EPiC Series in Computing
Volume 94, 2023, Pages 350–368
Proceedings of 24th International Conference on Logic for Programming, Artificial Intelligence and Reasoning

On the Complexity of Convex and Reverse Convex Prequadratic Constraints

Rodrigo Raya, Jad Hamza, and Viktor Kunčak

School of Computer and Communication Science
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{rodrigo.raya,viktor.kuncak}@epfl.ch, jad.hamza@protonmail.com

Abstract

Motivated by satisfiability of constraints with function symbols, we consider numerical inequalities on non-negative integers. The constraints we address are a conjunction of a linear system $Ax = b$ and an arbitrary number of (reverse) convex constraints of the form $x_i \geq x_j^d$ ($x_i \leq x_j^d$). We show that the satisfiability of these constraints is NP-complete even if the solution to the linear part is given explicitly. As a consequence, we obtain NP-completeness for an extension of certain quantifier-free constraints on sets with cardinalities and function images.

1 Introduction

Many satisfiability problems in logic naturally reduce to numerical constraints. This includes in particular two-variable logic with counting [41–43], as well as description logics with cardinality bounds [3,4]. In many of these cases, the resulting numerical constraints belong to linear integer arithmetic (LIA) whose satisfiability problem is in NP [18,24,37]. However, satisfiability in the presence of functions with multiple arguments naturally leads to multiplicative constraints [20,50]. Perhaps due to a negative answer to Hilbert's 10th problem [51], such multiplicative constraints are often avoided. Even the case of atoms $t' = t^2$ yields undecidability, because of the identity $2t_1t_2 = (t_1 + t_2)^2 - t_1^2 - t_2^2$. We show, however, that certain classes of such constraints can still be solved within the complexity class NP—arguably low complexity for logical constraints.

Prequadratic constraints. The main class of numerical constraints we consider extends LIA with atoms of the form $x \leq y^d$. It is a strict subset of the so-called prequadratic constraints, which also allow atoms of the form $x \leq yz$ and were first studied in [20]. Two decades ago, the authors of [20] sketched an argument that prequadratic constraints can be decided in NEXPTIME and conjectured that the complexity can be reduced to NP. However, no result showing membership in NP has appeared to date. In the meantime, an alternative method was used to settle the complexity for Tarskian constraints [34]. Nevertheless, other reductions to such non-linear inequalities remain of interest.

In [27], the authors prove the decidability of satisfiability of *monotone exponential Diophantine formulas* (LIA with atoms of the form $x \leq y^z$ and of the form $x \leq yz$). They do so by reducing it to the emptiness of *monotone AC-tree automata* (tree automata modulo associativity and commutativity), but they do not provide complexity bounds.

One application of non-linear inequalities is the satisfiability of set algebra with cardinality constraints and images of functions of multiple arguments [50], which is related to description logics [5]. Consider the constraint $A = f[B, C]$, which states that A is the image of a two-argument function f under sets B and C. Assume that all sets are non-empty. Then such f exists if and only if $|A| \leq |B||C|$, where the equality is reached only when f is injective. Denoting $|A|$ by x, $|B|$ by y and $|C|$ by z, we obtain constraints of the form $x \leq yz$. What is more, by picking fresh sets A, B, C, we can express arbitrary conjunctions of such constraints. In other words, solving numerical inequalities is necessary to check certain constraints of cardinalities and function images.

While we leave open the question of NP membership for the general case, $x \leq yz$, we solve it in the case of conjunctions of constraints of the form $x \leq y^2$, and, more generally, $x \leq y^d$ for any positive integer d. We also consider the dual case, $x \geq y^2$, and more generally, $x \geq y^d$. As an application, we describe logics that handle quantifier-free constraints on sets with cardinalities (QFBAPA) and (inverse) function images $S = f[P^d]$ ($S = f^{-1}[P^d]$). The atomic formula $S = f[P^d]$ ($S = f^{-1}[P^d]$) expresses that S is an (inverse) image of P^d under function f. As a consequence of the results shown for $x \leq y^d$ ($x \geq y^d$), under restrictions on multiple occurrences of f, the satisfiability problem of these logics with (inverse) function images is in NP.

We believe that such results are of interest because they compose with other constructions that preserve NP membership. In particular, in a recent analysis of array theories [45] we observed that the fragment of combinatory array logic [13] corresponds to the theory generated by a power structure with an arbitrary index set subject to QFBAPA constraints. Given that [45] shows a NP complexity bound for such product, it is natural to ask how far we can extend NP satisfiability results. The non-linear constraints we present in this paper can be applied to the case when the index set I is a power J^d, because image constraints with functions on subsets of J^d reduce to non-linear constraints whose complexity we consider.

Finally, we argue that non-linear inequalities are such a natural and fundamental problem that their complexity is of intrinsic interest. Once their complexity is understood, they are likely to find other applications.

(Non-)convexity. [48] has proven a NP complexity bound for certain classes of convex non-linear constraints. However, the class of numerical constraints considered in our Theorem 12 is different, since we do not bound the degree of the non-linear monomial. On the other hand, the class of numerical constraints considered in Theorem 14 is non-convex. Indeed, consider the constraint $x \leq y^2$. Both $(x, y) = (4, 2)$ and $(x, y) = (16, 4)$ satisfy the constraints, but the midpoint of the line segment connecting them is $(10, 3)$, which does not satisfy the constraint. In the operational research literature [22, 33, 35], these constraints receive the name of reverse convex, since the set of solutions is the complement of a convex set. We are not aware of any previous NP complexity bounds for non-convex constraints.

Organization of the paper. Section 2 introduces the classes of constraints that we solve. They are of the form $\varphi = L \wedge Q$ where L stands for linear constraints and Q for certain conjunctions of monomial inequalities. We also recall known facts on the structure of semilinear sets. Finally, Lemma 11 gives a normal form that is used in the rest of the paper. Section 3 proves a NP complexity bound when Q is a conjunction of constraints of the form $x \geq y^d$. Section 4 proves a NP complexity bound when Q is a conjunction of atoms of the form $x \leq y^d$.

Note that one cannot reduce either case to the other because non-negativity of numbers breaks the symmetry between $\leq$ and $\geq$ (in fact, one case has a small model property whereas the other one needs certificates that are not always actual values of integer variables). Section 5 states NP-hardness of both problems even under the assumption that the solution of the linear part is given explicitly. Section 6 gives the complexity of satisfiability for sets with cardinalities and (inverse) function images based on Sections 3 and 4. Section 7 concludes the paper.

2 Background and Initial Analysis

2.1 Basic definitions and facts

Families of linear arithmetic constraints. We now define the families of constraints that we discuss in the paper. In the following, $\mathbb{N}$ will denote the set of non-negative integer numbers. Our constraints can be fully expressed in the framework of relational logic [10, Chapter 4], that is, first-order logic without quantifiers. All the families of constraints we address extend linear arithmetic, a restriction of full arithmetic that omits multiplication.

Definition 1. *A linear arithmetic formula is a relational formula whose atoms are of the form $a_1x_1 + \ldots + a_nx_n \leq b$ where $a_1, \ldots, a_n$ and b are integer constants and $x_1, \ldots, x_n$ are non-negative integer variables.*

Note that we choose our variables over the non-negative integers since they represent cardinalities of sets. It is straightforward to reduce linear arithmetic constraints over the integers to those over non-negative integers by encoding each integer variable as the difference of two non-negative integer variables. As we mentioned, the satisfiability problem of linear arithmetic constraints is in NP. In this paper, we will show NP-completeness for the following extensions of linear arithmetic.

Definition 2. *A less-than-monomial (more-than-monomial) constraint is a relational conjunction whose atoms are linear arithmetic formulae or of the form $x \leq y^d$ ($x \geq y^d$) where x, y are variables, $d \geq 2$ is a non-negative integer that may be distinct for different atoms and y^d denotes the d^{th} power of y.*

We will refer to the non-linear part of less-than-monomial and more-than-monomial constraints as *monomial inequalities*. The non-linear restrictions of less-than-monomial constraints form a strict subset of the non-linearities in the prequadratic class [20].

Definition 3. *A set of Diophantine inequalities of the form $p(x_1, \ldots, x_n) \leq q(x_1, \ldots, x_n)$ between polynomials p and q over nonnegative integer variables $x_1, \ldots, x_n$ is prequadratic if every p is linear and every q is either linear or is a product of variables.*

By adding slack variables, we may transform any prequadratic constraint $p(x_1, \ldots, x_n) \leq q(x_1, \ldots, x_n)$ as a Diophantine equation $p(x_1, \ldots, x_n) + s = q(x_1, \ldots, x_n)$. Solving these equations over the integers was shown to be undecidable in a joint effort of Davis, Matiyasevich, Putnam and Robinson [32], which yielded a solution of the so-called Hilbert's tenth problem. Furthermore, it is easy to show that the analogous problem over the non-negative numbers is also undecidable [32, Section 1.3]. In our case, this yields at once the following corollary.

Corollary 4. *The satisfiability problem for relational conjunctions whose atoms are linear arithmetic formulae or of the form $x \leq y^d$, $x \geq y^d$ where x, y are variables, d is a non-negative integer that may be distinct for different atoms and y^d denotes the d^{th} power of y is undecidable.*

In Section 6, we will also make use of the quantifier-free fragment of BAPA [28, 29], termed QFBAPA, whose language allows to express Boolean algebra and cardinality constraints on sets. Figure 1 shows the syntax of the fragment: F presents the Boolean structure of the formula, A stands for the top-level constraints, B gives the Boolean restrictions and T the Presburger arithmetic terms. $\mathcal{U}$ stands for the universe of the interpretation and MAXC for its cardinality.

$$
\begin{aligned}
F &::= A \,|\, F_1 \wedge F_2 \,|\, F_1 \vee F_2 \,|\, \neg F \\
A &::= B_1 = B_2 \,|\, B_1 \subseteq B_2 \,|\, T_1 = T_2 \,|\, T_1 \leq T_2 \,|\, K \text{ dvd } T \\
B &::= x \,|\, \emptyset \,|\, \mathcal{U} \,|\, B_1 \cup B_2 \,|\, B_1 \cap B_2 \,|\, B^c \\
T &::= k \,|\, K \,|\, \text{MAXC} \,|\, T_1 + T_2 \,|\, K \cdot T \,|\, |B| \\
K &::= \ldots \,|\, -2 \,|\, -1 \,|\, 0 \,|\, 1 \,|\, 2 \,|\, \ldots
\end{aligned}
$$

Figure 1: QFBAPA's syntax

Note that QFBAPA constraints can also be seen as extending linear arithmetic restrictions. Indeed, as noted in [28, Section 2], the addition of the cardinality operator allows to express all Presburger arithmetic (i.e. the theory of the structure $\langle \mathbb{N}, 0, 1, +, \leq \rangle$) operations. In turn, these can be efficiently represented by linear arithmetic constraints. The relation K dvd T (divisibility by an integer constant K) and the term $K \cdot T$ (multiplication by an integer constant) are added to preserve the expressive power of full first-order Presburger arithmetic as in [44].

Semilinear sets. As a first step of our NP procedures, we will guess a normal form of the input constraint. The particular normal form is based on results on the structure of the sets defined by the linear part of the constraint.

Let $\mathbb{N}^n$ denote the direct product $\mathbb{N}$ taken n times. We will distinguish elements of $\mathbb{N}^n$ from those in $\mathbb{N}$ using bold font. If $\mathbf{x} \in \mathbb{N}^n$ then the sum norm and the infinite norm are defined as follows:

$$
\begin{aligned}
\|\mathbf{x}\|_1 &= \sum_{i=1}^{n} |x_i| \\
\|\mathbf{x}\|_\infty &= \max\{|x_1|, \ldots, |x_n|\}
\end{aligned}
$$

A subset $L \subseteq \mathbb{N}^n$ is *linear* if there exist members $\mathbf{a}, \mathbf{b}^1, \ldots, \mathbf{b}^m \in \mathbb{N}^n$ such that:

$$
L = \left\{ \mathbf{x} \middle| \exists \alpha_1, \ldots, \alpha_m \in \mathbb{N}.\ \mathbf{x} = \mathbf{a} + \sum_{i=1}^{m} \alpha_i \mathbf{b}^i \right\}
$$

The element $\mathbf{a}$ is called the *base vector* of L and the elements $\mathbf{b}^1, \ldots, \mathbf{b}^m$ are called the *step vectors* of L. We refer to both the base vectors and the step vectors as the *generators* of the set L.

S is *semilinear* [38, Definition 12] if it is the union of a finite number of linear sets. The base vectors (step vectors, generators) of S are defined as the union of the set of base vectors (step vectors, generators) of each of its linear parts [40].

In [19], it was shown that the sets definable by linear arithmetic formulas are precisely the semilinear sets. Every semilinear set can be written in the form $\{\mathbf{x} \in \mathbb{N}^n | F(\mathbf{x})\}$ where F is a linear arithmetic formula. Furthermore, it was shown in [15, 31] that when given in terms of a linear arithmetic formula F, the semilinear set defined by F can be represented using a set of generators whose coefficients are polynomially bounded in the size of F. [39, Theorem 2.13] derives the following normal form for F based on these facts.

Theorem 5. *Let F be a linear arithmetic formula of size s. Then there exist numbers $m, q_1, \ldots, q_m \in \mathbb{N}$ and vectors $\mathbf{a_i}, \mathbf{b_{ij}} \in \mathbb{N}^n$ for $1 \le j \le q_i, 1 \le i \le m$ with $\|\mathbf{a_i}\|_1, \|\mathbf{b_{ij}}\|_1 \le 2^{p(s)}$ with p polynomial such that F is equivalent to the formula:*

$$\exists \alpha_{11}, \ldots, \alpha_{mq_m} \in \mathbb{N}. \bigvee_{i=1}^{m} \left(\mathbf{x} = \mathbf{a_i} + \sum_{j=1}^{q_i} \alpha_{ij} \mathbf{b_{ij}} \right)$$

Finally, the integer analog of Carathéodory's theorem [17] allows to express any element of a semilinear set using polynomially many step vectors. It is formulated in terms of integer conic hulls.

Definition 6. *Given a subset $S \subseteq \mathbb{N}^n$, the integer conic hull of S is the set:*

$$int_{cone}(S) = \left\{ \sum_{i=1}^{t} \lambda_i s_i \middle| t \ge 0, s_i \in S, \lambda_i \in \mathbb{N} \right\}$$

Theorem 7. *Let $X \subseteq \mathbb{N}^n$ be a finite set of integer vectors and $\mathbf{b} \in int_{cone}(X)$. Then there exists a subset $X' \subseteq X$ such that $\mathbf{b} \in int_{cone}(X')$ and:*

$$|X'| \le 2n \log(4nM)$$

where $M = \max_{\mathbf{x} \in X} \|\mathbf{x}\|_\infty$.

Computational complexity. We assume basic definitions in the theory of computation [2, 47] such as NP-hardness and NP-completeness. We will use the notion of polynomial-time verifier which is equivalent to that of non-deterministic polynomial-time procedure with the difference that the non-deterministic computation is encoded as a *certificate*.

Definition 8. *A language $L \subseteq \{0,1\}^*$ is in NP if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine V, called the verifier for L such that for every $x \in \{0,1\}^*$, $x \in L$ if and only if there exists $C \in \{0,1\}^{p(|x|)}$ such that $V(x, C) = 1$. If $x \in L$ and $C \in \{0,1\}^{p(|x|)}$ satisfy $V(x, C) = 1$, then C is called a certificate for x.*

It is straightforward to generalise the notion of polynomial-time verifier so that it outputs a bit-string rather than a single bit. We use this notion to define NP-reductions which we use to combine the normal form of Lemma 11 with the polynomial-time verifiers of Theorem 12 and Theorem 14 into a single NP procedure.

Definition 9. *A language $L \subseteq \{0,1\}^*$ is NP-reducible to a language $L' \subseteq \{0,1\}^*$, written $L \le_{np} L'$, if there is a polynomial-time verifier V such that for every $x \in \{0,1\}^*$, $x \in L$ if and only if there exists a certificate C such that $V(x, C) \in L'$.*

Lemma 10. *The relation $\le_{np}$ satisfies the following properties:*

1. *If $L \le_{np} L'$ and $L' \in NP$ then $L \in NP$.*
2. *If $L \le_{np} L'$ and $L' \le_{np} L''$ then $L \le_{np} L''$.*

Proof. Ad 1), by hypothesis, we have a verifier V satisfying Definition 9 for $L \le_{np} L'$. We also have a verifier V' accepting L'. Then $V'' = V' \circ V$ is a verifier for L. Ad 2), if V satisfies Definition 9 for $L \le_{np} L'$ and V' satisfies Definition 9 for $L' \le_{np} L''$ then $V' \circ V$ satisfies Definition 9 for $L \le_{np} L''$. □

2.2 Normal Form of Constraints

The following lemma gives a normal form for linear arithmetic formulae conjoined with monomial inequality constraints. The resulting structure of the generators of the semilinear set is shown in Figure 2. This normal form will be used as input in Sections 3 and 4 to establish NP complexity bounds for the more-than-monomial and less-than-monomial constraints.

Lemma 11. *There is a NP-reduction mapping each satisfiable formula $(F \wedge Q)(\mathbf{x})$ where F is a linear arithmetic formula and Q is a conjunction of monomial inequalities to a satisfiable formula $(L \wedge Q')(\mathbf{x})$ where L is of the form $\exists \alpha_1, \ldots, \alpha_K \in \mathbb{N}.\ \mathbf{x} = \mathbf{a} + \sum_{i=1}^{K} \alpha_i \mathbf{b}^i$ and Q' is a conjunction of monomial inequalities obtained from Q by permuting its variables. Moreover, the reduction ensures that:*

1. *K is polynomial in the size of F.*
2. *If x_i, a_i and b^i_j are the coordinates of the vectors $\mathbf{x}, \mathbf{a}$ and $\mathbf{b}^i$ then:*
    - *$a_1 \leq \ldots \leq a_n$ and $b^i_1 \leq \ldots \leq b^i_n$ for all $i = 1, \ldots, K$.*
    - *for all satisfying assignments of $L \wedge Q'$, $x_1 \leq \ldots \leq x_n$.*
3. *$\mathbf{b}^i > \mathbf{b}^{i+1}$ for all $i = 1, \ldots, K$ where $>$ is the strict lexicographic order defined as $\mathbf{b} > \mathbf{b}' \equiv \exists k.\ b_k > b'_k \wedge \bigwedge_{1 \leq j < k} b_j = b'_j$.*
4. *$b^i_1 = 0$ for all $i = 1, \ldots, K$.*

Proof. The NP-reduction is the composition of two simpler ones.

The first reduction is based on the observation that for each satisfiable formula $(F \wedge Q)(\mathbf{x})$, we can choose a permutation $\sigma_{F\wedge Q}$ such that the formula $(F \wedge Q)(\mathbf{x}) \wedge x_{\sigma_{F\wedge Q}(1)} \leq \cdots \leq x_{\sigma_{F\wedge Q}(n)}$ or renaming variables $(F \wedge Q)(\sigma^{-1}_{F\wedge Q}(\mathbf{y})) \wedge y_1 \leq \ldots \leq y_n$ where $\sigma^{-1}(\mathbf{y}) = (y_{\sigma^{-1}(1)}, \ldots, y_{\sigma^{-1}(n)})$ is satisfiable. We define a polynomial-time verifier V that takes a certificate, interprets it as a permutation σ of the variables occurring in its input formula $(F \wedge Q)(\mathbf{x})$ and outputs the formula $(F \wedge Q)(\sigma^{-1}(\mathbf{y})) \wedge y_1 \leq \ldots \leq y_n$. V satisfies Definition 9: if $(F \wedge Q)(\mathbf{x})$ is satisfiable then some certificate encodes $\sigma^{-1}_{F\wedge Q}$ and if $(F \wedge Q)(\mathbf{x})$ is not satisfiable then for any certificate the formula $(F \wedge Q)(\sigma^{-1}(\mathbf{y})) \wedge y_1 \leq \ldots \leq y_n$ is still unsatisfiable. The permutation can be encoded in the certificate since it takes linear space on the input formula.

For the second reduction, we observe that since F is satisfiable, there must exist a satisfiable disjunct in its full disjunctive normal form. Such a disjunct can be encoded in the certificate of a polynomial-time verifier because it only takes linear space in the size of F [45, Lemma 1] and it can be written as a linear system $A\mathbf{y} \leq \mathbf{b}$. Such a linear system is itself a linear arithmetic formula. By Theorem 5, its satisfying assignments are of the form $\mathbf{y} = \mathbf{a} + \sum_j \alpha_j \mathbf{b}_j$ with $\|\mathbf{a}\|_1, \|\mathbf{b_j}\|_1 \leq 2^{p(s)}$, p polynomial and s the size of F. Let $\mathbf{y_{F\wedge Q}}$ be one such satisfying assignment. By Theorem 7, there is a polynomial q such that $\mathbf{y_{F\wedge Q}} = \mathbf{a_{F\wedge Q}} + \sum_{j=1}^{q(s)} \alpha^j_{F\wedge Q} \mathbf{b^j_{F\wedge Q}}$. We define a polynomial-time verifier V that takes a certificate C, interprets it as the list $A, \mathbf{a}, \mathbf{b^j}, \mathbf{b}$, checks $A\mathbf{a} \leq \mathbf{b}$, $A\mathbf{b^j} \leq \mathbf{0}$ and if successful outputs $\exists \alpha_1, \ldots, \alpha_{q(s)}.\mathbf{y} = \mathbf{a} + \sum_{j=1}^{q(s)} \alpha_j \mathbf{b^j} \wedge Q(\mathbf{y})$, satisfies Definition 9: if $(F \wedge Q)(\mathbf{y})$ is satisfiable, then there is some certificate encoding $\mathbf{a_{F\wedge Q}}, \mathbf{b^j_{F\wedge Q}}, A, \mathbf{b}$ and if $(F \wedge Q)(\mathbf{y})$ is not satisfiable then the formula $\exists \alpha_1, \ldots, \alpha_{q(s)}.\mathbf{y} = \mathbf{a} + \sum_{j=1}^{q(s)} \alpha_j \mathbf{b^j} \wedge Q(\mathbf{y}) \wedge A\mathbf{a} \leq \mathbf{b} \wedge \bigwedge A\mathbf{b^j} \leq \mathbf{0}$ is also unsatisfiable.

Composing the two reductions, we obtain a formula

$$\psi \equiv \left(\exists \alpha_1, \dots, \alpha_K \in \mathbb{N}. \mathbf{y} = \mathbf{a} + \sum_{i=1}^{q(s)} \alpha_i \mathbf{b}^i\right) \wedge y_1 \leq \dots \leq y_n \wedge Q'(\mathbf{y})$$

equisatisfiable with $(F \wedge Q)(\mathbf{x})$.

We now show items 2, 3 and 4.

For 2), observe that the first transformation ensures that $y_1 \leq \dots \leq y_n$ for any solution $\mathbf{y} = \mathbf{a} + \sum_{i=1}^{K} \alpha_i \mathbf{b}^i$. Taking all the $\alpha_i = 0$ yields $\mathbf{y} = \mathbf{a}$. This implies that $a_1 \leq \dots \leq a_n$. Now, for each i, take $\mathbf{y} = \mathbf{a} + \alpha_i \mathbf{b}^i$ by setting $\alpha_j = 0$ for $j \neq i$. Finally, the coordinates of $\mathbf{b}^i$ are increasing. By contradiction, if $b^i_j > b^i_k$ for $j < k$ then there is some α_i such that $y_j = a_j + \alpha_i b^i_j > a_k + \alpha_i b^i_k = y_k$. But we showed that the components of $\mathbf{y}$ are linearly ordered.

For 3), observe that there is no need to have two identical (or even linear dependent) vectors among $\mathbf{b}^i$ in ψ. So, we assume the vectors are distinct. As the order of vectors is not relevant either, we will henceforth assume that the order of vectors is chosen so that $i_1 < i_2$ implies $\mathbf{b}^{i_1} > \mathbf{b}^{i_2}$, i.e. $\mathbf{b}^1 > \dots > \mathbf{b}^K$.

4) is a consequence of the coefficients of the step vectors being linearly ordered. Indeed, if for some i we have that $b^i_1 \geq 1$ then, for all j, $b^i_j \geq b^i_1 \geq 1$. Setting $\alpha_j = 0$ for $j \neq i$ and letting α_i increase towards infinity, each prequadratic constraint $x_i \leq x^d_j$ becomes satisfied because the left-hand side grows linearly whereas the right-hand side grows at least quadratically. This implies that $x_1 = a_1$. $\square$

Figure 2 shows the structure of the matrix of step vectors that the verifier of Lemma 11 guesses. The matrix is syntactically similar to the row echelon form found in Gaussian elimination. Here, all zero rows are at the top and each zero value of a row appears to the right (but not necessarily strictly to the right) of its previous row.

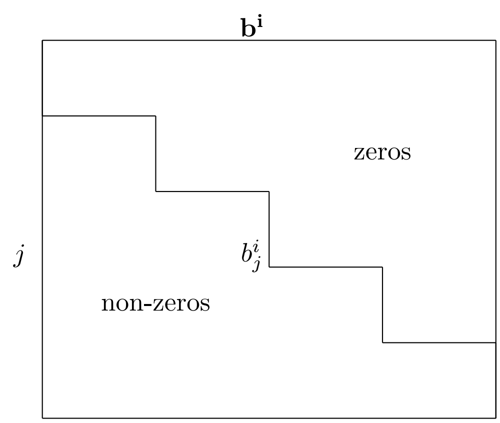

Figure 2: Vertical column arrangement of the step vectors $\mathbf{b}^1,\dots,\mathbf{b}^K$.

3 Satisfiability of Convex Monomials

This section proves an NP bound for more-than-monomial constraints. We assume the input formula is given in the normal form found in Lemma 11. Let m denote the largest constant appearing in the constraint, that is, the largest among all coordinates a_j and b^i_j. We first note that, in this case, it is not possible to find a polynomial bound on the size of minimal solutions, because there are systems whose minimal solutions are doubly exponential in m (and thus have an exponential number of bits). For example, consider the following system of n variables:

$$\begin{cases} x_1 \geq 2 \\ x_{i+1} \geq x_i^2 \quad \forall i \in \{1, \ldots, n-1\} \end{cases}$$

Consider any solution $x_1, \ldots, x_n$ of the above system. Then by induction it immediately follows that $x_i \geq 2^{2^{i-1}}$ for $1 \leq i \leq n$. Indeed, $x_1 \geq 2 = 2^{2^0}$ and if $x_i \geq 2^{2^{i-1}}$ for $i < n$ then:

$$x_{i+1} \geq x_i^2 \geq \left(2^{2^{i-1}}\right)^2 = 2^{2 \cdot 2^{i-1}} = 2^{2^i}$$

Despite the lack of small enough solutions, we show that the satisfiability problem can be solved in non-deterministic polynomial time by observing that satisfiability can be checked without exhibiting a specific solution. In Section 6, we apply this result to show a NP upper bound of a fragment of QFBAPA with (inverse) function images.

Theorem 12. *Satisfiability of more-than-monomial constraints is in NP.*

Proof. We can assume that the input formula is of the form specified in Lemma 11. Let m denote the maximum of the coefficients of the generators of the linear part. We introduce the notation j^* to refer to the column of the first zero entry for the j-th row, and $supp(j)$ to refer to the set of indices with non-zero values of the j-th row of the step vector matrix (see Figure 3):

$$j^* := \begin{cases} 0 & \text{if for every } 1 \leq i \leq K.\ b^i_j \neq 0 \\ i & \text{if } i \text{ is the least index such that } b^i_j = 0 \end{cases}$$
$$\text{supp}(j) := \{i \mid b^i_j \neq 0\} = [1, j^* - 1]$$

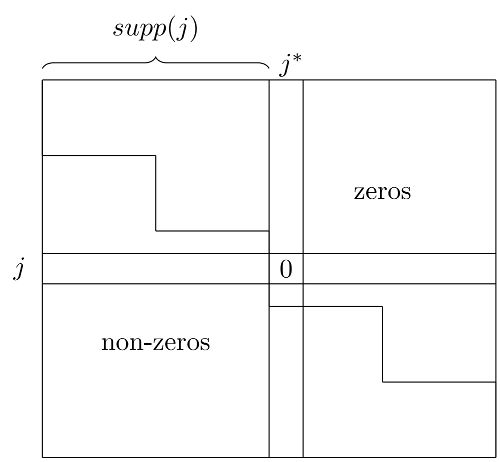

Figure 3: The support supp(j) and the critical value j^* of a row j in the vertical column arrangement of the step vectors $\mathbf{b}^1,\ldots,\mathbf{b}^K$.

The proof is based on three observations:

1) We can assume that Q contains only constraints of the form $x_k \geq x_j^d$ with $j < k$. If Q contains a constraint $x_k \geq x_j^d$ with $j \geq k$ then we would have $x_j \geq x_k \geq x_j^d \geq x_j$ and thus $x_j = x_k = 1$ or $x_j = x_k = 0$. Thus, these can be guessed and substituted by the NP procedure.

2) If there is $x_j^d \leq x_k \in Q$ such that $I = \mathrm{supp}(j) = \mathrm{supp}(k)$ then $\alpha_i \leq m$ for every $i \in I$. Towards a contradiction, assume that $\alpha_l \geq m+1$ for some $l \in I$. Note that since $l < j^*$, $b_j^l > 0$. Let $v_j = a_j + \alpha_l b_j^l$ and $v_k = a_k + \alpha_l b_k^l$. We have $v_j^d > v_k$ because:

$$v_j^d \geq \alpha_l^d > (\alpha_l - 1)(\alpha_l + 1)\alpha_l^{d-2} \geq m(\alpha_l + 1)\alpha_l^{d-2} \geq m(\alpha_l + 1) = m + \alpha_l m \geq v_k$$

It is also the case that $v_j \geq \alpha_l b_j^l \geq \alpha_l \geq m + 1$.

Since $(x_j, x_k) = (v_j, v_k) + \sum_{i \in I \setminus \{l\}} \alpha_i (b_j^i, b_k^i)$, we obtain a contradiction with the inequality $x_j^d \leq x_k$:

$$\begin{aligned}
x_j^d &= \Big(v_j + \sum_{i \in I \setminus \{l\}} \alpha_i b_j^i\Big)^d = \Big(v_j + \sum_{i \in \mathrm{supp}(j) \setminus \{l\}} \alpha_i b_j^i\Big)^d \\
&\geq v_j^d + \binom{d}{d-1} v_j^{d-1} \Bigg(\sum_{i \in \mathrm{supp}(j) \setminus \{l\}} \alpha_i\Bigg) + \Bigg(\sum_{i \in \mathrm{supp}(j) \setminus \{l\}} \alpha_i\Bigg)^d \\
&> v_j^d + v_j \sum_{i \in \mathrm{supp}(j) \setminus \{l\}} \alpha_i \geq v_k + (m+1) \sum_{i \in \mathrm{supp}(k) \setminus \{l\}} \alpha_i \\
&> v_k + \sum_{i \in \mathrm{supp}(k) \setminus \{l\}} \alpha_i b_k^i = x_k
\end{aligned}$$

3) Otherwise, for every $x_j^d \leq x_k \in Q$, $\mathrm{supp}(j) \subsetneq \mathrm{supp}(k)$. Then, x_j depends only on $b^1, \ldots, b^{j^*-1}$ while x_k depends also on a term $\alpha_{j^*} b_k^{j^*}$ where $b_k^{j^*} > 0$. We can thus extend any solution $(\alpha_1, \ldots, \alpha_{j^*-1})$ of constraints which only depends on $b^1, \ldots, b^{j^*-1}$ to a solution $(\alpha_1, \ldots, \alpha_{j^*})$ where $x_j^d \leq x_k$ also holds, by making α_{j^*} large enough.

These observations suggest the following NP algorithm.

On input $\langle L \wedge Q \rangle$ in normal form:

1. Compute the set B of inequalities $x_j^d \leq x_k \in Q$ such that $\mathrm{supp}(j) = \mathrm{supp}(k)$.
2. If $B = \emptyset$ then accept. Otherwise, non-deterministically guess $\alpha_1, \ldots, \alpha_l \leq m$ where $l = \max_{x_j^d \leq x_k \in B} (j^* - 1)$.
3. Accept iff $\alpha_1, \ldots, \alpha_l$ satisfy the inequalities $x_j^d \leq x_k \in Q$ with $k^* - 1 \leq l$.

If there is a solution to the constraints in Q then it is clear that the algorithm accepts. Conversely, if the algorithm accepts, we can construct a solution $(\alpha_1, \ldots, \alpha_l, \alpha_{l+1}^*, \ldots, \alpha_n^*)$ for Q as follows.

On input $\langle Q, B \rangle$:

1. Sort the inequalities $x_j^d \leq x_k \in Q \setminus B$ with $k > l$ by lexicographic order of the tuple (j, k) in a list $\mathcal{L}$.
2. While $\mathcal{L}$ is non-empty:

- Remove the first element (j,k) of $\mathcal{L}$ and find a coefficient α_{jk} for b^{j^*} such that $x_j^d \leq x_k$ is holds. This is possible since $\text{supp}(j) \subset \text{supp}(k)$.
- Repeat for all pairs of the form (j',k') with $\text{supp}(j') = \text{supp}(j)$. Since the step vectors are ordered lexicographically, these appear immediately after (j,k).
- Set $\alpha^*_{j^*} = \max_{j',k} \alpha_{j',k}$ and $\alpha^*_t = 0$ for any $l+1 \leq t < j$ previously left unset.

3. Set the remaining α^*_j to zero.

The result, $(\alpha_1, \ldots, \alpha_l, \alpha^*_{l+1}, \ldots, \alpha^*_n)$, satisfies Q by construction. □

The constraints we solve in Theorem 12 are of the form $x \geq y^d$ or equivalently $0 \geq f(x,y)$ where $f(x,y) = y^d - x$. This function is convex since it is the addition of a linear function (trivially convex) and the d^{th} power function (convex for having a positive semidefinite Hessian) [8, sections 3.1.4 and 3.2.1].

In [26], a related result is given for systems of s convex polynomial inequalities $f_i(x_1, \ldots, x_n) \leq 0, i = 1, \ldots, s$ where the $f_i(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_n]$ are convex polynomials in $\mathbb{R}^n$ with integral coefficients. It is formulated over the integers but one can add the linear constraints $-x_i \leq 0$ (which are trivially convex) to obtain an analogous result over the natural numbers.

Theorem 13 (Tarasov and Khachiyan (1980) [48])**.** *For a fixed $d \geq 1$ the problem of determining the consistency of systems of convex diophantine inequalities of degree at most d over the integers belongs to the class NP.*

While Theorem 13 allows for arbitrary convex inequalities, it fixes the degree that the polynomial-time verifier can handle. Our Theorem 12, on the other hand, focuses on monomial constraints but gives a single verifier for the entire class, over all degrees d.

4 Satisfiability of Non-Convex Monomials

This section proves a NP complexity bound for less-than-monomial constraints. Our proof shows a *small model property* for $(\alpha_1, \ldots, \alpha_n)$. If there is a solution, then there is a solution where $\alpha_i \leq m+1$ for $i \in \{1, \ldots, n\}$ where m is the maximum of the coefficients o fthe generators of the linear part.

The key insight of the proof is that we can avoid recomputation of the underlying linear set each time we substitute one fixed variable. Instead, we guess small coefficients α_i and show that if α_i is large enough, then the prequadratic constraints $x_l \leq x_j \cdot x_k$ where $b^i_l, b^i_j, b^i_k > 0$ are satisfiable. This follows from an inductive argument that is sketched in the fourth case distinction below.

Theorem 14. *Satisfiability of less-than-monomial constraints is in NP.*

Proof. We can assume that the input formula is of the form specified in Lemma 11. Let m denote the maximum of the coefficients of the generators of the linear part. We introduce the notation i_* to refer to the row of the last zero entry and $\text{null}(\mathbf{b}^i)$ to refer to the set of indices with zero values of the step vector $\mathbf{b}^i$ (see figure 4):

$$i_* = \begin{cases} 0 & \text{if } \text{null}(\mathbf{b}^i) = \emptyset \\ \max \text{null}(\mathbf{b}^i) & \text{if } \text{null}(\mathbf{b}^i) \neq \emptyset \end{cases}$$
$$\text{null}(\mathbf{b}^i) = \{j \mid \mathbf{b}^i_j = 0\}$$

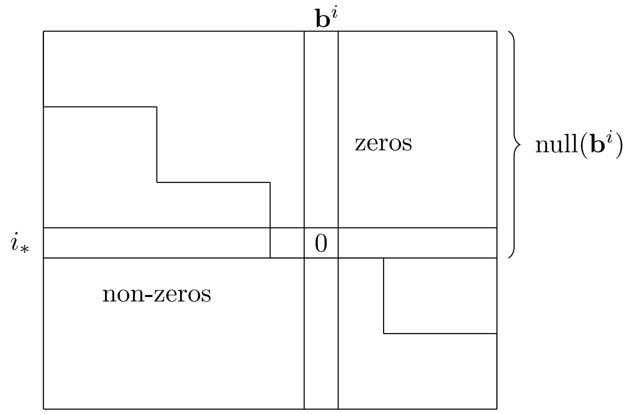

Figure 4: The nullity null($\mathbf{b}^i$) and the critical value i_* of a column $\mathbf{b}^i$ in the vertical column arrangement of the step vectors $\mathbf{b}^1,\ldots,\mathbf{b}^K$.

Given a solution $\mathbf{x^s} = \mathbf{a} + \sum_{i=1}^{K} \alpha_i \mathbf{b}^i$, our goal is to prove that there exists another solution $\mathbf{x^{s\prime}} = \mathbf{a} + \sum_{i=1}^{K} \alpha_i' \mathbf{b}^i$ of $L \wedge Q$ where $\max_i \alpha_i' \leq m+1$.

If $\max_i \alpha_i \leq m+1$ then we are done. Otherwise, let l be the smallest index such that $\alpha_l > m+1$. Since we assume a lexicographic order in the $\mathbf{b}^i$'s, if $i \leq i'$ then null($\mathbf{b}^i$) $\subseteq$ null($\mathbf{b}^{i'}$). This together with the linear order in the solutions $\mathbf{x^s}$ leads to a matrix of step vectors where $b^l_{l_*}$ separates the lower-left non-zero submatrix from the upper-right zero part.

We construct another solution $\mathbf{x^s}' = \mathbf{a} + \sum_{i=1}^{K} \alpha_i{}' \mathbf{b^i}$ with:

$$\alpha_i' = \begin{cases} \alpha_i & \text{if } i < l_* \\ m+1 & \text{if } i = l_* \\ 0 & \text{if } i > l_* \end{cases}$$

$\mathbf{x^s}'$ is a small solution in terms of $(\alpha_1, \ldots, \alpha_n)$ since $\|(\alpha_1', \ldots, \alpha_n')\|_\infty = m+1$. Furthermore, $x_j^{s'} \leq x_j^s$ for any j since if $j \leq l_*$ then:

$$x_j^s = a_j + \sum_{i=1}^{K} \alpha_i b_j^i = a_j + \sum_{i=1}^{l-1} \alpha_i b_j^i = a_j + \sum_{i=1}^{l-1} \alpha_i' b_j^i = a_j + \sum_{i=1}^{K} \alpha_i' b_j^i = x_j^{s'}$$

and if $l_* < j$:

$$x_j^{s'} = a_j + \sum_{i=1}^{K} \alpha_i' b_j^i = a_j + \sum_{i=1}^{l-1} \alpha_i b_j^i + (m+1) b_j^l <$$
$$< a_j + \sum_{i=1}^{l-1} \alpha_i b_j^i + \alpha_l b_j^l \leq a_j + \sum_{i=1}^{K} \alpha_i b_j^i = x_j^s$$

where we used that all base and step vector components and all coefficients are greater or equal than zero and $\alpha_l > m+1, b_j^i \geq 1$ for $i < l$.

Finally, we show that $\mathbf{x^s}'$ is a solution of Q. Given $x_j \leq x_k^d \in Q$, we show $x_j^{s'} \leq (x_k^{s'})^d$. Consider four cases:

1. $j \leq k$: the components of the solutions are linearly ordered and thus $x_j^{s'} \leq x_k^{s'} \leq x_k^{s'd}$.
2. $k < j \leq l_*$: $x_j^{s'} = x_j^s \leq (x_k^s)^d = (x_k^{s'})^d$.
3. $k \leq l_* < j$: $x_j^{s'} < x_j^s \leq (x_k^s)^d = (x_k^{s'})^d$
4. $l_* < k < j$: call $v_j = a_j + (m+1)b_j^l$ and $v_k = a_k + (m+1)b_k^l$.
   We show by finite induction on the natural number $t \leq l_*$ that:
   $$v_j + \sum_{i<t} \alpha_i' b_j^i \leq \Big(v_k + \sum_{i<t} \alpha_i' b_k^i\Big)^d$$
   (a) In the base case, $t = 0$ and we need to show $v_j \leq v_k^d$:
   $$v_j \leq m + (m+1)m \leq (m+1)^2 \leq (m+1)^d \leq (a_k + (m+1)b_k^l)^d = v_k^d$$
   (b) Assume that for $t < l$, we have:
   $$v_j + \sum_{i<t} \alpha_i' b_j^i \leq \Big(v_k + \sum_{i<t} \alpha_i' b_k^i\Big)^d$$
   then we need to show that:
   $$v_j + \sum_{i<t+1} \alpha_i' b_j^i \leq \Big(v_k + \sum_{i<t+1} \alpha_i' b_k^i\Big)^d$$
   Set $v_j' = v_j + \sum_{i<t} \alpha_i' b_j^i$ and $v_k' = v_k + \sum_{i<t} \alpha_i' b_k^i$. Then it suffices to show that:
   $$\begin{aligned} v_j' + \alpha_t' b_j^t &\leq v_k'^d + \alpha_t' m \\ &\leq v_k'^d + \alpha_t' v_k \\ &\leq v_k'^d + \binom{d}{d-1} v_k'^{d-1} \alpha_t' b_k^t \leq (v_k' + \alpha_t' b_k^t)^d \end{aligned}$$
   where in the second and third inequalities we have used that since $k > l_*$ and $t \leq i_*$ we have that $b_k^l, b_k^t \geq 1$.

Thus, the trivial NP procedure that guesses all the coefficients $\alpha_1, \ldots, \alpha_n \leq m+1$ and accepts if and only if $\mathbf{x}^\mathbf{s} = \mathbf{a} + \sum_{i=1}^K \alpha_i \mathbf{b}^i$ respects the inequalities $x_j \leq x_k^d \in Q$ shows the problem can be decided in NP. □

The function $f(x, y, z, \ldots) = y^d - x$ is convex as discussed in Section 3. The constraints of the form $f(x, y, z, \ldots) \geq 0$ are called reverse convex in the operations research literature. To the best of our knowledge this is the first complexity result for conjunctions of reverse convex constraints over the integers.

Note that it is key that, thanks to the inductive argument, we can disregard the remaining α's after α_l. These α's would be detrimental for an inequality $x_j \leq x_k^d$ with $k < l_* < j$. However, in the general case, we could furthermore have linear inequalities $x_j \leq x_k x_m$ with $m < l_* < k, j$ and we cannot guarantee that the α's after α_l are superfluous. Furthermore, the inductive argument would fail in the case that $b_j^i > 0$ but $b_k^i = b_m^i = 0$.

5 Satisfiability of Monomial Inequalities with Solved Linear Constraints

In previous sections, we have presented decision procedures that leveraged insights on the structure of the set of solutions of linear constraints in order to find solutions to restricted families of non-linear inequalities. It is thus natural to ask how hard it is to check satisfiability of the non-linear part when given the set of solutions to the linear constraints as input. The answer to this question is mixed. On the one hand, we observe that from the results of [36], it follows that for a single more-than-monomial constraint, satisfiability with the Hilbert basis given as input can be decided in polynomial time. This is no longer true when given arbitrary more-than-monomial or less-than-monomial constraints.

Theorem 15. *The more-than-monomial and less-than-monomial problems are NP-hard even when the linear part of constraint is given as input.*

The proof is deferred to the appendix.

6 Logical Consequences

Theorems 12 and 14 can be used to establish an NP complexity bound on some fragments of theories of relational logic since an unconstrained d-ary relation R on a set $\mathcal{U}$ exists if and only if $|R| \leq |\mathcal{U}|^d$. Let's consider as in [50], the theory of QFBAPA enriched with unary functions of sets and their inverse and direct function images $f^{-1}[B] = \{y|\exists x.x \in B \land y = f(x)\}$ and $f[B] = \{y|\exists x.x \in B \land y = f(x)\}$. Let's also allow for a set variable B, to form the Cartesian product B^d of B iterated d times. Then the satisfiability of the formula $S = f^{-1}[P^d]$ is equivalent to the satisfiability of the non-linear constraint $|P|^d \leq |S|$. Similarly, the satisfiability of the formula $S = f[P^d]$ is equivalent to the satisfiability of the non-linear constraint $|S| \leq |P|^d$.

As a result, we obtain a fragment that is strictly more expressive than the language of QFBAPA. It enriches the language of Figure 1 with top-level constraints of the form $S = f^{-1}[P^d]$ (QFBAPA-Fun) or $S = f[P^d]$ (QFBAPA-InvFun). Note that one cannot mix both kinds of constraints since as remarked in the introduction this would express Hilbert's 10th problem and would thus yield an undecidable fragment.

Theorem 16. *Satisfiability of QFBAPA-Fun and QFBAPA-InvFun is in NP.*

7 Conclusion

Non-linear Diophantine constraints have been widely investigated in mathematical optimisation and automated reasoning. Despite the number of applications of prequadratic [1, 12, 20, 45, 50] and more general constraints [16, 21, 23, 25, 30, 49, 52] there exist few classes in the literature with low complexity bounds making them suitable for integration in satisfiability modulo theory solvers [6,7,9,11,14,46]. In this work, we prove an optimal bound for a subfamily of prequadratic Diophantine constraints. We show that these constraints are useful in analyzing the cardinality of cartesian powers, which can be used in fragments of Boolean algebra with function and inverse images. We have remarked that in the case of a single monomial constraint, the complexity is polynomial when given the Hilbert basis of the linear part. On the other hand, we have shown that with arbitrary monomial constraints the problem becomes NP-hard even if the Hilbert basis of the linear part is given. The key of our development is the normal form of Section 2.2.

In future work, we plan to investigate larger classes of (non-)convex and general prequadratic constraints.

References

[1] Marcelo Arenas, Wenfei Fan, and Leonid Libkin. On the Complexity of Verifying Consistency of XML Specifications. *SIAM Journal on Computing*, 38(3):841–880, January 2008. doi:10.1137/050646895.

[2] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach.* Cambridge University Press, Cambridge ; New York, 2009.

[3] Franz Baader. Expressive cardinality constraints on ALCSCC concepts. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, page 1123–1130, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3297280.3297390.

[4] Franz Baader, Bartosz Bednarczyk, and Sebastian Rudolph. Satisfiability checking and conjunctive query answering in description logics with global and local cardinality constraints. In *Proceedings of the 32nd International Workshop on Description Logics, Oslo, Norway, June 18-21, 2019*, volume 2373 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[5] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.

[6] Haniel Barbosa, Clark Barrett, Martin Brain, Gereon Kremer, Hanna Lachnitt, Makai Mann, Abdalrhman Mohamed, Mudathir Mohamed, Aina Niemetz, Andres Nötzli, Alex Ozdemir, Mathias Preiner, Andrew Reynolds, Ying Sheng, Cesare Tinelli, and Yoni Zohar. cvc5: A Versatile and Industrial-Strength SMT Solver. In *Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science, pages 415–442, Cham, 2022. Springer International Publishing. doi:10.1007/978-3-030-99524-9_24.

[7] Cristina Borralleras, Daniel Larraz, Enric Rodríguez-Carbonell, Albert Oliveras, and Albert Rubio. Incomplete SMT Techniques for Solving Non-Linear Formulas over the Integers. *ACM Transactions on Computational Logic*, 20(4):25:1–25:36, August 2019. doi:10.1145/3340923.

[8] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 13th printing edition, 2004.

[9] Aaron Bradley and Zohar Manna. *Calculus of computation: decision procedures with applications to verification.* Springer, Berlin, 2007.

[10] Stanley N. Burris. *Logic for Mathematics and Computer Science.* Prentice Hall, Upper Saddle River, N.J, 1st edition edition, August 1997.

[11] Alessandro Cimatti, Alberto Griggio, Ahmed Irfan, Marco Roveri, and Roberto Sebastiani. Incremental Linearization for Satisfiability and Verification Modulo Nonlinear Arithmetic and Transcendental Functions. *ACM Transactions on Computational Logic*, 19(3):1–52, July 2018. doi:10.1145/3230639.

[12] Claire David, Leonid Libkin, and Tony Tan. Efficient reasoning about data trees via integer linear programming. *ACM Transactions on Database Systems*, 37(3):1–28, August 2012. doi:10.1145/2338626.2338632.

[13] Leonardo de Moura and Nikolaj Bjorner. Generalized, efficient array decision procedures. In *2009 Formal Methods in Computer-Aided Design*, pages 45–52, Austin, TX, November 2009. IEEE. doi:10.1109/FMCAD.2009.5351142.

[14] Leonardo de Moura and Nikolaj Bjørner. Z3: An Efficient SMT Solver. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 4963, pages 337–340. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. Series Title: Lecture Notes in Computer Science. doi:10.1007/978-3-540-78800-3_24.

[15] Eric Domenjoud. Solving systems of linear diophantine equations: An algebraic approach. In *Mathematical Foundations of Computer Science 1991*, volume 520 of *Lecture Notes in Computer Science*, pages 141–150, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg. doi:10.1007/3-540-54345-7_57.

[16] Andreas Eggers, Evgeny Kruglov, Stefan Kupferschmid, Karsten Scheibler, Tino Teige, and Christoph Weidenbach. Superposition Modulo Non-linear Arithmetic. In Cesare Tinelli and Viorica Sofronie-Stokkermans, editors, *Frontiers of Combining Systems*, Lecture Notes in Computer Science, pages 119–134, Berlin, Heidelberg, 2011. Springer. doi:10.1007/978-3-642-24364-6_9.

[17] Friedrich Eisenbrand and Gennady Shmonin. Carathéodory bounds for integer cones. *Operations Research Letters*, 34(5):564–568, September 2006. doi:10.1016/j.orl.2005.09.008.

[18] Joachim von zur Gathen and Malte Sieveking. A Bound on Solutions of Linear Integer Equalities and Inequalities. *Proceedings of the American Mathematical Society*, 72(1):155–158, 1978. Publisher: American Mathematical Society. doi:10.1090/s0002-9939-1978-0500555-0.

[19] Seymour Ginsburg and Edwin H. Spanier. Semigroups, Presburger formulas, and languages. *Pacific Journal of Mathematics*, 16(2):285–296, January 1966. Publisher: Pacific Journal of Mathematics, A Non-profit Corporation. doi:10.2140/pjm.1966.16.285.

[20] Robert Givan, David McCallester, Carl Witty, and Dexter Kozen. Tarskian Set Constraints. *Information and Computation*, 174(2):105–131, May 2002. doi:10.1109/lics.1996.561313.

[21] Eitan M Gurari and Oscar H Ibarra. An NP-Complete Number-Theoretic Problem. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 205–215, San Diego California USA, May 1978. doi:10.1145/800133.804349.

[22] Richard J. Hillestad and Stephen E. Jacobsen. Reverse convex programming. *Applied Mathematics and Optimization*, 6(1):63–78, March 1980. doi:10.1007/BF01442883.

[23] Dejan Jovanović and Leonardo de Moura. Solving Non-linear Arithmetic. In *Automated Reasoning*, Lecture Notes in Computer Science, pages 339–354, Berlin, Heidelberg, 2012. Springer. doi:10.1007/978-3-642-31365-3_27.

[24] Ravindran Kannan and Clyde L. Monma. On the Computational Complexity of Integer Programming Problems. In *Optimization and Operations Research*, Lecture Notes in Economics and Mathematical Systems, pages 161–172, Berlin, Heidelberg, 1978. Springer. doi:10.1007/978-3-642-95322-4_17.

[25] Wong Karianto, Aloys Krieg, and Wolfgang Thomas. On Intersection Problems for Polynomially Generated Sets. In *Automata, Languages and Programming*, volume 4052 of *Lecture Notes in Computer Science*, pages 516–527, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. doi:10.1007/11787006_44.

[26] L. G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, January 1980. doi:10.1016/0041-5553(80)90061-0.

[27] Naoki Kobayashi and Hitoshi Ohsaki. Tree automata for non-linear arithmetic. In *Rewriting Techniques and Applications, 19th International Conference, RTA 2008, Hagenberg, Austria, July 15-17, 2008, Proceedings*, volume 5117 of *Lecture Notes in Computer Science*, pages 291–305. Springer, 2008. doi:10.1007/978-3-540-70590-1_20.

[28] Viktor Kuncak, Huu Hai Nguyen, and Martin Rinard. Deciding Boolean Algebra with Presburger Arithmetic. *Journal of Automated Reasoning*, 36(3):213–239, April 2006. doi:10.1007/s10817-006-9042-1.

[29] Viktor Kuncak and Martin Rinard. Towards Efficient Satisfiability Checking for Boolean Algebra with Presburger Arithmetic. In *Automated Deduction – CADE-21*, Lecture Notes in Computer Science, pages 215–230, Berlin, Heidelberg, 2007. Springer. doi:10.1007/978-3-540-73595-3_15.

[30] L. G. Khachiyan. Convexity and Complexity in Polynomial Programming. In *Proceedings of the International Congress of Mathematicians*, pages 1569–1577, Warszawa, August 1983. North-Holland.

[31] Loïc Pottier. Minimal solutions for linear diophantine systems: bounds and algorithms. In

Rewriting Techniques and Applications, volume 488, Como, Italy, April 1991. doi:10.1007/3-540-53904-2_94.

[32] Jurij V. Matijasevic. *Hilbert's tenth problem.* Foundations of computing. MIT Press, Cambridge, 1993.

[33] Robert Meyer. The Validity of a Family of Optimization Methods. *SIAM Journal on Control*, 8(1):41–54, February 1970. Publisher: Society for Industrial and Applied Mathematics. URL: https://epubs.siam.org/doi/abs/10.1137/0308003, doi:10.1137/0308003.

[34] Pawel Mielniczuk and Leszek Pacholski. Tarskian Set Constraints Are in NEXPTIME. In *Proceedings of the 23rd International Symposium on Mathematical Foundations of Computer Science*, MFCS '98, pages 589–596, Berlin, Heidelberg, August 1998. Springer-Verlag. doi:10.1007/BFb0055809.

[35] Wiesława T. Obuchowska. Unboundedness in reverse convex and concave integer programming. *Mathematical Methods of Operations Research*, 72(2):187–204, October 2010. doi:10.1007/s00186-010-0315-4.

[36] Shmuel Onn. *Nonlinear Discrete Optimization*, volume 13 of *EMS Zurich Lectures in Advanced Mathematics*. European Mathematical Society (EMS), September 2010.

[37] Christos H. Papadimitriou. On the complexity of integer programming. *Journal of the ACM*, 28(4):765–768, October 1981. doi:10.1145/322276.322287.

[38] R. J. Parikh. Language generating devices. Technical Report, Research Laboratory of Electronics (RLE) at the Massachusetts Institute of Technology (MIT), January 1961. Accepted: 2010-04-01T22:52:42Z. URL: https://dspace.mit.edu/handle/1721.1/53500.

[39] Ruzica Piskac. *Decision Procedures for Program Synthesis and Verification.* PhD thesis, EPFL, Lausanne, 2011. doi:10.5075/epfl-thesis-5220.

[40] Ruzica Piskac and Viktor Kunčak. Linear Arithmetic with Stars. In *Computer Aided Verification*, Lecture Notes in Computer Science, pages 268–280, Berlin, Heidelberg, 2008. Springer. doi:10.1007/978-3-540-70545-1_25.

[41] Ian Pratt-Hartmann. Complexity of the two-variable fragment with counting quantifiers. *J. Log. Lang. Inf.*, 14(3):369–395, 2005. doi:10.1007/s10849-005-5791-1.

[42] Ian Pratt-Hartmann. Logics with counting and equivalence. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 76:1–76:10. ACM, 2014. doi:10.1145/2603088.2603117.

[43] Ian Pratt-Hartmann. The two-variable fragment with counting and equivalence. *Math. Log. Q.*, 61(6):474–515, 2015. doi:10.1002/malq.201400102.

[44] Mojżesz Presburger. Über der Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchen die Addition als einzige Operation hervortritt. In *Comptes-Rendus du ler Congres des Mathematiciens des Pays Slavs*, Varsovie, 1929.

[45] Rodrigo Raya and Viktor Kunčak. NP Satisfiability for Arrays as Powers. In *Verification, Model Checking, and Abstract Interpretation*, Lecture Notes in Computer Science, pages 301–318, Cham, 2022. Springer International Publishing. doi:10.1007/978-3-030-94583-1_15.

[46] Yasser Shoukry, Pierluigi Nuzzo, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, George J. Pappas, and Paulo Tabuada. SMC: Satisfiability Modulo Convex Optimization. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, pages 19–28, Pittsburgh Pennsylvania USA, April 2017. ACM. doi:10.1145/3049797.3049819.

[47] Michael Sipser. *Introduction to the theory of computation.* Cengage Learning, Boston Mass, 3rd, edition edition, 2013.

[48] S. P. Tarasov and L. G. Khachiyan. Bounds of Solutions and Algorithmic Complexity of Systems of Convex Diophantine Inequalities. *Soviet Matematics Doklady*, 255(2):5, 1980.

[49] Gaoyan Xie, Zhe Dang, and Oscar H. Ibarra. A Solvable Class of Quadratic Diophantine Equations

with Applications to Verification of Infinite-State Systems. In *Automata, Languages and Programming*, volume 2719 of *Lecture Notes in Computer Science*, pages 668–680, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. doi:10.1007/3-540-45061-0_53.

[50] Kuat Yessenov, Ruzica Piskac, and Viktor Kunčak. Collections, Cardinalities, and Relations. In *Verification, Model Checking, and Abstract Interpretation*, volume 5944 of *Lecture Notes in Computer Science*, pages 380–395, Berlin, Heidelberg, 2010. Springer. doi:10.1007/978-3-642-11319-2_27.

[51] Yuri Matijasevich. Enumerable sets are diophantine. *Soviet Mathematics Doklady*, 11(2):5, 1970.

[52] Harald Zankl and Aart Middeldorp. Satisfiability of Non-linear (Ir)rational Arithmetic. In *Logic for Programming, Artificial Intelligence, and Reasoning*, Lecture Notes in Computer Science, pages 481–500, Berlin, Heidelberg, 2010. Springer. doi:10.1007/978-3-642-17511-4_27.

A Appendix: proof of the statements in Section 5

A.1 One Monomial Inequality

We start with the case where there is a single monomial inequality and the linear part has been solved in the normal form suggested, i.e. we have:

$$\begin{cases} x_k \geq x_j^l \\ \mathbf{x} = \mathbf{a} + \sum \alpha_i \mathbf{b^i} \end{cases}$$

If $\operatorname{supp}(j) \neq \operatorname{supp}(k)$ then we know there is a solution. If $\operatorname{supp}(j) = \operatorname{supp}(k)$ then, by the second observation in the proof of Theorem 12, a solution necessarily lies in the ball $B(0, m + 2nm^2 \log(4m))$. Theorem 3.12 in [36] shows that in contrast to Sections A.2 and A.3, this instance can be solved in polynomial time:

Theorem 17 (Onn [36])**.** *There is an algorithm that, given $A \in \mathbb{Z}^{m\times n}$, $\mathcal{G}(A), l, u \in \mathbb{Z}^n, b \in \mathbb{Z}^m$ and separable convex $f : \mathbb{Z}^n \to \mathbb{Z}$ presented by comparison oracle, solves in time polynomial in $\langle A, \mathcal{G}(A), l, u, b, \hat{f}\rangle$ the problem* $\min \{f(x) : x \in \mathbb{Z}^n, Ax = b, l \leq x \leq u\}$.

Here $\mathcal{G}(A)$ stands for the so-called Graver basis which is a generalisation of the notion of Hilbert basis for the non-positive orthants. On the other hand, $\hat{f}$ stands for the maximum of f over the compact domain $l \leq x \leq u$. The theorem guarantees that the minimisation problem can be solved in polynomial time in the size of the parameters. Since we are interested in the solution in a ball, the maximum of the function $\hat{f}$ is simply a constant and can be ignored. Then, we would minimise the function $f(x) = x_j^l - x_k$. If the minimum value is ≤ 0 then we accept, otherwise we reject.

A.2 More-Than-Monomial Constraints

Consider a family of more-than-monomial constraints:

$$\begin{cases} \{x_k \geq x_j^{d_i}\}_{i=1,\dots q, n_i \in \mathbb{N}, d_i \geq 2} \\ \mathbf{x} = \mathbf{a} + \sum \alpha_i \mathbf{b^i} \end{cases}$$

To show NP-hardness we reduce from the circuit satisfiability problem [2]:

Definition 18. *CKT-SAT is the decision problem which for a given n-input circuit C determines whether there exists $u \in \{0,1\}^n$ such that $C(u) = 1$.*

Theorem 19. *More-than-monomial is NP-hard.*

Proof. We reduce CKT-SAT to more-than-monomial. In order to ease the translation, we assume that the circuit to which the reduction is applied is given in terms of NAND gates. It is known that NAND gates are universal, that is, any circuit can be represented in terms of this operation. Since translating each basic gate requires only a constant number of NAND gates, one further observes that the translation of a Boolean circuit into an equivalent NAND-based circuit increases size by a constant multiplicative factor, which is irrelevant for complexity considerations.

First, we observe that we can encode each NAND gate with polynomially many more-than-monomial constraints.

Let $g : z = \neg(x \wedge y)$ be a NAND gate. We introduce four variables $\alpha_0, \alpha_1, \alpha_2, \alpha_3$. The index i of α_i translated to a two-digit binary number corresponds to each possible valuation of x, y. We add the equalities $x = \alpha_2 + \alpha_3, y = \alpha_1 + \alpha_3, z = \alpha_1 + \alpha_2 + \alpha_3$.

We impose for each $i, j \in \{0, 1, 2, 3\}$ $(i \neq j)$ the restriction that $\alpha_i + \alpha_j \leq 1$. This ensures that at most one coefficient α_i is set to one. This restriction can be enforced with more-than-monomial constraints by adding variables u_{ij}, v_{ij} with $i \neq j$ such that $u_{ij} = \alpha_i + \alpha_j, v = 3, u_{ij}^2 \leq v_{ij}$.

Similarly, we impose the restriction that $1 \leq \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3$. This ensures that at least one coefficient is satisfied. This restriction can be enforced by adding variables r, s such that $r = 1, s = \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3, r^2 \leq s$.

$$\begin{pmatrix} x \\ y \\ z \\ u_{01} \\ \vdots \\ u_{32} \\ v \\ r \\ s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 3 \\ 1 \\ 0 \end{pmatrix} + \alpha_0 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \alpha_1 \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{1}$$

In summary, the linear set of equation 1 together with the prequadratic constraints $r^2 \leq s, u_{ij}^2 \leq v$ where $i < j$ and $i, j \in \{0, \ldots, 3\}$ encode the operation of g.

Second, we encode the rest of the circuit. For each new gate, we add a new diagonal block to the step vectors. Each block repeats the pattern shown in equation 1.

We may reuse any of the variables x, y, z in other gates. To do so, we need to encode equality between two variables of the left hand side. Since we will later enforce that each variable is zero-one valued, this can be done using more-than-monomial constraints: to say that x and y are equal it suffices to impose that $x^2 \leq y$ and $y^2 \leq x$. In the zero-one valued case, this implies that $x = y$.

The last step of the transformation ensures that all variables, either those labelling wires in the original circuit or those added later, are zero-one valued. In particular, for the coefficients α_i of the linear set, we first introduce equations $t = \alpha_i$. Finally, we add the inequalities $x_i^2 \leq x_i$ for all the resulting variables.

The transformation can be clearly done in polynomial time and the correctness is ensured by construction. Thus, more-than-monomial is NP-hard even when the underlying linear set is explicitly given, as we wanted to show. □

A.3 Less-Than-Monomial Constraints

Now assume that we are given a family of monomials:

$$\begin{cases} \{x_j \leq x_k^{d_i}\}_{i=1,\ldots q, d_i \in \mathbb{N}, d_i \geq 2} \\ \mathbf{x} = \mathbf{a} + \sum g_i \mathbf{b^i} \end{cases}$$

Theorem 20. *Less-than-monomial is NP-hard.*

Proof. It suffices to modify slightly the construction above. To enforce $x_j \in \{0, 1\}$, it suffices to set $x_i = 1$ and $x_j \leq x_i^2$. To enforce $\alpha_i + \alpha_j \leq 1$ it suffices to write $u_{ij} = \alpha_i + \alpha_j, v = 1, u_{ij} \leq v^2$. To enforce that $1 \leq \alpha_0 + \alpha_1 + \alpha_3 + \alpha_3$ we simply set $r = 1, s = \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3, r \leq s^2$. □