Polynomial-Time Tractable Problems over the p-Adic Numbers

Manuel Bodirsky
Institut für Algebra, Technische Universität Dresden, Germany

Arno Fehm
Institut für Algebra, Technische Universität Dresden, Germany

Abstract

We study the computational complexity of fundamental problems over the p-adic numbers $\mathbb{Q}_p$ and the p-adic integers $\mathbb{Z}_p$. Guépin, Haase, and Worrell [9] proved that checking satisfiability of systems of linear equations combined with valuation constraints of the form $v_p(x) = c$ for $p \geq 5$ is NP-complete (both over $\mathbb{Z}_p$ and over $\mathbb{Q}_p$), and left the cases $p = 2$ and $p = 3$ open. We solve their problem by showing that the problem is NP-complete for $\mathbb{Z}_3$ and for $\mathbb{Q}_3$, but that it is in P for $\mathbb{Z}_2$ and for $\mathbb{Q}_2$. We also present different polynomial-time algorithms for solvability of systems of linear equations in $\mathbb{Q}_p$ with either constraints of the form $v_p(x) \leq c$ or of the form $v_p(x) \geq c$ for $c \in \mathbb{Z}$. Finally, we show how our algorithms can be used to decide in polynomial time the satisfiability of systems of (strict and non-strict) linear inequalities over $\mathbb{Q}$ together with valuation constraints $v_p(x) \geq c$ for several different prime numbers p simultaneously.

2012 ACM Subject Classification Theory of computation → Logic and verification; Theory of computation → Complexity theory and logic

Keywords and phrases p-adic numbers, existential theory, linear theory, constraint satisfaction, linear program feasibility, NP-hardness, polynomial-time algorithm

Digital Object Identifier 10.4230/LIPIcs.MFCS.2025.25

Related Version *Full Version*: `https://arxiv.org/abs/2504.13536`, `arXiv:2504.13536` [6]

Funding *Manuel Bodirsky*: The author received funding from the ERC (Grant Agreement no. 101071674, POCOCOP). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency.
Arno Fehm: The author was funded by the Deutsche Forschungsgemeinschaft (DFG) - 404427454

Acknowledgements The authors would like to thank Philip Dittmann and Pierre Touchard for helpful discussions regarding quantifier elimination for $\mathbb{Q}_p$.

1 Introduction

The satisfiability problem for systems of polynomial equations is an immensely useful computational problem; however, it has a quite bad worst-time complexity: it is NP-hard in arbitrary fields, undecidable over $\mathbb{Z}$ [14], not known to be decidable over $\mathbb{Q}$, and not known to be in NP for $\mathbb{R}$ [15]. In contrast, the satisfiability problem for systems of *linear* equations has a much better computational complexity: it can be solved in polynomial time over $\mathbb{R}$ and, equivalently, over $\mathbb{Q}$, and even over $\mathbb{Z}$ (see, e.g., [16]). It is therefore natural to search for meaningful extensions of the satisfiability problem for linear systems that retain some of the pleasant computational properties; in particular, extensions that remain in the complexity class P. It is also interesting to search for meaningful restrictions of the satisfiability problem for systems of polynomial equations that are no longer computationally hard.

50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025).
Editors: Paweł Gawrychowski, Filip Mazowiecki, and Michał Skrzypczak; Article No. 25; pp. 25:1–25:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

One of the well-studied expansions of linear systems is the expansion by linear *inequalities*. Note that $x \leq y$ can be expressed over $\mathbb{R}$ by $\exists z(x + z^2 = y)$ (and it can also be expressed over $\mathbb{Q}$ and $\mathbb{Z}$, but we then need a different formula), so this expansion can also be viewed as a restriction of the mentioned problem for systems of polynomial equations. The satisfiability problem for linear inequalities is known to be NP-complete over $\mathbb{Z}$, but remains in P over $\mathbb{Q}$ and $\mathbb{R}$ (e.g., via the ellipsoid method; see, e.g., [16]).

Other interesting, but less well-known expansions of the linear existential theory of $\mathbb{Z}$ and $\mathbb{Q}$ come from p-adic valuations v_p, for p a prime number: For $x \in \mathbb{Z}$, one defines $v_p(x) := \sup\{j : p^j | x\} \in \mathbb{N} \cup \{\infty\}$, and one extends this to $\mathbb{Q}$ by $v_p(\frac{a}{b}) := v_p(a) - v_p(b)$. The complexity of the satisfiability problem for systems of linear equalities combined with valuation constraints of the form $v_p(x) = c$ for $c \in \mathbb{Z}$ has been studied by Guépin, Haase, and Worrell [9]. Their results show that the problem over $\mathbb{Q}$ is in NP, even if the constants c are represented in binary and p is part of the input. This is remarkable, because for any $x = \frac{a}{b} \in \mathbb{Q}$ that satisfies $v_p(x) = c > 0$, the number a has exponential size in c, i.e., doubly exponential size in the input size. So we cannot simply guess and verify a solution in binary representation.

The results of Guépin, Haase, and Worrell are actually stated in a different setting: they phrase their result over the *p-adic numbers*. The p-adic valuation gives rise to a (non-archimedean) *absolute value*, defined for $x \in \mathbb{Q}$ by $|x|_p := p^{-v_p(x)}$. The *field of p-adic numbers* $\mathbb{Q}_p$ is the completion of $\mathbb{Q}$ with respect to $|\cdot|_p$, similarly as $\mathbb{R}$ is defined to be the completion of $\mathbb{Q}$ with respect to the standard absolute value. The *ring of p-adic integers* is the subring $\mathbb{Z}_p$ of $\mathbb{Q}_p$ with domain $\{x \in \mathbb{Q}_p \mid v_p(x) \geq 0\}$, where v_p denotes the natural extension of the p-adic valuation to $\mathbb{Q}_p$. Guépin, Haase, and Worrell [9] phrase their mentioned results as satisfiability problems over $\mathbb{Q}_p$; however, the problems are equivalent to the respective problems over $\mathbb{Q}$; see Proposition 6. They then use their algorithm to prove that the entire existential theory of $\mathbb{Q}_p$ in a suitable (linear) language is in NP.

Guépin, Haase, and Worrell moreover obtain some hardness results: they prove that the satisfiability problem for systems of linear equations over $\mathbb{Q}_p$ and over $\mathbb{Z}_p$ with valuation constraints of the form $v_p(x) = c$ is NP-hard for $p \geq 5$. They also state: *"While we believe it to be the case, it remains an open problem whether an NP lower bound can also be established for the cases* $p = 2, 3$*."* [9, Remark 23].

We solve this problem and prove that satisfiability is NP-complete in the case $p = 3$ for both $\mathbb{Q}_p$ and $\mathbb{Z}_p$. For $p = 2$, however, we prove containment in P. Interestingly, our algorithm can also cope with constraints of the form $v_p(x) \geq c$, even if p is larger than 2 (Theorem 18). We also find an algorithm that can test the satisfiability of linear systems for $\mathbb{Q}_p$ in the presence of constraints of the form $v_p(x) \leq c$ (Proposition 8); it is the combination of both upper and lower valuation bounds that makes the problem hard.

Our algorithm can also be used for the satisfiability problem for valuation constraints in combination with linear inequalities over $\mathbb{Q}$. We prove that the satisfiability of systems of (weak and strict) linear inequalities together with various valuation constraints, for instance of the form $v_p(x) \geq c$, can be decided in polynomial time (Theorem 31). We do allow valuation constraints for different primes in the input; we allow binary representations of all coefficients in the input. The proof uses the fact that linear programming is in P [16, Section 13], and the approximation theorem for finitely many inequivalent absolute values for $\mathbb{Q}$ ([13, Ch. XII, Thm. 1.2]).

Related Work. The computational complexity for satisfiability problems of semilinear expansions of linear inequalities over $\mathbb{Q}$ (equivalently: over $\mathbb{R}$) has been studied in [3]. The results there state that every expansion of the satisfiability problem for linear inequalities by other semilinear relations is NP-hard, unless all relations $R \subseteq \mathbb{Q}^n$ are *essentially convex*, i.e., have the property that for any two $a, b \in R$, all but finitely many rational points on the line segment between a and b are also contained in R; moreover, if all relations are essentially convex, then the satisfiability problem is in P [3, Theorem 5.2]. This result has later been generalised to expansions of linear equalities instead of inequalities [12]. Valuation constraints are clearly not essentially convex; however, they are also not semilinear, and not even semialgebraic, and hence are not covered by the results from [3] and from [12].

Different computational tasks for the p-adic numbers have been studied by Dolzmann and Sturm [5], and more recently by Haase and Mansutti [10]: they showed that whether a given system of linear equations with valuation constraints (where the valuation constraints in [10] are more expressive than the ones from [5], which are more expressive than ours) has a solution in $\mathbb{Q}_p$ for *all* prime numbers p is in coNExpTime.

Another recent results is a polynomial-time algorithm for the *dyadic feasibility problem* [1], which is the problem of testing the satisfiability of systems of linear inequalities over $\mathbb{Z}[\frac{1}{2}]$; it is unclear how to reduce this problem to the problems studied here and vice versa.

2 Preliminaries

We recall some well-known facts about p-adic numbers, see e.g. [8], and how we treat them from a logic and a computational point of view. We write $\mathbb{P} \subseteq \mathbb{N}$ for the set of all prime numbers and we let $p \in \mathbb{P}$.

2.1 $\mathbb{Q}_p$ and $\mathbb{Z}_p$

As $\mathbb{Q}_p$ is by definition the completion of $\mathbb{Q}$ with respect to the p-adic absolute value $|.|_p$, it is a metric space whose topology is the *p-adic topology*. The p-adic absolute value on $\mathbb{Q}_p$ gives rise to the p-adic valuation $v_p(x) = -\log_p |x|_p$. It satisfies the following basic properties:

▶ **Lemma 1.** *For all $a, b \in \mathbb{Q}_p$ we have*

- $v_p(a \cdot b) = v_p(a) + v_p(b)$, *and*
- $v_p(a + b) \geq \min(v_p(a), v_p(b))$, *with equality if* $v_p(a) \neq v_p(b)$.

The set $\mathbb{Z}_p = \{x \in \mathbb{Q}_p : v_p(x) \geq 0\}$ forms a subring of $\mathbb{Q}_p$ called the *ring of p-adic integers*. Its unique maximal ideal is generated by p, and $\mathbb{Z}_p/p^n\mathbb{Z}_p \cong \mathbb{Z}/p^n\mathbb{Z}$ for every $n \in \mathbb{N}$. This implies the following fact, which we will use several times:

▶ **Lemma 2.** *For every $x \in \mathbb{Q}_p \setminus \{0\}$ with $n = v_p(x)$ there exists a unique $i \in \{1, \ldots, p-1\}$ such that $v_p(x - ip^n) > n$.*

This further implies that every p-adic number has a unique *p-adic expansion*:

▶ **Lemma 3.** *Every $0 \neq x \in \mathbb{Q}_p$ with $n = v_p(x)$ is the limit (in the p-adic topology) of a unique series of the form $\sum_{i=n}^{\infty} x_i p^i$ with $x_i \in \{0, \ldots, p-1\}$ for every i.*

As usual, we let (see Figure 1)

$$\mathbb{Z}_{(p)} := \mathbb{Z}_p \cap \mathbb{Q} = \{x \in \mathbb{Q} : v_p(x) \geq 0\} = \left\{\frac{a}{b} : a, b \in \mathbb{Z}, p \nmid b\right\}.$$

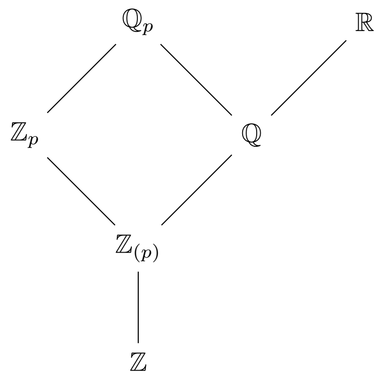

Figure 1 Inclusions between the number domains studied in this article.

2.2 The structure $\mathfrak{Q}_p$

It will be convenient for some of our results and proofs to take a logic perspective on the p-adic numbers; for an introduction to first-order logic, see [11]. A *signature* is a set τ of relation and function symbols, each equipped with an *arity*, which is a natural number. A *(first-order) structure* $\mathfrak{S}$ of signature τ consists of a set (the domain, typically denoted by the corresponding capital roman letter S), a function $f^{\mathfrak{S}} \colon S^k \to S$ for each function symbol $f \in \tau$ of arity $k \in \mathbb{N}$ (the case $k = 0$ is allowed; in this case, we refer to f as a *constant symbol*), and a relation $R^{\mathfrak{S}} \subseteq S^k$ for each relation symbol $R \in \tau$ of arity k; we then say that f *denotes* $f^{\mathfrak{S}}$, and R *denotes* $R^{\mathfrak{S}}$.

A *reduct* of $\mathfrak{S}$ is a structure obtained from $\mathfrak{S}$ by taking a subset of the signature. If $\mathfrak{R}$ is a reduct of $\mathfrak{S}$, then $\mathfrak{S}$ is called an *expansion* of $\mathfrak{R}$. A *substructure* of $\mathfrak{S}$ is a structure $\mathfrak{S}'$ with the same signature τ as $\mathfrak{S}$ and domain $S' \subseteq S$ such that for every function symbol $f \in \tau$ of arity k, the function $f^{\mathfrak{S}'}$ is the restriction of $f^{\mathfrak{S}}$ to $(S')^k$, and for every relation symbol $R \in \tau$ of arity k, the relation $R^{\mathfrak{S}'}$ equals $R^{\mathfrak{S}} \cap (S')^k$.

A *first-order τ-formula* is a formula built from first-order quantifiers $\forall, \exists$, Boolean connectives $\wedge, \vee, \neg$, and atomic formulas that are built from variables, the equality symbol $=$, and the symbols from τ in the usual way; for a proper definition, we refer to any standard introduction to mathematical logic or model theory, such as [11].

▶ Remark 4. Often when p-adic numbers are treated from a logic perspective, they are introduced as "two-sorted structures", with one sort for the p-adic numbers and one sort for the values, i.e., $\mathbb{Z} \cup \{\infty\}$, and a function symbol v for the valuation. For our purposes, however, usual first-order structures (as introduced above) are sufficient.

We work with the structure $\mathfrak{Q}_p$ which has the domain $\mathbb{Q}_p$ and the signature

$$\{+, 1\} \cup \{\leq^p_c, \geq^p_c, =^p_c, \neq^p_c, \mid c \in \mathbb{Z}\},$$

where

- $+$ is a binary function symbol that denotes the addition operation of p-adic numbers as introduced above;
- 1 is a constant symbol which denotes $1 \in \mathbb{Z}_{(p)} = \mathbb{Q}_p \cap \mathbb{Z}$ as introduced above;
- $\leq^p_c$ is a unary relation symbol that denotes the unary relation $\{x \in \mathbb{Q}_p \mid v_p(x) \leq c\}$; $\geq^p_c$, $=^p_c$, and $\neq^p_c$ are defined analogously.

Sometimes, we specify structures as tuples; e.g., we write

$$\mathfrak{Q}_p = (\mathbb{Q}_p; +, 1, (\leq^p_c)_{c\in\mathbb{Z}}, (\geq^p_c)_{c\in\mathbb{Z}}, (=^p_c)_{c\in\mathbb{Z}}, (\neq^p_c)_{c\in\mathbb{Z}})$$

and do not distinguish between function and relation symbols and the respective functions and relations. Atomic formulas that are built from the relations $\leq^p_c$, $\geq^p_c$, $=^p_c$, and $\neq^p_c$ will be called *valuation constraints*. For $c \in \mathbb{Z}$, we also use the symbols $<^p_c$ as a shortcut for $\leq^p_{c-1}$, and $>^p_c$ as a shortcut for $\geq^p_{c+1}$.

2.3 Primitive Positive Formulas and CSPs

A formula is called *primitive positive* if it is of the form

$$\exists x_1, \dots, x_n(\psi_1 \wedge \dots \wedge \psi_m)$$

where $\psi_1, \dots, \psi_m$ are atomic. In *primitive existential* formulas, $\psi_1, \dots, \psi_m$ are allowed to be negated atomic formulas as well, and *existential* formulas are disjunctions of primitive existential formulas. We use the concepts of primitive positive (and primitive existential, and existential) sentences, theories, definitions, definability, etc., as in the case of first-order logic (see, e.g., [11]), but restricting to primitive positive (primitive existential, and existential) formulas.

The computational problem of deciding the truth of a given primitive positive sentence φ in a fixed structure $\mathfrak{S}$ is called the *constraint satisfaction problem (CSP)* of $\mathfrak{S}$. We refer to the quantifier-free part $\psi_1 \wedge \dots \wedge \psi_m$ of φ as the *instance* of CSP($\mathfrak{S}$) (i.e., the existential quantifiers will be left implicit), and a satisfying assignment to the variables will also be called a *solution* to φ.

If the signature of $\mathfrak{S}$ is infinite, then the computational problem is not yet well-defined, because we still have to specify how to represent the symbols from the signature in the input; the choice of the representation can have an impact on the complexity of the CSP. For the structure $\mathfrak{Q}_p$ introduced above, a natural representation is to represent the relation symbols $\leq^p_c$, $\geq^p_c$, $=^p_c$, and $\neq^p_c$ by the binary encoding of $p \in \mathbb{P}$ and $c \in \mathbb{Z}$. Note that $v_p(x) \leq c$ holds if and only if $v_p(x) \leq v_p(p^c)$. It will turn out that our polynomial-time algorithms can handle the constants c stored in binary (which makes p^c a doubly exponentially large number). The hardness results, however, always make use of only finitely many symbols in the signature, and hence hold independently from the choice of the representation. We will therefore allow binary representations for the values c in the valuation constraints, since this allows the strongest formulations of our results.

We will determine the computational complexity of CSP($\mathfrak{S}$) for all reducts of $\mathfrak{Q}_p$ (Theorem 27 and 28).

2.4 Primitive positive interpretations

Primitive positive interpretations can be used to obtain complexity reductions between CSPs. For $d \geq 1$, a d-dimensional *primitive positive interpretation* of a structure $\mathfrak{A}$ in a structure $\mathfrak{B}$ is given by a partial function I from B^d to A such that the preimages under I of the following sets are primitively positively definable in $\mathfrak{B}$:

- A and the equality relation $=_A$ on A,
- each relation of $\mathfrak{A}$, and
- each graph of a function of $\mathfrak{A}$.

▶ **Lemma 5** (see, e.g., [2, Theorem 3.1.4]). *Let $\mathfrak{A}$ be a structure with a finite signature and a primitive positive interpretation in a structure $\mathfrak{B}$. Then $\mathfrak{B}$ has a reduct $\mathfrak{B}'$ with a finite signature such that there is a polynomial-time reduction from* $\mathrm{CSP}(\mathfrak{A})$ *to* $\mathrm{CSP}(\mathfrak{B}')$.

3 $\mathbb{Q}$ versus $\mathbb{Q}_p$

Note that the structure $\mathfrak{Q}_p$ has a substructure with domain $\mathbb{Q}$. All our algorithms in Section 4 and hardness proofs in Section 5 can be stated equivalently over the uncountable field $\mathbb{Q}_p$ or over $\mathbb{Q}$. This is possible due to the following fact, which is a consequence of a result of Weispfenning [17].

▶ **Proposition 6.** *For every $p \in \mathbb{P}$, the structure $\mathfrak{Q}_p$ and its substructure with domain $\mathbb{Q}$ have the same first-order theory.*

Proof. Let τ be the signature $\{+, \cdot, 0, 1, \pi, \mathrm{div}\}$ where π is a constant symbol and div is a binary relation symbol. Weispfenning [17] introduces a certain first-order τ-theory, which he calls T_{DVF_p}; both $\mathbb{Q}$ and $\mathbb{Q}_p$ give rise to models of T_{DVF_p} if π is interpreted as p and a div b if and only if $v_p(a) < v_p(b)$. He then proves that T_{DVF_p} admits quantifier elimination for *linear* formulas [17, Theorem 3.6]. That is, every σ-formula, for $\sigma := \{+, 0, 1, \pi, \mathrm{div}\}$ (where the symbol for multiplication is missing, which is why these formulas are called "linear"), is over T_{DVF_p} equivalent to a quantifier-free σ-formula. Clearly, every atomic formula in the signature of $\mathfrak{Q}_p$ can be defined by a σ-formula over $\mathbb{Q}_p$. Let φ be a first-order sentence in the signature of $\mathfrak{Q}_p$, and let φ' be the first-order σ-sentence obtained from φ by replacing all atomic formulas by their defining σ-formula. Then φ' is either equivalent to $0 = 0$ over T_{DVF_p} or it is equivalent to $0 = 1$ over T_{DVF_p}. It follows that either both $\mathfrak{Q}_p$ and its substructure with domain $\mathbb{Q}$ satisfy φ, or both $\mathfrak{Q}_p$ and its substructure with domain $\mathbb{Q}$ satisfy $\neg\varphi$, which is what we wanted to show. ◀

▶ **Corollary 7.** *For each $p \in \mathbb{P}$, the existential theory of $\mathfrak{Q}_p$ and the existential theory of the expansion of $(\mathbb{Q}; +, 1)$ by all relations of the form $\leq_c^p$, $\geq_c^p$, $=_c^p$ and $\neq_c^p$, for $c \in \mathbb{Z}$, are in NP.*

Proof. By Proposition 6, these two existential theories are equal, so the claim follows from [9, Proposition 21], where it is proven that the existential theory of $\mathbb{Q}_p$ in a more expressive language is in NP. ◀

4 Algorithms

We first discuss how to measure the size of input instances to the computational problems studied in this text. For $a, b \in \mathbb{N} \setminus \{0\}$ coprime, define $h(\pm\frac{a}{b}) := 1 + \log|a| + \log|b|$, and define $h(0) := 1$. Occasionally we might allow special coefficients like ∞ or $-\infty$; we set $h(\infty) = h(-\infty) := 1$. For matrices $A_1, \ldots, A_r$ with coefficients in $\mathbb{Q}$ we let

$$C(A_1, \ldots, A_r) := s + \sum_{k=1}^{r} \sum_{i,j} h(a_{kij}),$$

where s is the maximal number of rows or columns of one of the $A_k = (a_{kij})_{i,j}$. This is our measure of size of a computational problem that is given by a set of rational matrices. A rational number p in the input is interpreted as the matrix $p \in \mathbb{Q}^{1\times 1}$, and a finite set $D = \{d_1, \ldots, d_r\} \subseteq \mathbb{Q}$ is interpreted as the matrix $D = (d_1, \ldots, d_r) \in \mathbb{Q}^{1\times r}$. For example, the input size of the algorithm in Proposition 8 below is $C(A, b, p, c, D_1, \ldots, D_n)$.

We now present two algorithms. The first one, essentially for constraints of the form $v_p(x) \leq c$, is straightforward. The second one, mainly for constraints of the form $v_p(x) \geq c$, is more involved. In both settings, among all such valuation constraints on the same variable, there is a most restrictive one, which can easily be identified (in polynomial time), and therefore our algorithms are only formulated for one valuation constraint of the form $v_p(x) \leq c$ (or of the form $v_p(x) \geq c$) per variable.

▶ **Proposition 8.** *There is a polynomial time algorithm that decides, given $m, n \in \mathbb{N}$, $p \in \mathbb{P}$, $c \in (\mathbb{Z} \cup \{\infty\})^n$, $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$, and finite sets $D_1, \ldots, D_n \subseteq \mathbb{Z}$, whether there exists $x \in \mathbb{Q}^n$ with $Ax = b$ such that $v_p(x_j) \leq c_j$ and $v_p(x_j) \notin D_j$ for $j = 1, \ldots, n$.*

Proof. Let $L := \{x \in \mathbb{Q}^n : Ax = b\}$ be the solution space of the system of linear equations. If $L = \emptyset$, the algorithm outputs NO. Otherwise write

$$L = \left\{ y_0 + \sum_{k=1}^{d} \lambda_k y_k : \lambda_1, \ldots, \lambda_d \in \mathbb{Q} \right\} \tag{4.1}$$

with $y_1, \ldots, y_d \in \mathbb{Q}^n$ linearly independent. One can check whether $L = \emptyset$ and otherwise compute such $d \in \mathbb{N}$ and $y_0, \ldots, y_d$ in polynomial time: It is possible to compute one solution $y_0 \in \mathbb{Q}^n$ of $Ax = b$ in polynomial time [16, Corollary 3.3a]. Moreover, we can transform A by elementary row operations into a matrix A' in row echelon form in polynomial time [16, Theorem 3.3], and from A' we can read off the rank d of A and a basis $y_1, \ldots, y_d$ of

$$\{x \in \mathbb{Q}^n : Ax = 0\} = \{x \in \mathbb{Q}^n : A'x = 0\}.$$

If $c_j = \infty$ let $C_j = (\mathbb{Z} \cup \{\infty\}) \setminus D_j$, otherwise let $C_j = (-\infty, c_j] \setminus D_j$, so that the algorithm has to decide whether there exists $x \in L$ with $v_p(x_j) \in C_j$ for every j. If for some j we have that $v_p(y_{0,j}) \notin C_j$ and $y_{k,j} = 0$ for every $k = 1, \ldots, d$, then every $x \in L$ satisfies $v_p(x_j) = v_p(y_{0,j}) \notin C_j$, and the algorithm outputs NO. Otherwise, the algorithm outputs YES. To see that this is the correct answer, assume now that for every j we have $v_p(y_{0,j}) \in C_j$ or $y_{k,j} \neq 0$ for some k. Let $c'_j = \sup(\mathbb{Z} \setminus C_j) \in \mathbb{Z} \cup \{\infty\}$, where we set $c'_j := \infty$ if $C_j = \mathbb{Z} \cup \{\infty\}$, and let

$$e := \max\{|v_p(y_{k,j})| : k = 0, \ldots, d; j = 1, \ldots, n; y_{k,j} \neq 0\} + \max\{0, -c'_1, \ldots, -c'_n\} + 1.$$

We claim that

$$x := y_0 + \sum_{k=1}^{d} p^{-2ke} y_k$$

is a solution to all the constraints. For each j let

$$K_j = \{k \in \{0, \ldots, d\} : y_{k,j} \neq 0\}.$$

If $K_j \setminus \{0\} = \emptyset$, then, by our assumption, $v_p(x_j) = v_p(y_{0,j}) \in C_j$. Otherwise,

$$-e(2k+1) = -2ke - e < v_p(p^{-2ke} y_{k,j}) < -2ke + e = -e(2k-1)$$

for every $k \in K_j$, so that the $v_p(p^{-2ke} y_{k,j})$ for $k \in K_j$ are pairwise distinct, and therefore, with $k_j := \max K_j$,

$$v_p(x_j) = v_p\Big(\sum_{k=0}^{k_j} p^{-2ke} y_{k,j}\Big) = -2k_j e + v_p(y_{k_j,j}) < c'_j$$

by the choice of e. This shows in particular that $v_p(x_j) \in C_j$, as required. ◀

▶ Remark 9. We might not be able to compute a solution in the usual binary representation, as already for the single constraint $v_p(x) \leq c$ the smallest solution (with respect to the p-adic absolute value $|x|_p := p^{-v_p(x)}$) is p^{-c}. The algorithm not only works for the p-adic valuation on $\mathbb{Q}$ but for arbitrary so-called discrete valuations on a computable field K in which a solution of a given linear equation, a basis for the solution space of a homogeneous linear equation, and the valuation of an element can be computed; the resulting algorithm has a polynomial running time if these computations can be performed in polynomial time.

For our second algorithm we need some preparations. As the algorithm achieves a stronger result, we just mention without proof that the usual Hermite normal form allows to check in polynomial time whether $Ax = b$ has a solution $x \in \mathbb{Z}_{(p)}^n$ (see, e.g., [16, Chapter 5]). However, already checking for solutions x with $x_j \in \mathbb{Z}_{(p)}$ for $1 \leq j \leq r$ and $x_j \in \mathbb{Q}$ for $r+1 \leq j \leq n$ requires new ideas. Also, if we want to allow constraints of the form $v_p(x_j) \geq c_j$ rather than just $v_p(x_j) \geq 0$, one could replace x_j by $x_j p^{-c_j}$, but only as long as p^{c_j} is polynomial in the input size. This would be the case if the c_j would be coded in unary, but if the c_j are coded in binary, as is our convention (see above), replacing x_j by $x_j p^{-c_j}$ will blow up the coefficients of the linear equation exponentially. We therefore do *not* replace x_j by $x_j p^{-c_j}$ but instead do some extra bookkeeping, exploiting the fact that although we might not be able to compute finite sums of elements of the form $x_j p^{c_j}$ in polynomial time, we can at least compute their valuations.

▶ **Lemma 10.** *There is a polynomial-time algorithm which, given $p \in \mathbb{P}$, $n \in \mathbb{N}$, and pairs $(a_1, c_1), \ldots, (a_n, c_n) \in \mathbb{Q} \times \mathbb{Z}$, computes $v_p(\sum_{i=1}^n a_i p^{c_i}) \in \mathbb{Z} \cup \{\infty\}$.*

Proof. First remove all (a_i, c_i) with $a_i = 0$ from the list. If $n = 0$ then output ∞. Replace each (a_i, c_i) by $(a_i p^{-v_p(a_i)}, c_i + v_p(a_i))$ to assume that $v_p(a_i) = 0$. Let $c = \min_i c_i$. If there exists a unique i_0 with $c = c_{i_0}$, then output $v_p(\sum_i a_i p^{c_i}) = c$. Otherwise assume without loss of generality that $c = c_1 = c_2$. Then $a_1 p^{c_1} + a_2 p^{c_2} = (a_1 + a_2) p^c$. Remove (a_1, c_1) and (a_2, c_2) from the list and append $(a_1 + a_2, c)$. Repeating this process will terminate after at most n steps. ◀

We also need a certain row echelon form. Before we give the definition, we present two motivating examples.

▶ **Example 11.** Suppose we want check whether a linear equation

$$a_1 x_1 + \cdots + a_n x_n = b \tag{4.2}$$

with $a_1, \ldots, a_n, b \in \mathbb{Q}$ has a solution $x \in \mathbb{Q}^n$ with $v_p(x_j) \geq 0$ for every $j \in \{1, \ldots, n\}$. Such an x exists if and only if $v_p(b) \geq \min_j v_p(a_j)$: For any such x,

$$\begin{aligned} v_p(b) = v_p(a_1 x_1 + \cdots + a_n x_n) &\geq \min\{v_p(a_1) + v_p(x_1), \ldots, v_p(a_n) + v_p(x_n)\} \\ &\geq \min_j v_p(a_j), \end{aligned}$$

and conversely, if $v_p(a_{j_0}) \leq v_p(b)$ for some j_0, we can let $x_{j_0} := a_{j_0}^{-1} b$ and set the other x_j to 0 (unless $a_{j_0} = 0$, in which case $b = 0$ and we can let $x = 0$).

▶ **Example 12.** Suppose we are given a nonempty set $X \subseteq \mathbb{Z}_{(p)}^{n-1}$ and want to check whether for some $(x_2, \ldots, x_n) \in X$ there exists $x_1 \in \mathbb{Z}_{(p)}$ satisfying (4.2). As long as $a_1 \neq 0$, we can solve for x_1 and obtain

$$x_1 = a_1^{-1}\big(b - \sum_{j=2}^n a_j x_j\big).$$

However, computing $v_p(x_1)$ can be difficult from just the values $v_p(x_j)$ for $j = 2, \dots, n$, since we are only guaranteed

$$v_p\Big(a_1^{-1}(b - \sum_{j=2}^{n} a_j x_j)\Big) \geq \min\{v_p(b) - v_p(a_1), \min_{j=2,\dots,n}(v_p(a_j) - v_p(a_1) + v_p(x_j))\}$$

and it can happen that the inequality is strict. The right hand side is certainly nonnegative as long as $v_p(a_1) \leq v_p(b)$ and $v_p(a_1) \leq v_p(a_j)$ for every j. And in fact, when $v_p(a_1) \leq v_p(a_j)$ for every j, the condition $v_p(a_1) \leq v_p(b)$ is also necessary for the left hand side to be nonnegative: If $v_p(a_1) > v_p(b)$, then $v_p(b) - v_p(a_1) < 0$ but $v_p(a_j) - v_p(a_1) + v_p(x_j) \geq 0$ for every j, so the inequality is actually an equality. Therefore, as long as a_1 has minimal valuation among the a_i, for any $(x_2, \dots, x_n) \in X$ there exists $x_1 \in \mathbb{Z}_{(p)}$ satisfying (4.2) if and only if $v_p(a_1) \leq v_p(b)$. This criterion easily generalizes to systems of several equations $Ax = b$ where A is in row echelon form and each pivot element has minimal valuation in its row. This is what Definition 13 below expresses in the special case of the function $f(a, j) = v_p(a)$.

▶ **Definition 13.** *A* pivot function *is a function*

$$f : \mathbb{Q} \times \mathbb{N} \to \mathbb{Q} \cup \{\infty, -\infty\}$$

such that $f(a, j) = \infty$ if and only if $a = 0$. For a pivot function f, we say that a matrix $A = (a_{ij})_{i,j} \in \mathbb{Q}^{m \times n}$ is in f-minimal row echelon form *if the following two conditions are satisfied.*

(a) *A is in row echelon form, i.e., setting $j_i := \inf\{j : a_{ij} \neq 0\}$ for $i \in \{1, \dots, n\}$, there exists $k \in \{0, \dots, m\}$ such that $j_1 < \dots < j_k < j_{k+1} = \dots = j_m = \infty$.*

(b) *Each pivot element a_{i,j_i} of A minimizes f within its row in the sense that for each $i \in \{1, \dots, k\}$,*

$$f(a_{ij_i}, j_i) = \min\{f(a_{ij}, j) : j = j_i, \dots, n\}.$$

▶ **Example 14.** To explain why we need more general functions f than just $f(a, j) = v_p(a)$, suppose we replace the conditions $v_p(x_j) \geq 0$ in Example 12 by $v_p(x_j) \geq c_j$ for some c_j. Rewriting this as $v_p(x_j p^{-c_j}) \geq 0$ we see that we could instead consider the matrix $A' = (a'_{ij})_{i,j}$ given by $a'_{ij} = a_{ij} p^{c_j}$ and apply the criterion from Example 12. However, the numbers p^{c_j} have exponential representation size. This can be avoided by replacing the condition that each pivot element $a_{ij_i} p^{c_{j_i}}$ of A' minimizes the function v_p within its row by the condition that each pivot element a_{ij_i} of A minimizes the function $f(a_{ij}, j) := v_p(a_{ij}) + c_j$ within its row, where the second argument indicates the column.

We write $\mathrm{GL}_m(\mathbb{Q})$ for the general linear group of degree m over the field $\mathbb{Q}$, i.e., the group of all invertible matrices in $\mathbb{Q}^{m \times m}$. If $\sigma \in S_n$ is a permutation, then $P_\sigma = (\delta_{i,\sigma(i)})_{i,j} \in \mathrm{GL}_n(\mathbb{Q})$ denotes the corresponding permutation matrix. For a pivot function f and $\sigma \in S_n$, we write f_σ for the pivot function given by

$$f_\sigma(a, j) := \begin{cases} f(a, \sigma^{-1}(j)) & \text{if } j \in \{1, \dots, n\} \\ f(a, j) & \text{otherwise.} \end{cases}$$

If S is a set, then S^* denotes the set of non-empty words over the alphabet S, i.e., the set of finite sequences of elements of S.

▶ **Lemma 15.** *Let* $f\colon \mathbb{Q} \times \mathbb{N} \times (\mathbb{Z} \cup \{-\infty\})^* \to \mathbb{Q} \cup \{\infty, -\infty\}$ *and assume that for each* $c \in (\mathbb{Z} \cup \{-\infty\})^*$*, the map* f_c *defined by* $(a, j) \mapsto f(a, j, c)$ *is a pivot function. For every* $m, n \in \mathbb{N}$*,* $A \in \mathbb{Q}^{m \times n}$*, and* $c \in (\mathbb{Z} \cup \{-\infty\})^*$ *there exist* $U \in \mathrm{GL}_m(\mathbb{Q})$ *and* $\sigma \in S_n$ *such that* UAP_σ *is in* $(f_c)_\sigma$*-minimal row echelon form. If* f *is computable in polynomial time, then such* U *and* P_σ *can be computed in polynomial time.*

Proof. We describe how to get U and P_σ in terms of elementary row and column operations, where the only elementary column operations allowed are swapping two columns. If $A = 0$, then we are done. Otherwise, possibly swap two rows to assume that $a_{1j} \neq 0$ for some j. Choose $k \in \{1, \ldots, n\}$ such that $f_c(a_{1k}, k) = \min\{f_c(a_{1j}, j) : j = 1, \ldots, n\}$ (which implies in particular that $a_{1k} \neq 0$, since $f_c(0, k) = \infty$ by assumption). If $k \neq 1$, then swap the first column with the k-th column. Add multiples of the first row to the other rows to achieve that $a_{i1} = 0$ for every $i > 1$. Reduce the fractions in the entries of the matrix. Now take the $(m-1) \times (n-1)$-submatrix with rows $i = 2, \ldots, m$ and columns $j = 2, \ldots, n$, and iterate (extending each of the following row and column operations to the whole matrix). It is well-known that the representation size of the involved numbers stays polynomial (see, e.g., [16, Theorem 3.3]). This process terminates after at most $\max\{m, n\}$ steps, and the resulting matrix is of the desired form. ◀

In the following, if $x \in \mathbb{Q}^n$ and $c \in (\mathbb{Z} \cup \{-\infty\})^n$, we will write $v_p(x) \geq c$ if $v_p(x_j) \geq c_j$ for every $j \in \{1, \ldots, n\}$.

▶ Remark 16. Note that if $B = UAP_\sigma$ for some $U \in \mathrm{GL}_m(\mathbb{Q})$ and $\sigma \in S_n$, then $Ax = b$ has a solution $x \in \mathbb{Q}^n$ such that $v_p(x) \geq c$ if and only if $By = Ub$ has a solution $y \in \mathbb{Q}^n$ such that $v_p(y) \geq P_\sigma^{-1} c$ (the map $x \mapsto P_\sigma^{-1} x$ is a bijection between the solutions to the first system and the solutions to the second system).

The following result allows constraints of the form $v_p(x) \geq c$.

▶ **Theorem 17.** *There is a polynomial-time algorithm that decides, given* $m, n \in \mathbb{N}$*,* $p \in \mathbb{P}$*,* $c \in (\mathbb{Z} \cup \{-\infty\})^n$*,* $A \in \mathbb{Q}^{m \times n}$*, and* $b \in \mathbb{Q}^m$*, whether there exists* $x \in \mathbb{Q}^n$ *with* $Ax = b$ *such that* $v_p(x) \geq c$*.*

In the case $p = 2$, we can additionally treat constraints of the form $v_2(x) = c$. The tuple δ encodes which constraint applies to which variable. The following is a generalisation of Theorem 17.

▶ **Theorem 18.** *There is a polynomial-time algorithm that decides, given* $m, n \in \mathbb{N}$*,* $p \in \mathbb{P}$*,* $c \in (\mathbb{Z} \cup \{-\infty\})^n$*,* $\delta \in \{0, 1\}^n$*,* $A \in \mathbb{Q}^{m \times n}$*, and* $b \in \mathbb{Q}^m$*, whether there exists* $x \in \mathbb{Q}^n$ *with* $Ax = b$ *such that* $v_p(x) \geq c$ *and, in the case* $p = 2$*,* $\delta_j = 1$*, and* $c_j \neq -\infty$*, also* $v_p(x_j) = c_j$*.*

Proof. We can assume that if $\delta_j = 1$ for some j, then $p = 2$ and $c_j \neq -\infty$. Define the pivot function

$$f(a, j) := v_p(a) + c_j + \frac{\delta_j}{2},$$

where we use the convention $\infty + (-\infty) := \infty$. Clearly, f is computable in polynomial time (as a function of a, j, p and c and δ). By Lemma 15 we can compute U and P_σ in polynomial time such that UAP_σ is in f_σ-minimal row echelon form. We may replace A by UAP_σ, b by Ub, c by $P_\sigma^{-1} c$, and δ by $P_\sigma^{-1} \delta$ (adapting the idea from Remark 16 appropriately in the case $p = 2$), and henceforth assume without loss of generality that $\sigma = \mathrm{id}$ and that A is already in f-minimal row echelon form.

Let k and $j_1, \dots, j_k$ be as in Definition 13. The algorithm then outputs YES if

1. $b_i = 0$ for every $i \in \{k+1, \dots, m\}$, and
2. $v_p(a_{ij_i}) + c_{j_i} + \delta_{j_i} \le v_p(b_i - \sum_{j \ge j_i} \delta_j a_{ij} p^{c_j})$ for every $i \in \{1, \dots, k\}$,

and otherwise it outputs NO. Note that Condition (2) can be checked in polynomial time by Lemma 10. Since A is in row echelon form, Condition (1) holds if and only if there exists $x \in \mathbb{Q}^n$ with $Ax = b$. To see that the algorithm gives the correct answer, we have to show that (1) and (2) holds if and only if there exists $x \in \mathbb{Q}^n$ with $Ax = b$, $v_p(x) \ge c$ and $v_p(x_j) = c_j$ for every j with $\delta_j = 1$.

For the forward direction, we assume that (1) and (2) hold and construct x as follows. For each $j \in \{1, \dots, n\} \setminus \{j_1, \dots, j_k\}$, let $x_j := p^{c_j}$ if $c_j \in \mathbb{Z}$, and otherwise let $x_j := 0$. For $i = k, \dots, 1$ define x_{j_i} iteratively by

$$x_{j_i} := a_{ij_i}^{-1} \cdot \Big(b_i - \sum_{j > j_i} a_{ij} x_j \Big). \tag{4.3}$$

Since A is in row echelon form, the so constructed x satisfies $Ax = b$. Moreover, for each $j \notin \{j_1, \dots, j_k\}$ one has by definition that $v_p(x_j) \ge c_j$ and $v_p(x_j) = c_j$ if $\delta_j = 1$, and one needs to show that the latter conditions also hold for $j \in \{j_1, \dots, j_k\}$, using that A is in f-minimal row echelon form.

The full proof can be found in the arXiv-version of the paper [6]. ◀

5 NP-hardness and reductions

For a set A and $a \in A$, we use $\neq_a$ as a relation symbol for the unary relation $A \setminus \{a\}$, and later write $x \neq a$ instead of $\neq_a(x)$.

▶ **Lemma 19.** *Let G be a finite cyclic group of order $n \ge 3$. Then $\mathrm{CSP}(G; +, \neq_0)$ is NP-hard. In particular, the primitive existential theory of $(G; +)$ is NP-hard.*

Proof. The primitive positive formula $\exists e, z (e + e = e \wedge y + z = e \wedge x + z \neq 0)$ defines the binary relation $\neq$ over G. A finite graph with vertices $[n]$ and edges $E \subseteq [n]^2$ can be colored with $n = |G|$ colors if and only if $\bigwedge_{(i,j) \in E} x_i \neq x_j$ is satisfiable in G. For $n \ge 3$, the graph coloring problem is NP-hard [7, Section 4], so the claim follows from Lemma 5. ◀

▶ **Lemma 20.** *For every prime number p and every $e \in \mathbb{N}$ the structure $(\mathbb{Z}/p^e\mathbb{Z}; +, \neq_0)$ has a primitive positive interpretation in $(\mathbb{Z}_p; +, <^p_e)$.*

Proof. The quotient map $\gamma \colon \mathbb{Z}_p \to \mathbb{Z}_p / p^e \mathbb{Z}_p \cong \mathbb{Z}/p^e\mathbb{Z}$ does the job: As $\gamma^{-1}(0) = p^e \mathbb{Z}_p$ is primitively positively definable in $(\mathbb{Z}_p; +)$, also the pullback of the graph of $+$ is primitively positively definable in $(\mathbb{Z}_p; +)$. Finally, $v_p(x) < e$ is a primitive positive definition in $(\mathbb{Z}_p; +, <^p_e)$ of the unary relation $\gamma^{-1}(\neq_0) = \mathbb{Z}_p \setminus p^e \mathbb{Z}_p$. ◀

▶ **Proposition 21.** *The primitive positive theory of $\mathrm{CSP}(\mathbb{Z}_p; +, =^p_0)$ is NP-hard for $p \ge 3$, and $\mathrm{CSP}(\mathbb{Z}_p; +, \le^p_1)$ is NP-hard for all prime numbers p.*

Proof. If $p \ge 3$, then $(\mathbb{Z}/p\mathbb{Z}; +, \neq_0)$ is NP-hard by Lemma 19. Moreover, by Lemma 20 it has a primitive positive interpretation in $(\mathbb{Z}_p; +, =^p_0) = (\mathbb{Z}_p; +, <^p_1)$ and so $\mathrm{CSP}(\mathbb{Z}_p; +, =^p_0)$ is NP-hard by Lemma 5.

If p is an arbitrary prime number, then $(\mathbb{Z}/p^2\mathbb{Z}; +)$ is cyclic of order $p^2 \ge 3$ and we have that $\mathrm{CSP}(\mathbb{Z}/p^2\mathbb{Z}; +, \neq_0)$ is NP-hard by Lemma 19. The structure $(\mathbb{Z}/p^2\mathbb{Z}; +, \neq_0)$ has a primitive positive interpretation in $(\mathbb{Z}_p; +, <^p_2)$ by Lemma 20, and hence $(\mathbb{Z}_p; +, <^p_2) = (\mathbb{Z}_p; +, \le^p_1)$ is NP-hard by Lemma 5. ◀

Let c be a positive integer. In primitive positive formulas over structures whose signature contains $+$ and 1, we use cy as a shortcut for $\underbrace{y + \cdots + y}_{c \text{ times}}$, and c as a shortcut for $c1$. We also freely use the term $x + c$ for $c \in \mathbb{Z}$; if $c = 0$, then this can be replaced by x, and if $c < 0$, then this can be rewritten into a proper primitive positive formula by introducing a new existentially quantified variable y, replacing $x + c$ by y, and adding a new conjunct $x = y + |c|$.

▶ **Lemma 22.** *For $p \geq 3$, the primitive positive formula*

$$\exists y, z\big(v_p(y) = 0 \wedge v_p(z) = 0 \wedge x = y + z\big)$$

defines the relation $\geq_0^p$ in $(\mathbb{Q}_p; +, =_0^p)$. The primitive positive formula

$$\exists y, z\big(v_2(y) = 0 \wedge v_2(z) = 0 \wedge 2x = y + z\big)$$

defines the relation $\geq_0^2$ in $(\mathbb{Q}_2; +, =_0^2)$.

Proof. First let $p \geq 3$. Suppose that $x \in \mathbb{Q}_p$ is such that $v_p(x) \geq 0$. Let $i_0 \in \{0, \ldots, p-1\}$ be such that $v_p(x - i_0) > 0$ (Lemma 2). Since $p \geq 3$, there exists $i \in \{1, \ldots, p-1\} \setminus \{i_0\}$, and $x = (x - i) + i$ with $v_p(x - i) = 0$ and $v_p(i) = 0$. Then setting y to $x - i$ and z to i, all the three conjuncts of the given formula are satisfied. Conversely, if $v_p(y) = v_p(z) = 0$, then $v_p(y + z) \geq 0$.

For $p = 2$, if $x \in \mathbb{Q}_2$ is such that $v_2(x) \geq 0$, then $2x = (2x - 1) + 1$ with $v_2(2x - 1) = 0$ and $v_2(1) = 0$. Conversely, if $y, z \in \mathbb{Q}_2$ are such that $v_2(y) = v_2(z) = 0$, then $v_2(y + z) = v_2((y-1) + (z-1) + 2) \geq \min\{v_2(y-1), v_2(z-1), v_2(2)\} > 0$ by Lemma 2, so if $2x = y + z$, then $1 + v_2(x) = v_2(2x) = v_2(y + z) \geq 1$, hence $v_2(x) \geq 0$. ◀

The following solves an open problem from [9, Remark 23] for $p = 3$; the NP-hardness for $p \geq 5$ was already shown in [9, Prop. 22].

▶ **Corollary 23.** *Let $p \geq 3$ be prime. Then $\mathrm{CSP}(\mathbb{Q}_p; +, =_0^p)$ is NP-hard.*

Proof. Note that $(\mathbb{Z}_p; +, =_0^p)$ has a primitive positive interpretation in $(\mathbb{Q}_p; +, =_0^p)$, because $\geq_0^p$ is primitive positive definable in $(\mathbb{Q}_p; +, =_0^p)$ by Lemma 22. Since $\mathrm{CSP}(\mathbb{Z}_p; +, =_0^p)$ is NP-hard by Proposition 21, the statement follows from Lemma 5. ◀

▶ **Lemma 24.** *Let $c \in \mathbb{Z}$. The relation $=_c^2$ has the primitive positive definition*

$$\exists y\big(v_2(y) \geq 0 \wedge x = 2^c + 2^{c+1}y\big)$$

in $(\mathbb{Q}_2; +, 1, \geq_0^2)$, and in $(\mathbb{Z}_2; +, 1)$ the primitive positive definition

$$\exists y(x = 2^c + 2^{c+1}y).$$

Proof. If $v_2(x) = c$, then $x = 2^c + 2^{c+1}y$ with $v_2(y) \geq 0$, i.e., $y \in \mathbb{Z}_2$ (Lemma 2). Conversely, if $x = 2^c + 2^{c+1}y$ with $v_2(y) \geq 0$, then $v_2(x) = \min\{v_2(2^c), v_2(2^{c+1}y)\} = c$. ◀

Note that the primitive positive formula in Lemma 24 has exponential representation size, since 2^{c+1} is a doubly exponentially large number. However, in all hardness proofs where we use this formula, c will be a constant and hence the length of the formula will be a constant as well.

▶ **Lemma 25.** *For all $p \in \mathbb{P}$, the relation $\neq_0^p$ has the primitive positive definition*

$$\bigwedge_{i=1}^{p-1} v_p(x-i) \leq 0$$

in $(\mathbb{Q}_p; +, 1, \leq_0^p)$, and in $(\mathbb{Z}_p; +)$ the primitive positive definition $\exists y(py = x)$.

Proof. If $v_p(x) > 0$, then $v_p(x-i) = v_p(i) = 0$ for every $1 \leq i < p$, and if $v_p(x) < 0$, then $v_p(x-i) = v_p(x) < 0$ for every i. Conversely, if $v_p(x) = 0$ there exists $i_0 \in \{1, \ldots, p-1\}$ with $v_p(x - i_0) > 0$ (Lemma 2). In $\mathbb{Z}_p$, $v_p(x) \neq 0$ just means $v_p(x) \geq 1$, i.e., $x = py$ with $y \in \mathbb{Z}_p$. ◀

▶ **Lemma 26.** *Let $d \in \mathbb{Z}$. Then $\leq_d^p$ has the primitive positive definition*

$$\bigwedge_{i=1}^{p-1} v_p(x + ip^{d+1}) \neq d+1$$

in $(\mathbb{Q}_p; +, 1, \neq_{d+1}^p)$ for $p \geq 3$, and in $(\mathbb{Q}_2; +, 1, \neq_d^2)$ the primitive positive definition

$$v_2(x + 2^d) \neq d.$$

Proof. First let $p \geq 3$. If $v_p(x) \leq d$, then $v_p(x + ip^{d+1}) = v_p(x) < d+1$ for every $i = 1, \ldots, p-1$. Conversely, if $v_p(x) > d$, then either $v_p(x) > d+1$, in which case $v_p(x + ip^{d+1}) = d+1$ for every $i = 1, \ldots, p-1$, or $v_p(x) = d+1$. In this case, there exists (exactly) one $i_0 \in \{1, \ldots, p-1\}$ with $v_p(x + i_0 p^{d+1}) > d+1$ (Lemma 2), and $v_p(x + ip^{d+1}) = v_p(p^{d+1}) = d+1$ for all $i \in \{1, \ldots, p-1\} \setminus \{i_0\}$. Such an i exists by the assumption that $p \geq 3$.

Now let $p = 2$. If $v_2(x) < d$, then $v_2(x + 2^d) = v_2(x) < d$, and if $v_2(x) = d$, then $v_2(x + 2^d) > d$ (Lemma 2). Conversely, if $v_2(x) > d$, then $v_2(x + 2^d) = d$. ◀

▶ **Theorem 27.** *Let $p \in \mathbb{P}$ be such that $p \geq 3$. Let $\mathfrak{R}$ be a reduct of $\mathfrak{Q}_p$ whose signature τ contains $\{+, 1\}$. Then $\mathrm{CSP}(\mathfrak{R})$ is in P if $\mathfrak{R}$ is a reduct of one of the structures*

$$(\mathbb{Q}_p; +, 1, (\leq_c^p)_{c \in \mathbb{Z}}, (\neq_c^p)_{c \in \mathbb{Z}}) \tag{5.1}$$
$$(\mathbb{Q}_p; +, 1, (\geq_c^p)_{c \in \mathbb{Z}}), \tag{5.2}$$

and is NP-complete otherwise.

Proof. The containment of $\mathrm{CSP}(\mathfrak{R})$ in NP follows from Corollary 7. If τ contains $=_c^p$ for some $c \in \mathbb{Z}$, then the relation $=_0^p$ is primitively positively definable in $\mathfrak{R}$ and $\mathrm{CSP}(\mathfrak{R})$ is NP-hard by Corollary 23 and Lemma 5. So suppose that τ does not contain $=_c^p$ for any $c \in \mathbb{Z}$. If $\mathfrak{R}$ does not contain $\geq_c^p$ for any $c \in \mathbb{Z}$, then $\mathfrak{R}$ is a reduct of the structure in (5.1). In this case, the polynomial-time tractability of $\mathrm{CSP}(\mathfrak{R})$ follows from Proposition 8 and Proposition 6. So suppose that $\mathfrak{R}$ contains $\geq_c^p$ for some $c \in \mathbb{Z}$. If τ also contains $\leq_d^p$ for some $d \in \mathbb{Z}$, then the relation $=_0^p$ is primitively positively definable as well, and we are again done. If τ contains $\neq_c^p$ for some $c \in \mathbb{Z}$, then $\leq_{c-1}^p$ is primitively positively definable in $\mathfrak{R}$ by Lemma 26, and we are in a case that we have already treated. Otherwise, τ contains neither of $\neq_c^p$, $\leq_c^p$, and $=_c^p$ for any $c \in \mathbb{Z}$, and hence $\mathfrak{R}$ is a reduct of the structure (5.2). The polynomial-time tractability in this case follows from Theorem 18 and Proposition 6. ◀

▶ **Theorem 28.** *Let $\mathfrak{R}$ be a reduct of $\mathfrak{Q}_2$ whose signature τ contains $\{+, 1\}$. Then* CSP($\mathfrak{R}$) *is in P if $\mathfrak{R}$ is a reduct of one of the structures*

$$(\mathbb{Q}_2; +, 1, (\leq_c^2)_{c\in\mathbb{Z}}, (\neq_c^2)_{c\in\mathbb{Z}}) \tag{5.3}$$

$$(\mathbb{Q}_2; +, 1, (=_c^2)_{c\in\mathbb{Z}}, (\geq_c^2)_{c\in\mathbb{Z}}), \tag{5.4}$$

and is NP-complete otherwise.

Proof. The containment of CSP($\mathfrak{R}$) in NP follows again from Corollary 7. If τ contains neither $\neq_c^2$ nor $\leq_c^2$ for any $c \in \mathbb{Z}$, then $\mathfrak{R}$ is a reduct of the structure in (5.4), and the polynomial-time tractability of CSP($\mathfrak{R}$) follows from Theorem 18 and Proposition 6. Otherwise, the relation $\leq_1^2$ is primitively positively definable in $\mathfrak{R}$ by Lemma 26. If additionally $\geq_0^2$ is primitively positively definable in $\mathfrak{R}$, then the structure $(\mathbb{Z}_2; +, \leq_1^2)$ has a primitive positive interpretation in $\mathfrak{R}$, and the NP-hardness of CSP($\mathfrak{R}$) follows from Proposition 21 via Lemma 5. If not, then by Lemma 22 we may assume that τ contains neither $\geq_c^p$ nor $=_c^p$ for any $c \in \mathbb{Z}$. In this case, $\mathfrak{R}$ is a reduct of the structure in (5.3), and the polynomial-time tractability of CSP($\mathfrak{R}$) follows from Proposition 8 and Proposition 6. ◀

6 Combining several primes, and the ordering

The complexity classification results for reducts of $\mathfrak{Q}_p$ from Theorems 27 and 28 translate to complexity classification results for expansions of $(\mathbb{Q}; +, 1)$ by relations from

$$\tau_p := \{\leq_c^p, \geq_c^p, =_c^p, \neq_c^p \mid c \in \mathbb{Z}\},$$

for fixed $p \in \mathbb{P}$, via Proposition 6. Interestingly, we can even derive results about expansions of $(\mathbb{Q}; +, 1)$ by relations from $\bigcup_{p\in\mathbb{P}} \tau_p$. Moreover, we may also obtain results about expansions of $(\mathbb{Q}; +, 1, <)$ and of $(\mathbb{Q}; +, 1, \leq)$. The key to this is the following consequence of the approximation theorem for absolute values. As in the introduction, define $|x|_p := p^{-v_p(x)}$ for $x \in \mathbb{Q}$.

▶ **Lemma 29.** *Let $m, n, r \in \mathbb{N}$, $\epsilon > 0$, $A \in \mathbb{Q}^{m\times n}$, $b \in \mathbb{Q}^m$, and let $p_1, \dots, p_r$ be distinct prime numbers. For each $i \in \{0, \dots, r\}$ let $x^{(i)} \in \mathbb{Q}^n$ be such that $Ax^{(i)} = b$. Then there exists $x \in \mathbb{Q}^n$ with $Ax = b$ such that for every $j \in \{1, \dots, n\}$ and $i \in \{1, \dots, r\}$ we have $|x_j - x_j^{(0)}| < \epsilon$ and $|x_j - x_j^{(i)}|_{p_i} < \epsilon$.*

Proof. Write the solution space $L \subseteq \mathbb{Q}^n$ of $Ax = b$ as in (4.1). The map $L \to \mathbb{Q}^d$, $y_0 + \sum_{k=1}^d \lambda_k y_k \mapsto (\lambda_1, \dots, \lambda_d)$ is a homeomorphism with respect to the real topology and with respect to each p-adic topology. We can therefore assume without loss of generality that $L = \mathbb{Q}^n$, i.e., that $A = 0$ and $b = 0$. The claim is then precisely the statement of the approximation theorem for finitely many inequivalent absolute values on a field K ([13, Ch. XII, Thm. 1.2]) in the case $K = \mathbb{Q}$, applied for each $j \in \{1, \dots, n\}$. ◀

Let $\mathfrak{Q}$ be the expansion of $(\mathbb{Q}; +, 1)$ by new relations for $\tau := \{<\} \cup \bigcup_{p\in\mathbb{P}} \tau_p$.

▶ **Proposition 30.** *Let φ be a conjunction of atomic $(\{+, 1\} \cup \tau)$-formulas. Let $\varphi_<$ be all conjuncts of φ formed with the symbol $<$, let φ_p be all conjuncts of φ formed with symbols from τ_p, and let $\varphi_=$ be all the conjuncts formed with $=$. Then φ is satisfiable in $\mathfrak{Q}$ if and only if $\varphi_= \wedge \varphi_<$ is satisfiable in $\mathfrak{Q}$ and $\varphi_= \wedge \varphi_p$ is satisfiable in $\mathfrak{Q}$ for each $p \in \mathbb{P}$.*

Proof. The forward implication is trivial. For the converse, let $s^< \in \mathbb{Q}^n$ be a satisfying assignment for $\varphi_= \wedge \varphi_<$, let P denote the (finite) set of prime numbers such that φ contains symbols from τ_p, and for each $p \in P$ let $s^{(p)} \in \mathbb{Q}^n$ be a satisfying assignment for $\varphi_= \wedge \varphi_p$.

The set $U_< \subseteq \mathbb{Q}^n$ of satisfying assignments for $\varphi_<$ is open in the real topology, and the set U_p of satisfying assignments for φ_p is open in the p-adic topology, for each p. In particular, there exists $\epsilon > 0$ such that the whole box $\{y \in \mathbb{Q}^n : |y_j - s_j^<| < \epsilon$ for every $j\}$ is contained in $U_<$, and similarly $\{y \in \mathbb{Q}^n : |y_j - s_j^{(p)}|_p < \epsilon$ for every $j\} \subseteq U_p$ for every $p \in P$. Therefore, by Lemma 29, there exists $s \in \mathbb{Q}^n$ such that s satisfies $\varphi_=$ and $s \in U_< \cap \bigcap_{p \in P} U_p$, hence s is a satisfying assignment for φ. ◀

Proposition 30 only works for strict inequalities, and the corresponding statement would be false for weak inequalities. Algorithmically, however, there is a way to reduce the problem to the satisfiability problem for strict inequalities, and we obtain the following result.

▶ **Theorem 31.** *Let $\mathfrak{R}$ be a reduct of $(\mathfrak{Q}, \leq)$ whose signature τ contains $\{1, +\}$. If τ contains*
- *$=_c^p$ for some $c \in \mathbb{Z}$ and $p \in \mathbb{P}$ with $p \geq 3$, or*
- *$\geq_{c_1}^p$ and a relation from $\{\leq_{c_2}^p, \neq_{c_2}^p\}$ for some $c_1, c_2 \in \mathbb{Z}$ and $p \in \mathbb{P}$ with $p \geq 3$,*
- *a relation from $\{\geq_{c_1}^2, =_{c_1}^2\}$ and a relation from $\{\leq_{c_2}^2, \neq_{c_2}^2\}$ for some $c_1, c_2 \in \mathbb{Z}$,*

then CSP($\mathfrak{R}$) *is NP-complete; otherwise,* CSP($\mathfrak{R}$) *is in P.*

Proof. If τ contains a symbol of the form $=_c^p$ for some $p \geq 3$, or contains $\geq_c^p$ and a symbol of the form $\leq_c^p$ or $\neq_c^p$, then the NP-hardness of CSP($\mathfrak{R}$) follows from Theorem 27 and Proposition 6. Moreover, if the signature contains a symbol of the form $=_c^2$ or $\geq_c^2$ and a symbol of the form $\leq_p^2$ or $\neq_c^p$, then the NP-hardness of CSP($\mathfrak{R}$) follows from Theorem 28 and Proposition 6. Otherwise, let φ be an instance of CSP($\mathfrak{R}$). Similarly to Proposition 30 let
- $\varphi_<$ be the conjuncts of φ formed with the symbol $<$,
- $\varphi_\leq$ the conjuncts formed with $\leq$,
- φ_p the conjuncts formed with symbols from τ_p, and
- $\varphi_=$ the conjuncts formed with $=$.

Let P be the set of $p \in \mathbb{P}$ such that a symbol from τ_p occurs in φ. For any instance ψ denote by $\psi^<$ the instance obtained by replacing all $\leq$ by $<$.

We first check with known methods whether there is a solution for $\varphi_0 := \varphi_= \wedge \varphi_< \wedge \varphi_\leq$ (see, e.g., [16, final remark in Section 13.4]). If there is no solution, then output NO. Otherwise, let Ψ be the set of conjuncts of $\varphi_\leq$. We then test for each $\psi \in \Psi$ whether the formula $\varphi_0 \wedge \psi^<$ is still satisfiable (again, using known methods). If $\varphi_0 \wedge \psi^<$ is unsatisfiable, then every solution of φ_0 must satisfy the formula $\psi^=$ obtained from ψ by replacing $\leq$ with $=$. We then recursively run the entire algorithm on the formula where we replace the conjunct ψ by $\psi^=$. Otherwise, if for every $\psi \in \Psi$, the formula $\varphi_0 \wedge \psi^<$ has a solution s_ψ, then $\varphi_0^<$ has a solution $s^<$ as well. This is clear if $\Psi = \emptyset$; otherwise, we note that the function $f\colon \mathbb{Q}^k \to \mathbb{Q}$ given by $(x_1, \ldots, x_k) \mapsto \frac{1}{k} \sum_{i=1}^k x_i$ applied componentwise preserves $+$, 1, $\leq$, and strongly preserves $<$ in the sense that $f(x_1, \ldots, x_k) < f(y_1, \ldots, y_k)$ if $x_1 \leq y_1, \ldots, x_k \leq y_k$ and $x_i < y_i$ for at least one $i \in \{1, \ldots, k\}$. This shows that we may take $s^< := \frac{1}{|\Psi|} \sum_{\psi \in \Psi} s_\psi$.

We run the polynomial-time algorithm from Theorem 28 on $\varphi_= \wedge \varphi_2$ and for each $p \in P \setminus \{2\}$ the polynomial-time algorithm from Theorem 27 on $\varphi_= \wedge \varphi_p$. If one of these algorithms returns NO, then φ is unsatisfiable by Proposition 6. If all of the algorithms return YES, then $\varphi^<$ has a solution by Proposition 6 and Proposition 30, and therefore also φ has a solution.

Finally, CSP($\mathfrak{Q}$) is in NP as can be shown by repeating the argument from the previous paragraphs for an instance φ of CSP($\mathfrak{Q}$) and using Corollary 7 instead of the polynomial-time algorithms. ◀

Table 1 An overview of polynomial-time tractability and NP-hardness for systems of linear equations with valuation constraints.

	$\mathbb{Q}_p$, $p \geq 3$	$\mathbb{Q}_p$, $p = 2$	$\mathbb{Z}_p$, $p \geq 3$	$\mathbb{Z}_p$, $p = 2$
$\emptyset$	P: Gauss algorithm		P: Hermite normal form	
$v_p(x) \geq c$	P: 18			
$v_p(x) = 0$	NP-hard: def. $\mathbb{Z}_p$ 22	P: reduce to $v_p(x) \geq 0$ 24	NP-hard: 21	P: reduce to $\emptyset$ 24
$v_p(x) = c$	NP-hard: solves $v_p(x) = 0$	P: 18	NP-hard: solves $v_p(x) = 0$	P: 18
$v_p(x) \leq 0$	P: special case of $v_p(x) \leq c$		NP-hard: same as $v_p(x) = 0$	P: same as $v_p(x) = 0$
$v_p(x) \leq 1$	P: special case of $v_p(x) \leq c$		NP-hard: 21	
$v_p(x) \leq c$	P: 8		NP-hard: solves $v_p(x) \leq 1$	
$v_p(x) \neq 0$	P: 8 or reduce to $v_p(x) \leq 0$ via 25		P: reduces to $\emptyset$ via 25	
$v_p(x) \neq c$	P: 8		NP-hard: def. $v_p(x) \leq 1$ via 26	

7 Conclusions and an open problem

We have presented polynomial-time algorithms for the satisfiability problem of systems of linear equalities combined with various valuation constraints. For such systems, the satisfiability in $\mathbb{Q}_p$ is equivalent to satisfiability in $\mathbb{Q}$ (Proposition 6). We also prove the matching NP-hardness results, answering open questions from [9] (Theorem 27 and Theorem 28; also see Table 1). Our results can be combined with the polynomial-time tractability result for the satisfiability of (strict and weak) linear inequalities over $\mathbb{Q}$, and we may even solve valuation constraints for different prime numbers simultaneously (Theorem 31). Our polynomial-time tractability result for linear inequalities with valuation constraints of the form $v_2(x) = c$, for constants $c \in \mathbb{Z}$ given in binary, would also follow from a positive answer to the following question, which remains open.

▶ **Question 32.** Is there a polynomial-time algorithm for the satisfiability problem of systems of weak linear inequalities where the coefficients of the inequalities are of the form 2^c where c is represented in binary?

Such an algorithm would also imply a polynomial-time algorithm for mean-payoff-games (see [4] for related reductions) which is a problem currently not known to be in P.

References

1 Ahmad Abdi, Gérard Cornuéjols, Bertrand Guenin, and Levent Tunçel. Dyadic linear programming and extensions. *Mathematical Programming*, 2024. doi:10.1007/s10107-024-02146-4.

2 Manuel Bodirsky. *Complexity of Infinite-Domain Constraint Satisfaction*. Lecture Notes in Logic (52). Cambridge University Press, Cambridge, United Kingdom; New York, NY, 2021. doi:10.1017/9781107337534.

3 Manuel Bodirsky, Peter Jonsson, and Timo von Oertzen. Essential convexity and complexity of semi-algebraic constraints. *Logical Methods in Computer Science*, 8(4), 2012. An extended abstract about a subset of the results has been published under the title *Semilinear Program Feasibility* at ICALP'10. doi:10.2168/LMCS-8(4:5)2012.

4 Manuel Bodirsky, Georg Loho, and Mateusz Skomra. Reducing stochastic games to semidefinite programming. In *the proceedings of the International Colloquium on Automata, Languages,*

and Programming, ICALP, 2025. https://arxiv.org/abs/2411.09646. doi:10.48550/arXiv.2411.09646.

5 Andreas Dolzmann and Thomas Sturm. P-adic constraint solving. In *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation*, ISSAC '99, pages 151–158, New York, NY, USA, 1999. Association for Computing Machinery. doi:10.1145/309831.309894.

6 Arno Fehm and Manuel Bodirsky. Polynomial-time tractable problems over the p-adic numbers, 2025. doi:10.48550/arXiv.2504.13536.

7 Michael Garey and David Johnson. *A guide to NP-completeness.* CSLI Press, Stanford, 1978.

8 Fernando Gouvêa. *p-adic Numbers.* Springer, 1997.

9 Florent Guépin, Christoph Haase, and James Worrell. On the existential theories of Büchi arithmetic and linear p-adic fields. In *Logic in Computer Science, LICS*. IEEE, 2019. doi:10.1109/LICS.2019.8785681.

10 Christoph Haase and Alessio Mansutti. On Deciding Linear Arithmetic Constraints Over p-adic Integers for All Primes. In Filippo Bonchi and Simon J. Puglisi, editors, *46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021)*, volume 202 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 55:1–55:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2021.55.

11 Wilfrid Hodges. *A shorter model theory.* Cambridge University Press, Cambridge, 1997.

12 Peter Jonsson and Johan Thapper. Constraint satisfaction and semilinear expansions of addition over the rationals and the reals. *CoRR*, abs/1506.00479, 2015. arXiv:1506.00479.

13 Serge Lang. *Algebra.* Springer, 2002. Revised third edition.

14 Yuri Matiyasevich. Enumerable sets are Diophantine. *Doklady Akademii Nauk SSSR*, 191:279–282, 1970.

15 Marcus Schaefer and Daniel Štefankovič. Fixed points, Nash equilibria, and the existential theory of the reals. *Theory of Computing Systems*, pages 1–22, 2015.

16 Alexander Schrijver. *Theory of Linear and Integer Programming.* Wiley - Interscience Series in Discrete Mathematics and Optimization, 1998.

17 Volker Weispfenning. The complexity of linear problems in fields. *Journal of Symbolic Computation*, 5(1):3–27, 1988. doi:10.1016/S0747-7171(88)80003-8.