial number of iterations is guaranteed, as $|P|$, while being nonnegative, strictly decreases in each iteration. The transformation step invokes a PTime procedure for ordinary linear programming (as detailed in (18) and (19) below). Here is a pseudo-code of the algorithm:

Algorithm 1 (Almost-all-Poly-Ineq-Solv)

Input: A polynomially-parametrised system P.
$\Gamma \leftarrow \emptyset$
repeat
    if $\textsc{hd}(P) \cup \Gamma$ (18) is non-solvable **then** ▷ see Claim 29
        report non-solvability of P
    else
        if $\textsc{hd}_>(\mathcal{E}) \cup \textsc{hd}(P \setminus \mathcal{E}) \cup \Gamma$ (19) is solvable for all $\mathcal{E} \in P$ **then** ▷ see Claims 30, 31
            report solvability of P
        else
            choose any $\mathcal{E} \in P$ **such that** $\textsc{hd}_>(\mathcal{E}) \cup \textsc{hd}(P \setminus \mathcal{E}) \cup \Gamma$ (19) is non-solvable ▷ see Claim 32
            $P \leftarrow (P \setminus \mathcal{E}) \cup \textsc{tl}(\mathcal{E})$
            $\Gamma \leftarrow \Gamma \cup \textsc{hd}_=(\mathcal{E})$
until solvability or non-solvability of P is reported

Transformation step. The step, defined by the body of the **repeat** loop, proceeds as follows. If the ordinary system

$$\textsc{hd}(P) \ \cup \ \Gamma \tag{18}$$

is non-solvable, non-solvability of $P \cup \Gamma$ is reported. This is correct due to Claim 29. Otherwise, knowing that (18) is solvable, the algorithm checks, for every $\mathcal{E} \in P$, whether the strengthened system

$$\textsc{hd}_>(\mathcal{E}) \ \cup \ \textsc{hd}(P \setminus \mathcal{E}) \ \cup \ \Gamma, \tag{19}$$

obtained from (18) by replacing the inequality $\textsc{hd}(\mathcal{E})$ with $\textsc{hd}_>(\mathcal{E})$, is also solvable. If this is the case, solvability of $P \cup \Gamma$ is reported. This is correct due to Claim 30 combined with the following one:

Claim 31. *Solvability of* (19) *for every inequality* $\mathcal{E}$ *in* P*, implies solvability of*

$$\textsc{hd}_>(P) \ \cup \ \Gamma. \tag{20}$$

Proof. Let m be the number of inequalities in P, and suppose that for every inequality $\mathcal{E}$ in P, the system (19) has a solution, $\mathbf{x}_{\mathcal{E}}$. All $\mathbf{x}_{\mathcal{E}}$ are thus solutions of (18), and since the solution set of (18) is convex, the average of all these solutions $\frac{1}{m} \cdot \sum_{\mathcal{E} \in P} \mathbf{x}_{\mathcal{E}}$ is then a solution of $\textsc{hd}_>(P) \ \cup \ \Gamma$. □

Otherwise, we know that some inequality $\mathcal{E}$ in P is *degenerate*, namely (19) is non-solvable. In other words, the equality $\textsc{hd}_=(\mathcal{E})$ is implied by (18). The algorithm chooses a degenerate inequality $\mathcal{E} \in P$ and creates a new instance $P' \cup \Gamma'$, where

$$P' \ = \ (P \setminus \mathcal{E}) \ \cup \ \textsc{tl}(\mathcal{E}) \qquad\qquad \Gamma' \ = \ \Gamma \ \cup \ \textsc{hd}_=(\mathcal{E}).$$

In words, P' is obtained from P by replacing $\mathcal{E}$ with $\textsc{tl}(\mathcal{E})$, and Γ' is obtained from Γ by adding $\textsc{hd}_=(\mathcal{E})$. As $|\textsc{tl}(\mathcal{E})| < |\mathcal{E}|$, we have $|P'| < |P|$, as required. This completes description of the transformation step.

Correctness. By Claim 26 we derive:

Claim 32. *Systems* $P \cup \Gamma$ *and* $P' \cup \Gamma'$ *have the same almost-all-solutions.*

Proof. In one direction, consider an almost-all-solution $\mathbf{x}$ of $P' \cup \Gamma'$. It is trivially a solution of Γ. Furthermore, being a solution of $\textsc{hd}_=(\mathcal{E})$ and of $\textsc{tl}(\mathcal{E})(n)$ for almost all $n \in \mathbb{N}$, by Claim 26 it is a solution of $\mathcal{E}(n)$ for almost all n, and hence an almost-all-solution of P.

Conversely, consider an almost-all-solution $\mathbf{x}$ of $P \cup \Gamma$. By Claim 29, it is a solution of $\textsc{hd}(P) \cup \Gamma$ and hence, as $\mathcal{E}$ is degenerate, also a solution of $\textsc{hd}_=(\mathcal{E})$. Therefore $\mathbf{x}$ is a solution of Γ'. Furthermore, being a solution of $\textsc{hd}_=(\mathcal{E})$ and of $\mathcal{E}(n)$ for all sufficiently large $n \in \mathbb{N}$, by Claim 26 it is also a solution of $\textsc{tl}(\mathcal{E})(n)$ for all sufficiently large $n \in \mathbb{N}$, and hence an almost-all-solution of P'. □

Complexity. We note that the main loop of the algorithm always terminates, at latest when $P = \emptyset$, as in this case the system (19) is vacuously solvable for all $\mathcal{E} \in P$. Solvability of (18) in line 4 is checked by one solvability test of an ordinary system of inequalities. Solvability of (19) in line 7 is also checkable in polynomial time due to the following claim applied to $Q = \textsc{hd}(P) \cup \Gamma$:

Claim 33. *Given an ordinary system Q of linear inequalities and $\mathcal{E} \in Q$, one can check, in PTime, solvability of $\mathcal{E}_> \cup (Q \setminus \mathcal{E})$, where $\mathcal{E}_>$ is the strict strengthening of $\mathcal{E}$.*

Proof. We invoke ordinary linear programming twice (in PTime, see e.g. [29, Section 8.7]). Let $\mathcal{E}$ be of the form $a_1 \cdot x_1 + \ldots + a_k \cdot x_k \geq b$. If Q is non-solvable, the algorithm reports non-solvability of $\mathcal{E}_> \cup (Q \setminus \mathcal{E})$. Otherwise, the algorithm computes the supremum $M \in \mathbb{Q} \cup \{\infty\}$ of the objective function

$$S(x_1, \ldots, x_k) = a_1 \cdot x_1 + \ldots + a_k \cdot x_k,$$

constraint by $Q \setminus \mathcal{E}$, by invoking ordinary linear programming. By solvability of Q we know that $M \geq b$. If $M > b$, the algorithm reports solvability, otherwise it reports non-solvability. □

Number of iterations of transformation step is polynomial (as $|P|$ decreases in each iteration) and hence so is the number of inequalities in Γ. In consequence, the number of invocations of ordinary linear programming is polynomial in each transformation step, and hence polynomial in total, and each its instance of ordinary linear programming is also polynomial. Summing up, our decision procedure for Almost-all-Poly-Ineq-Solv works in PTime.

The proof of Theorem 25 is thus completed. □

Remark 34. We do not need any explicit bound on the threshold value of n_0 guaranteeing that every almost-all-solution of P is a solution of $P(n)$ for every integer $n \geq n_0$. On the other hand, an exponential bound is derivable from our algorithm. Assuming P has an almost-all-solution, P has also an almost-all-solution $\mathbf{x}$ which is at most exponentially large, e.g., a solution of an ordinary system (20). Substituting $\mathbf{x}$ into $P(n)$ yields a system of univariate polynomial inequalities, and one can take as threshold n_0 any integer larger than all nonnegative roots of all polynomials appearing in the system. As roots of univariate polynomials are polynomially bounded, we deduce the bound for n_0. ◁

Example 35. Recall two polynomially-parametrised inequalities (8) in Example 4 in Section 1. They have the same head inequality $x \geq 1$, which is trivially solvable, and hence the algorithm reports solvability after the first iteration. Both the tail inequalities, $-x \geq 1$ and $0 \geq 1$, are ordinary (non-parametrised).

The following instance P_0 admits three iterations of the main loop of the algorithm:

$$\begin{aligned} n^2 \cdot x \; - \; n^2 \cdot y \; + \; n \cdot z \; &\geq \; 0 \\ -n \cdot x \; + \; (n+3) \cdot y \quad\quad &\geq \; 0 \end{aligned}$$

The head inequalities of these two inequalities are $x - y \geq 0$ and $-x + y \geq 0$, respectively. Therefore the system $\textsc{hd}(P_0)$ is equivalent to $x = y$ and hence solvable, while $\textsc{hd}_>(P_0)$ is not, and both inequalities in P_0 are degenerate. Supposing the first one is chosen by the algorithm, after the first iteration we get the following systems P_1 (left) and Γ_1 (right):

$$\begin{aligned} n \cdot z &\geq 0 \\ -n \cdot x + (n+3) \cdot y &\geq 0 \end{aligned} \qquad\qquad x - y = 0$$

In the second iteration, the system $\textsc{hd}(P_1) \cup \Gamma_1$ (left) is solvable but the system $\textsc{hd}_>(P_1) \cup \Gamma_1$ (right) is not:

$$\begin{aligned} z &\geq 0 \\ -x + y &\geq 0 \\ x - y &= 0 \end{aligned} \qquad\qquad \begin{aligned} z &> 0 \\ -x + y &> 0 \\ x - y &= 0 \end{aligned}$$

The algorithm picks up the second inequality in P_1, the only degenerate one, and sets P_2 (left) and Γ_2 (right):

$$\begin{aligned} n \cdot z &\geq 0 \\ 3 \cdot y &\geq 0 \end{aligned} \qquad\qquad \begin{aligned} x - y &= 0 \\ -x + y &= 0 \end{aligned}$$

In the last third iteration, the system $\textsc{hd}_>(P_2) \cup \Gamma_2$ (obtained by replacing the inequality $n \cdot z \geq 0$ by $z > 0$) is solvable, and hence solvability of P_0 is reported. ◁

6 FINITELY SETWISE-SUPPORTED SETS

In this section we introduce the novel concept of *setwise-support*, playing a central role in the proofs of Theorems 18 and 23. In short, we replace *pointwise* stabilisers by *setwise* ones.

For any $T \subseteq_{\text{fin}} \mathbb{A}$ consider the set of all atom automorphisms that preserve T as a set only (called *setwise-T-automorphisms*):

$$\textsc{Aut}_{\{T\}} = \{ \pi \in \textsc{Aut} \mid \pi(T) = T \}.^{7}$$

Accordingly, we define *setwise-T-orbits* as equivalence classes with respect to the action of $\textsc{Aut}_{\{T\}}$: two sets (elements) x, y are in the same setwise-T-orbit if $\pi(x) = y$ for some $\pi \in \textsc{Aut}_{\{T\}}$. We have

$$\textsc{Aut}_T \subseteq \textsc{Aut}_{\{T\}} \subseteq \textsc{Aut},$$

and hence every equivariant orbit splits into finitely many setwise-T-orbits, each of which splits in turn into finitely many T-orbits. A set X is *setwise-T-supported* if $\pi(X) = X$ for all $\pi \in \textsc{Aut}_{\{T\}}$. Equivalently, X is a union of setwise-T-orbits. Note that each setwise-T-supported set is T-supported, but the opposite implication is not true. When T is irrelevant, we speak of finitely setwise-supported sets. Finally notice that a setwise-T-supported set is not necessarily setwise-T'-supported for $T \subseteq T'$, which distinguishes setwise-support from standard support.

Example 36. Let $T = \{\alpha, \beta\} \subseteq \mathbb{A}$. The vector $\mathbf{v}$, defined in Example 12 in Section 3, is not setwise-T-supported. Indeed,

$$\pi(\mathbf{v})(\alpha, \chi) = \mathbf{v}(\beta, \chi) \neq \mathbf{v}(\alpha, \chi)$$

[7] $\textsc{Aut}_{\{T\}}$ is often called the *setwise stabilizer* of T.

for any $\chi \notin T$ and $\pi \in \textsc{Aut}_{\{T\}}$ that swaps α and β but preserves all other atoms. The *averaged* vector $\mathbf{v}'$ defined by

$$\mathbf{v}'(\alpha\chi) = \mathbf{v}'(\chi\alpha) = -1.5 \qquad \mathbf{v}'(\alpha\beta) = \mathbf{v}'(\beta\alpha) = 3$$
$$\mathbf{v}'(\beta\chi) = \mathbf{v}'(\chi\beta) = -1.5 \qquad \mathbf{v}'(\chi\gamma) = 0,$$

for $\chi, \gamma \in \mathbb{A} \setminus \{\alpha, \beta\}$, is setwise-$T$-supported. Notice that $\mathbf{v}'$, is not setwise-$(T \cup \{\gamma\})$-supported, for $\gamma \notin T$. ◁

Clearly, with the size of T increasing towards infinity, the number of T-orbits included in one equivariant orbit may increase towards infinity as well. The crucial property of setwise-T-supported sets is that they do not suffer from this unbounded growth: the number of setwise-T-orbits included in a fixed equivariant orbit is bounded, no matter how large T is. We will need this property for setwise-T-orbits $U \subseteq \mathbb{A}^{(n)}$, $n \in \mathbb{N}$, and it follows immediately by Lemma 37. Intuitively speaking, each such setwise-T-orbit is determined by a subset $I \subseteq \{1, \ldots, n\}$ of positions which is filled by arbitrary pairwise different atoms from T, the remaining positions $\{1, \ldots, n\} \setminus I$ are filled by arbitrary atoms from $\mathbb{A} \setminus T$ (cf. Lemma 8 in Section 2).

Lemma 37. *Let $T \subseteq_{fin} \mathbb{A}$ of size $|T| \geq n$. Each setwise-T-orbit $U \subseteq \mathbb{A}^{(n)}$ is of the form*

$$U = \left\{ a \in \mathbb{A}^{(n)} \;\middle|\; \Pi_{n,I}(a) \in T^{(\ell)},\;\; \Pi_{n,\{1,\ldots,n\}\setminus I}(a) \in \left(\mathbb{A} \setminus T\right)^{(n-\ell)} \right\}, \tag{21}$$

for some $I \subseteq \{1, \ldots, n\}$ of size ℓ.

Proof. Consider any tuple $t = (\alpha_1, \ldots, \alpha_n) \in \mathbb{A}^{(n)}$. Let $I = \{\, i \in \{1, \ldots, n\} \mid \alpha_i \in T \,\}$ denote the positions in t filled by atoms from T. By applying all setwise-T-automorphisms to t, we obtain all tuples, where positions from I are arbitrarily filled by elements of T, and positions outside of I are arbitrarily filled by elements of $\mathbb{A} \setminus T$. □

Notation 38. Given a finitary vector $\mathbf{x} : C \to_{\text{fin}} \mathbb{R}$ and an equivariant orbit $U \subseteq C$, we write

$$\mathbf{x}^{\Sigma}(U) \;=\; \sum_{c \in U} \mathbf{x}(c)$$

to denote for the sum of $\mathbf{x}(c)$ ranging over all $c \in U$. This yields the finite *orbit-sum* vector

$$\mathbf{x}^{\Sigma} : \textsc{orbits}\,(C) \to \mathbb{R}$$

mapping the equivariant orbits included in C to $\mathbb{R}$.

A key observation is that a solvable equivariant system necessarily has a finitely setwise-supported solution:

Lemma 39. *If an equivariant system of inequalities $(\mathbf{A}, \mathbf{t})$ has a finitary T-supported solution $\mathbf{x}$ then it also has a finitary setwise-T-supported one $\mathbf{y}$ such that $\mathbf{x}^{\Sigma} = \mathbf{y}^{\Sigma}$.*

Proof. Let $\mathbf{x} : C \to_{\text{fs}} \mathbb{R}$ be a solution of the system, namely $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{t}$. Let $T = \text{supp}(\mathbf{x})$ and $n = |T|$. As $(\mathbf{A}, \mathbf{t})$ is equivariant, atom automorphisms preserve being a solution, namely for every $\rho \in \textsc{Aut}$, the vector $\rho(\mathbf{x})$ is also a solution: $\mathbf{A} \cdot \rho(\mathbf{x}) \;\geq\; \mathbf{t}$. Consider $\textsc{Aut}_{\mathbb{A}\setminus T}$, the subgroup of atom automorphisms that only permute T and preserve all other atoms. Knowing that the size of $\textsc{Aut}_{\mathbb{A}\setminus T}$ is $n!$, we have

$$\mathbf{A} \cdot \Big(\sum_{\rho \in \textsc{Aut}_{\mathbb{A}\setminus T}} \rho(\mathbf{x}) \Big) \;\geq\; n! \cdot \mathbf{t},$$

and hence the vector $\mathbf{y}$ defined by averaging (cf. Example 36)

$$\mathbf{y} = \frac{1}{n!} \cdot \sum_{\rho \in \mathrm{Aut}_{\mathbb{A} \setminus T}} \rho(\mathbf{x}) \tag{22}$$

is also a solution of the system, namely $\mathbf{A} \cdot \mathbf{y} \geq \mathbf{t}$. We notice that for finitary $\mathbf{x}$, the vector $\mathbf{y}$ is finitary as well. By the very definition, the averaging (22) preserves the orbit-sum: $\mathbf{x}^{\Sigma} = \mathbf{y}^{\Sigma}$. Furthermore, we claim that the vector $\mathbf{y}$ is setwise-T-supported. To prove this, we fix an arbitrary $\pi \in \mathrm{Aut}_{\{T\}}$, aiming at showing that $\pi(\mathbf{y}) = \mathbf{y}$. It factors through $\pi = \sigma \circ \rho$ for some $\rho \in \mathrm{Aut}_{\mathbb{A} \setminus T}$ and $\sigma \in \mathrm{Aut}_T$. Indeed, ρ acts as π on T but is identity elsewhere, while σ acts as π outside of T but is identity on T. A crucial but simple observation is that, by the very construction of $\mathbf{y}$, we have

$$\rho(\mathbf{y}) = \mathbf{y}. \tag{23}$$

Indeed, as $\mathbf{y}$ is defined by averaging over all $\rho' \in \mathrm{Aut}_{\mathbb{A} \setminus T}$,

$$\rho\Big(\sum_{\rho' \in \mathrm{Aut}_{\mathbb{A} \setminus T}} \rho'(\mathbf{x}) \Big) = \sum_{\rho' \in \mathrm{Aut}_{\mathbb{A} \setminus T}} \rho \circ \rho'(\mathbf{x}) = \sum_{\rho' \in \mathrm{Aut}_{\mathbb{A} \setminus T}} \rho'(\mathbf{x})$$

which implies $\rho(\mathbf{y}) = \mathbf{y}$. Moreover, as action of atom automorphisms commutes with support, we have

$$\mathrm{supp}(\rho'(\mathbf{x})) = \rho'(\mathrm{supp}(\mathbf{x}))$$

for every $\rho' \in \mathrm{Aut}$, and therefore

$$\mathrm{supp}(\rho'(\mathbf{x})) = \mathrm{supp}(\mathbf{x})$$

for every $\rho' \in \mathrm{Aut}_{\mathbb{A} \setminus T}$. Therefore T supports the right-hand side of (22), which means that $\mathrm{supp}(\mathbf{y}) \subseteq T$ and implies

$$\sigma(\mathbf{y}) = \mathbf{y}. \tag{24}$$

By (23) and (24) we obtain $\pi(\mathbf{y}) = \mathbf{y}$, as required.

Finally, the equality $\mathbf{x}^{\Sigma} = \mathbf{y}^{\Sigma}$ follows directly by (22). □

Example 40. Recall the system of inequalities from Examples 3 and 5. Its finitary solutions correspond to finite directed graphs, whose vertices and edges are labeled by real numbers satisfying constraints (5) and (6). According to Lemma 39, if such a directed graph existed, there would also exist a directed clique, where labels of all vertices are pairwise equal, and labels of all edges are pairwise equal as well, which satisfying constraints (5) and (6). In particular, all edges incoming to a vertex would carry the same value as all outgoing edges. This requirement is clearly contradictory with constraints (5) and (6), and hence the system has no finitary solutions. ◁

In the next section we rely on the fact that existence of a setwise-S-supported solution implies existence of such a solution for any support larger than S. The fact follows immediately from Lemma 39, since every setwise-S-supported vector is trivially T-supported, for every superset T of S:

Corollary 41. *If an equivariant system of inequalities* $(\mathbf{A}, \mathbf{t})$ *has a finitary setwise-S-supported solution* $\mathbf{x}$, *then for every superset T of S of size $|T| = |S| + 1$, the system* $(\mathbf{A}, \mathbf{t})$ *has a finitary setwise-T-supported solution* $\mathbf{y}$ *such that* $\mathbf{x}^{\Sigma} = \mathbf{y}^{\Sigma}$.

7 DECIDABILITY OF REAL SOLVABILITY

In this section we prove Theorem 18 by a reduction of Fin-Ineq-Solv($\mathbb{R}$) to Poly-Ineq-Solv (cf. Example 4 in Section 1).

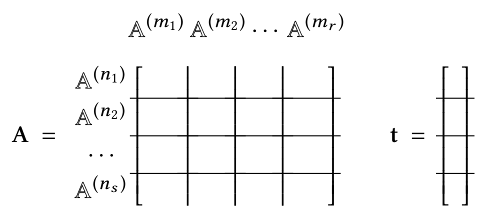
7.1 Preliminaries

Consider an orbit-finite system of inequalities given by a matrix $\mathbf{A} : B \times C \to_{\text{fs}} \mathbb{Z}$ and a target vector $\mathbf{t} : B \to_{\text{fs}} \mathbb{Z}$.

Lemma 42. *W.l.o.g. we can assume that B and C are disjoint unions of equivariant orbits $\mathbb{A}^{(k)}$, $k \in \mathbb{N}$:*

$$B = \mathbb{A}^{(n_1)} \uplus \ldots \uplus \mathbb{A}^{(n_s)} \qquad\qquad C = \mathbb{A}^{(m_1)} \uplus \ldots \uplus \mathbb{A}^{(m_r)} \tag{25}$$

(see the figure below), and that $\mathbf{A}$ and $\mathbf{t}$ are equivariant. The size blow-up is exponential in atom dimension, but polynomial when atom dimension is fixed.



(The proof is in Section A.4.) Note that this includes the case of finite systems, namely $n_1 = \ldots = n_s = m_1 = \ldots = m_r = 0$.

7.2 Idea of the reduction

Suppose only finitary T-*supported* solutions are sought, for a fixed $T \subseteq_{\text{fin}} \mathbb{A}$. Fin-Ineq-Solv($\mathbb{R}$) reduces then to a finite system of inequalities $(\mathbf{A}', \mathbf{t}')$ obtained from $(\mathbf{A}, \mathbf{t})$ as follows:

(1) Keep only columns indexed by T-tuples (= elements of *finite* T-orbits) $c \in C$, discarding all other columns.
(2) Pick arbitrary representatives of *all* T-orbits included in B, and keep only rows of $\mathbf{A}$ and entries of $\mathbf{t}$ indexed by the representatives, discarding all others.

The system $(\mathbf{A}', \mathbf{t}')$ is solvable if and only if the original one $(\mathbf{A}, \mathbf{t})$ has a finitary T-supported solution. Indeed, discarding unknowns as in (1) is justified as a finitary T-supported solution of $(\mathbf{A}, \mathbf{t})$ assigns 0 to each non-T-tuple. Discarding inequalities as in (2) is also justified. Indeed, each inequality in the original system is obtained by applying some atom T-automorphism to an inequality in $(\mathbf{A}', \mathbf{t}')$, while atom T-automorphisms preserve T-supported solutions of $(\mathbf{A}', \mathbf{t}')$, which implies that every T-supported solutions of $(\mathbf{A}', \mathbf{t}')$ is also a solution of all inequalities in the original system.

The above reduction yields no algorithm yet, as we do not know a priori any bound on size of T, and the size of $(\mathbf{A}', \mathbf{t}')$ depends on the number of T-orbits and hence grows unboundedly when T grows. We overcome this difficulty by using setwise-T-orbits instead of T-orbits, and relying on Lemmas 39 and 37. The latter one guarantees that the number of setwise-T-orbits is constant - independent of T. Once we additionally merge (sum up) all columns indexed by elements of the same setwise-T-orbit, we get $\mathbf{A}'$ of size independent of T.

This still does not yield an algorithm, as entries of $\mathbf{A}'$ change when T grows. We however crucially discover that the growth of the entries of $\mathbf{A}'$ is *polynomial* in $n = |T|$, for sufficiently large n. Therefore, $\mathbf{A}'$ is a matrix of polynomials in one unknown n, and solvability of $(\mathbf{A}, \mathbf{t})$ is equivalent to solvability of $(\mathbf{A}', \mathbf{t}')$ for some value $n \in \mathbb{N}$. As argued in Section 5, the latter solvability is decidable.

7.3 Reduction of Fin-Ineq-Solv($\mathbb{R}$) to Almost-all-Poly-Ineq-Solv

Let us fix an equivariant system $(\mathbf{A}, \mathbf{t})$. We construct a finite system P_2 of polynomially-parametrised inequalities such that $(\mathbf{A}, \mathbf{t})$ has a finitary solution if and only if $P_2(n)$ has a solution for almost all $n \in \mathbb{N}$.

Let us denote by $d = \max\{n_1, \ldots, n_s, m_1, \ldots, m_r\}$ the maximal atom dimension of orbits included in B and C.

Let $T \subseteq_{\text{fin}} \mathbb{A}$ be an arbitrary finite subset of atoms. Both B and C split into setwise-T-orbits, refining (25):

$$B = B_1 \uplus \ldots \uplus B_N \qquad C = C_1 \uplus \ldots \uplus C_{M'}. \tag{26}$$

Let $C_1, \ldots, C_M$ be the *finite* setwise-T-orbits among $C_1, \ldots, C_{M'}$ (clearly, M and N may depend on T). Importantly, by Lemma 37, N and M do not depend on T as long as $|T| \geq d$. In fact $M = r$, the number of orbits included in C, as by Lemma 37 we deduce:

Lemma 43. *Assuming $|T| \geq \ell$, the equivariant orbit $\mathbb{A}^{(\ell)}$ includes exactly one finite setwise-T-orbit, namely $T^{(\ell)}$.*

Our reduction proceeds in two steps: first, we derive a finite polynomially-parametrised system P_1, and then we transform it further to a monotonic system P_2. Monotonicity of P_2 guarantees correctness of reduction.

Step 1 (finite polynomially-parametrised system). Our construction is parametric in T. Let $b_1, \ldots, b_N$ be arbitrarily chosen representatives of setwise-T-orbits included in B. Given $\mathbf{A}$ and $\mathbf{t}$, we define an $N \times M$ matrix $\mathbf{A}_1(T)$ and a vector $\mathbf{t}_1(T) \in \mathbb{Z}^N$ as follows:

1. Pick columns of $\mathbf{A}(T)$ indexed by elements of all finite setwise-T-orbits included in C, and discard other columns; this yields a matrix $\mathbf{A}'(T)$ with finitely many columns (number thereof depending on T).
2. Merge (sum up) columns of $\mathbf{A}'(T)$ indexed by elements of the same setwise-T-orbit; this yields a matrix $\mathbf{A}''(T)$ with M columns (M independent of T).
3. Pick N rows of $\mathbf{A}''$, indexed by $b_1, \ldots, b_N$, and discard other rows; this yields an $N \times M$ matrix $\mathbf{A}_1(T)$.
4. Likewise pick the corresponding entries of $\mathbf{t}$ and discard others, thus yielding a finite vector $\mathbf{t}_1(T) \in \mathbb{Z}^N$.

For $b \in B$ and $C_j \subseteq C$, $j \in \{1, \ldots, M\}$, we write $\mathbf{A}^\Sigma(b, C_j)$ for the finite sum ranging over elements of C_j:

$$\mathbf{A}^\Sigma(b, C_j) = \sum_{c \in C_j} \mathbf{A}(b, c),$$

which allows us to formally define the $B \times M$ matrix $\mathbf{A}''(T)$, the $N \times M$ matrix $\mathbf{A}_1(T)$ and the vector $\mathbf{t}_1(T) \in \mathbb{Z}^N$:

$$\mathbf{A}''(T)(b, j) = \mathbf{A}^\Sigma(b, C_j) \qquad \mathbf{A}_1(T)(i, j) = \mathbf{A}''(T)(b_i, j) = \mathbf{A}^\Sigma(b_i, C_j) \qquad \mathbf{t}_1(T)(i) = \mathbf{t}(b_i). \tag{27}$$

Example 44. We explain how the system (8) in Example 4 in Section 1 is obtained from the system (1) in Example 1. Fix a non-empty $T \subseteq_{\text{fin}} \mathbb{A}$. The set $\mathbb{A}$ includes just one finite setwise-T-orbit, namely T. Therefore the matrix $\mathbf{A}'(T)$ has $|T|$ columns, $\mathbf{A}''(T)$ has just one column, and the system $(\mathbf{A}_1(T), \mathbf{t}_1(T))$ has just one unknown. Furthermore, the set $\mathbb{A}$ includes two setwise-T-orbits, the finite one T plus the infinite one $\mathbb{A} \setminus T$, and therefore the system $(\mathbf{A}_1(T), \mathbf{t}_1(T))$ has two inequalities. Pick arbitrary representatives of the setwise-T-orbits, $b_1 \in T$ and $b_2 \in (\mathbb{A} \setminus T)$. We have

$$\mathbf{A}_1(T)(1,1) \;=\; \sum_{c \in T} \mathbf{A}(b_1, c) = |T| - 1 \qquad\qquad \mathbf{A}_1(T)(2,1) \;=\; \sum_{c \in T} \mathbf{A}(b_2, c) = |T|.$$

Replacing $|T|$ with n yields the system $(\mathbf{A}_1(T), \mathbf{t}_1(T))$:

$$\begin{bmatrix} n-1 \\ n \end{bmatrix} \cdot x \;\geq\; \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{28}$$

which happens to be monotonic. In general, the system obtained so far needs not be monotonic, but we will ensure monotonicity in the subsequent step. ◁

The choice of representatives b_i is irrelevant, and hence $\mathbf{A}_1(T)$ and $\mathbf{t}_1(T)$ are well defined, since rows of $\mathbf{A}''$ indexed by any two elements of B belonging the same setwise-T-orbit are equal, and likewise the corresponding entries of $\mathbf{t}$:

Lemma 45. *If $b, b' \in B$ are in the same setwise-T-orbit, then $\mathbf{t}(b) = \mathbf{t}(b')$ and $\mathbf{A}^{\Sigma}(b, C_j) = \mathbf{A}^{\Sigma}(b', C_j)$ for every $j \in \{1, \ldots, M\}$.*

Proof. Let $\pi \in \mathrm{Aut}_{(T)}$ be such that $\pi(b) = b'$. As $\mathbf{t}$ is equivariant, it is necessarily constant on the whole equivariant orbit to which b and b' belong (cf. Lemma 13), and hence $\mathbf{t}(b') = \mathbf{t}(b)$.

For the second point fix $j \in \{1, \ldots, M\}$. As $\mathbf{A}$ is equivariant, it is constant over the orbit included in $B \times C$ to which (b, c) belongs, for every $c \in C$, and hence $\mathbf{A}(b, c) = \mathbf{A}(\pi(b), \pi(c))$. This implies

$$\sum_{c \in C_j} \mathbf{A}(b, c) = \sum_{c \in C_j} \mathbf{A}(\pi(b), \pi(c)) = \sum_{c \in C_j} \mathbf{A}(b', \pi(c)).$$

Since π is a setwise-T-automorphism, when restricted to the setwise-T-orbit C_j it is a bijection $C_j \to C_j$, and hence the two sums below differ only by the order of summation and are thus equal:

$$\sum_{c \in C_j} \mathbf{A}(b', \pi(c)) = \sum_{c \in C_j} \mathbf{A}(b', c).$$

The two above equalities imply the claim, namely $\sum_{c \in C_j} \mathbf{A}(b, c) = \sum_{c \in C_j} \mathbf{A}(b', c)$. □

Notation 46. Let $T \subseteq_{\text{fin}} \mathbb{A}$. Due to Lemma 43, the set of finite setwise-T-orbits $\{C_1, \ldots, C_M\}$ included in C is in bijection with the set $\mathrm{orbits}\,(C) = \{U_1, \ldots, U_M\}$ of equivariant orbits included in C. W.l.o.g. assume $C_j \subseteq U_j$ for $j = 1 \ldots M$. Take any finitary setwise-T-supported vector $\mathbf{x} : C \to_{\text{fin}} \mathbb{R}$. It is non-zero only inside finite setwise-T-orbits C_j, which implies

$$\mathbf{x}^{\Sigma}(C_j) \;=\; \mathbf{x}^{\Sigma}(U_j)$$

for $j = 1 \ldots M$ (cf. Notation 38). Furthermore, $\mathbf{x}$ is constant inside each C_j, which allows us to write $\dot{\mathbf{x}}(C_j)$ (cf. Notation 14). For notational convenience we slightly relax Notations 14 and 38 from now on, and treat the orbit-value and orbit-sum vectors as M-tuples, $\dot{\mathbf{x}}, \ \mathbf{x}^{\Sigma} \ \in \ \mathbb{R}^M$, with obvious meaning $\dot{\mathbf{x}}(j) = \dot{\mathbf{x}}(C_j)$ and $\mathbf{x}^{\Sigma}(j) = \mathbf{x}^{\Sigma}(C_j)$. We note the (obvious) relation between $\dot{\mathbf{x}}$ and $\mathbf{x}^{\Sigma}$:

$$\mathbf{x}^{\Sigma}(j) \;=\; |C_j| \cdot \dot{\mathbf{x}}(j). \tag{29}$$

The following lemma, being a cornerstone of correctness of the whole reduction, is now not difficult to prove:

Lemma 47. *Let $|T| \geq d$ and $\mathbf{x} : C \to_{fin} \mathbb{R}$ a finitary setwise-T-supported vector. The following conditions are equivalent:*

- *$\mathbf{x}$ is solution of $(\mathbf{A}, \mathbf{t})$;*
- *$\dot{\mathbf{x}}$ is a solution of $P_1(T) = (\mathbf{A}_1(T), \mathbf{t}_1(T))$.*

Proof. Take any setwise-T-supported vector $\mathbf{x} : C \to_{\text{fin}} \mathbb{R}$, and let $\mathbf{x}'$ be the restriction of $\mathbf{x}$ to $C' = C_1 \uplus \ldots \uplus C_M$. We argue that the following four conditions are equivalent, which implies the claim:

(1) $\mathbf{x}$ is solution of $(\mathbf{A}, \mathbf{t})$;
(2) $\mathbf{x}'$ is solution of $(\mathbf{A}'(T), \mathbf{t})$;
(3) $\dot{\mathbf{x}}$ is solution of $(\mathbf{A}''(T), \mathbf{t})$;

(4) $\dot{\mathbf{x}}$ is a solution of $(\mathbf{A}_1(T), \mathbf{t}_1(T))$.

First, as $\mathbf{x}$ is finitary, we have $\mathbf{x}(c) = 0$ for all $c \notin C'$, and hence $\mathbf{A}'(T) \cdot \mathbf{x}' = \mathbf{A} \cdot \mathbf{x}$. This implies equivalence of (1) and (2). Second, as $\mathbf{A}''$ is obtained from $\mathbf{A}'$ by summing columns over a setwise-T-orbit where the vector $\mathbf{x}$, being setwise-T-supported, is constant, we have $\mathbf{A}''(T) \cdot \dot{\mathbf{x}} = \mathbf{A}'(T) \cdot \mathbf{x}'$. This implies equivalence of (2) and (3). Finally, (3) implies (4) as $(\mathbf{A}_1(T), \mathbf{t}_1(T))$ is obtained from $(\mathbf{A}''(T), \mathbf{t})$ by removing inequalities. For the reverse implication, we recall that Lemma 45 shows that $\mathbf{A}''(b, j) = \mathbf{A}''(b_i, j)$ and $\mathbf{t}(b) = \mathbf{t}(b_i)$ for every $i \in \{1, \ldots, N\}$ and $b \in B_i$, and therefore $\mathbf{A}''(T)$ contains the same inequalities as $\mathbf{A}_1(T)$. In consequence, (4) implies (3). □

The function $T \mapsto P_1(T)$ is equivariant, i.e., invariant under action of atom automorphisms. In consequence, the entries of $\mathbf{A}_1(T)$ and $\mathbf{t}_1(T)$ do not depend on the set T itself, but only on its size $|T|$. Indeed, if $|T| = |T'|$ then $\pi(T) = T'$ for some atom automorphism π, and hence $\pi(P_1(T)) = P_1(T')$. Since the system $P_1(T)$ is atom-less we have also $\pi(P_1(T)) = P_1(T)$, which implies $P_1(T) = P_1(T')$. We may thus meaningfully write $P_1(|T|) = (\mathbf{A}(|T|), \mathbf{t}(|T|))$, i.e., $P_1(n) = (\mathbf{A}_1(n), \mathbf{t}_1(n))$ for $n \in \mathbb{N}$ (cf. Example 44).

We argue that the dependence on $|T|$ is polynomial, as long as $|T| \geq 2d$:

Lemma 48. *There are univariate polynomials $p_{ij}(n) \in \mathbb{Z}[n]$ such that $\mathbf{A}_1(n)(i, j) = p_{ij}(n)$ for $n \geq 2d$.*

Proof. Let $n = |T|$. Fix a setwise-T-orbits $B_i \subseteq B$ and a finite setwise-T-orbit $C_j \subseteq C$. Each of them is included in a unique equivariant orbit, say:

$$B_i \subseteq B' = \mathbb{A}^{(p)} \qquad C_j \subseteq C' = \mathbb{A}^{(\ell)}$$

(cf. the partitions (25)). Recall Lemma 37: B_i is determined by the subset $I \subseteq \{1, \ldots, p\}$ of positions where atoms of T appear in tuples belonging to B_i. Let $m = |I|$. On the other hand $C_j = T^{(\ell)}$ (cf. Lemma 43). Note that $m = |T \cap \mathrm{supp}(b_i)|$.

We are going to demonstrate that the value $\mathbf{A}^{\Sigma}(b_i, C_j)$ is polynomially depending on $n = |T|$. We will use the polynomials $n^{(w)}$ of degree w, for $w \leq d$, defined by

$$n^{(w)} = n \cdot (n-1) \cdot \ldots \cdot (n - w + 1). \tag{30}$$

In the special case of $w = 0$, we put $n^{(w)} = 1$. The value $n^{(w)}$ can be interpreted as follows:

Claim 49. *For $n \geq w$, $n^{(w)}$ is equal to the number of arrangements of w items chosen from n objects into a sequence.*

Denote by $\mathcal{D}$ the set of equivariant orbits $U \subseteq B' \times C'$. For $U \in \mathcal{D}$, we put $U(b_i, C_j) := \left\{ c \in C_j \;\middle|\; (b_i, c) \in U \right\}$. As $\mathbf{A}$ is equivariant, the value $\mathbf{A}(b_i, c)$ depends only on the orbit to which (b_i, c) belongs. We write $\mathbf{A}(U)$, for $U \in \mathcal{D}$, and get:

Claim 50. $\mathbf{A}^{\Sigma}(b_i, C_j) = \sum_{U \in \mathcal{D}} \mathbf{A}(U) \cdot |U(b_i, C_j)|$.

By Lemma 10 in Section 2, orbits $U \subseteq B' \times C'$ are in one-to-one correspondence with partial injections $\iota : \{1, \ldots, p\} \to \{1, \ldots, \ell\}$. We write U_ι for the orbit corresponding to ι. Let $\mathrm{dom}(\iota) = \{ x \mid \iota(x) \text{ is defined} \}$ denote the domain of ι.

Claim 51. *$U_\iota(b_i, C_j) \neq \emptyset$ if and only if $\mathrm{dom}(\iota) \subseteq I$.*

Indeed, recall again Lemma 10 which yields $U_\iota(b_i, C_j) = \left\{ c \in C_j \;\middle|\; \forall x, y : b_i(x) = c(y) \iff \iota(x) = y \right\}$. If $\mathrm{dom}(\iota) \subseteq I$, the set $U_\iota(b_i, C_j)$ contains tuples $c \in C_j$ with fixed values on positions $J = \{ \iota(x) \mid x \in \mathrm{dom}(\iota) \}$, namely

$$b_i(x) = c(\iota(x)), \tag{31}$$

and arbitrary other atoms from T elsewhere, and therefore is nonempty. If there is $x \in \mathrm{dom}(\iota) \setminus I$ then $b_i(x) \notin T$ and therefore no $c \in C_j$ satisfies (31). Claim 51 is thus proved.

Claim 52. *Let $k = |\mathrm{dom}(\iota)|$ be the number of pairs related by ι. If $U_\iota(b_i, C_j) \neq \emptyset$ then $|U_\iota(b_i, C_j)| = (n-m)^{(\ell-k)}$.*

According to (31), tuples $c \in U_\iota(b_i)$ have fixed values on k positions in J. The remaining $\ell - k$ positions in tuples $c \in U_\iota(b_i, C_j)$ are filled arbitrarily using $n - m$ atoms from $T \setminus \mathrm{supp}(b_i)$. Due to the assumption that $n \geq 2d$, we have $n - m \geq d \geq \ell - k$, and therefore using Claim 49 (for $w = \ell - k$) we deduce $|U_\iota(b_i, C_j)| = (n-m)^{(\ell-k)}$, thus proving Claim 52.

Once $b_i \in B_i$ and $U \in \mathcal{D}$ are fixed, the values k, ℓ and m are fixed too, and the formula of Claim 52 is an univariate polynomial of degree $\ell - k$. The formula of Claim 50 yields the required polynomial[8] $\mathbf{A}(n)(i,j) = p_{ij}(n)$ and hence the proof of Lemma 48 is completed. □

Relying on Lemma 48 we get a polynomially-parametrised system $P_1(n) = (\mathbf{A}_1(n), \mathbf{t}_1(n))$.

Step 2 (monotonicity). The system P_1 constructed so far, does *not* have to be monotonic in general. As an immediate corollary of Lemma 47 and Corollary 41, we only know that If $P_1(n)$ has a solution for $n \geq d$, then $P_1(n+1)$ has a (potentially different) solution. We slightly modify the system P_1 in order to achieve monotonicity.

Before defining formally the new system $P_2(n) = (\mathbf{A}_2(n), \mathbf{t}_2(n))$, we point to our objective: we aim at replacing the orbit-value vector $\dot{\mathbf{x}}$ in Lemma 47 by the orbit-sum vector $\mathbf{x}^\Sigma$, as in Lemma 55 below. In other terms, we want the solutions $\mathbf{y}_1$ and $\mathbf{y}_2$ of $P_1(n)$ and $P_2(n)$, respectively, differ on position j by the multiplicative factor of $|C_j|$, namely

$$\mathbf{y}_2(j) \;=\; |C_j| \cdot \mathbf{y}_1(j) \tag{32}$$

for $j = 1, \ldots, M$ (cf. (29)). The size $|C_j|$ of the setwise-T-orbit C_j, where $|T| = n$, is equal to

$$|C_j| \;=\; n^{(e_j)}, \tag{33}$$

where $C_j \subseteq \mathbb{A}^{(e_j)}$, i.e., e_j is the atom dimension of the equivariant orbit including C_j, assuming $|T| \geq e_j$. These considerations lead to the following formal definition of P_2:

$$\mathbf{A}_2(n)(i,j) \;=\; \mathbf{A}_1(n)(i,j) \cdot \frac{n^{(d)}}{n^{(e_j)}} \qquad\qquad \mathbf{t}_2(i) \;=\; \mathbf{t}_1(i) \cdot n^{(d)} \tag{34}$$

where $\mathbf{A}_1(n)(i,j) = p_{ij}(n)$. We rely on the following fact:

Claim 53. $n^{(w)} \cdot (n-w)^{(u)} \;=\; n^{(w+u)}$.

By the claim, all coefficients in (34) are polynomials, namely: $\mathbf{A}_2(n)(i,j) \;=\; p_{ij}(n) \cdot (n - e_j)^{(d-e_j)}$, since $e_j \leq d$. It remains to conclude that the systems $P_1(n)$ and $P_2(n)$ have the same solutions modulo (32):

Lemma 54. *Let $n \geq d$ and let $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^M$ satisfy $\mathbf{y}_2(j) \;=\; n^{(e_j)} \cdot \mathbf{y}_1(j)$ for $j = 1, \ldots, M$. Then $\mathbf{y}_1$ is a solution of $P_1(n)$ if and only if $\mathbf{y}_2$ is a solution of $P_2(n)$.*

Combination of (29), (33), and Lemmas 47 and 54 yields:

Lemma 55. *Let $|T| = n \geq d$ and $\mathbf{x} : C \to_{fin} \mathbb{R}$ a finitary setwise-T-supported vector. The following conditions are equivalent:*

- *$\mathbf{x}$ is solution of $(\mathbf{A}, \mathbf{t})$;*
- *$\mathbf{x}^\Sigma$ is a solution of $P_2(n)$.*

[8] This confirms, in particular, that $\mathbf{A}(T)(i,j)$ is independent from the actual set T, and only depend on its size $n = |T|$.

Example 56. The system (8) in Example 4 is obtained from (28) in Example 44 by applying the definition (34). Indeed, $M = d = e_1 = 1$ and hence $\mathbf{A}$ stays unchanged, while the right-hand side vector $\mathbf{t}$ gets multiplied by $n^{(1)} = n$. ◁

Lemma 57 (Monotonicity). *Let $n \geq d$. Every solution of $P_2(n)$ is also a solution of $P_2(n+1)$.*

Proof. Suppose $\mathbf{y}$ is a solution of $P_2(n)$, for $n \geq d$. Let $T \subseteq_{\text{fin}} \mathbb{A}$ be any subset of atoms of size $|T| = n$. Let $\mathbf{x}$ be the finitary setwise-T-supported vector uniquely determined by

$$\mathbf{x}^{\Sigma} \;=\; \mathbf{y}. \tag{35}$$

By Lemma 55, $\mathbf{x}$ is a solution of $(\mathbf{A}, \mathbf{t})$. We apply Corollary 41 to obtain another finitary solution $\mathbf{x}'$ of $(\mathbf{A}, \mathbf{t})$, setwise-T'-supported by some T' of size $|T|' = n + 1$, and having the same orbit-summation mapping:

$$\mathbf{x}^{\Sigma} \;=\; (\mathbf{x}')^{\Sigma}. \tag{36}$$

Equalities (35) and (36) imply $(\mathbf{x}')^{\Sigma} = \mathbf{y}$. By Lemma 55 again, $(\mathbf{x}')^{\Sigma} = \mathbf{y}$ is a solution of $P_2(n+1)$, as required. □

Combining Lemmas 39, 48, 55 and 57 we derive correctness of reduction (the constraint $n \geq 2d$ is inherited from the assumption in Lemma 48):

Corollary 58. *The following conditions are equivalent:*

- $(\mathbf{A}, \mathbf{t})$ *has a finitary solution,*
- $P_2(n)$ *has a solution for some integer* $n \geq 2d$,
- $P_2(n)$ *has a solution for almost all* $n \in \mathbb{N}$.

Reduction of Fin-Ineq-Solv($\mathbb{R}$) to Almost-all-Poly-Ineq-Solv is thus completed.

Complexity. It remains to argue that P_2 is computable from $(\mathbf{A}, \mathbf{t})$, and estimate the computational complexity. Computability of P_2 follows immediately from computability of P_1, which we focus now on:

Lemma 59. *The system P_1 is computable from* $(\mathbf{A}, \mathbf{t})$.

Proof. Indeed, it is enough to range over representations of setwise-T-orbits B_i and C_j of B and C, respectively (such representations are given by Lemma 37), and for each pair of such orbits proceed with computations outlined in the proof of Lemma 48, applied to an arbitrarily chosen representative $b_i \in B_i$. □

By Corollary 58 and Lemma 59, Fin-Ineq-Solv($\mathbb{R}$) reduces to Almost-all-Poly-Ineq-Solv.

Concerning computational complexity, the number of setwise-T-orbits included in an equivariant orbit $\mathbb{A}^{(\ell)}$ is exponential in ℓ (cf. Lemma 37). That is why the size of P_2 may be exponential in atom dimension d of $(\mathbf{A}, \mathbf{t})$. On the other hand, the size of P_2 is only polynomial (actually, linear) in the number of orbits included in $B \times C$. In consequence, for fixed atom dimension we get a polynomial-time reduction and hence, relying on Theorem 25, the decision procedure for Fin-Ineq-Solv($\mathbb{R}$) in PTime. Without fixing atom dimension, we get an exponential-time reduction and hence the decision procedure is in ExpTime.

The same complexity bounds apply to the algorithm for the optimisation problem presented in Section 8.

8 OPTIMISATION PROBLEMS

In this section we prove Theorem 23: we introduce a maximisation variant of Poly-Ineq-Solv and routinely adapt the decision procedure of Section 5, as well as the reduction of Section 7.3, to the maximisation setting.

8.1 Polynomially-parametrised maximisation problem

We consider a maximisation problem, whose instance (P, S) consists of a finite system of polynomially-parametrised inequalities P as in (12), and an *ordinary* (non-parametrised) objective function S given by a linear map

$$S(x_1, \ldots, x_k) = a_1 \cdot x_1 + \ldots + a_k \cdot x_k.$$

As in Section 5.1, by an *almost-all-solution* of a system P we mean in this section a solution of $P(n)$ for almost all $n \in \mathbb{N}$. We define the *supremum* of a monotonic instance (P, S) as

$$\sup(P, S) \;:=\; \sup\{\, S(\mathbf{x}) \mid \mathbf{x} \text{ is an almost-all-solution of } P\,\},$$

under a proviso that $\sup(P, S) = -\infty$ if P has no almost-all-solutions. Referring to the standard terminology, we can say that the system is *infeasible* if $sup(P, S) = -\infty$, and it is *unbounded* if $\sup(P, S) = \infty$. Interestingly, the supremum can not be irrational (see Corollary 63 below).

In this section we study the problem of computing the supremum of monotonic instances:

Almost-all-Poly-Ineq-Max:
Input: *An instance* (P, S).
Output: *The supremum of* (P, S).

The problem generalises ordinary (non-parametrised) linear programming, and can be solved similarly to Almost-all-Poly-Ineq-Solv (of which it is a strengthening):

Theorem 60. Almost-all-Poly-Ineq-Max *is in PTime.*

Proof. Let (P_0, S) be an instance. The algorithm is essentially the same as Algorithm 1 for Almost-all-Poly-Ineq-Solv in the proof of Theorem 25, and proceeds by iterating the transformation step until either unsolvability or solvability is reported. Recall that solution set is preserved by the transformation step (Claim 32). If unsolvability is reported, the algorithm returns $-\infty$. If solvability is reported — let $P \cup \Gamma$ be the system examined in the last iteration — the decision procedure computes and returns $\sup(\textsc{hd}(P) \cup \Gamma, S)$, the supremum of S constrained by the ordinary system of inequalities $\textsc{hd}(P) \cup \Gamma$, by invoking any PTime procedure for ordinary linear programming.

Correctness follows by the two claims formulated below. First, since solution set is preserved by the transformation step, we have:

Claim 61. $\sup(P_0, S) = \sup(P \cup \Gamma, S)$.

Second, the supremum does not change if the polynomially-parametrised constraints P are replaced by the overapproximation $\textsc{hd}(P)$:

Claim 62. $\sup(P \cup \Gamma, S) = \sup(\textsc{hd}(P) \cup \Gamma, S)$.

For the claim it is enough to prove the inequality $\sup(\textsc{hd}_{>}(P) \cup \Gamma, S) \geq \sup(\textsc{hd}(P) \cup \Gamma, S)$ as, according to Claims 29 and 30, we have $\sup(\textsc{hd}_{>}(P) \cup \Gamma, S) \leq \sup(P \cup \Gamma, S) \leq \sup(\textsc{hd}(P) \cup \Gamma, S)$. Take any solution $\mathbf{y}$ of $\textsc{hd}(P) \cup \Gamma$, and any solution $\mathbf{x}$ of $\textsc{hd}_{>}(P) \cup \Gamma$ (we rely here on solvability of the latter system). For every $k \in \mathbb{N}$, the vector $\mathbf{x}_k = \frac{\mathbf{x}+k\mathbf{y}}{k+1}$ is a solution of $\textsc{hd}_{>}(P) \cup \Gamma$, and $S(\mathbf{x}_k)$ tends to $S(\mathbf{y})$ when k tends to ∞. Hence $\sup(\textsc{hd}_{>}(P) \cup \Gamma, S) \geq \sup(\textsc{hd}(P) \cup \Gamma, S)$, as required. ◻

By Claims 61 and 62 in the proof of Theorem 60, the supremum of a monotonic instance, if not $-\infty$ nor ∞, is equal to the supremum of an ordinary linear program and hence is rational:

Corollary 63. *The supremum of a monotonic instance (P, S) belongs to $\mathbb{Q} \cup \{-\infty, +\infty\}$.*

Remark 64. As illustrated in Example 4 in Section 1, the objective function S may not achieve its supremum over the almost-all-solutions of P. Once the supremum $s \in \mathbb{Q}$ is computed, one can easily check if S achieves its supremum, by adding to the system an equation $S(x_1, \ldots, x_k) = s$ and checking if the system is still solvable. ◁

8.2 Reduction of Fin-Ineq-Max($\mathbb{R}$) to Almost-all-Poly-Ineq-Max

We only sketch the reduction as it amounts to a slight adaptation of the reduction of Section 7.3. The input of Fin-Ineq-Max($\mathbb{R}$) consists of a system $(\mathbf{A}, \mathbf{t})$ and an integer vector $\mathbf{s} : C \to_{\mathrm{fs}} \mathbb{Z}$ representing the objective function

$$S(\mathbf{x}) = \mathbf{s} \cdot \mathbf{x},$$

and we ask for the supremum of values $S(\mathbf{x})$, for $\mathbf{x}$ ranging over finitary solutions of $(\mathbf{A}, \mathbf{t})$. This value we denote as $\sup(\mathbf{A}, \mathbf{t}, \mathbf{s})$. In addition to Lemma 42 we show (the proof is in Section A.4):

Lemma 65. *W.l.o.g. we may assume that* $\mathbf{s}$ *is equivariant.*

We proceed by adapting the reduction of Section 7.3: given an instance $(\mathbf{A}, \mathbf{t}, \mathbf{s})$ of Fin-Ineq-Max($\mathbb{R}$) we compute a monotonic instance (P_2, S') of Poly-Ineq-Max, where the finite system $P_2(n) = (\mathbf{A}_2(n), \mathbf{t}_2(n))$ of polynomially-parametrised inequalities is exactly as in Section 7.3, and the objective function is

$$S'(x_1, \ldots, x_k) = a_1 \cdot x_1 + \ldots + a_M \cdot x_M, \tag{37}$$

where $a_j = \dot{\mathbf{s}}(C_j)$ for $j = 1 \ldots M$ (recall Notations 14 and 46). More concisely, the vector $\mathbf{a} = a_1 \ldots a_M$ is defined as $\mathbf{a} = \dot{\mathbf{s}}$. We apply Lemmas 39 and 55 to obtain:

Lemma 66. $\sup(\mathbf{A}, \mathbf{t}, \mathbf{s}) = \sup(P_2, S')$.

Proof. Let $\mathbf{x} : C \to_{\mathrm{fin}} \mathbb{R}$. By equivariance of S and the definition of S' we have the equality $S(\mathbf{x}) = S'(\mathbf{x}^{\Sigma})$, that is, the value of the objective function $S(\mathbf{x})$ depends only on the orbit-sum vector $\mathbf{x}^{\Sigma} : \text{orbits}\,(C) \to \mathbb{R}$. As Lemmas 39 and 55 preserve orbit-sum, we deduce that for every $T \subseteq_{\mathrm{fin}} \mathbb{A}$ of size $|T| = n \geq 2d$, the values of S on finitary T-supported solutions of $(\mathbf{A}, \mathbf{t})$ are the same as the values of S' on solutions of $P_2(n)$. By Lemma 57, the solutions of $P_2(n)$ for some $n \geq 2d$ are exactly the same as the almost-all-solutions of P_2. In consequence, the two suprema are equal. □

Example 67. To illustrate the reduction, consider the modification of the system in Example 3:

$$\sum_{\alpha \in \mathbb{A}} \alpha \;\geq\; 1 \qquad\qquad \sum_{\beta \in \mathbb{A}} \alpha\beta \;-\; \alpha \;-\; 2 \cdot \sum_{\beta \in \mathbb{A}} \beta\alpha \;\geq\; 0 \qquad (\alpha \in \mathbb{A}). \tag{38}$$

It enforces, for each vertex $\alpha \in \mathbb{A}$, the sum of values assigned to all outgoing edges to be larger than *double* the sum of values assigned to all ingoing edges, plus the value assigned to the vertex α. The indexing sets $B = \mathbb{A} \cup \{*\}$ and $C = \mathbb{A} \cup \mathbb{A}^{(2)}$ and the shape of the matrix (7) are the same. We identify the singletons $\{*\} = \mathbb{A}^{(0)}$. We consider maximisation of triple the sum of values assigned to edges: $S(\mathbf{x}) = \mathbf{s} \cdot \mathbf{x}$, where $\mathbf{s} = 3 \cdot \mathbf{1}_{\mathbb{A}^{(2)}}$, or $S(\mathbf{x}) = 3 \cdot \sum_{\alpha\beta \in \mathbb{A}^{(2)}} \mathbf{x}(\alpha\beta)$.

According to Lemma 43, the set C includes exactly 2 finite setwise-T-orbits, namely $T \subseteq \mathbb{A}$ and $T^{(2)} \subseteq \mathbb{A}^{(2)}$, and therefore the system computed by the reduction has 2 unknowns, x_1 and x_2. By Lemma 37, for any nonempty $T \subseteq_{\mathrm{fin}} \mathbb{A}$, the set B includes 3 setwise-T-orbits, namely T, $\mathbb{A} \setminus T$ and $\{*\}$, and therefore the system P_1 computed in the first step has 3 inequalities:

$$\begin{aligned} -x_1 \ - \ (n-1) \cdot x_2 \ &\geq \ 0 && (T) \\ 0 \ &\geq \ 0 && (\mathbb{A} \setminus T) \\ n \cdot x_1 \ &\geq \ 1 && (\{*\}) \end{aligned} \tag{39}$$

For instance, the coefficient $-(n-1)$ in the first inequality arises as:

$$\mathbf{A}(U_{\text{out}}) \cdot |U_{\text{out}}(\alpha, T^{(2)})| \ + \ \mathbf{A}(U_{\text{in}}) \cdot |U_{\text{in}}(\alpha, T^{(2)})| \ = \ 1 \cdot (n-1) - 2 \cdot (n-1) \ = \ -(n-1)$$

(cf. Claim 50), for some arbitrary $\alpha \in T$ and the following two orbits included in $\mathbb{A} \times \mathbb{A}^{(2)}$:

$$U_{\text{out}} = \{(\alpha, \alpha\beta) \,|\, \beta \neq \alpha\}\,, \quad U_{\text{in}} = \{(\alpha, \beta\alpha) \,|\, \beta \neq \alpha\}\,.$$

Likewise, the coefficient n in the last inequality arises as $\mathbf{A}(U) \cdot |O(*, T)| = 1 \cdot n = n$, for the orbit $U = \{*\} \times \mathbb{A}$. According to (34), the system P_2 is obtained from (39) by multiplying all occurrences of x_1 by $(n-1)^{(1)} = n-1$, and by multiplying all right-hand sides by $n^{(2)} = n(n-1)$ (the trivial second inequality is omitted):

$$\begin{aligned} -(n-1) \cdot x_1 \ - \ (n-1) \cdot x_2 \ &\geq \ 0 \\ n(n-1) \cdot x_1 \ &\geq \ n(n-1) \end{aligned} \tag{40}$$

Finally, the objective function produced by the reduction, as in (37), is $S'(x_1, x_2) \ = \ \mathbf{s}(\mathbb{A}^{(2)}) \cdot x_2 \ = \ 3 \cdot x_2$. It achieves -3 as its supremum, as the system (40) is equivalent to the ordinary system (its head):

$$x_1 \geq 1 \qquad\qquad x_2 \leq -x_1.$$

For every $n \geq 2$, the optimal solution $x_1 = 1$, $x_2 = -1$ corresponds, via the constructions of Section 7.3, to a clique of n vertices where each vertex is assigned $\frac{1}{n}$, and each edge is assigned $-\frac{1}{n(n-1)}$. ◁

9 UNDECIDABILITY OF INTEGER SOLVABILITY

We prove Theorem 19 by showing undecidability of Fin-Ineq-Solv($\mathbb{Z}$). We proceed by reduction from the reachability problem of counter machines.

We conveniently define a *d-counter machine* as a finite set of instructions I, where each instruction is a function

$$i : \{1 \ldots d\} \to \mathbb{Z} \cup \{\textsc{zero}\}$$

that specifies, for each counter $k \in \{1, \ldots, d\}$, either the additive update of k (if $i(k) \in \mathbb{Z}$) or the zero-test of k (if $i(k) = \textsc{zero}$). Configurations of M are nonnegative vectors $c \in \mathbb{N}^d$, and each instruction induces steps between configurations: $c \xrightarrow{i} c'$ if $c'(k) = c(k) + i(k)$ whenever $i(k) \in \mathbb{N}$, and $c'(k) = c(k) = 0$ whenever $i(k) = \textsc{zero}$. A run of M is defined as a finite sequence of steps

$$c_0 \xrightarrow{i_1} c_1 \xrightarrow{i_2} \ldots \xrightarrow{i_n} c_n. \tag{41}$$

The reachability problem asks, given a machine M and two its configurations, a source c_0 and a target c_f, if M admits a run from c_0 to c_f. The problem is undecidable, as counter machines can easily simulate classical Minsky machines.[9]

[9] A d-counter machine resembles a vector addition system with zero tests. A Minsky machine with n states and k counters can be simulated by an $(n+k)$-counter machine, by encoding control states into additional counters.

For $k \in \{1, \dots, d\}$ we denote by $\textsc{zero}(k) = \{\, i \in I \mid i(k) = \textsc{zero} \,\}$ the set of instructions that zero-test counter k, and $\textsc{upd}(k) = \{\, i \in I \mid i(k) \in \mathbb{Z} \,\}$ the set of instructions that update counter k.

Given a d-counter machine M and two configurations c_0, c_f, we construct an orbit-finite system of inequalities $S = (\mathbf{A}, \mathbf{t})$ such that M admits a run from c_0 to c_f if and only if S has a finitary nonnegative integer solution. (Nonegativeness is *enforced* by adding inequalities $x \geq 0$ for all unknowns x.) We describe construction of S gradually, on the way giving intuitive explanations and sketching the proof of the if direction.

The system S has unknowns $e_{\alpha\beta}$ indexed by pairs of distinct atoms $\alpha\beta \in \mathbb{A}^{(2)}$, and contains the following inequalities:

$$e_{\alpha\beta} \leq 1 \qquad (\alpha\beta \in \mathbb{A}^{(2)}). \tag{42}$$

Therefore, in every solution the unknowns $e_{\alpha\beta}$ define a directed graph G, where atoms are vertices, $e_{\alpha\beta} = 1$ encodes an edge from α to β and $e_{\alpha\beta} = 0$ encodes a non-edge. In case of a finitary solution, the graph G is finite (when atoms with no adjacent edges are dropped). Let us fix two distinct atoms $\iota, \zeta \in \mathbb{A}$. The system S contains the following further equations and inequalities:

$$\sum_{\beta \neq \alpha} e_{\beta\alpha} = \sum_{\beta \neq \alpha} e_{\alpha\beta} \leq 1 \qquad (\alpha \in \mathbb{A} \setminus \{\iota, \zeta\}) \tag{43}$$

enforcing that in-degree of every vertex, except for ι and ζ, is the same as its out-degree, and equal 0 or 1, and also

$$\sum_{\beta \neq \iota} e_{\beta\iota} = 0 \qquad \sum_{\beta \neq \iota} e_{\iota\beta} = 1 \qquad \sum_{\beta \neq \zeta} e_{\beta\zeta} = 1 \qquad \sum_{\beta \neq \zeta} e_{\zeta\beta} = 0 \tag{44}$$

enforcing that in-degree of ι and out-degree of ζ are 0, while out-degree of ι and in-degree of ζ are 1. Thus atoms split into three categories: inner nodes (with in- and out-degree equal 1), end nodes (ι and ζ) and non-nodes (with in- and out-degree equal 0). Therefore, the graph G defined by a finitary solution consists of a directed path from ι to ζ plus a number of vertex disjoint directed cycles. The path will be used below to encode a run of M: each edge, intuitively speaking, will be assigned a configuration of M, while each inner node will be assigned an instruction of M.

The system S has also unknowns $t_{i\alpha}$ indexed by instructions $i \in I$ of M and atoms $\alpha \in \mathbb{A}$, and the following equations:

$$\sum_{i \in I} t_{i\alpha} = \sum_{\beta \neq \alpha} e_{\alpha\beta} \qquad (\alpha \in \mathbb{A} \setminus \{\iota, \zeta\}). \tag{45}$$

Therefore in every finitary solution, for each inner node α of the above-defined graph G, there is exactly one instruction $i \in I$ such that $t_{i\alpha}$ equals 1 (intuitively, this instruction i is *assigned* to node α), and $t_{i\alpha}$ equals 0 for all other instructions. (This applies to *all* inner nodes of G, both those on the path as well as those on cycles.) For non-nodes α, all $t_{i\alpha}$ are necessarily equal 0. Note that the values of unknowns $t_{i\iota}$ and $t_{i\zeta}$ are unrestricted, as they are irrelevant.

Finally, the system S contains unknowns $c_{\alpha\beta\gamma k}$ indexed by $\alpha\beta\gamma \in \mathbb{A}^{(3)}$ and $k \in \{1, \dots, d\}$. The following inequalities:

$$c_{\alpha\beta\gamma k} \leq e_{\alpha\beta} \qquad (\alpha\beta\gamma \in \mathbb{A}^{(3)}, k \in \{1 \dots d\}) \tag{46}$$

enforce that, whatever atom γ is, the value of unknown $c_{\alpha\beta\gamma k}$ may be 0 or 1 when $\alpha\beta$ is an edge (i.e., when $e_{\alpha\beta} = 1$), but $c_{\alpha\beta\gamma k}$ is forcedly 0 when $\alpha\beta$ is a non-edge (i.e., when $e_{\alpha\beta} = 0$). The underlying intuition is that for each $k \in \{1, \dots, d\}$, we represent the kth coordinate of the configuration *assigned* to the edge $\alpha\beta$ by the (necessarily finite) sum

$$\sum_{\gamma \notin \{\alpha, \beta\}} c_{\alpha\beta\gamma k}. \tag{47}$$

(In particular, configurations assigned to non-edges are necessarily zero on all coordinates.) In agreement with this intuition, we add to S the requirement that the configuration assigned to the edge outgoing from ι is the source c_0, and the configuration assigned to the edge incoming to ζ is the target c_f:

$$\sum_{\beta,\gamma\neq\iota} c_{\iota\beta\gamma k} = c_0(k) \qquad \sum_{\beta,\gamma\neq\zeta} c_{\beta\zeta\gamma k} = c_f(k) \qquad (k \in \{1,\ldots,d\}).$$

Furthermore, in order to enforce correctness of encoding of a run of M, we add to S equations that relate, intuitively speaking, two consecutive configurations. Recall that, due to (43)–(44) and (46), for every $\alpha \in \mathbb{A} \setminus \{\iota,\zeta\}$, unknowns $c_{\beta\alpha\gamma k}$ may be positive for at most one $\beta \in \mathbb{A}$; likewise unknowns $c_{\alpha\beta\gamma k}$. We add to S the following equations:

$$\sum_{\beta,\gamma\neq\alpha} c_{\beta\alpha\gamma k} + \sum_{i\in\text{UPD}(k)} i(k)\cdot t_{i\alpha} = \sum_{\beta,\gamma\neq\alpha} c_{\alpha\beta\gamma k} \qquad (\alpha \in \mathbb{A}\setminus\{\iota,\zeta\}, k\in\{1\ldots d\}). \tag{48}$$

These equalities say that for every inner node or non-node α (i.e., every atom except for the end nodes ι and ζ), on every coordinate k, the configuration incoming to α differs from the configuration outgoing from α exactly by the sum

$$\sum_{i\in\text{UPD}(k)} i(k)\cdot t_{i\alpha}$$

ranging over those instructions i of M that update counter k. Remembering that for each α there is at most one instruction i satisfying $t_{i\alpha} \neq 0$, we get that the configurations differ on coordinate k by exactly $i(k)$ (if i updates counter k) or the configuration are equal on coordinate k (if i zero-tests counter k, or there is no instruction i such that $t_{i\alpha} \neq 0$).

In order to deal with zero tests, we add to S not just the inequalities (46), but the following strengthening thereof:

$$c_{\alpha\beta\gamma k} + \sum_{i\in\text{ZERO}(k)} t_{i\alpha} \leq e_{\alpha\beta} \qquad (\alpha\beta\gamma \in \mathbb{A}^{(3)}, k\in\{1\ldots d\}). \tag{49}$$

In consequence, for every edge $\alpha\beta$, if the instruction i assigned to α updates counter k, (49) does not restrict further the kth coordinate of the configuration assigned to $\alpha\beta$. But if the instruction i assigned to α zero-tests counter k, the sum

$$\sum_{i\in\text{ZERO}(k)} t_{i\alpha}$$

equals 1 and therefore the kth coordinate of the configuration assigned to $\alpha\beta$, encoded by (47), is necessarily 0 (the same applies also to the configuration incoming to α, due to inequalities (48) below). The above considerations apply to *all* edges of G, both those on the path as well as those on cycles. As a further consequence, for a non-edge $\alpha\beta$, the configuration assigned at $\alpha\beta$, encoded by (47), is necessarily the zero configuration.

The construction of S is thus completed, and it remains to argue towards its correctness:

Lemma 68. *M admits a run from c_0 to c_f if and only if S has a finitary nonnegative integer solution.*

Proof. For the 'if' direction, given a finitary nonnegative integer solution of S, we consider the graph G determined by values of unknowns $e_{\alpha\beta}$, as discussed in the course of construction, consisting of inner nodes and two end nodes, and having the form of a finite directed path plus (possibly) a number of directed cycles. By the construction of S, each edge of G has assigned a configuration of M, and each inner node has assigned an instruction of M, so that the configuration on the edge outgoing from an inner node is exactly the result of executing its instruction on the configuration assigned to the incoming edge. (As above, this applies to *all* inner nodes and edges of G, both those on the path as well as those on cycles.) Ignoring the cycles of G, we conclude that the sequence of configurations and instructions along the path of G is a run of M from c_0 to c_f.

For the 'only if' direction, given a run of M from c_0 to c_n as in (41), one constructs a solution of S in the form of a sole path involving end nodes $\alpha_0 = \iota, \alpha_{n+1} = \zeta$, n inner nodes $\alpha_1, \ldots, \alpha_n$, and $n+1$ edges $\alpha_j\alpha_{j+1}$. Thus unknowns $e_{\alpha_j,\alpha_{j+1}}$ are equal 1. The values of unknowns $t_{i\alpha_j}$ are determined by instructions i_j used in the run, and the values of the unknowns $c_{\alpha_j\alpha_{j+1}\gamma k}$, for sufficiently many fresh atoms γ, are determined by configurations c_j. All other unknowns are equal 0. □

Remark 69. The proof does not adapt to Fin-Nonneg-Eq-Solv($\mathbb{Z}$). Indeed, the standard way of transforming inequalities into equations involves adding an infinite set of additional unknowns, that might be all non-zero. ◁

10 CONCLUSIONS

As two main contributions, we show two contrasting results: decidability of orbit-finite linear programming, and undecidability of orbit-finite integer linear programming. For decidability, we invent a novel concept of setwise-T-orbit, and provide a reduction to a finite but polynomially-parametrised linear programming. In addition to the decidability of the latter problem, we show that it can be solved in ExpTime, and even in PTime for every fixed atom dimension. We thus match, in case of fixed atom dimension, the complexity of classical linear programming.

We consider non-strict inequalities for presentation only, and our decision procedures may be straightforwardly adapted to to mixed systems of strict and non-strict inequalities.

We leave a number of intriguing open questions, all of them except the last one referring to linear programming:

Question 1. In this paper we only consider finitely supported solutions. We do not know the decidability status of linear programming when this restriction is dropped (like in [23]). It is decidable for finitary inequalities, where existence of a solution implies existence of an equivariant one [33].

Question 2. We exclusively consider equality atoms, and extension to richer structures seems highly non-trivial. In the important case of *ordered atoms*, we are currently only able to prove decidability of Eq-Solv($\mathbb{F}$), for any commutative ring $\mathbb{F}$.

Question 3. It is very natural to ask if the classical duality of linear programs extends to the orbit-finite setting. According to our initial observations this is indeed the case, under the restriction that either (v) vertical vectors of a matrix and the target vector are finitary, or (h) horizontal vectors of a matrix and objective function are finitary. Whenever the primary program satisfies one of the conditions (v), (h), the dual one satisfies the other one.

Question 4. Solution sets of orbit-finite systems are not always finitely generated. Therefore an interesting question arises if one can compute a representation of solution sets that would enable testing for equality or inclusion of such sets? For instance, the solution set of the system of inequalities $\sum_{\alpha\in\mathbb{A}\setminus\{\beta\}} \alpha \geq \beta \quad (\beta \in \mathbb{A})$, in matrix form

$$\begin{bmatrix} -1 & 1 & 1 & \cdots \\ 1 & -1 & 1 & \cdots \\ 1 & 1 & -1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \cdot \mathbf{x} \quad \geq \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

is not equal to the cone generated by (=non-negative linear combinations of) an orbit-finite set.

Question 5. We would be happy to know if our general ExpTime upper complexity bound is tight.

Question 6. Concerning integer linear programming, an intriguing research task is to identify the decidability borderline. For instance, we suspect decidability in case when all inequalities are finitary. The reduction proving undecidability produces a system of atom dimension 3, and it is unclear if the dimension can be lowered to 2. In case of atom dimension 1 we suspect decidability (along the lines of [20]).

ACKNOWLEDGMENTS

We are grateful to Mikołaj Bojańczyk, Lorenzo Clemente, Damian Niwiński and Asia Fijalkow for fruitful discussions.

We acknowledge partial support of NCN Opus grant 2019/35/B/ST6/02322 (Arka Ghosh and Sławomir Lasota), of NCN Preludium grant 2022/45/N/ST6/03242 (Arka Ghosh), and of ERC Starting grant INFSYS, agreement no. 950398 (Piotr Hofman and Sławomir Lasota).

REFERENCES

[1] Albert Atserias, Anuj Dawar, and Joanna Ochremiak. 2021. On the Power of Symmetric Linear Programs. *J. ACM* 68, 4 (2021), 26:1–26:35.

[2] Saugata Basu, Richard Pollack, and Marie-Francoise Roy. 2006. *Algorithms in Real Algebraic Geometry*. Springer-Verlag.

[3] Michael Ben-Or, Dexter Kozen, and John H. Reif. 1986. The Complexity of Elementary Algebra and Geometry. *J. Comput. Syst. Sci.* 32, 2 (1986), 251–264.

[4] Mikołaj Bojańczyk. 2011. Data Monoids. In *Proc. STACS 2011 (LIPIcs, Vol. 9)*. 105–116.

[5] Mikołaj Bojańczyk. 2019. Slightly Infinite Sets. (2019). https://www.mimuw.edu.pl/~bojan/paper/atom-book

[6] Mikołaj Bojańczyk, Bartek Klin, and Slawomir Lasota. 2011. Automata with Group Actions. In *Proc. LICS 2011*. 355–364.

[7] Mikołaj Bojańczyk, Bartek Klin, and Slawomir Lasota. 2014. Automata theory in nominal sets. *Log. Methods Comput. Sci.* 10, 3 (2014).

[8] Mikołaj Bojańczyk, Bartek Klin, Sławomir Lasota, and Szymon Toruńczyk. 2013. Turing Machines with Atoms. In *Proc. LICS'13*. 183–192.

[9] Mikołaj Bojańczyk, Bartek Klin, and Joshua Moerman. 2021. Orbit-Finite-Dimensional Vector Spaces and Weighted Register Automata. In *Proc. LICS*. IEEE, 1–13.

[10] Mikołaj Bojańczyk and Szymon Toruńczyk. 2012. Imperative Programming in Sets with Atoms. In *Proc. FSTTCS 2012*, Vol. 18. 4–15.

[11] Mikolaj Bojanczyk and Szymon Torunczyk. 2018. On computability and tractability for infinite sets. In *Proc. LICS 2018*, Anuj Dawar and Erich Grädel (Eds.). ACM, 145–154.

[12] Lorenzo Clemente and Slawomir Lasota. 2015. Reachability Analysis of First-order Definable Pushdown Systems. In *Proc. CSL 2015 (LIPIcs, Vol. 41)*, Stephan Kreutzer (Ed.). 244–259.

[13] Thomas Colcombet. 2015. Unambiguity in Automata Theory. In *Proc. DCFS 2015 (LNCS, Vol. 9118)*. Springer, 3–18.

[14] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms, 3rd Edition*. MIT Press.

[15] Wojciech Czerwiński and Łukasz Orlikowski. 2022. Reachability in Vector Addition Systems is Ackermann-complete. In *Proc. FOCS 2021*. IEEE, 1229–1240.

[16] Arka Ghosh, Piotr Hofman, and Slawomir Lasota. 2022. Solvability of orbit-finite systems of linear equations. In *Proc. LICS'22*. ACM, 11:1–11:13.

[17] Arka Ghosh, Piotr Hofman, and Slawomir Lasota. 2023. Orbit-finite linear programming. In *LICS*. 1–14.

[18] Utkarsh Gupta, Preey Shah, S. Akshay, and Piotr Hofman. 2019. Continuous Reachability for Unordered Data Petri Nets is in PTime. In *Proc. FOSSACS 2019 (Lecture Notes in Computer Science, Vol. 11425)*, Mikolaj Bojanczyk and Alex Simpson (Eds.). Springer, 260–276.

[19] Piotr Hofman and Slawomir Lasota. 2018. Linear Equations with Ordered Data. In *Proc. CONCUR 2018*. 24:1–24:17.

[20] Piotr Hofman, Jérôme Leroux, and Patrick Totzke. 2017. Linear combinations of unordered data vectors. In *Proc. LICS 2017*. 1–11.

[21] Piotr Hofman and Jakub Różycki. 2022. Linear equations for unordered data vectors in $[D]^k \to Z^d$. *Logical Methods in Computer Science* Volume 18, Issue 4 (2022).

[22] Khadijeh Keshvardoost, Bartek Klin, Slawomir Lasota, Joanna Ochremiak, and Szymon Torunczyk. 2019. Definable isomorphism problem. *Log. Methods Comput. Sci.* 15, 4 (2019).

[23] Bartek Klin, Eryk Kopczyński, Joanna Ochremiak, and Szymon Toruńczyk. 2015. Locally Finite Constraint Satisfaction Problems. In *Proc. LICS 2015*. 475–486.

[24] Bartek Klin, Slawomir Lasota, Joanna Ochremiak, and Szymon Torunczyk. 2014. Turing machines with atoms, constraint satisfaction problems, and descriptive complexity. In *Proc. CSL-LICS 2014*, Thomas A. Henzinger and Dale Miller (Eds.). ACM, 58:1–58:10.

[25] Bartek Klin, Slawomir Lasota, Joanna Ochremiak, and Szymon Torunczyk. 2016. Homomorphism Problems for First-Order Definable Structures. In *Proc. FSTTCS 2016 (LIPIcs, Vol. 65)*, Akash Lal, S. Akshay, Saket Saurabh, and Sandeep Sen (Eds.). 14:1–14:15.

[26] Sławomir Lasota. 2022. Improved Ackermannian lower bound for the Petri nets reachability problem. In *Proc. STACS 2022 (LIPIcs, Vol. 219)*. 46:1–46:15.

[27] Ranko Lazic, Thomas Christopher Newcomb, Joël Ouaknine, A. W. Roscoe, and James Worrell. 2008. Nets with Tokens which Carry Data. *Fundam. Inform.* 88, 3 (2008), 251–274.

[28] Jérôme Leroux. 2022. The Reachability Problem for Petri Nets is Not Primitive Recursive. In *Proc. FOCS 2021*. IEEE, 1241–1252.
[29] Christos H. Papadimitriou and Kenneth Steiglitz. 1982. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall.
[30] A. M. Pitts. 2013. *Nominal Sets: Names and Symmetry in Computer Science*. Cambridge Tracts in Theoretical Computer Science, Vol. 57. Cambridge University Press.
[31] Manuel Silva Suárez, Enrique Teruel, and José Manuel Colom. 1996. Linear Algebraic and Linear Programming Techniques for the Analysis of Place or Transition Net Systems. In *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets*. 309–373.
[32] Alfred Tarski. 1948. *A Decision Method for Elementary Algebra and Geometry*. University of California Press.
[33] Szymon Toruńczyk. 2023. Solvability of orbit-finite systems of finite inequalities. (2023). Personal communication.
[34] Jan Martijn E. M. van der Werf, Boudewijn F. van Dongen, Cor A. J. Hurkens, and Alexander Serebrenik. 2008. Process Discovery Using Integer Linear Programming. In *Proc. Petri Nets 2008 (Lecture Notes in Computer Science, Vol. 5062)*, Kees M. van Hee and Rüdiger Valk (Eds.). Springer, 368–387.
[35] Dinh Van Le and Tim Römer. 2021. Theorems of Carathéodory, Minkowski-Weyl, and Gordan up to symmetry.

A MISSING PROOFS

We start by introducing notation useful in proving Theorems 17 and 22. For subsets $P \subseteq \textsc{Lin}(B)$ and $\mathbb{F} \subseteq \mathbb{R}$, we define $\textsc{Fin-Span}_{\mathbb{F}}(P) \subseteq \textsc{Lin}(B)$ as the set of all linear $\mathbb{F}$-combinations of vectors from P:

$$\textsc{Fin-Span}_{\mathbb{F}}(P) \;=\; \{ q_1 \cdot \mathbf{p}_1 + \ldots + q_k \cdot \mathbf{p}_k \mid k \geq 0,\ q_1, \ldots, q_k \in \mathbb{F},\ \mathbf{p}_1, \ldots, \mathbf{p}_k \in P \}.$$

Recall that given a matrix $\mathbf{A} \in \textsc{Lin}(B \times C)$ with rows B and columns C, we can define a partial operation of multiplication of $\mathbf{A}$ by a vector $\mathbf{v} \in \textsc{Lin}(C)$ in an expected way:

$$(\mathbf{A} \cdot \mathbf{v})(b) = \mathbf{A}(b, \_) \cdot \mathbf{v}$$

for every $b \in B$. The result $\mathbf{A} \cdot \mathbf{v} \in \textsc{Lin}(B)$ is well-defined if $\mathbf{A}(b, \_) \cdot \mathbf{v}$ is well-defined for all $b \in B$. For $c \in C$ we denote by $\mathbf{A}(\_, c) \in \textsc{Lin}(B)$ the corresponding (column) vector. The multiplication $\mathbf{A} \cdot \mathbf{v}$ can be also seen as an *orbit-finite* linear combination of column vectors $\mathbf{A}(\_, c)$, for $c \in C$, with coefficients given by $\mathbf{v}$. This allows us to define the *span* of $\mathbf{A}$ seen as a C-indexed orbit-finite set of vectors $\mathbf{A}(\_, c) \in \textsc{Lin}(B)$:

$$\textsc{Span}_{\mathbb{F}}(\mathbf{A}) := \{ \mathbf{A} \cdot \mathbf{v} \mid \mathbf{v} : C \to_{\text{fs}} \mathbb{F},\ \mathbf{A} \cdot \mathbf{v} \text{ well-def.} \}.$$

Therefore, a system of inequalities $(\mathbf{A}, \mathbf{t})$ has a solution if $\textsc{Span}_{\mathbb{F}}(\mathbf{A})$ contains some vector $\mathbf{u} \geq \mathbf{t}$. When $\mathbf{v}$ is finitary, well-definedness is vacuous, and we may define:

$$\textsc{Fin-Span}_{\mathbb{F}}(\mathbf{A}) := \{ \mathbf{A} \cdot \mathbf{v} \mid \mathbf{v} : C \to_{\text{fin}} \mathbb{F} \} = \textsc{Fin-Span}_{\mathbb{F}}(P)$$

for $P = \{ \mathbf{A}(\_, c) \mid c \in C \}$ the set of column vectors of $\mathbf{A}$. Therefore, a system of inequalities $(\mathbf{A}, \mathbf{t})$ has a finitary solution if $\textsc{Fin-Span}_{\mathbb{F}}(\mathbf{A})$ contains some vector $\mathbf{u} \geq \mathbf{t}$.

A.1 Proof of Theorems 17 and 22 (Section 3)

Recall that we consider supremum of a maximisation problem to be $-\infty$ if the constraints in the problem are infeasible. Therefore proving that two maximisation problems have the same supremum also proves that the underlying systems of inequalities are equisolvable. In consequence, Theorem 22 implies 17, and hence we concentrate in the sequel on proving the former one.

The proof of mutual reductions between $\textsc{Ineq-Max}(\mathbb{F})$ and $\textsc{Nonneg-Eq-Max}(\mathbb{F})$ amounts to lifting of standard arguments from finite to orbit-finite systems, and checking that all constructed objects are finitely supported. We include the reductions here mostly in order to get acquainted with orbit-finite systems. One of the remaining two reductions builds on results of [16].

Reduction of Ineq-Max($\mathbb{F}$) **to** Nonneg-Eq-Max($\mathbb{F}$). Consider an instance $(\mathbf{A}, \mathbf{t}, \mathbf{s})$ of Ineq-Max($\mathbb{F}$), supported by S, where $\mathbf{A} : B \times C \to_{\mathrm{fs}} \mathbb{F}$, $\mathbf{t} : B \to_{\mathrm{fs}} \mathbb{F}$ and $\mathbf{s} : C \to_{\mathrm{fs}} \mathbb{F}$. We construct an instance $(\mathbf{A}', \mathbf{t}, \mathbf{s}')$ of Nonneg-Eq-Max($\mathbb{F}$), with the same target vector $\mathbf{t}$, and $\mathbf{A}' : B \times (C \uplus C \uplus B) \to_{\mathrm{fs}} \mathbb{F}$, $\mathbf{s}' : (C \uplus C \uplus B) \to_{\mathrm{fs}} \mathbb{F}$, such that

$$\mathrm{supremum}(\mathbf{A}', \mathbf{t}, \mathbf{s}') = \mathrm{supremum}(\mathbf{A}, \mathbf{t}, \mathbf{s}).$$

In the new system, we double each variable x into x_+ and x_-, and we add a fresh variable per each equation. The matrix $\mathbf{A}'$ of the new system is a composition of $\mathbf{A}$, $-\mathbf{A}$, and the diagonal matrix $B \times B \to_{\mathrm{fs}} \mathbb{F}$ with -1 in the diagonal:

$$\mathbf{A}' = \left[\begin{array}{c|c|ccc} & & -1 & & \\ \mathbf{A} & -\mathbf{A} & & \ddots & \\ & & & & -1 \end{array} \right]$$

Similarly, $\mathbf{s}'$ is defined as the composition of $\mathbf{s}$, $-\mathbf{s}$ and the zero vector $B \to_{\mathrm{fs}} \mathbb{F}$:

$$\mathbf{s}' = \left[\begin{array}{c|c|ccc} \mathbf{s} & -\mathbf{s} & 0 & \cdots & 0 \end{array} \right]$$

$\mathbf{A}'$ and $\mathbf{s}'$ are thus supported by S.

Any vector $\mathbf{x}' : (C \uplus C \uplus B) \to_{\mathrm{fs}} \mathbb{F}$ can be written as

$$\mathbf{x}' = (\mathbf{x}_+ | \mathbf{x}_- | \mathbf{y}),$$

where $\mathbf{x}_+, \mathbf{x}_- : C \to_{\mathrm{fs}} \mathbb{F}$ and $\mathbf{y} : B \to_{\mathrm{fs}} \mathbb{F}$. If any such non-negative vector $\mathbf{x}'$ satisfies the above constructed system of constraints, i.e. if we have

$$\mathbf{A}' \cdot (\mathbf{x}_+ | \mathbf{x}_- | \mathbf{y}) = \mathbf{t}, \tag{50}$$

then then vector $\mathbf{x}_+ - \mathbf{x}_-$, supported by $\mathrm{supp}(\mathbf{x}')$, is a solution of $(\mathbf{A}, \mathbf{t})$, namely

$$\mathbf{A} \cdot (\mathbf{x}_+ - \mathbf{x}_-) \geq \mathbf{A} \cdot (\mathbf{x}_+ - \mathbf{x}_-) - \mathbf{y} = \mathbf{A}' \cdot (\mathbf{x}_+ | \mathbf{x}_- | \mathbf{y}) = \mathbf{t}.$$

Furthermore, by the very definition of $\mathbf{s}'$ we have

$$\mathbf{s}' \cdot (\mathbf{x}_+ | \mathbf{x}_- | \mathbf{y}) = \mathbf{s} \cdot (\mathbf{x}_+ - \mathbf{x}_-), \tag{51}$$

which implies $\mathrm{supremum}(\mathbf{A}', \mathbf{t}, \mathbf{s}') \leq \mathrm{supremum}(\mathbf{A}, \mathbf{t}, \mathbf{s})$.

In the opposite direction, given a finitely supported vector $\mathbf{x}$ such that $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{t}$, we define a non-negative vector $\mathbf{x}' = (\mathbf{x}_+ | \mathbf{x}_- | \mathbf{y})$ supported by $\mathrm{supp}(\mathbf{x}) \cup S$ as follows:

$$\mathbf{x}_+(c) = \begin{cases} \mathbf{x}(c) & \text{if } \mathbf{x}(c) \geq 0, \\ 0 & \text{otherwise;} \end{cases} \qquad \mathbf{x}_-(c) = \begin{cases} -\mathbf{x}(c) & \text{if } \mathbf{x}(c) < 0, \\ 0 & \text{otherwise;} \end{cases} \qquad \mathbf{y}(c) = (\mathbf{A} \cdot \mathbf{x} - \mathbf{t})(c).$$

Then $\mathbf{x} = \mathbf{x}_+ - \mathbf{x}_-$ and $\mathbf{A}' \cdot (\mathbf{x}_+ | \mathbf{x}_- | \mathbf{y}) = \mathbf{A} \cdot \mathbf{x} - \mathbf{y} = \mathbf{t}$. The equality (51) holds again, which implies $\mathrm{supremum}(\mathbf{A}, \mathbf{t}, \mathbf{s}) \leq \mathrm{supremum}(\mathbf{A}', \mathbf{t}, \mathbf{s}')$.

Reduction of Nonneg-Eq-Max($\mathbb{F}$) **to** Ineq-Max($\mathbb{F}$). For any orbit-finite system of linear equations supported by S:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{t},$$

its nonnegative solutions are exactly solutions of the following system of linear inequalities, also supported by S:

$$\mathbf{A}\cdot\mathbf{x} \geq \mathbf{t} \qquad \mathbf{A}\cdot\mathbf{x} \leq \mathbf{t} \qquad \mathbf{x} \geq 0.$$

This implies an easy reduction from Nonneg-Eq-Max($\mathbb{F}$) to Ineq-Max($\mathbb{F}$).

Remark 70. The above two reductions preserve row-finiteness, i.e., transform a system of finite equations (inequalities) to a system of finite inequalities (equations), or vice versa. ◁

Reduction of Fin-Ineq-Max($\mathbb{F}$) **to** Ineq-Max($\mathbb{F}$)**.** Consider an instance $(\mathbf{A}, \mathbf{t}, \mathbf{s})$ of Fin-Ineq-Max($\mathbb{F}$) supported by S, where $\mathbf{A} : B \times C \to_{\mathrm{fs}} \mathbb{F}$. We construct an instance $(\mathbf{A}', \mathbf{t}', \mathbf{s}')$ of Ineq-Max($\mathbb{F}$) as follows. The new system of inequalities $\mathbf{A}' \cdot \mathbf{x}' \geq \mathbf{t}'$ is obtained by extending the column index C by one additional variable y and extending the system by one inequality:

$$\mathbf{A}' = \left[\begin{array}{ccc|c} & & & 0 \\ & \mathbf{A} & & \vdots \\ & & & 0 \\ \hline 1 & \cdots & 1 & -1 \end{array}\right] \qquad \mathbf{t}' = \left[\begin{array}{c} \mathbf{t} \\ \hline 0 \end{array}\right]$$

and the new objective function $\mathbf{s}'$ is defined as expected:

$$\mathbf{s}' = \left[\begin{array}{c|c} \mathbf{s} & 0 \end{array}\right].$$

The so constructed instance is supported by S, and its solutions have the form $\mathbf{x}' = (\mathbf{x}, y)$, where

$$\mathbf{A}\cdot\mathbf{x} \geq \mathbf{t} \qquad \sum_{c \in C} \mathbf{x}(c) \geq y.$$

Any such finitely supported solution is necessarily finitary. This implies supremum$(\mathbf{A}, \mathbf{t}, \mathbf{s})$ = supremum$(\mathbf{A}', \mathbf{t}', \mathbf{s})$.

Reduction of Ineq-Max($\mathbb{F}$) **to** Fin-Ineq-Max($\mathbb{F}$)**.** We rely on the following result of [16][10]:

Claim 71 ([16] Claim 20). *Let $\mathbb{F} \in \{\mathbb{Z}, \mathbb{R}\}$. Given an S-supported orbit-finite matrix $\mathbf{M}$ one can effectively construct an S-supported orbit-finite matrix $\widetilde{\mathbf{M}}$ such that* $\textsc{Span}_{\mathbb{F}}(\mathbf{M}) = \textsc{Fin-Span}_{\mathbb{F}}\left(\widetilde{\mathbf{M}}\right)$.

Consider an instance $(\mathbf{A}, \mathbf{t}, \mathbf{s})$ of Ineq-Max($\mathbb{F}$), and apply the above claim to the matrix $\mathbf{M}$ (left) in order to get the matrix $\widetilde{\mathbf{M}}$ (right),

$$\mathbf{M} = \left[\begin{array}{c} \mathbf{A} \\ \hline \mathbf{s} \end{array}\right] \qquad \widetilde{\mathbf{M}} = \left[\begin{array}{c} \mathbf{A}' \\ \hline \mathbf{s}' \end{array}\right]$$

such that

$$\textsc{Span}_{\mathbb{F}}(\mathbf{M}) = \textsc{Fin-Span}_{\mathbb{F}}\left(\widetilde{\mathbf{M}}\right). \tag{52}$$

This yields an instance $(\mathbf{A}', \mathbf{t}, \mathbf{s}')$ of Fin-Ineq-Max($\mathbb{F}$), supported by supp$(\mathbf{A}, \mathbf{t}, \mathbf{s})$.

[10] The result, as shown in [16], holds for any commutative ring $\mathbb{F}$.

By the equality (52), for every $r \in \mathbb{R}$ we have the following: there exists a finitely supported vector $\mathbf{x}$ such that $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{t}$ and $\mathbf{s} \cdot \mathbf{x} = r$ if and only if there exists a finitary vector $\mathbf{x}'$ such that $\mathbf{A}' \cdot \mathbf{x}' \geq \mathbf{t}$ and $\mathbf{s}' \cdot \mathbf{x}' = r$. In consequence,

$$\text{supremum}(\mathbf{A}, \mathbf{t}, \mathbf{s}) = \text{supremum}(\mathbf{A}', \mathbf{t}, \mathbf{s}').$$

Theorem 22 is thus proved.

A.2 Proof of Theorem 20 (Section 3)

We consider cases $\mathbb{F} = \mathbb{R}$ and $\mathbb{F} = \mathbb{Z}$ separately.

Case $\mathbb{F} = \mathbb{R}$. Decidability of Fin-Nonneg-Eq-Solv($\mathbb{R}$) follows by a direct reduction of Fin-Nonneg-Eq-Solv($\mathbb{R}$) to Nonneg-Eq-Solv($\mathbb{R}$) (similar to the reduction of Fin-Ineq-Max($\mathbb{F}$) to Ineq-Max($\mathbb{F}$)) and Theorem 18.

Case $\mathbb{F} = \mathbb{Z}$. Decidability of Fin-Nonneg-Eq-Solv($\mathbb{Z}$) follows by results of [16] and [21].

Let $(\mathbf{A}, \mathbf{t})$ be an instance of Fin-Nonneg-Eq-Solv($\mathbb{Z}$), where $\mathbf{A} : B \times C \to_{\text{fs}} \mathbb{Z}$, and consider the set of column vectors

$$P = \{ \mathbf{A}(\_, c) \mid c \in C \} \subseteq \textsc{Lin}(B)$$

of $\mathbf{A}$. Then the system of equations $\mathbf{A} \cdot \mathbf{x} = \mathbf{t}$ has a finite non-negative integer solution if and only if

$$\mathbf{t} \in \textsc{Fin-Span}_{\mathbb{N}}(P) . \tag{53}$$

We rely on Theorem 3.3 of [16] which says that $\textsc{Lin}(B)$ has an orbit-finite basis. Let $\widehat{B} \subseteq \textsc{Lin}(B)$ be such a basis. This implies that there exists a linear isomorphism $\varphi : \textsc{Lin}(B) \to \textsc{Fin-Lin}(\widehat{B})$. In consequence, (53) is equivalent to

$$\varphi(\mathbf{t}) \in \textsc{Fin-Span}_{\mathbb{N}}(\varphi(P)) . \tag{54}$$

By Remark 11.16 of [21] we can compute a finite set of vectors $\left\{\mathbf{t}_1', \ldots, \mathbf{t}_k'\right\} \subseteq \textsc{Fin-Lin}(\widehat{B})$ and an orbit-finite subset $P' \subseteq \varphi(P)$ such that (54) holds if and only if

$$\mathbf{t}_i' \in \textsc{Fin-Span}_{\mathbb{Z}}(P') \tag{55}$$

for some $i \in \{1, \ldots, k\}$. The question (55) is nothing but finitary integer solvability of an orbit-finite system of equations, which is decidable using Theorem 6.1 of [16].

A.3 Proof of Theorem 24 (Section 5)

We show decidability of Poly-Ineq-Solv by encoding the problem into real arithmetic, i.e., first-order theory of $(\mathbb{R}, +, \cdot, 0, 1, \leq)$. We say that a real arithmetic formula $\varphi(x_1, \ldots, x_k)$ with free variables $x_1, \ldots, x_k$ *defines* the set of all valuations $\{x_1, \ldots, x_k\} \to \mathbb{R}$ satisfying it. When the order of free variables is fixed, we naturally identify the set defined by φ with a subset of $\mathbb{R}^k$.

Claim 72. *Every real arithmetic formula $\varphi(x)$ with one free variable, defines a finite union of (possibly infinite) disjoint intervals.*

Proof. By quantifier elimination [32], the formula $\varphi(x)$ is equivalent to a quantifier-free formula $\overline{\varphi}(x)$ with constants, namely $\varphi(x)$ and $\overline{\varphi}(x)$ define the same set. Therefore $\overline{\varphi}(x)$ is a Boolean combination of inequalities $p(x) \geq 0$, for univariate polynomials $p \in \mathbb{R}[x]$, and validity of $\overline{\varphi}(x)$ depends only on the sign of $p(x)$, for (finitely many) polynomials that appear in $\overline{\varphi}(x)$. This implies the claim. ◻

Consider a fixed system P of polynomially-parametrised inequalities over unknowns $x_1, \ldots, x_k$, and let n range over *reals*, not just over nonnegative integers. For each $n \in \mathbb{R}$, we get the system $P(n)$ of linear inequalities with *real* coefficients. Let

$$\sigma_P(n, x_1, \ldots, x_k) \tag{56}$$

be the conjunction of inequalities in P, each of the form (12); it is thus a quantifier-free real arithmetic formula which says that a tuple $\mathbf{x} = x_1, \ldots, x_k$ is a solution of $P(n)$. The existential real arithmetic formula $\psi(n) \equiv \exists \mathbf{x} : \sigma_P(n, \mathbf{x})$ with one free variable n, says that $P(n)$ has a real solution. Thus Poly-Ineq-Solv has positive answer exactly when $\psi(n)$ is true for some $n \in \mathbb{N}$.

Evaluating real arithmetic formulas of fixed quantifier alternation depth is doable in ExpTime [3], [2, Theorem 14.16]. In order to decide Poly-Ineq-Solv, the algorithm evaluates the closed formula

$$\exists \tilde{n} : \forall n : n > \tilde{n} \Longrightarrow \psi(n)$$

and answers positively if the formula is true. Otherwise, we know that the set D defined by ψ, being a finite union of intervals (cf. Claim 72), is bounded from above. The algorithm computes an integer upper bound m_0 of D, by evaluating closed existential formulas

$$\varphi_m \equiv \exists n : n > m \ \wedge \ \psi(n),$$

for increasing nonnegative integer constants $m = 0, 1, \ldots$, until φ_m eventually evaluates to false. Finally, the algorithm evaluates the formula $\psi(m)$ for all nonnegative integers m between 0 and m_0, and answers positively if $\psi(m)$ is true for some such m; otherwise the algorithm answers negatively.

A.4 Proofs of Lemmas 42 and 65 (Sections 7 and 8)

We sketch the proofs only, as they amount to a slightly tedious but entirely standard exercise in sets with atoms.

Consider an instance $(\mathbf{A}, \mathbf{t}, \mathbf{s})$ of the maximisation problem Fin-Ineq-Max($\mathbb{R}$). Let $S = \text{supp}(\mathbf{A}, \mathbf{t}, \mathbf{s})$, and let $\mathbf{A} : B \times C \to_{\text{fs}} \mathbb{Z}$. Thus the row and column index sets B and C are necessarily supported by S. We want to effectively transform the instance into another one $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{t}}, \widetilde{\mathbf{s}})$, where the row and column index sets are disjoint unions of sets of the form $\mathbb{A}^{(\ell)}$ (non-repeating tuples of atoms of a fixed length), as in (25) in Section 7.1. Moreover, the transformation should preserve the supremum:

$$\text{supremum}(\mathbf{A}, \mathbf{t}, \mathbf{s}) = \text{supremum}(\widetilde{\mathbf{A}}, \widetilde{\mathbf{t}}, \widetilde{\mathbf{s}}). \tag{57}$$

Recall that we consider supremum of a maximisation problem to be $-\infty$ if the constraints are infeasible. Therefore proving that two maximisation problems have the same supremum also proves that the underlying systems of inequalities are equisolvable.

We proceed in two steps. First we show that the row and column index sets B and C may be assumed to be disjoint unions of sets of the form $(\mathbb{A} \setminus S)^{(\ell)}$. As mentioned in Section 4, B and C are assumed to be given as finite union of S-orbits of the form $(\mathbb{A} \setminus S)^{(n)}/_G$ where $n \in \mathbb{N}$ and G is a subgroup of S_n, the group of all permutations of the set $\{1, \ldots, n\}$. Consider the partition of B and C into S-orbits:

$$B = B_1 \uplus \cdots \uplus B_k \qquad\qquad C = C_1 \uplus \cdots \uplus C_\ell,$$

where

$$B_i = (\mathbb{A} \setminus S)^{(p_i)}/_{G_i} \qquad \text{and} \qquad C_j = (\mathbb{A} \setminus S)^{(q_j)}/_{H_j}$$

for some $p_i, q_j \in \mathbb{N}$ and subgroups G_i and H_j of respectively S_{p_i} and S_{p_j}. Let f_i and g_j be the quotient maps

$$f_i : (\mathbb{A} \setminus S)^{(p_i)} \to B_i \qquad g_j : (\mathbb{A} \setminus S)^{(q_j)} \to C_j.$$

Notice that for every i, j and $x \in B_i$ and $y \in C_j$

$$|f_i^{-1}(x)| = |G_i| \qquad |g_j^{-1}(y)| = |H_j|. \tag{58}$$

We put:

$$B' \ = \ (\mathbb{A} \setminus S)^{(p_1)} \uplus \cdots \uplus (\mathbb{A} \setminus S)^{(p_k)} \qquad C' \ = \ (\mathbb{A} \setminus S)^{(q_1)} \uplus \cdots \uplus (\mathbb{A} \setminus S)^{(q_\ell)} \tag{59}$$

and define maps $f : B' \to B$ and $g : C' \to C$ by disjoint unions of $f_1, \ldots, f_k$ and $g_1, \ldots, g_\ell$, respectively:

$$f = f_1 \uplus \cdots \uplus f_k \qquad g = g_1 \uplus \cdots \uplus g_\ell.$$

Both the maps are surjective. We write $(f, g) : B' \times C' \to B \times C$ for the product of the two maps. Finally, we define a matrix $\mathbf{A}' : (B' \times C') \to_{\text{fs}} \mathbb{Z}$ and vectors $\mathbf{t}' \in \textsc{Lin}(B')$ and $\mathbf{s}' \in \textsc{Lin}(C)'$ by pre-composing with the above-defined maps:

$$\mathbf{A}' = \mathbf{A} \circ (f, g) \qquad \mathbf{t}' = \mathbf{t} \circ f \qquad \mathbf{s}' = \mathbf{s} \circ g. \tag{60}$$

Lemma 73. $\text{supremum}(\mathbf{A}, \mathbf{t}, \mathbf{s}) = \text{supremum}(\mathbf{A}', \mathbf{t}', \mathbf{s}')$.

Proof. Define two functions $F : \textsc{Lin}(C) \to \textsc{Lin}(C')$ and $G : \textsc{Lin}(C') \to \textsc{Lin}(C)$ as follows:

$$F(\mathbf{x}) \ : \ c' \mapsto \frac{\mathbf{x}(g(c'))}{|H_i|}, \text{ where } g(c') \in C_i \qquad G(\mathbf{x}') \ : \ c \mapsto \sum_{g(c')=c} \mathbf{x}'(c').$$

Both F and G are supported by S. By the very definition of F and G, together with (58), we deduce the following two facts, assuming either $\mathbf{x}' = F(\mathbf{x})$ or $\mathbf{x} = G(\mathbf{x}')$, where $\mathbf{x} \in \textsc{Lin}(C)$ and $\mathbf{x}' \in \textsc{Lin}(C')$. First, the value of $\mathbf{A} \cdot \mathbf{x}$ is well-defined if and only if the value of $\mathbf{A}' \cdot \mathbf{x}'$ is so, and in such case

$$\mathbf{A} \cdot \mathbf{x} \geq \mathbf{t} \iff \mathbf{A}' \cdot \mathbf{x}' \geq \mathbf{t}'.$$

Second, the value of $\mathbf{s} \cdot \mathbf{x}$ is well defined if and only if the value of $\mathbf{s}' \cdot \mathbf{x}'$ is so, and in such case $\mathbf{s} \cdot \mathbf{x} = \mathbf{s}' \cdot \mathbf{x}'$. The two facts prove the lemma. □

The instance $(\mathbf{A}', \mathbf{t}', \mathbf{s}')$ is supported by S.

As the second step we transform the instance $(\mathbf{A}', \mathbf{t}', \mathbf{s}')$ further so that the row and column index sets B and C are disjoint unions of sets of the form $\mathbb{A}^{(\ell)}$. Let $h : \mathbb{A} \to \mathbb{A} \setminus S$ be an arbitrarily chosen bijection. Since atoms from S do not appear in tuples belonging to B' or C', the map h induces two further bijective maps

$$f : \widetilde{B} \to B' \qquad g : \widetilde{C} \to C',$$

where

$$\widetilde{B} = \mathbb{A}^{(p_1)} \uplus \cdots \uplus \mathbb{A}^{(p_k)} \qquad \widetilde{C} = \mathbb{A}^{(q_1)} \uplus \cdots \uplus \mathbb{A}^{(q_\ell)}$$

(cf. (59)). We define a matrix $\widetilde{\mathbf{A}} : \widetilde{B} \times \widetilde{C} \to_{\text{fs}} \mathbb{Z}$ and two vectors $\widetilde{\mathbf{t}} : \widetilde{B} \to_{\text{fs}} \mathbb{Z}$ and $\widetilde{\mathbf{s}} : \widetilde{C} \to_{\text{fs}} \mathbb{Z}$ by pre-composing with the two above-defined maps, similarly as in (60):

$$\widetilde{\mathbf{A}} = \mathbf{A}' \circ (f, g) \qquad \widetilde{\mathbf{t}} = \mathbf{t}' \circ f \qquad \widetilde{\mathbf{s}} = \mathbf{s}' \circ g.$$

Knowing that $\mathbf{A}'$, $\mathbf{t}'$ and $\mathbf{s}'$ are all supported by S, we deduce that the so defined instance $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{t}}, \widetilde{\mathbf{s}})$ is equivariant and independent from the choice of the bijection $h : \mathbb{A} \to \mathbb{A} \setminus S$. The size blowup is exponential only in atom dimension of $(\mathbf{A}, \mathbf{t}, \mathbf{s})$, and hence polynomial when atom dimension if fixed.

Lemma 74. $\text{supremum}(\mathbf{A}', \mathbf{t}', \mathbf{s}') = \text{supremum}(\widetilde{\mathbf{A}}, \widetilde{\mathbf{t}}, \widetilde{\mathbf{s}})$.

Proof. Similarly as before, assuming $\mathbf{x}' = g(\widetilde{\mathbf{x}})$ for some vectors $\mathbf{x}' \in \textsc{Lin}(C')$ and $\widetilde{\mathbf{x}} \in \textsc{Lin}(\widetilde{C})$, we deduce the following two facts. First, the value of $\mathbf{A}' \cdot \mathbf{x}'$ is well-defined if and only if the value of $\widetilde{\mathbf{A}} \cdot \widetilde{\mathbf{x}}$ is so, and in such case

$$\mathbf{A}' \cdot \mathbf{x}' \geq \mathbf{t}' \iff \widetilde{\mathbf{A}} \cdot \widetilde{\mathbf{x}} \geq \widetilde{\mathbf{t}}.$$

Second, the value of $\mathbf{s}' \cdot \mathbf{x}'$ is well defined if and only if the value of $\widetilde{\mathbf{s}} \cdot \widetilde{\mathbf{x}}$ is so, and in such case $\mathbf{s}' \cdot \mathbf{x}' = \widetilde{\mathbf{s}} \cdot \widetilde{\mathbf{x}}$. The two facts prove the lemma. □

The last two lemmas prove Lemmas 42 and 65.