HORN VERSUS FULL FIRST-ORDER: COMPLEXITY DICHOTOMIES IN ALGEBRAIC CONSTRAINT SATISFACTION

MANUEL BODIRSKY, PETER JONSSON, AND TIMO VON OERTZEN

CNRS/LIX, École Polytechnique, France
e-mail address: bodirsky@lix.polytechnique.fr

Department of Computer and System Science, Linköping University, Linköping, Sweden
e-mail address: peter.jonsson@liu.se

Max-Planck-Institute for Human Development, Berlin, Germany
e-mail address: vonoertzen@mpib-berlin.mpg.de

Abstract. We study techniques for deciding the computational complexity of infinite-domain constraint satisfaction problems. For certain basic algebraic structures Δ, we prove definability theorems of the following form: for every first-order expansion Γ of Δ, either Γ has a quantifier-free Horn definition in Δ, or there is an element d of Γ such that all non-empty relations in Γ contain a tuple of the form $(d, \ldots, d)$, or all relations with a first-order definition in Δ have a primitive positive definition in Γ. The results imply that several families of constraint satisfaction problems exhibit a complexity dichotomy: the problems are either polynomial-time solvable or NP-hard depending on the choice of the allowed relations. As concrete examples, we investigate fundamental algebraic constraint satisfaction problems. The first class consists of all relational structures with a first-order definition in $(\mathbb{Q}; +)$ that contain the relation $\{(x, y, z) \in \mathbb{Q}^3 \mid x + y = z\}$. The second class is the affine variant of the first class. In both cases, we obtain full dichotomies by utilising our general methods.

1. Introduction

Constraint satisfaction problems (CSPs) are computational problems that appear in almost every area of computer science such as artificial intelligence, graph algorithms, scheduling, combinatorics, and computer algebra. Depending on the type of constraints that are allowed in the input instances of a CSP, the computational complexity of a CSP is usually polynomial (and we will call such CSPs *tractable*) or NP-hard. In the last decade, a lot of progress was made to find general criteria that imply that a CSP is tractable, or that it is NP-hard. Such results have been obtained for constraint languages over finite domains [10, 12, 13, 19], but also for constraint languages over infinite domains that are ω-categorical (for formal definitions of these concepts see Section 2). For example, it has been shown that for every structure Γ with a first-order definition in $(\mathbb{Q}; <)$, the problem CSP(Γ) is in P if it falls into one of nine classes, and it is NP-hard otherwise [7].

Lately, many researchers have been fascinated by a conjecture due to Feder and Vardi [16] which is known as the *dichotomy conjecture*. This conjecture says that every CSP with a finite domain constraint language is either tractable (i.e., in P) or NP-complete. According to a well-known result by Ladner [22], there are NP-*intermediate* computational problems, i.e., problems in NP that are neither tractable nor NP-complete (unless P=NP). But the problems that are given in Ladner's construction are extremely artificial. It is a curious fact that there are so few candidates for natural NP-intermediate problems.

Any outcome of the dichotomy conjecture is probably surprising: a negative answer would finally provide relatively natural NP-intermediate problems, which would be of interest in complexity theory. A positive answer probably comes with a criterion which describes the NP-hard CSPs (and it would probably even provide algorithms for the tractable CSPs). But then we would have a rich catalogue of computational problems where the computational complexity is known. Such a catalogue would be a valuable tool for deciding the complexity of computational problems in the mentioned application areas: since CSPs are abundant, one might derive algorithmic results by reducing the problem of interest to a known tractable CSP, and one might derive hardness results by reducing a known NP-hard CSP to the problem of interest.

In this article, we study two fundamental classes of infinite domain constraint languages, and show that the corresponding CSPs do exhibit a complexity dichotomy. To the best of our knowledge, this is the first systematic complexity result for classes of structures that are not ω-categorical.

The first class consists of all first-order relational expansions of $(\mathbb{Q}; \{(x, y, z) \in \mathbb{Q}^3 \mid x + y = z\})$; by a *first-order relational expansion of* Γ we mean the structure obtained from Γ by adding relations to Γ that are first-order definable in Γ. The second class is an affine version of the first class: it consists of all relational first-order expansions of the structure $(\mathbb{Q}; \{(x, y, z, w) \in \mathbb{Q}^4 \mid x - y + z = w\}$. That none of these structures is ω-categorical follows from the theorem by Engeler, Ryll-Nardzewski, and Svenonius (cf. Theorem 6.3.1 in [18]). It is even the case that the corresponding CSPs cannot be formulated by any ω-categorical template: the basic proof idea is presented in [1, Proposition 1]; also see [4].

Our results follow from theorems about primitive positive definability: we show that for every relation R with a first-order definition in $(\mathbb{Q}; +)$, either R has a quantifier-free Horn definition in $(\mathbb{Q}; +, 0)$, or R contains the tuple $(0, \ldots, 0)$, or all relations with a first-order definition in $(\mathbb{Q}; +)$ have a primitive positive definition in $(\mathbb{Q}; +, R)$. The analogous result also holds for the affine case. The techniques that we use to prove these two definability theorems are more general than the two classification results, and they are very different in nature. One technique applies for structures 'that have little structure'; to be precise, for *all* structures Γ where $=$ and $\neq$ are the only primitive positive definable non-trivial binary relations (Section 5). In particular, they apply to structures with a 2-transitive automorphism group. The other technique applies for structures 'with a lot of structure'; informally, it applies whenever we can find a primitive positive definition for the line between two points in $\mathbb{Q}^k$ (Section 4).

We would like to mention that the authors have recently also shown a complexity dichotomy for (semi-linear or semi-algebraic) expansions of the structure $(\mathbb{R}; \{(x, y, z) \in \mathbb{R}^3 \mid x + y = z\}, \leq, \{1\})$ (the constraint language of linear program feasibility), which might look similar [5]. However, there are some important differences between this result and the results presented here. The first is that the constraint satisfaction problems in [5] are more expressive since they contain the unary relation $\{1\}$ (which together with the relation $\{(x, y, z) \in \mathbb{R}^3 \mid x + y = z\}$ allows us to express all rational constants) and inequality (inequality makes those structures much more unwildly from a model-theoretic point of view), and that they are over the real numbers and not the rationals (which makes a real difference when we look at semi-algebraic expansions of the structure above). Also, the results in [5] do *not* come with a 'Horn versus full first-order' definability result in the sense of this article. However, the most important difference between [5] and the results in this article is the completely different proof technique: the proofs here are based on logic definability, whereas the proofs in [5] are based on geometric arguments.

The rest of this article is organised as follows: in Section 2, we provide some background material on constraint satisfaction and logic. A tractability result for templates that have a quantifier-free Horn definition in $(\mathbb{Q}; +, 0)$ is presented in Section 3. The classification result for $(\mathbb{Q}; +, 0)$ can be found in Section 4 while the results for the affine case are collected in Section 5. Finally, a number of open questions and directions for future work can be found in Section 6. It may seem peculiar that we consider the structure $(\mathbb{Q}; +, 0)$ instead of $(\mathbb{Q}; +)$. This is due to certain technicalities concerning quantifier elimination and the details will be explained in Section 4.2.

2. Preliminaries

A first-order formula is called *primitive positive*[1] if it is of the form

$$\exists x_1, \ldots, x_n. \psi_1 \wedge \ldots \wedge \psi_m$$

where ψ_i are atomic formulas, i.e., formulas of the form $x = y$ or $R(x_{i_1}, \ldots, x_{i_k})$ with R the relation symbol for a k-ary relation from Γ. We call such a formula a *pp-formula.* As usual, formulas without free variables are called *sentences.*

Let $\Gamma = (D; R_1, \ldots, R_n)$ be a relational structure with domain D (which will usually be infinite) and finitely many relations $R_1, \ldots, R_n$. The *constraint satisfaction problem for* Γ (in short CSP(Γ)) is the computational problem to decide whether a given primitive positive sentence Φ involving relation symbols for the relations in Γ is true in Γ. The conjuncts in a pp-formula Φ are called the *constraints* of Φ. We refer to Γ as the *constraint language* (it is also often called the *template*) of CSP(Γ).

We say that a first-order formula ϕ *defines* a relation R in Γ when $\phi(a_1, \ldots, a_k)$ holds in Γ iff $(a_1, \ldots, a_k) \in R$. If ϕ is primitive positive, we call R *primitive positive definable (pp-definable)* over Γ. The following simple but important result explains the importance of primitive positive definability for constraint satisfaction problems; it was first stated in [20] for finite structures Γ only, but the proof there also works for arbitrary infinite structures.

Lemma 2.1 (from [20])**.** *Let Γ be a relational structure and Γ' be an expansion of this structure by a pp-definable relation R over Γ. Then CSP(Γ) is polynomial-time equivalent to CSP(Γ').*

Lemma 2.1 will be used extensively in the sequel and we will not make explicit references to it. Another important class of formulas are *Horn* formulas; a first-order formula whose quantifier-free part is in conjunctive normal form is Horn if and only if each clause contains at most one positive literal. A relation R is called quantifier-free Horn definable over Γ if there exists a quantifier-free Horn formula that defines R in Γ. Note that Lemma 2.1 does not hold if we replace 'pp-definable' with 'Horn definable'.

[1]Our terminology is standard; all notions that are not explicitly introduced in the article can be found in standard text books, e.g., in [18].

By choosing an appropriate structure Γ, many computational problems that have been studied in the literature can be formulated as CSP(Γ) (see e.g. [1,11,12]). It turns out very often that the structure Γ can be chosen to be ω-categorical. A countable structure Γ is called *ω-categorical* if all models of the set of first-order sentences that are true in Γ are isomorphic to Γ. A famous example of an ω-categorical structure is $(\mathbb{Q}; <)$. The condition of ω-categoricity is interesting for constraint satisfaction, because the so-called *universal-algebraic approach*, which is currently intensively studied for finite constraint languages, applies—at least in principle—also to ω-categorical structures (see e.g. [7] for an application of the universal-algebraic approach to CSPs for constraint languages over infinite domains). In this article, we demonstrate that systematic complexity classification can be performed for constraint languages over infinite domains even if the constraint languages are not ω-categorical.

Example. Let Γ denote the structure $(\mathbb{Q}; R)$ where

$$R = \{(x, y, u, v) \in \mathbb{Q}^4 \mid (x = y \ \vee \ y = u + v) \wedge x \neq u\}).$$

It is fairly straightforward (by an argument similar to the one presented in [1]) to show that CSP(Γ) *cannot* be formulated with an ω-categorical template (in fact, there exists a general result that completely characterises the CSPs that can be formulated with ω-categorical templates [4]). One can show that the relations $\{(x, y) \in \mathbb{Q}^2 \mid x \neq y\}$ and $\{(x, y, z) \in \mathbb{Q}^3 \mid x = y + z\}$ have pp-definitions in Γ; merely note that $x \neq u \Leftrightarrow \exists y, v.R(x, y, u, v)$ and $y = u + v \Leftrightarrow \exists x.R(x, y, u, v) \wedge x \neq y$. The computational complexity of CSP(Γ) can thus be determined by our classification result (Corollary 4.6).

We will sometimes consider the automorphism group Aut(Γ) of a constraint language Γ over a domain D, i.e., the group formed by the set of all automorphisms of Γ with respect to functional composition. An *orbit* of $Aut(\Gamma)$ on D^2 is a set of the form $\{(\alpha(a), \alpha(b)) \mid \alpha \in Aut(\Gamma)\}$, for some $a, b \in D$. We note that pairs from the same orbit satisfy the same first-order formulas.

Let D be an arbitrary infinite set and arbitrarily choose an element $d \in D$. The complexity of CSP(Γ) where Γ has a first-order definition in $(D; =)$ (so-called *equality languages*) has been classified in [6]. If R is first-order definable in $(D; =)$ and $(d, \ldots, d) \in R$, then $(d', \ldots, d') \in R$ for every $d' \in D$. Thus, the exact choice of d is irrelevant when stating the following theorem.

Theorem 2.2 (from [6])**.** Let Γ be a constraint language with a first-order definition in $(D; =)$. Then, all relations in Γ have a quantifier-free Horn definition in $(D; =)$, or all non-empty relations in Γ contain the tuple $(d, \ldots, d)$, or else every first-order definable relation in $(D; =)$ has a pp-definition in Γ. In the last case, CSP(Γ) is NP-complete.

Instead of using Theorem 2.2 in its full generality, it will often be sufficient to use a simple corollary. For any set D, let S_D denote the relation

$$\{(x, y, z) \in D^3 \mid y \neq z \wedge (x = y \vee x = z)\} .$$

Corollary 2.3. Let D be an infinite set. Every first-order definable relation in $(D; =)$ has a pp-definition in $(D; S_D)$.

Proof. The relation S_D has a first-order definition in $(D; =)$ and does not contain the tuple (d, d, d). It is easy to verify that S_D has no quantifier-free Horn definition in $(D; =)$ so every first-order definable relation in $(D; =)$ has a pp-definition in $(D; S_D)$ by Theorem 2.2. $\square$

Note that we have defined the CSP only for templates with *relational*; this is the standard definition of CSPs in the literature. However, it will be convenient to also use structures involving function symbols to state our results on primitive positive definability. The reason is that with these functions, the algebraic structures under consideration have quantifier elimination, which will be needed in our approach. Quantifier elimination is discussed in detail in Section 4 and 5.

3. Tractability

For all structures Γ with finite relational signature and a quantifier-free Horn definition in $(\mathbb{Q}; +, 0)$, the problem $\mathrm{CSP}(\Gamma)$ can be solved in polynomial time. This follows from a more general algorithmic result in [21] combined with the observation that the solution spaces considered in [21] always contain a rational point when they are non-empty. However, the algorithm presented there solves a linear number of linear programs, and thus the best known algorithms have a rather high worst-case running time. We present a more efficient algorithm for the special case that is relevant in this article. We denote by $O^{\sim}(f(N))$ the class of all functions of asymptotic growth at most $f(N)$ up to poly-logarithmic factors.

Proposition 3.1. Let Γ be a relational structure whose relations have a quantifier-free Horn definition in $(\mathbb{Q}; +, 0)$. Then there is an algorithm that solves $\mathrm{CSP}(\Gamma)$ in time $O^{\sim}(N^4)$ where N is the size of the input.

The algorithm we present in the proof of Proposition 3.1 is a combination of general techniques in constraint satisfaction [2,14] and a polynomial-time implementation of Gaussian elimination on rational data. Since the input of $\mathrm{CSP}(\Gamma)$ consists of a primitive positive sentence whose atomic formulas are of the form $R(x_1, \ldots, x_k)$ where R is quantifier-free Horn definable over $(\mathbb{Q}; +, 0)$, we can as well assume that the input to our problem consists of a set of Horn clauses over $(\mathbb{Q}; +, 0)$.

We have to make some remarks about the worst-case running time of the Gaussian elimination algorithm. It is well-known that Gaussian elimination requires $O(n^2 m)$ many arithmetic operations on rational numbers, where m is the number of equations and n is the number of variables. In our algorithm, we have to solve a linear number of linear equation systems $S_1, \ldots, S_m$; however, system S_{i+1} is obtained from system S_i by adding a single linear equation. Since the Gaussian algorithm can be presented in such a way that it computes a system in triangular form, adding successively equation by equation, the overall costs for solving $S_1, \ldots, S_m$ equals the cost to solve S_m with Gaussian elimination.

Also recall that the size of the numbers involved when performing the Gaussian elimination algorithm might grow exponentially when implemented without care. However, when we use the Euclidean algorithm to shorten the coefficients during the elimination process, the Gaussian elimination algorithm can be shown to be polynomial [15]. We are only interested in deciding solvability of linear equation systems, and not constructing solutions, and so we even have *linear* bounds (in the input size) on the representation size of all numbers involved in deciding solvability for linear equation systems over the rational numbers with Gaussian elimination (see [25], proof of Theorem 3.3). Finally, we remark that the most costly arithmetic operation that has to be performed on rational numbers during the elimination process is multiplication, and multiplication can be performed in time $O(s \log s \log\log s)$, where s denotes the representation size of the two rational numbers (in bits). Hence, the overall running time for solving $S_1, \ldots, S_m$ with the discussed implementation of the Gaussian elimination algorithm is in $O^{\sim}(N^4)$, where N denotes the representation size of the input.

We will show that our algorithm for CSP(Γ) can be implemented such that it has the same overall asymptotic worst-case complexity.

```
Solve(Φ)
// Input: an instance Φ of CSP(Γ)
// where all relations in Γ have a quantifier-free Horn definition in (ℚ; +, 0)
// Output: satisfiable if Φ is true in Γ, unsatisfiable otherwise
Let 𝒞 be the set of all Horn-clauses from each constraint in Φ
Let 𝒰 be the subset of 𝒞 that only contains clauses with a single positive literal.
If 𝒰 is unsatisfiable then return unsatisfiable.
Do
    For all negative literals ¬φ in clauses from 𝒞
        If 𝒰 implies φ, then delete the negative literal ¬φ from all clauses in 𝒞.
    If 𝒞 contains an empty clause, then return unsatisfiable.
    If 𝒞 contains a clause with a single positive literal ψ, then add {ψ} to 𝒰.
Loop until no literal has been deleted
Return satisfiable.
```

Figure 1: An algorithm for the constraint satisfaction problem for Γ when Γ has a quantifier-free Horn definition in $(\mathbb{Q}; +, 0)$.

Proof of Proposition 3.1. We first discuss the correctness of the algorithm shown in Figure 1, and then explain how to implement the algorithm such that it achieves the desired running time.

When $\mathcal{U}$ logically implies ϕ then the negative literal $\neg\phi$ is never satisfied and can be deleted from all clauses without affecting the set of solutions. Since this is the only way in which literals can be deleted from clauses, it is clear that if one clause becomes empty the instance is unsatisfiable.

If the algorithm terminates with *satisfiable*, then no negation of an inequality is implied by $\mathcal{U}$. If r is the rank of the linear equation system defined by $\mathcal{U}$, we can use the Gaussian elimination algorithm as described above to eliminate r of the variables from all literals in the remaining clauses. By multiplying the equations with appropriate constants, we can assume that all coefficients are integer. For each of the remaining inequalities, consider the sum of absolute values of all coefficients. Let S be one plus the maximum of the this sum over all the remaining inqualities. Then setting the i-th variable to S^i satisfies all clauses.

To see this, take any inequality, and assume that i is the highest variable index in this inequality. Order the inequality in such a way that the variable with highest index is on one side and all other variables on the other side of the $\neq$ sign. The absolute value on the side with the i-th variable is at least S^i. The absolute value on the other side is less than $S^i - S$, since all variables have absolute value less than S^{i-1} and the sum of all coefficients is less than $S - 1$ in absolute value. Hence, both sides of the inequality have different absolute value, and the inequality is satisfied. Since all remaining clauses have at least one inequality, all constraints are satisfied.

We finally explain how to implement the algorithm such that it runs in time $O^{\sim}(N^4)$. To decide whether $\mathcal{U}$ implies an equality ϕ, we compute in each iteration of the main loop the triangular normal form for the linear equation system determined by $\mathcal{U}$ as described before the statement of the proposition. The overall time to do this is in $O^{\sim}(N^4)$.

We then maintain for each negative literal an equation where we eliminate as many variables as possible using the computed triangular form. If one of the equations becomes trivial (i.e. is of the form $a = a$) we conclude that the equation is implied by $\mathcal{U}$. The total time for doing this is also bounded by $O^{\sim}(N^4)$ by a very similar argument as given before the statement of the proposition. With appropriate data structures, the time needed for removing negated literals $\neg\phi$ from all clauses when ϕ is implied by $\mathcal{U}$ is linearly bounded in the input size since each literal can be removed at most once. $\square$

4. The Rational Numbers with Addition

In this section we prove results about pp-definability in first-order expansions Γ of $(\mathbb{Q};+)$. Our first result in Section 4.1 says that if Γ contains a relation R such that $R(x,\dots,x)$ is false for any $x \in \mathbb{Q}$, then $\neq$ is pp-definable in Γ. We then show in Section 4.2 that if $\neq$ is pp-definable in Γ, then either every relation in Γ is quantifier-free Horn definable in $(\mathbb{Q};+,0)$, or every relation with a first-order definition in $(\mathbb{Q};+)$ is pp-definable in Γ. From these results we can then derive in Section 4.3 a complexity classification of CSPs for first-order expansions of $(\mathbb{Q};\{(x,y,z) \in \mathbb{Q}^3 \mid x+y=z\})$. For increased readability, we let R_+ denote the relation $\{(x,y,z) \in \mathbb{Q}^3 \mid x+y=z\})$.

4.1. PP-definability of Inequality.

We begin by identifying the unary relations that have first-order definitions in $(\mathbb{Q};+)$.

Lemma 4.1. For any structure Γ with a first-order definition in $(\mathbb{Q};+)$, the first-order definable unary relations in Γ are members of $\{\mathbb{Q}, \mathbb{Q}\setminus\{0\}, \{0\}, \emptyset\}$.

Proof. Let R be a unary relation with a first-order definition in $(\mathbb{Q};+)$. The statement is clear if R does not contain any element distinct from 0, so let a be from $\mathbb{Q}\setminus\{0\}$. We have to show that $R = \mathbb{Q}$ or $R = \mathbb{Q}\setminus\{0\}$. Observe that for any $c \in \mathbb{Q}$, $c \neq 0$, the mapping $x \mapsto cx$ is an automorphism of Γ. Hence, for any $b \neq 0$ there is an automorphism of $(\mathbb{Q};+)$ that maps a to b. Automorphisms preserve first-order formulas, so $b \in R$ and the claim follows. $\square$

Note that $x = 0$ is equivalent to $x + x = x$ which implies that the relation $\{0\}$ is pp-definable over $(\mathbb{Q};+)$; thus we can use 0 freely as a constant symbol in pp-definitions over Γ.

Proposition 4.2. Let Γ be a first-order expansion of $(\mathbb{Q};+)$ containing a non-empty relation R such that $R(x,\dots,x)$ is false for any $x \in \mathbb{Q}$. Then $\neq$ is pp-definable in Γ.

Proof. Observe that if the set $\mathbb{Q}\setminus\{0\}$ has a pp-definition $\phi(u)$ in Γ, then the pp-formula

$$\exists u.\ \phi(u) \wedge x + u = y$$

defines $x \neq y$ over Γ.

Let S be a non-empty pp-definable relation in Γ of minimal arity such that $S(x,\dots,x)$ defines the empty set. Let k be the arity of S. First, assume that $S(x_1,x_2,\dots,x_k) \wedge x_1 = x_2$ is satisfiable. Then the $(k-1)$-ary relation $S'(x_2,\dots,x_k)$ defined by $S(x_2,x_2,\dots,x_k)$ is non-empty, and $S'(x,\dots,x)$ defines the empty set; this is in contradiction to the choice of S.

Assume next that $S(x_1, \ldots, x_k) \wedge x_1 = x_2$ is unsatisfiable. Define the unary relation $T(x)$ by

$$\exists x_3, \ldots, x_k.\, S(x, 0, x_3, \ldots, x_k)$$

and the unary relation $U(y)$ by

$$\exists x_1, x_3, \ldots, x_k.\, S(x_1, y, x_3, \ldots, x_k)\ .$$

By Lemma 4.1, both T and U are from $\{\mathbb{Q}, \mathbb{Q} \setminus \{0\}, \{0\}, \emptyset\}$. The relation T cannot be equal to $\{0\}$ or to $\mathbb{Q}$ since this contradicts the assumption that $S(x_1, x_2, \ldots, x_k) \wedge x_1 = x_2$ is unsatisfiable. If T is equal to $\mathbb{Q} \setminus \{0\}$, then by the initial observation $\neq$ is pp-definable in Γ and we are done. We conclude that $T = \emptyset$ and hence $0 \notin U$. Since U is non-empty, it must be the case that $U = \mathbb{Q} \setminus \{0\}$ and, again by the initial observation, $\neq$ is pp-definable in Γ. □

4.2. **PP-definability in Non-Horn Expansions.** The structure $(\mathbb{Q}; +, 0)$ admits *quantifier elimination*, i.e., every relation with a first-order definition in $(\mathbb{Q}; +, 0)$ also has a quantifier-free definition over $(\mathbb{Q}; +, 0)$. This follows from the more general fact that the first-order theory of torsion-free divisible abelian groups admits quantifier elimination (see e.g. Theorem 3.1.9 in [23]; the statement there is for the signature $\{+, -, 0\}$, but there is an explicit comment that having $-$ in the signature is not necessary). Having the constant symbol 0 is a necessary technical detail here, since we cannot eliminate quantifiers for false sentences in $(\mathbb{Q}; +)$, such as $\exists x. x \neq x$. (If we have the constant symbol 0, then this sentence is equivalent to the sentence $0 \neq 0$, which is quantifier-free.)

The following lemma allows us to freely use certain expressions in pp-definitions over $(\mathbb{Q}; R_+)$.

Lemma 4.3. The relation $\{(x_1, \ldots, x_l) \mid r_1 x_1 + \ldots + r_l x_l = 0\}$ is pp-definable in $(\mathbb{Q}; R_+)$ for arbitrary $r_1, \ldots, r_l \in \mathbb{Q}$.

Proof. First observe that we can assume that $r_1, \ldots, r_l$ are integers, because we can multiply the equation $r_1 x_1 + \cdots + r_l x_l = 0$ by the least common multiple of the denominators of $r_1, \ldots, r_l$ and obtain an equivalent equation.

The proof is by induction on l. We first show how to express equations of the form $r_1 x_1 + r_2 x_2 = x_3$. By setting x_1 and x_2 to 0, this will solve the case $l = 1$, and by setting x_3 to 0 this will also solve the case $l = 2$ (recall that 0 is pp-definable in $(\mathbb{Q}; R_+)$).

For positive r_1, r_2, the formula $r_1 x_1 + r_2 x_2 = x_3$ is equivalent to

$$\begin{aligned} \exists u_1, \ldots, u_{r_1}, v_1, \ldots, v_{r_2}. \quad & u_1 = x_1 \wedge \bigwedge_{i=1}^{r_1 - 1} x_1 + u_i = u_{i+1} \\ & \wedge\ v_1 = x_2 \wedge \bigwedge_{i=1}^{r_2 - 1} x_2 + v_i = v_{i+1} \\ & \wedge\ u_{r_1} + v_{r_2} = x_3. \end{aligned}$$

If we replace in this expression all atomic formulas of the form $x + y = z$ by $R_+(x, y, z)$, we obtain the desired pp-definition. The cases that one or two of r_1 and r_2 are negative can be handled analogously.

Now suppose that $l > 2$. By the inductive assumption, there is a pp-definition ϕ_1 for $r_1 x_1 + r_2 x_2 = u$ and a pp-definition ϕ_2 for $r_3 x_3 + \cdots + r_l x_l = v$. Then $\exists u, v. \phi_1 \wedge \phi_2 \wedge R_+(u, v, 0)$ is a pp-definition for $r_1 x_1 + \cdots + r_l x_l = 0$ in $(\mathbb{Q}; R_+)$. □

We now consider the concept of *reduced* formulas; this idea was introduced in [3]. We call a quantifier-free formula ϕ *reduced* if and only if it is not logically equivalent to any of its subformulas, i.e., there is no formula ψ obtained from ϕ by deleting literals or clauses such that ψ and ϕ are logically equivalent. Clearly, such a reduced definition of R always exists because we can find one by successively removing literals and clauses from ϕ.

Recall that $S_{\mathbb{Q}}$ was defined in Section 2 as $\{(x, y, z) \in \mathbb{Q}^3 \mid y \neq z \wedge (x = y \vee x = z)\}$.

Lemma 4.4. If R is first-order, but not quantifier-free Horn definable in $(\mathbb{Q}; +, 0)$, then $S_{\mathbb{Q}}$ has a pp-definition in $(\mathbb{Q}; R, R_+, \neq)$.

Proof. Let $T(x, y) \subsetneq \mathbb{Q}^2$ be the binary relation defined by $x \neq 0 \wedge (y = 0 \vee x = y)$. We first prove that T has a pp-definition in $(\mathbb{Q}; R, R_+, \neq)$. Let ϕ be a reduced first-order definition of R, and let C be a clause of ϕ with two positive literals l_1 and l_2. Because ϕ is reduced, there are $p, q \in R$ such that p satisfies l_1 and does not satisfy any other literal in C, and q satisfies l_2 but does not satisfy any other literal in C.

We claim that the following formula is logically equivalent to $x \neq 0 \wedge (y = 0 \vee x = y)$.

$$\exists z_1, \ldots, z_k. \quad x \neq 0 \quad \wedge \quad \bigwedge_{i=1}^{k} z_i = p_i x + (q_i - p_i) y \quad \wedge$$
$$\bigwedge_{l \in C \setminus \{l_1, l_2\}} \neg l \quad \wedge \quad R(z_1, \ldots, z_k).$$

We note that this formula is equivalent to a primitive positive formula over $(Q; R, R_+, \neq)$ by Lemma 4.3. Now, arbitrarily choose $x \neq 0$. Suppose that $y = 0$. Then the assignment $z_1 = p_1 x, \ldots, z_k = p_k x$ obviously satisfies the first line in the pp-formula. Recall that $p \in R$ and p does not satisfy any literal in C except for l_1. The function $f(a) = x \cdot a$ is in $\mathrm{Aut}(\mathbb{Q}; +)$ whenever $x \neq 0$. Consequently, $f \in \mathrm{Aut}(\mathbb{Q}; R, R_+, \neq)$, too, and the second line in the formula is satisfied as well. Now suppose that $x = y$. Then the assignment $z_1 = q_1 x, \ldots, z_k = q_k x$ obviously satisfies the first line in the pp-formula. By construction, $q \in R$ and q does not satisfy all literals in C except for l_1. Again we conclude that the second line in the formula is satisfied.

For the opposite direction, suppose that $x, y \in \mathbb{Q}$ satisfy the pp-formula. Because of the first line of the formula, $x \neq 0$. Let $z_1, \ldots, z_k$ be the k elements whose existence is asserted in the first line of the formula. Because the formula contains the conjunct $R(z_1, \ldots, z_k)$, the clause C in ϕ is satisfied by $z_1, \ldots, z_k$. Since $z_1, \ldots, z_k$ also satisfy the conjunction of all negated literals in C except for the positive literals l_1 and l_2, at least one of these two literals l_1 and l_2 must be satisfied by $z_1, \ldots, z_k$.

Assume $l_1(x_1, \ldots, x_k) \equiv \sum_{i=1}^{k} c_i x_i = 0$ and $l_2(x_1, \ldots, x_k) \equiv \sum_{i=1}^{k} d_i x_i = 0$. By the choice of p and q, we know that $\sum_{i=1}^{k} c_i p_i = 0$, $\sum_{i=1}^{k} c_i q_i \neq 0$, $\sum_{i=1}^{k} d_i q_i = 0$, and $\sum_{i=1}^{k} d_i p_i \neq 0$. Also note that $z_i = p_i(x - y) + q_i y$ when $1 \leq i \leq k$; this follows immediately from the fact that $z_i = p_i x + (q_i - p_i) y$ by the first line in the formula.

Suppose first that l_1 is satisfied by $z_1, \ldots, z_k$, i.e. $\sum_{i=1}^{k} c_i z_i = 0$. Now,

$$\begin{aligned}\sum_{i=1}^{k} c_i z_i =& \sum_{i=1}^{k} c_i(p_i x - p_i y + q_i y) \\ =& x \cdot \sum_{i=1}^{k} c_i p_i - y \cdot \sum_{i=1}^{k} c_i p_i + y \cdot \sum_{i=1}^{k} c_i q_i = y \cdot \sum_{i=1}^{k} c_i q_i.\end{aligned}$$

Consequently, $y \cdot \sum_{i=1}^{k} c_i q_i = 0$. The point q does not satisfy l_1, so $\sum_{i=1}^{k} c_i q_i \neq 0$. Thus, $y = 0$ and we are done in this case.

Suppose instead that l_2 is satisfied by $z_1, \ldots, z_k$, i.e. $\sum_{i=1}^{k} d_i z_i = 0$. We see that

$$\sum_{i=1}^{k} d_i z_i = \sum_{i=1}^{k} d_i(p_i x - p_i y + q_i y) = (x - y) \cdot \sum_{i=1}^{k} d_i p_i$$

and $(x - y) \cdot \sum_{i=1}^{k} d_i p_i = 0$. Hence, $x - y = 0$ and $y = x$.

Finally, we prove that $S_{\mathbb{Q}}(u, v, w)$ has the following pp-definition in $(\mathbb{Q}; R_+, T)$:

$$\exists x, y.\ R_+(x, v, w)\ \wedge\ R_+(y, v, u)\ \wedge\ T(x, y).$$

Assume first that $(u, v, w) \in S_{\mathbb{Q}}$. Note that $x = w - v$ is not equal to 0 because $v \neq w$. If $u = v$, then $y = 0$, and if $u = w$, then $x = w - v = u - v = y$ so $T(x, y)$ is satisfied.

Conversely, suppose that $(x, y) \in \mathbb{Q}^2$ satisfies the pp-formula above. The formula $T(x, y)$ implies that $x \neq 0$ and hence $w \neq v$. Moreover, $T(x, y)$ implies that $y = 0$ or $x = y$. If $y = 0$, then $u = v$ and $(u, v, w) \in S_{\mathbb{Q}}$. If $x = y$, then $w - v = u - v$ and hence $u = w$. Again (u, v, w) is in $S_{\mathbb{Q}}$. $\square$

We will now use Lemma 4.4 in order to prove the following definability result.

Theorem 4.5. Let Γ be first-order expansion of $(\mathbb{Q}; +)$. Then, either

- Γ has a quantifier-free Horn definition in $(\mathbb{Q}; +, 0)$, or
- every non-empty relation of Γ contains a tuple of the form $(0, \ldots, 0)$, or
- every first-order definable relation in $(\mathbb{Q}; +)$ has a pp-definition in Γ.

Proof. Suppose that there is a non-empty k-ary relation R of Γ that does not contain the tuple $(0, \ldots, 0)$. Then the $(k + 1)$-ary relation $R'(x_1, \ldots, x_{k+1})$ defined by $R(x_1, \ldots, x_k) \wedge x_{k+1} = 0$ is non-empty, and the relation defined by $R'(x, \ldots, x)$ is empty. So we can apply Proposition 4.2 and find that $\neq$ is pp-definable in $(\mathbb{Q}; +, R')$ and hence also in Γ. So assume in the following without loss of generality that Γ contains the relation $\neq$.

Suppose that one of the relations of Γ does not have a quantifier-free Horn definition in $(\mathbb{Q}; +, 0)$. Lemma 4.4 implies that the relation $S_{\mathbb{Q}}$ has a pp-definition in Γ, and Corollary 2.3 implies that every relation with a first-order definition in $(\mathbb{Q}; =)$ has a pp-definition in Γ.

Let R be a relation with a first-order definition in $(\mathbb{Q}; +)$. We have already remarked that R has a quantifier-free definition ϕ in $(\mathbb{Q}; +, 0)$. To find a pp-definition for R in Γ, we introduce for every atomic formula $s = t$ of ϕ two new variables x_s and x_t. We then replace $s = t$ by $x_s = x_t$. The resulting formula consists of a boolean combination of atomic formulas of the form $x = y$, which we know has a pp-definition ϕ' in Γ. Consider the conjuntion of ϕ' with $x_s = s$ and $x_t = t$ over all atomic formulas of the form $s = t$ in ϕ, and let ϕ'' be the formula obtained from this conjunction by existentially quantifying over all new variables. It is straightforward to verify that the resulting formula is a pp-definition of R in Γ. $\square$

4.3. **Complexity Classification.** Theorem 4.5 has consequences for the computational complexity of constraint satisfaction.

Corollary 4.6. Let Γ be a first-order expansion of $(\mathbb{Q}; R_+)$. Then, CSP(Γ) is in P if all relations in Γ have a quantifier-free Horn definition over $(\mathbb{Q}; +, 0)$, or if all non-empty relations contain a tuple of the form $(0, \ldots, 0)$. Otherwise, CSP(Γ) is NP-hard.

Proof. If all relations in Γ have a quantifier-free Horn definition over $(\mathbb{Q}; +, 0)$, then Proposition 3.1 implies that CSP(Γ) is in P. If all non-empty relations contain a tuple of the form $(0, \ldots, 0)$, CSP(Γ) is also in P, since it suffices to test on a given input whether it contains a constraint that denotes the empty relation in Γ, in which case the input is unsatisfiable. Otherwise, the input has the solution that maps every variable to 0.

If neither of these two conditions holds, consider the expansion of Γ by the function $+$. Theorem 4.5 implies that $S_{\mathbb{Q}}$ has a pp-definition ϕ in this expansion. Note that when t is a term that appears in ϕ, then the expression $x = t$ has a pp-definition ψ over $(\mathbb{Q}; R_+)$ by Lemma 4.3. We replace t by a new variable x in ϕ, add ψ as a conjunct to ϕ, and existentially quantify x. We repeat this procedure until ϕ contains no more occurrences of the function symbol $+$, and obtain a primitive positive definition of $S_{\mathbb{Q}}$ in Γ. It follows from Theorem 2.2 that the constraint satisfaction problem for Γ is NP-hard. $\square$

5. Affine Structures over the Rational Numbers

We will now consider affine additive structures over $\mathbb{Q}$, i.e., structures with a first-order definition in $(\mathbb{Q}; f)$ where $f : \mathbb{Q}^3 \to \mathbb{Q}$ is given by $f(a, b, c) = a - b + c$. This structure is very similar to the structure of Section 4: we begin by studying the definability of $\neq$ (Section 5.1) and of the relation $S_{\mathbb{Q}}$ (Section 5.2). Using quantifier-elimination for non-sentences in $(\mathbb{Q}; f)$, we combine these results to obtain a certain result on pp-definability (Section 5.3). Finally, we completely classify the complexity of the corresponding CSPs (Section 5.4). The main proof in Section 5.2, however, is very different from the corresponding proof in Section 4.2.

5.1. **PP-definability of Inequality.** In affine structures, there are only four first-order definable binary relations.

Lemma 5.1. Let Γ be a structure with a first-order definition in $(\mathbb{Q}; f)$. Then there are four first-order definable binary relations: the empty relation, the full relation, the relation $\neq$, and the relation $=$.

Proof. Since first-order formulas are preserved by automorphisms, it suffices to show that Aut(Γ) has precisely two orbits on $\mathbb{Q}^2$, namely

$$O_1 = \{(x, x) \mid x \in \mathbb{Q}\} \quad \text{and} \quad O_2 = \{(x, y) \mid x, y \in \mathbb{Q}, x \neq y\} .$$

These two orbits clearly partition $\mathbb{Q}^2$. It is obvious that O_1 is an orbit, because for every $c \in \mathbb{Q}$, the mapping $x \mapsto x + c$ is an automorphism of $(\mathbb{Q}; f)$ and hence of Γ. To see that O_2 is an orbit of pairs of rationals, we apply linear interpolation: let $(a, b) \in O_2$ and $(c, d) \in O_2$ be arbitrarily chosen. The mapping $x \mapsto \frac{c-d}{a-b}(x - a) + c$ maps (a, b) to (c, d) and it is an automorphism of $(\mathbb{Q}; f)$, and hence of Γ. $\square$

In the proof of Lemma 5.1 we have in fact verified that the automorphism group of Γ is *2-transitive*; a structure Γ is called k-transitive, for $k \geq 1$, if there is only one orbit of k-tuples having pairwise distinct entries with respect to the componentwise action of the automorphism group of Γ on k-tuples.

Lemma 5.2. Let R be a non-empty relation that is first-order definable in $(\mathbb{Q}; f)$. If $(0, ..., 0) \notin R$, then $\neq$ is pp-definable in $\{R\}$.

Proof. Let m denote the arity of R. By 2-transitivity, it follows that $(x, \ldots, x) \notin R$ for any $x \in \mathbb{Q}$. Since R is non-empty, we conclude that $m > 1$. Consider the following pp-definitions of binary relations $R_1, \ldots, R_m$.

$$\begin{aligned}
R_1(x, y) &\equiv \exists z_1, \ldots, z_{m-2}.R(x, y, z_1, \ldots, z_{m-2}) \\
R_2(x, y) &\equiv \exists z_1, \ldots, z_{m-3}.R(x, x, y, z_1, \ldots, z_{m-3}) \\
R_3(x, y) &\equiv \exists z_1, \ldots, z_{m-4}.R(x, x, x, y, z_1, \ldots, z_{m-4}) \\
&\vdots \\
R_{m-3}(x, y) &\equiv \exists z_1, z_2.R(x, \ldots, x, y, z_1, z_2) \\
R_{m-2}(x, y) &\equiv \exists z_1.R(x, \ldots, x, y, z_1) \\
R_{m-1}(x, y) &\equiv R(x, \ldots, x, y) \\
R_m(x, y) &\equiv R(x, \ldots, x)
\end{aligned}$$

Let i be the least number such that R_i is empty; such an i exists since R_m is empty. Note that $i > 1$ since R and, consequently, R_1 are non-empty. Now consider the relation R_{i-1} and note that it is non-empty due to the choice of i. We know (from Lemma 5.1) that the binary relations that are pp-definable in $\{R\}$ is a subset of $\mathbb{Q}^2$, $\neq$, $=$, and $\emptyset$. If R_{i-1} equals $\mathbb{Q}^2$ or the equality relation, then R_i would be non-empty which leads to a contradiction. Hence R_{i-1} equals the relation $\neq$. □

5.2. **PP-definability in Non-Horn Expansions.** The central step of the classification is the following result concerning pp-definability.

Lemma 5.3. Let Γ be a relational structure over an infinite domain D such that the set of pp-definable binary relations in Γ is exactly $\{D^2, \neq, =, \emptyset\}$. Suppose that Γ contains a relation Q such that there are pairwise distinct $1 \leq i, j, k, l \leq n$ for which the following conditions hold:

(1) $Q(x_1, \ldots, x_n) \wedge x_i \neq x_j$ is satisfiable;
(2) $Q(x_1, \ldots, x_n) \wedge x_k \neq x_l$ is satisfiable;
(3) $Q(x_1, \ldots, x_n) \wedge x_i \neq x_j \wedge x_k \neq x_l$ is unsatisfiable.

Then S_D has a pp-definition in Γ.

We simplify the proof of Lemma 5.3 by first proving a slightly restricted version:

Lemma 5.4. Let Γ be a relational structure over an infinite domain D such that such that the pp-definable binary relations in Γ are exactly $\{D^2, \neq, =, \emptyset\}$. Suppose that Γ contains a relation Q such that there are $1 \leq i, j, k \leq n$ for which the following conditions hold:

(1) $Q(x_1, \ldots, x_n) \wedge x_i \neq x_j$ is satisfiable;

(2) $Q(x_1, \ldots, x_n) \wedge x_i \neq x_k$ is satisfiable;
(3) $Q(x_1, \ldots, x_n) \wedge x_i \neq x_j \wedge x_i \neq x_k$ is unsatisfiable.

Then S_D has a pp-definition in Γ.

Proof. The indices i, j, k must be pairwise distinct, so suppose for the sake of notation that $i = 1$, $j = 2$, $k = 3$. Consider the relation R defined by

$$R(x_1, x_2, x_3) \equiv \exists x_4, \ldots, x_n.Q(x_1, \ldots, x_n) \wedge x_2 \neq x_3.$$

We first note that R is a non-empty relation: $Q(x_1, \ldots, x_n)$ is satisfiable so the only way of making R empty is that every tuple $(s_1, \ldots, s_n)$ in Q satisfies $s_2 = s_3$. This is impossible since we know that there exists a tuple $(s_1, \ldots, s_n) \in Q$ such that $s_1 \neq s_2$. This implies $s_1 \neq s_3$ and contradicts the third condition.

By condition (3), it is clear that $R \subseteq S_D$. Define $R'(x_1, x_3)$ to be $R(x_1, x_1, x_3)$. By conditions (2) and (3), $R(x_1, x_2, x_3) \wedge x_1 \neq x_3$ is satisfiable while $R(x_1, x_2, x_3) \wedge x_1 \neq x_2 \wedge x_1 \neq x_3$ is not satisfiable. Thus, $R(x_1, x_2, x_3) \wedge x_1 = x_2$ has to be satisfiable and R' is non-empty. Furthermore, $R' \subseteq (\neq)$ since if $(a, a) \in R'$, then $(a, a, a) \in R$ which contradicts the definition of R. By Lemma 5.1, $R' = (\neq)$ so, for all $a \neq b$, $R(a, a, b)$ holds. Similarly, $R(a, b, a)$ holds for all $a \neq b$ by considering $R''(x, y) \equiv R(x, y, x)$. We have shown that $R = S_D$. ☐

Proof of Lemma 5.3. Assume for notational simplicity that $i = 1, j = 2, k = 3$, and $l = 4$. Define the 4-ary relation R by

$$R(x_1, x_2, x_3, x_4) \equiv \exists x_5, \ldots, x_n.Q(x_1, \ldots, x_n)$$

and consider the formula $\phi = R(x, y, x', y') \wedge R(z', y', z, y) \wedge x' \neq z'$. We claim that $\phi \wedge x \neq y$ and $\phi \wedge y \neq z$ are satisfiable while $\phi \wedge x \neq y \wedge y \neq z$ is not satisfiable. Then we can apply Lemma 5.4 and are done. First we make an observation:

<u>Observation 1.</u> Define relation R_1 such that

$$R_1(u, v) \equiv \exists x, y.R(x, y, u, v) \wedge x \neq y.$$

We know that $R(x, y, u, v) \wedge x \neq y$ is satisfiable so R_1 is a non-empty relation. Since $R_1(u, v) \wedge u \neq v$ is not satisfiable, we conclude that R_1 is a non-empty subset of the equality relation. Consequently, R_1 is the equality relation. Analogously, define R_2 such that

$$R_2(u, v) \equiv \exists z, y.R(u, v, z, y) \wedge z \neq y$$

and note that R_2 is the equality relation, too.

We now prove that $\phi \wedge x \neq y \wedge y \neq z$ is not satisfiable. By using Observation 1, it follows that any solution s satisfies $x' = y'$ and $y' = z'$ — this is impossible due to the clause $x' \neq z'$.

Next, we prove that $\phi \wedge x \neq y$ is satisfiable; the case $\phi \wedge y \neq z$ is symmetric. Consider the relation

$$U(u, v) \equiv \exists w.R(w, u, v, v) \wedge w \neq u.$$

By the conditions on R, we know that U is non-empty. Since U is binary, we also know that U either is the equality relation, the inequality relation, or the full relation. We conclude that U is non-empty and symmetric.

By Observation 1, the clause $x \neq y$ has the effect that every solution s must satisfy $x' = y'$. By the definition of ϕ, the solution also has to satisfy $x' \neq z'$ which implies that

$y' \neq z'$. Observation 1 now tells us that $z = y$ and we conclude that every solution satisfies $x' = y'$ and $z = y$. We define

$$\phi' = R(x, y, x', x') \wedge R(z', y', z, z) \wedge x' \neq z' \wedge x \neq y$$

Thus, ϕ' is satisfiable if and only if $\phi \wedge x \neq y$ is satisfiable. We will now construct a concrete satisfying assignment s to the variables of ϕ'.

Arbitrarily choose a tuple $(a, b) \in U$ and let $s(y) = a$, $s(x') = b$. By the conditions on U, there exists an element c such that $(c, a, b, b) \in R$ and $c \neq a$; we let $s(x) = c$. Furthermore, we know that $s(x') = s(y')$ and $s(z) = s(y)$ so $s(y') = b$ and $s(z) = a$. At this point, we see that the assignment s satisfies the clauses $R(x, y, x', x')$ and $x \neq y$.

We know that $(a, b) \in U$ so $(b, a) \in U$, too, and there exists a value d such that $(d, b, a, a) \in R$ and $d \neq b$. Now, let $s(z') = d$ and note that $R(z', y', z, z)$ is satisfied by s. Finally, $s(x') = b \neq d = s(z')$ so the clause $x' \neq z'$ is satisfied and the proof is completed. □

5.3. **Classification Result.** We are now almost ready to prove our pp-definability result for affine structures. However, we first need to prove that we can eliminate quantifiers in $(\mathbb{Q}; f)$ for formulas that have at least one free variable ('non-sentences'). We have been unable to find a reference for this fact; however, this should be considered to be well-known. We give the argument here for the convenience of the reader. We say that a first-order theory T has quantifier-elimination for non-sentences if for every first-order formula ϕ with at least one free variable, there exists a quantifier-free formula ψ such that $T \models \forall \bar{x}.\ \phi(\bar{x}) \Leftrightarrow \psi(\bar{x})$.

Theorem 5.5 (Theorem 8.4.7 in [17])**.** A theory T has quantifier-elimination for non-sentences if and only if for all models Γ and Δ of T, and for all non-empty sequences $\bar{a}$ from Γ and embeddings e from the substructure of Γ generated by $\bar{a}$ into Δ, there exists an extension Δ' of Δ that satisfies T and an embedding of Γ into Δ' that extends e.

Let T be the first-order theory of $(\mathbb{Q}; f)$. In every model of T, say with domain G, if we arbitrarily fix a point (the precise choice of the point is not of importance since $(\mathbb{Q}; f)$ is 1-transitive) and make it the constant 0, we can define a binary function $+$ by $x + y = f(x, 0, y)$ in $(\mathbb{Q}; f, 0)$. It is straightforward to verify that the structure $(G; +, 0)$ is a torsion-free divisible abelian group, and that $(x, y, z) \mapsto x - y + z$ defines f in $(G; +, 0)$. Let Γ and Δ be two models of T, and let $(G; +, 0)$ and $(D; +, 0)$ be two torsion-free divisible abelian groups obtained from Γ and Δ as above. Let e be an embedding of a substructure of Γ generated by a non-empty sequence $\bar{a}$ of elements of Γ into Δ.

We choose an element a_0 from $\bar{a}$, and let $\bar{b}$ be the sequence given by $b_i = a_i - a_0$. Define g to be the function $x \mapsto e(x + a_0) - e(a_0)$ from the substructure of Γ generated by $\bar{b}$ to Δ. Note that g is in fact a function defined on the substructure generated by $\bar{b}$ in $(G; +, 0)$, and moreover g is an embedding from this substructure into $(D; +, 0)$. Since torsion-free divisible abelian groups have quantifier-elimination, and by the reverse implication of Theorem 5.5, there exists a torsion-free divisible abelian group $(D'; +, 0)$ and an extension g' of g that is an embedding of $(G; +, 0)$ into $(D'; +, 0)$. Let Δ' be the model of T that is definable in $(D'; +, 0)$. Then e' given by $e'(x) = g'(x) + e(a_0)$ is an extension of e that embeds Γ into Δ'. This proves quantifier-elimination for non-sentences for T via Theorem 5.5.

Theorem 5.6. Let Γ be a first-order expansion of $(\mathbb{Q}; f)$. Then, either

- every at least unary relation in Γ has a quantifier-free Horn definition in $(\mathbb{Q}; f)$, or
- every non-empty relation of Γ contains a tuple of the form $(0, \ldots, 0)$, or

- every first-order definable relation in $(\mathbb{Q}; f)$ has a pp-definition in Γ.

Proof. Suppose that there is a non-empty k-ary relation R in Γ that does not contain the tuple $(0, \ldots, 0)$. Lemma 5.2 shows that $\neq$ is pp-definable in Γ We therefore assume in the following without loss of generality that Γ contains the relation $\neq$.

Let R be an n-ary relation in Γ, for $n \geq 1$, that does not have a quantifier-free Horn definition in $(\mathbb{Q}; f)$. Since $(\mathbb{Q}; f)$ has quantifier-elimination for non-sentences, there is a reduced definition $\phi(x_1, \ldots, x_n)$ of R in $(\mathbb{Q}; f)$ (see Section 4). There must be a clause C in ϕ with at least two positive literals l_1 and l_2, where l_1 is of the form $t_1 = t_2$ and l_2 is of the form $t_3 = t_4$ for terms t_1, t_2, t_3, t_4 involving the function symbol f and variables. Let $S(x_1, \ldots, x_n, x_{n+1}, x_{n+2}, x_{n+3}, x_{n+4})$ be the relation defined by

$$\begin{aligned} &\phi(x_1, \ldots, x_n) \ \wedge \bigwedge_{l \in C \setminus \{l_1, l_2\}} \neg l \\ &\wedge\ x_{n+1} = f(t_1, t_2, x_{n+3}) \ \wedge \ x_{n+2} = f(t_3, t_4, x_{n+4}) \ , \end{aligned}$$

To see that the relation S is pp-definable in Γ, note that the formula $\neg l$, where l is of the form $s_1 = s_2$ for terms s_1, s_2, has a pp-definition $\exists u, v.\ u = s_1 \wedge v = s_2 \wedge u \neq v$. We claim that S is non-empty. Arbitrarily choose an assignment $s : \{x_1, \ldots, x_n\} \to \mathbb{Q}$ that satisfies $\phi(x_1, \ldots, x_n) \ \wedge \ \bigwedge_{l \in C \setminus \{l_1, l_2\}} \neg l$, and that satisfies the literal l_1, but not the literal l_2 (such an s exists since ϕ is reduced). Let d_3 be the value of t_3 under the assignment s, and d_4 the value of t_4 under s. Then the extension s' of s that maps $(x_{n+1}, x_{n+2}, x_{n+3}, x_{n+4})$ to $(0, d_3, 0, d_4)$ satisfies the entire formula: $x_{n+1} = f(t_1(\overline{x_1}), t_2(\overline{x_2}), x_{n+3})$ is satisfied by s' because $0 = t_1 - t_2 + 0$, which holds since s' satisfies literal l_1. The assignment s' also satisfies $x_{n+2} = f(t_3, t_4, x_{n+4})$, because $d_3 = t_3 - t_4 + d_4$ due to the choice of d_3 and d_4.

We claim that S satisfies the conditions of Lemma 5.3 (which is applicable due to Lemma 5.1) with respect to the arguments indexed by $n+1, n+3, n+2$, and $n+4$. By the argument above, it is easy to see that $S(x_1, \ldots, x_{n+4}) \wedge x_{n+1} \neq x_{n+3}$ is satisfiable: merely observe that $d_3 \neq d_4$ since s does not satisfy literal l_2. Analogously, $S(x_1, \ldots, x_{n+4}) \wedge x_{n+2} \neq x_{n+4}$ is satisfiable, too.

Let ϕ' be the formula $S(x_1, \ldots, x_{n+4}) \wedge x_{n+1} \neq x_{n+3} \wedge x_{n+2} \neq x_{n+4}$. If ϕ' is satisfiable, then there exists a solution $s : \{x_1, \ldots, x_{n+4}\} \to \mathbb{Q}$. It satisfies either l_1 or l_2. Assume first it satisfies literal l_1. Hence, s satisfies that $t_1 = t_2$, and the constraint $x_{n+1} = f(t_1, t_2, x_{n+3})$ then implies that $x_{n+1} = x_{n+3}$, in contradiction with the constraint $x_{n+1} \neq x_{n+3}$. Similarly, we obtain a contradiction when s satisfies l_2. Therefore ϕ' is not satisfiable. We can now apply Lemma 5.3 and obtain that $S_{\mathbb{Q}}$ is pp-definable over $(\mathbb{Q}; S, \neq)$. The relation S is pp-definable over $(\mathbb{Q}; R, f, \neq)$ and, consequently, $S_{\mathbb{Q}}$ is pp-definable over Γ.

Let S be an arbitrary relation with a first-order definition ϕ in $(\mathbb{Q}; f)$. By quantifier-elimination for $(\mathbb{Q}; f)$ there is a quantifier-free first-order definition ϕ of S over $(\mathbb{Q}; f)$. We can now proceed in the same way as at the end of the proof of Theorem 4.5, using the relation $S_{\mathbb{Q}}$ and the function symbol f, to produce a pp-formula over Γ that is equivalent to ϕ. □

5.4. **Complexity Classification.** The next corollary is a direct consequence of Proposition 3.1, Theorem 2.2, and Theorem 5.3.

Corollary 5.7. Let Γ be a structure with a first-order definition in $(\mathbb{Q}; R_f)$. If every at least unary relation in Γ has a quantifier-free Horn definition in $(\mathbb{Q}; f)$, or if every non-empty relation contains a tuple of the form $(0, \ldots, 0)$, then CSP(Γ) is in P. Otherwise, CSP(Γ) is NP-hard.

Proof. If every at least unary relation in Γ has a quantifier-free Horn definition in $(\mathbb{Q}; f)$, then Proposition 3.1 implies that CSP(Γ) can be solved in polynomial time. If every non-empty relation contains a tuple of the form $(0, \ldots, 0)$, then containment in P is trivial. Suppose that some at least unary relation in Γ does not have a quantifier-free Horn definition in $(\mathbb{Q}; f)$, and some non-empty relation does not contain a tuple of the form $(0, \ldots, 0)$. Lemma 5.2 now shows that the relation $\neq$ is pp-definable in Γ. Then, by Lemma 5.3, the relation $S_{\mathbb{Q}}$ has a pp-definition ϕ in Γ. Hardness of CSP(Γ) follows from Theorem 2.2. □

6. Concluding Remarks

We have presented classification results for certain algebraic constraint satisfaction problems, and the results are to a large extent based on dichotomy results for logical definability. We feel that the results and ideas presented in this article can be extended in many different directions. Hence, it seems worthwhile to provide some concrete suggestions for future work.

The results and proof techniques in Section 4 appear to be generalisable to many different templates defined over various structures. One example is the natural and important class of structures that are definable in *Presburger arithmetic* [24], i.e., structures that are first-order definable over $(\mathbb{Z}; +, 1)$. Studying the full theory of Presburger arithmetic is probably too difficult with current methods, but it is possible to approach related theories. For instance, the following result can be obtained by slightly modifying Corollary 4.6.

Corollary 6.1. Let Γ be a relational structure with a quantifier-free first-order definition in $(\mathbb{Z}; +)$ that contains the relation $\{(x, y, z) \in \mathbb{Z}^3 \mid x + y = z\}$. Then, CSP($\Gamma$) is in P if all relations in Γ have a quantifier-free Horn definition over $(\mathbb{Z}; +)$, or if all non-empty relations contain a tuple of the form $(0, \ldots, 0)$. Otherwise, CSP(Γ) is NP-hard.

There is an important difference between this result and a full classification result: we have replaced *first-order definability* with *quantifier-free first-order definability* in the statement of the result, and the reason is that $(\mathbb{Z}; +)$ does not admit quantifier elimination. Is there still a complexity dichotomy if we look at the class of CSPs with a template that is first-order definable in $(\mathbb{Z}; +)$? This appears to be a non-trivial question.

The results presented in Section 5 have strong connections with earlier work on the complexity of *disjunctive* constraints [9, 14]. We say that $\neq$ is *1-independent* with respect to a τ-structure Γ if and only if for every primitive positive τ-formula ϕ with free variables x, y, z, w the following holds: if $\phi \wedge x \neq y$ and $\phi \wedge z \neq w$ are satisfiable, then so is $\phi \wedge x \neq y \wedge z \neq w$. Assume that CSP($\Gamma$) is tractable and let Γ' denote the set of all relations that can be defined by (quantifier-free) conjunctions of disjunctions over Γ containing at most one literal that is not of the form $x \neq y$. The following has been shown in [9, 14].

Theorem 6.2. Let Γ and Γ' be defined as above, and assume that P $\neq$ NP. Then CSP(Γ') is tractable if and only if $\neq$ is 1-independent with respect to Γ.

This result does not imply our result since it only makes a statement about a constraint language Γ' of the form described above.

We have already mentioned that the structures studied in this article are in general not ω-categorical. However, torsion-free divisible abelian groups such as $(\mathbb{Q}; +)$ and all structures first-order definable in such groups are *strongly minimal* (Corollary 3.1.11 in [23]), i.e., every subset that is definable with parameters in Γ is either finite or cofinite. It follows (see e.g. Corollary 6.1.12 in [23]) that all those structures are *uncountably categorical*, i.e., have only one model (up to isomorphism) for each uncountable cardinal. This is interesting from a constraint satisfaction point of view because of the following preservation theorem.

Theorem 6.3 (from [4])**.** Let Γ be an uncountably categorical structure with a countable relational signature and a domain of cardinality larger than 2^ω. Then a first-order definable relation R has a pp-definition in Γ if and only if R is preserved by all *infinitary* polymorphisms of Γ.

Note that this theorem is weaker than the corresponding theorem for ω-categorical structures [8], because we have to assume that R is first-order definable, and that R is not only preserved by the finitary, but also by the infinitary polymorphisms of Γ. Since our classification result is purely in terms of primitive positive definability of first-order definable relations, it is an interesting question to describe the polymorphisms that guarantee tractability for structures Γ with a first-order definition in $(\mathbb{R}; +)$ (Theorem 6.3 shows that such polymorphisms do exist).

Acknowledgements

We want to thank Barnaby Martin for comments on an earlier version of the article, Martin Hils for discussions around quantifier elimination, and the anonymous reviewers for constructive criticism. Peter Jonsson is partially supported by the *Center for Industrial Information Technology* (Ceniit) under grant 04.01 and by the *Swedish Research Council* (VR) under grants 2006-4532 and 621-2009-4431. Manuel Bodirsky has received funding from the *European Research Council* under the European Community's Seventh Framework Programme (FP7/2007-2013 Grant Agreement no. 257039).

References

[1] M. Bodirsky. Constraint satisfaction problems with infinite templates. In H. Vollmer, editor, *Complexity of Constraints (a collection of survey articles)*, pages 196–228. Springer, LNCS 5250, 2008.

[2] M. Bodirsky, H. Chen, J. Kara, and T. von Oertzen. Maximal infinite-valued constraint languages. *Theoretical Computer Science*, 410:1684–1693, 2009.

[3] M. Bodirsky, H. Chen, and M. Pinsker. The reducts of equality up to primitive positive interdefinability. *Journal of Symbolic Logic*, 75(4):1249–1292, 2010.

[4] M. Bodirsky, M. Hils, and B. Martin. On the scope of the universal-algebraic approach to constraint satisfaction. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS-2010)*, 2010.

[5] M. Bodirsky, P. Jonsson, and T. von Oertzen. Essential convexity and complexity of semi-algebraic constraints. *Preprint*, 2010.

[6] M. Bodirsky and J. Kára. The complexity of equality constraint languages. *Theory of Computing Systems*, 43(2):136–158, 2008.

[7] M. Bodirsky and J. Kára. The complexity of temporal constraint satisfaction problems. *Journal of the ACM*, 57(2), 2009. An extended abstract appeared in the proceedings of STOC'08.

[8] M. Bodirsky and J. Nešetřil. Constraint satisfaction with countable homogeneous templates. *Journal of Logic and Computation*, 16(3):359–373, 2006.
[9] M. Broxvall, P. Jonsson, and J. Renz. Disjunctions, independence, refinements. *Artificial Intelligence*, 140(1/2):153–173, 2002.
[10] A. Bulatov. A graph of a relational structure and constraint satisfaction problems. In *Proceedings of the 19th IEEE Symposium on Logic in Computer Science (LICS-2004)*, pages 448–457, 2004.
[11] A. Bulatov, P. Jeavons, and A. Krokhin. The complexity of constraint satisfaction: an algebraic approach (a survey paper). *In: Structural Theory of Automata, Semigroups and Universal Algebra (Montreal, 2003), NATO Science Series II: Mathematics, Physics, Chemistry*, 207:181–213, 2005.
[12] A. Bulatov, A. Krokhin, and P. Jeavons. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34:720–742, 2005.
[13] A. Bulatov and V. Dalmau. A simple algorithm for Mal'tsev constraints. *SIAM Journal on Computing*, 36(1):16–27, 2006.
[14] D. Cohen, P. Jeavons, P. Jonsson, and M. Koubarakis. Building tractable disjunctive constraints. *Journal of the ACM*, 47(5):826–853, 2000.
[15] J. Edmonds. System of distinct representatives and linear algebra. *Journal of Research of the National Bureau of Standards*, 718(4):242–245, 1967.
[16] T. Feder and M. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: a study through Datalog and group theory. *SIAM Journal on Computing*, 28:57–104, 1999.
[17] W. Hodges. *Model theory.* Cambridge University Press, 1993.
[18] W. Hodges. *A shorter model theory.* Cambridge University Press, 1997.
[19] P. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard. Tractability and learnability arising from algebras with few subpowers. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS-2007)*, pages 213–224, 2007.
[20] P. Jeavons, D. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, 1997.
[21] P. Jonsson and C. Bäckström. A unifying approach to temporal constraint reasoning. *Artificial Intelligence*, 102(1):143–155, 1998.
[22] R. E. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM*, 22(1):155–171, 1975.
[23] D. Marker. *Model Theory: An Introduction.* Springer, New York, 2002.
[24] M. Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. *Comptes Rendus du I congres de Mathématiciens des Pays Slaves*, pages 92–101, 1929.
[25] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley - Interscience Series in Discrete Mathematics and Optimization, 1998.