

Constraint Satisfaction and Semilinear Expansions of Addition over the Rationals and the Reals

Peter Jonsson, Johan Thapper

► **To cite this version:**

Peter Jonsson, Johan Thapper. Constraint Satisfaction and Semilinear Expansions of Addition over the Rationals and the Reals. *Journal of Computer and System Sciences*, Elsevier, 2016, 82 (5), pp.912 - 928. 10.1016/j.jcss.2016.03.002 . hal-01796722

HAL Id: hal-01796722

<https://hal-upec-upem.archives-ouvertes.fr/hal-01796722>

Submitted on 21 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Constraint Satisfaction and Semilinear Expansions of Addition over the Rationals and the Reals

Peter Jonsson¹

*Department of Computer and Information Science, Linköpings universitet
SE-58183 Linköping, Sweden*

Johan Thapper*

*Laboratoire d'Informatique Gaspard-Monge, Université Paris-Est Marne-la-Vallée
5 boulevard Descartes, F-77420 Champs-sur-Marne, France*

Abstract

A semilinear relation is a finite union of finite intersections of open and closed half-spaces over, for instance, the reals, the rationals, or the integers. Semilinear relations have been studied in connection with algebraic geometry, automata theory, and spatiotemporal reasoning. We consider semilinear relations over the rationals and the reals. Under this assumption, the computational complexity of the constraint satisfaction problem (CSP) is known for all finite sets containing $R_+ = \{(x, y, z) \mid x + y = z\}$, \leq , and $\{1\}$. These problems correspond to expansions of the linear programming feasibility problem. We generalise this result and fully determine the complexity for all finite sets of semilinear relations containing R_+ . This is accomplished in part by introducing an algorithm, based on computing affine hulls, which solves a new class of semilinear CSPs in polynomial time. We further analyse the complexity of linear optimisation over the solution set and the existence of integer solutions.

Keywords: Constraint satisfaction problems, Semilinear sets, Algorithms, Computational complexity

*Corresponding author.

Email addresses: `peter.jonsson@liu.se` (Peter Jonsson), `thapper@u-pem.fr` (Johan Thapper)

¹The first author was partially supported by the Swedish Research Council (VR) under grant 621-2012-3239.

1. Introduction

We work over a ground set (or domain) X , which will either be the rationals, \mathbb{Q} , or the reals, \mathbb{R} . We say that a relation $R \subseteq X^k$ is *semilinear* if it can be represented as a finite union of finite intersections of open and closed half-spaces in X^k . Alternatively, R is semilinear if it is first-order definable in $\{R_+, \leq, \{1\}\}$ where $R_+ = \{(x, y, z) \in X^3 \mid x + y = z\}$ [1]. Semilinear relations appear in many different contexts within mathematics and computer science: they are, for instance, frequently encountered in algebraic geometry, automata theory, spatiotemporal reasoning, and computer algebra. Semilinear relations have also attained a fair amount of attention in connection with *constraint satisfaction problems* (CSPs). In a CSP, we are given a set of variables, a (finite or infinite) domain of values, and a finite set of constraints. The question is whether or not we can assign values to the variables so that all constraints are satisfied. From a complexity theoretical viewpoint, solving general constraint satisfaction problems is obviously a hard problem. Various ways of refining the problem can be adopted to allow a more meaningful analysis. A common refinement is that of introducing a *constraint language*; a finite set Γ of allowed relations. One then considers the problem $\text{CSP}(\Gamma)$ in which all constraint in the input must be members of Γ . This parameterisation of constraint satisfaction problems has proved to be very fruitful for CSPs over both finite and infinite domains. Since Γ is finite, the computational complexity of such a problem does not depend on the actual representation of the constraints.

The complexity of finite-domain CSPs has been studied for a long time and a powerful algebraic toolkit has gradually formed [2]. Much of this work has been devoted to the *Feder-Vardi conjecture* [3, 4] which posits that every finite-domain CSP is either polynomial-time solvable or NP-complete. Infinite-domain CSPs, on the other hand, constitute a much more diverse set of problems. In fact, *every* computational problem is polynomial-time equivalent to an infinite-domain CSP [5]. Obtaining a full understanding of their computational complexity is thus out of the question, and some further restriction is necessary. In this article, this restriction will be to study semilinear relations and constraint languages.

A relation $R \subseteq X^k$ is said to be *essentially convex* if for all $p, q \in R$ there are only finitely many points on the line segment between p and q that are not in R . A constraint language Γ is said to be *essentially convex* if every relation in Γ is essentially convex. The main motivation for this study is the following result:

Theorem 1 (Bodirsky et al. [6]). *Let Γ be a finite set of semilinear relations over \mathbb{Q} or \mathbb{R} such that $\{R_+, \leq, \{1\}\} \subseteq \Gamma$. Then,*

1. $CSP(\Gamma)$ is polynomial-time solvable if Γ is essentially convex, and NP-complete otherwise; and
2. the problem of optimizing a linear polynomial over the solution set of $CSP(\Gamma)$ is polynomial-time solvable if and only if $CSP(\Gamma)$ is polynomial-time solvable (and NP-hard otherwise).

One may suspect that there are semilinear constraint languages Γ such that $CSP(\Gamma) \in P$ but Γ is not essentially convex. This is indeed true and we identify two such cases. In the first case, we consider relations with large “cavities”. It is not surprising that the algorithm for essentially convex relations (and the ideas behind it) cannot be applied in the presence of such highly non-convex relations. Thus, we introduce a new algorithm which solves CSPs of this type in polynomial time. It is based on computing affine hulls and the idea of improving an easily representable upper bound on the solution space by looking at one constraint at a time; a form of “local consistency” method. In the second case, we consider relations R that are not necessarily essentially convex, but look essentially convex when viewed from the origin. That is, any points p and q that witness a not essentially convex relation lie on a line that passes outside of the origin. We show that we can remove all such holes from R to find an equivalent constraint language that is essentially convex, and thereby solve the problem in polynomial time.

Combining these algorithmic results with matching NP-hardness results and the fact that $CSP(\Gamma)$ is always in NP for a semilinear constraint language Γ (cf. Theorem 5.2 in Bodirsky et al. [6]) yields a dichotomy:

Theorem 2. *Let Γ be a finite set of semilinear constraints that contains R_+ . Then, $CSP(\Gamma)$ is either in P or NP-complete.*

Our result immediately generalises the first part of Theorem 1. It also generalises another result by Bodirsky et al. [7] concerning expansions of $\{R_+\}$ with relations that are first-order definable in $\{R_+\}$. One may note that this class of relations is a severely restricted subset of the semilinear relations since it admits quantifier elimination over the structure $\{+, \{0\}\}$, where $+$ denotes the binary addition function. 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 [8]). One may thus alternatively view relations that are first-order definable in $\{R_+\}$ as finite unions of sets defined by homogeneous linear systems of equations.

We continue by generalising the second part of Theorem 1, too: if Γ is semilinear and contains $\{R_+, \{1\}\}$, then the problem of optimising a linear polynomial over the solution set of $CSP(\Gamma)$ is polynomial-time solvable if

and only if $\text{CSP}(\Gamma)$ is polynomial-time solvable (and NP-hard otherwise). We also study the problem of finding integer solutions to $\text{CSP}(\Gamma)$ for certain semilinear constraint languages Γ . Here, we obtain some partial results but a full classification remains elusive. Our results shed some light on the *scalability property* introduced by Jonsson and Lööw [9].

This article has the following structure. We begin by formally defining constraint satisfaction problems and semilinear relations together with some terminology and minor results in Section 2. The algorithms and tractability results are presented in Section 3 while the hardness results can be found in Section 4. By combining the results from Sections 3 and 4, we prove Theorem 2 in Section 5. We partially generalise Theorem 2 to optimisation problems in Section 6, and we study the problem of finding integer solutions in Section 7. Finally, we discuss some obstacles to further generalisations in Section 8. This article is a revised and extended version of a conference paper [10].

2. Preliminaries

2.1. Constraint satisfaction problems

Let $\Gamma = \{R_1, \dots, R_n\}$ be a finite set of finitary relations over some domain X (which will usually be infinite). We refer to Γ as a *constraint language*. In order to avoid some uninteresting trivial cases, we will assume that all constraint languages are non-empty and contain non-empty relations only.

A first-order 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 each ψ_i is an atomic formula, i.e., either $x = y$ or $R(x_{i_1}, \dots, x_{i_k})$ with R the relation symbol for a k -ary relation from Γ . We call such a formula a *pp-formula*. Note that not all variables have to be existentially quantified; if they are, then we say that the formula is a *sentence*. Given a pp-formula Φ , we let $\text{Vars}(\Phi)$ denote the set of variables appearing in Φ . The atomic formulas $R(x_{i_1}, \dots, x_{i_k})$ in a pp-formula Φ are also called the *constraints* of Φ .

The *constraint satisfaction problem for a constraint language Γ* ($\text{CSP}(\Gamma)$ for short) is the following decision problem:

Problem: $\text{CSP}(\Gamma)$, where Γ is a finite set of relations over a domain X .
Input: A primitive positive sentence Φ over Γ .
Output: ‘yes’ if Φ is true in Γ , ‘no’ otherwise.

The exact representation of the relations in Γ is unessential since we exclusively study finite constraint languages.

A relation $R(x_1, \dots, x_k)$ is *pp-definable from Γ* if there exists a quantifier-free pp-formula φ over Γ such that

$$R(x_1, \dots, x_k) \equiv \exists y_1, \dots, y_n . \varphi(x_1, \dots, x_k, y_1, \dots, y_n).$$

The set of all relations that are pp-definable over Γ is denoted by $\langle \Gamma \rangle$. The following easy but important result explains the role of primitive positive definability for studying the computational complexity of CSPs. We will use it extensively in the sequel without making explicit references.

Lemma 1 (Jeavons [11]). *Let Γ be a constraint language and $\Gamma' \subseteq \langle \Gamma \rangle$ a finite subset. Then $\text{CSP}(\Gamma')$ is polynomial-time reducible to $\text{CSP}(\Gamma)$.*

Let $\Gamma = \{R_1, \dots, R_k\}$ and $\Gamma' = \{R'_1, \dots, R'_k\}$ be two constraint languages such that R_i and R'_i are relations of the same arity, for all $i = 1, \dots, k$. Given an instance Φ of $\text{CSP}(\Gamma)$, let Φ' denote the instance where each occurrence of a relation R_i is replaced by R'_i . We say that $\text{CSP}(\Gamma)$ is *equivalent* to $\text{CSP}(\Gamma')$ if Φ is true in Γ if and only if Φ' is true in Γ' . It is clear that if $\text{CSP}(\Gamma)$ and $\text{CSP}(\Gamma')$ are equivalent CSPs, then they have the same complexity (up to a trivial linear-time transformation).

2.2. Semilinear relations

The domain, X , of every relation in this article will be the set of rationals, \mathbb{Q} , or the set of reals, \mathbb{R} . In all cases, the set of coefficients, Y , will be the set of rationals, but in order to avoid confusion, we will still make this explicit in our notation. We define the following sets of relations.

- $LE_X[Y]$ denotes the set of linear equalities over X with coefficients in Y .
- $LI_X[Y]$ denotes the set of (strict and non-strict) linear inequalities over X with coefficients in Y .

Sets defined by finite conjunctions of inequalities from $LI_X[Y]$ are called *linear sets* or *linear relations*. The set of *semilinear sets* or *semilinear relations*, $SL_X[Y]$, is defined to be the set of finite unions of linear sets. We will refer to $SL_{\mathbb{Q}}[\mathbb{Q}]$ and $SL_{\mathbb{R}}[\mathbb{Q}]$ as semilinear relations over \mathbb{Q} and \mathbb{R} , respectively. One should be aware of the representation of objects in $LE_X[Y]$ and $LI_X[Y]$ compared to $SL_X[Y]$. In $LE_X[Y]$ and $LI_X[Y]$, we view the equalities and inequalities as syntactic objects which we can use for building logical formulas. Now, recall the definition of a linear set: it is defined by a *conjunction* of inequalities. However, a linear set is not a logical formula, it is a subset of X^k . The same thing holds for semilinear sets: they are defined by *unions* of

linear sets and should thus not be viewed as logical formulas. This distinction has certain advantages when it comes to terminology and notation but it also emphasises a difference in the way we view and use these objects. The objects in $LE_X[Y]$ and $LI_X[Y]$ are often used in a logical context (such as pp-definitions) while semilinear relations are typically used in a geometric context.

Given a relation R of arity k , let $R|_X = R \cap X^k$ and $\Gamma|_X = \{R|_X \mid R \in \Gamma\}$. We demonstrate that $\text{CSP}(\Gamma)$ and $\text{CSP}(\Gamma|_{\mathbb{Q}})$ are equivalent as constraint satisfaction problems whenever $\Gamma \subseteq SL_{\mathbb{R}}[\mathbb{Q}]$. Thus, we will exclusively concentrate on relations from $SL_{\mathbb{Q}}[\mathbb{Q}]$ in the sequel. Let $\Gamma \subseteq SL_{\mathbb{R}}[\mathbb{Q}]$ and let Φ be an instance of $\text{CSP}(\Gamma)$. Construct an instance Φ' of $\text{CSP}(\Gamma|_{\mathbb{Q}})$ by replacing each occurrence of R in Φ by $R|_{\mathbb{Q}}$. If Φ' has a solution, then Φ has a solution since $R|_{\mathbb{Q}} \subseteq R$ for each $R \in \Gamma$. If Φ has a solution, then it has a rational solution by Lemma 3.7 in Bodirsky et al. [6] so Φ' has a solution, too.

The following lemma is a direct consequence of our definitions: this particular property is often referred to as *o-minimality* in the literature [12].

Lemma 2. *Let $R \in SL_X[Y]$ be a unary semilinear relation. Then, R can be written as a finite union of open, half-open, and closed intervals with endpoints in $Y \cup \{-\infty, \infty\}$ together with a finite set of points in Y .*

The set of semilinear relations can also be defined as those relations that are first-order definable in $\{R_+, \leq, \{1\}\}$ [1]. In particular, $SL_X[Y]$ is closed under pp-definitions.

Lemma 3 (Bodirsky et al. [7, Lemma 4.3]). *Let $r_1, \dots, r_k, r \in \mathbb{Q}$. The relation $\{(x_1, \dots, x_k) \in \mathbb{Q}^k \mid r_1x_1 + \dots + r_kx_k = r\}$ is pp-definable in $\{R_+, \{1\}\}$ and it is pp-definable in $\{R_+\}$ if $r = 0$. Furthermore, the pp-formulas that define the relations can be computed in polynomial time.*

It follows that $LE_{\mathbb{Q}}[\mathbb{Q}] \subseteq \langle \{R_+, \{1\}\} \rangle$ and $LI_{\mathbb{Q}}[\mathbb{Q}] \subseteq \langle \{R_+, <, \leq, \{1\}\} \rangle$. One may also note that every homogeneous linear equation (with coefficients from \mathbb{Q}) is pp-definable in $\{R_+\}$.

2.3. Unary semilinear relations

For a rational c , and a unary relation $U \subseteq \mathbb{Q}$, let $c \cdot U = \{c \cdot x \mid x \in U\} \in \langle \{R_+, U\} \rangle$. When $c = -1$, we will also write $-U$ for $(-1) \cdot U$.

Given a relation $R \subseteq \mathbb{Q}^k$ and two distinct points $a, b \in \mathbb{Q}^k$, we define

$$\mathcal{L}_{R,a,b}(y) \equiv \exists x_1, \dots, x_k \cdot R(x_1, \dots, x_k) \wedge \bigwedge_{i=1}^k x_i = (1-y) \cdot a_i + y \cdot b_i.$$

The relation $\mathcal{L}_{R,a,b}$ is a parameterisation of the intersection between the relation R and a line through the points a and b . Note that $\mathcal{L}_{R,a,b}$ is a member

of $\langle LE_{\mathbb{Q}}[\mathbb{Q}] \cup \{R\} \rangle$ so, by Lemma 3, $\mathcal{L}_{R,a,b}$ is a member of $\langle \{R_+, \{1\}, R\} \rangle$, too.

A k -ary relation R is *bounded* if there exists an $a \in \mathbb{Q}$ such that $R \subseteq [-a, a]^k$. A unary relation U is *unbounded in one direction* if U is not bounded, but there exists an $a \in \mathbb{Q}$ such that one of the following holds: $U \subseteq [a, \infty)$; or $U \subseteq (-\infty, a]$. A unary relation is called a *BNU* (for *bounded, non-constant, and unary*) if it is bounded and contains more than one point.

Lemma 4. *Let U be a unary relation in $SL_{\mathbb{Q}}[\mathbb{Q}]$ that is unbounded in one direction. Then,*

1. $\langle \{R_+, \{1\}, U\} \rangle$ contains a BNU.
2. if, in addition, U contains both positive and negative elements, then $\langle \{R_+, U\} \rangle$ contains a non-empty bounded unary relation.

PROOF. (1) By Lemma 2, there exists an $a > 0$ such that either

- (i) $(-\infty, -a] \cap U = \emptyset$ and $[a, \infty) \subseteq U$; or
- (ii) $(-\infty, -a] \subseteq U$ and $[a, \infty) \cap U = \emptyset$.

Assume that (i) holds. (The remaining case follows by considering $-U$.) By choosing a rational $b > 2a$, it is not hard to see that the relation

$$U'(x) \equiv \exists y . y = b - x \wedge U(x) \wedge U(y)$$

is bounded and contains an interval. The result then follows from Lemma 3.

(2) Assume that (i) holds and let $c \in U$ be a negative element. (The remaining case follows by considering $-U$.) Then,

$$U''(x) \equiv \exists y . ay = cx \wedge U(x) \wedge U(y)$$

is bounded and contains the element a . The result again follows from Lemma 3. \square

For a unary semilinear relation $T \subseteq \mathbb{Q}$, and a rational $\delta > 0$, let $T + \mathcal{I}(\delta)$ denote the set of unary semilinear relations U such that $T \subseteq U$ and for all $x \in U$, there exists a $y \in T$ with $|x - y| < \delta$.

Example 1. *The set $\{-1, 1\} + \mathcal{I}(\frac{1}{2})$ contains all unary relations U such that $\{-1, 1\} \subseteq U \subseteq (-\frac{3}{2}, -\frac{1}{2}) \cup (\frac{1}{2}, \frac{3}{2})$.*

Lemma 5. *Let $U \neq \emptyset$ be a bounded unary semilinear relation such that $U \subseteq (\varepsilon, \infty)$ for some $\varepsilon > 0$. Then, $\langle \{R_+, U\} \rangle$ contains a relation $U_\delta \in \{1\} + \mathcal{I}(\delta)$, for every rational $\delta > 0$.*

PROOF. Let $U^+ = \sup U$ and $U^- = \inf U$. By Lemma 2, there exist elements $p^+, p^- \in U$ with $p^+ > U^+(1 + \delta)^{-1}$ and $p^- < U^-(1 + \delta)$. The relation $U_\delta := (p^-)^{-1} \cdot U \cap (p^+)^{-1} \cdot U$ is pp-definable in $\{R_+, U\}$ and satisfies: $1 \in U_\delta$, $\sup U_\delta \leq U^+(p^+)^{-1} < 1 + \delta$, and $\inf U_\delta \geq U^-(p^-)^{-1} > 1 - \delta U^-(p^-)^{-1} \geq 1 - \delta$. \square

Lemma 6. *Let U be a bounded unary semilinear relation such that $U \cap (-\varepsilon, \varepsilon) = \emptyset$ for some $\varepsilon > 0$ and $U \cap -U \neq \emptyset$. Then, $\langle \{R_+, U\} \rangle$ contains a relation $U_\delta \in \{-1, 1\} + \mathcal{I}(\delta)$, for every rational $\delta > 0$.*

PROOF. Let $T = U \cap -U$. The proof then follows using a similar construction as in the proof of Lemma 5. \square

2.4. Essential convexity

Let R be a k -ary relation over \mathbb{Q} . The relation R is *convex* if for all $p, q \in R$, R contains all points on the line segment between p and q . We say that R is *essentially convex* if for all $p, q \in R$ there are only finitely many points on the line segment between p and q that are not in R .

We say that R *excludes an interval* if there are $p, q \in R$ and real numbers $0 < \delta_1 < \delta_2 < 1$ such that $p + (q - p)y \notin R$ whenever $\delta_1 \leq y \leq \delta_2$. Note that we can assume that δ_1, δ_2 are rational numbers, since we can choose any two distinct rational numbers $\gamma_1 < \gamma_2$ between δ_1 and δ_2 instead of δ_1 and δ_2 .

If R is *not* essentially convex, and if p and q are such that there are infinitely many points on the line segment between p and q that are not in R , then we say that p and q *witness* that R is not essentially convex. Due to Lemma 2, we conclude that a semilinear relation is essentially convex if and only if it does not exclude an interval. We say that a constraint language is essentially convex if all its relations are essentially convex.

Theorem 3 (Bodirsky et al. [6, Theorems 5.1 and 5.4]). *If Γ is a finite set of essentially convex semilinear relations, then $\text{CSP}(\Gamma)$ is in P .*

3. Tractability

In this section, we present our two main sources of tractability. Section 3.1 contains a new algorithm for semilinear constraint languages Γ containing $\{R_+, \{1\}\}$ and such that $\langle \Gamma \rangle$ does not contain a BNU. In Section 3.2, we extend the applicability of Theorem 3 from essentially convex semilinear constraint languages to a certain class of semilinear CSPs that are not essentially convex.

3.1. Affine consistency

Instead of computing the exact solution set to a CSP instance, our approach will be to reduce an upper bound on this set as far as possible. In particular, we will maintain a representation of an *affine subspace* that is guaranteed to contain the solution set, and repeatedly intersect this subspace with every constraint in order to attempt to reduce it further. This can be seen as a form of *local consistency*. If we manage to reduce the upper bound to an empty set, then we are certain that the instance is unsatisfiable. We will show that under certain conditions, the converse holds; if the upper bound is non-empty, then there are necessarily solutions. To formalise this idea, we will need some definitions.

For a subset $S \subseteq \mathbb{Q}^n$, let $\text{aff}(S)$ denote the *affine hull of S in \mathbb{Q}^n* :

$$\text{aff}(S) = \left\{ \sum_{i=1}^k x_i p_i \mid k \geq 1, x_i \in \mathbb{Q}, p_i \in S, \sum_{i=1}^k x_i = 1 \right\}.$$

An *affine subspace* is a subset $S \subseteq \mathbb{Q}^n$ for which $\text{aff}(S) = S$. The points $p_1, \dots, p_k \in \mathbb{Q}^n$ are said to be *affinely independent* if $x_1 p_1 + \dots + x_k p_k = 0$ with $x_1 + \dots + x_k = 0$ implies $x_1 = \dots = x_k = 0$. The dimension, $\dim(S)$, of a set $S \subseteq \mathbb{Q}^n$ is defined to be one less than the maximum number of affinely independent points in S .

We define a notion of consistency for sets of semilinear constraints which we call *affine consistency*. Let V be a finite set of variables and let $n = |V|$. A set of constraints $R_i(x_{i_1}, \dots, x_{i_k})$ with $\{x_{i_1}, \dots, x_{i_k}\} \subseteq V$ is *affinely consistent with respect to a non-empty affine subspace $\emptyset \neq A \subseteq \mathbb{Q}^V$* if $\text{aff}(\hat{R}_i \cap A) = A$ for all i , where $\hat{R}_i := \{(x_1, \dots, x_n) \in \mathbb{Q}^V \mid (x_{i_1}, \dots, x_{i_k}) \in R_i\}$.

Algorithm 1: Affine consistency

Input: A set of constraints $\{R_i(x_{i_1}, \dots, x_{i_k})\}$ over variables V

Output: “yes” if establishing affine consistency among the constraints results in a non-empty affine subspace, “no” otherwise

```

1  $A := \mathbb{Q}^V$ 
2 repeat
3   foreach constraint  $R_i(x_{i_1}, \dots, x_{i_k})$  do
4      $A := \text{aff}(\hat{R}_i \cap A)$ 
5   end
6 until  $A$  does not change
7 if  $A \neq \emptyset$  then return “yes” else return “no”

```

To find an affine subspace A with respect to which a given set of constraints is affinely consistent, it suffices to initialise $A := \mathbb{Q}^V$ and repeatedly apply the

operation $A := \text{aff}(\hat{R}_i \cap A)$ with each of the constraints until A does not change. Algorithm 1 carries out this procedure, which we refer to as *establishing affine consistency*, and answers “yes” if the resulting affine subspace is non-empty and “no” otherwise. In the rest of this section, we show that this algorithm correctly solves $\text{CSP}(\Gamma)$ when $\{R_+, \{1\}\} \subseteq \Gamma$ is a semilinear constraint language such that $\langle \Gamma \rangle$ does not contain a BNU. Furthermore, we show that the algorithm can be implemented to run in polynomial time when applied to constraint languages of this kind.

We begin by proving a technical lemma which is the basis for these results.

Lemma 7. *Let $P = P_1 \cup \dots \cup P_k, Q = Q_1 \cup \dots \cup Q_l \in SL_{\mathbb{Q}}[\mathbb{Q}]$ be two n -ary relations and $P_1, \dots, P_k, Q_1, \dots, Q_l$ linear sets. Assume that neither $\langle LE_{\mathbb{Q}}[\mathbb{Q}] \cup \{P\} \rangle$ nor $\langle LE_{\mathbb{Q}}[\mathbb{Q}] \cup \{Q\} \rangle$ contains a BNU. If $\text{aff}(P) = \text{aff}(Q) =: A$, then $\text{aff}(P_i \cap Q_j) = \text{aff}(P \cap Q) = A$ for some i and j .*

PROOF. The proof is by induction on the dimension $d = \dim(A)$. For $d = 0$, both P and Q consist of a single point p . Clearly, $P_i = \{p\}$ for some i and $Q_j = \{p\}$ for some j . Now assume that $d > 0$ and that the lemma holds for all P', Q' with $\text{aff}(P') = \text{aff}(Q') = A'$ and $\dim(A') < d$. Let p_0, p_1, \dots, p_d be $d + 1$ affinely independent points in P and let q_0, q_1, \dots, q_d be $d + 1$ affinely independent points in Q . For $1 \leq i \leq d$, consider the lines L_i^p through p_0 and p_i , and the lines L_i^q through q_0 and q_i . Let $H = \{y \in \mathbb{Q}^n \mid \alpha \cdot y = 0\}$ ($\alpha \in \mathbb{Q}^n$) be a hyperplane in \mathbb{Q}^n through the origin that is not parallel to any of the lines L_i^p or L_i^q . Then, H intersects each of the $2d$ lines. Let $H(c) = \{y \in \mathbb{Q}^n \mid \alpha \cdot y = c\}$ and let $B(c) = \{y \in \mathbb{Q}^n \mid \alpha \cdot y \notin [-c, c]\}$.

Let $T = \mathcal{L}_{P, p_0, p_i} \in \langle LE_{\mathbb{Q}}[\mathbb{Q}] \cup \{P\} \rangle$. Since P contains p_0 and p_i , it follows that T contains 0 and 1. Therefore, T is not a constant, hence it is unbounded. By Lemma 4(1), T is unbounded in both directions. By Lemma 2, $(-\infty, -c) \cup (c, \infty) \subseteq T$ for some large enough constant $c > 0$. Therefore, $B(c_i^p) \cap L_i^p \subseteq P$, for some positive constant c_i^p . An analogous argument shows that $B(c_j^q) \cap L_j^q \subseteq Q$, for some positive constant c_j^q . Let c' be a positive constant such that $p_0, q_0 \notin B(c')$ and let $c = \max\{c', c_i^p, c_j^q \mid 1 \leq i, j \leq d\}$. This ensures that for any $x > c$, $H(x) \cap P$ intersects the lines L_i^p in d affinely independent points, and $H(x) \cap Q$ intersects the lines L_j^q in d affinely independent points.

Let $P'(x) = H(x) \cap P$, $P'_i(x) = H(x) \cap P_i$, $Q'(x) = H(x) \cap Q$, and $Q'_j(x) = H(x) \cap Q_j$. We now have $\text{aff}(P'(x)) = \text{aff}(Q'(x)) = A'(x)$ with $\dim(A'(x)) = \dim(A) - 1$, for every $x > c$. By induction on $P'(x) = P'_1(x) \cup \dots \cup P'_k(x)$ and $Q'(x) = Q'_1(x) \cup \dots \cup Q'_l(x)$, it follows that $\text{aff}(H(x) \cap (P_{i(x)} \cap Q_{j(x)})) = \text{aff}(P'_{i(x)}(x) \cap Q'_{j(x)}(x)) = A'(x)$ for some $i(x)$ and $j(x)$. Thus, to every $x > c$, we associate a pair $(i(x), j(x))$. But there are only finitely many such pairs,

so there exist distinct $x_1, x_2 > c$ with $i(x_1) = i(x_2) = i'$ and $j(x_1) = j(x_2) = j'$. Since $A'(x_1), A'(x_2) \subseteq \text{aff}(P_{i'}(x) \cap Q_{j'}(x))$, $A'(x_1) \cap A'(x_2) = \emptyset$, and $\dim(A'(x_2)) = d - 1 \geq 0$, it follows that $\text{aff}(P_{i'} \cap Q_{j'})$ strictly contains $A'(x_1)$, so we have $A'(x_1) \subset \text{aff}(P_{i'} \cap Q_{j'}) \subseteq \text{aff}(P \cap Q) \subseteq A$, and $\dim(A'(x_1)) = \dim(A) - 1$. Therefore we have the equalities $\text{aff}(P_{i'} \cap Q_{j'}) = \text{aff}(P \cap Q) = A$. The lemma follows. \square

Algorithm 2: Calculate $\text{aff}(R \cap A)$

Input: A semilinear relation $R = R_1 \cup \dots \cup R_k$ and an affine subspace A .

Output: A set of inequalities defining $\text{aff}(R \cap A)$, or \perp if $\text{aff}(R \cap A) = \emptyset$.

```

1 Find  $i$  that maximises  $d_i := \dim(\text{aff}(R_i \cap A))$ .
2 if  $\text{aff}(R_i \cap A) = \emptyset$  then return  $\perp$ 
3 Let  $I$  be the set of inequalities for  $R_i$  and  $J$  be the set of inequalities
  for  $A$ .
4  $S := I \cup J$ 
5 foreach inequality  $\iota \in I \cup J$  do
6   | if  $\dim(\text{aff}(S \setminus \{\iota\})) = d_i$  then
7   |   |  $S := S \setminus \{\iota\}$ 
8   | end
9 end
10 return  $S$ 

```

For a semilinear relation R , we let $\text{size}(R)$ denote the *representation size* of R , i.e., the number of bits needed to describe the arities and coefficients of each inequality in some fixed definition of R .

Lemma 8. *Let $R \in SL_{\mathbb{Q}}[\mathbb{Q}]$ be a relation such that $\langle LE_{\mathbb{Q}}[\mathbb{Q}] \cup \{R\} \rangle$ does not contain a BNU and let $A \subseteq \mathbb{Q}^n$ be an affine subspace. Algorithm 2 computes a set of linear inequalities S defining $\text{aff}(R \cap A)$ in time polynomial in $\text{size}(R) + \text{size}(A)$ and with $\text{size}(S) \leq \text{size}(R) + \text{size}(A)$.*

PROOF. Let $R = R_1 \cup \dots \cup R_k$ be the representation of R as the union of linear sets R_i . By Lemma 7, there exists an i such that $\text{aff}(R \cap A) = \text{aff}(R_i \cap A)$ and since $\text{aff}(R_j \cap A) \subseteq \text{aff}(R \cap A)$ for all j , the algorithm will find such an i on line 1 by simply comparing the dimensions of these sets. If $\text{aff}(R \cap A) = \emptyset$, then the algorithm returns \perp , signalling that the affine hull is empty.

Otherwise, the affine hull of a non-empty polyhedron can always be obtained as a subset of its defining inequalities (cf. Schrijver [13, Section 8.2]).

Here, some of the inequalities may be strict, but it is not hard to see that removing them does not change the affine hull. If $\iota \in I \cup J$ is an inequality that cannot be removed without increasing the dimension of the affine hull, then it is clear that ι still cannot be removed after the loop. Hence, after the loop, no inequality in S can be removed without increasing the dimension of the affine hull. It follows that S itself defines an affine subspace, A_S , and $A_S = \text{aff}(A_S) = \text{aff}(R_i \cap A) = \text{aff}(R \cap A)$.

Using the ellipsoid method, we can determine the dimension of the affine hull of a polyhedron defined by a system of linear inequalities in time polynomial in the representation size of the inequalities [13, Corollary 14.1f]. To handle strict inequalities on line 1, we can perturb these by a small amount, while keeping the representation sizes polynomial, to obtain a system of non-strict inequalities with the same affine hull. The algorithm does at most $|I \cup J| + k$ affine hull calculations. The total time is thus polynomial in $\text{size}(R) + \text{size}(A)$. Finally, the set S is a subset of $I \cup J$, so $\text{size}(A_S) \leq \text{size}(R) + \text{size}(A)$. \square

Theorem 4. *Let $\{R_+, \{1\}\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language. If there is no BNU in $\langle \Gamma \rangle$, then Algorithm 1 correctly solves $\text{CSP}(\Gamma)$ and can be implemented to run in polynomial time.*

PROOF. First, we show that the algorithm terminates with A equal to the affine hull of the solution space of the constraints. Assume that the input consists of the constraints $R_i(x_{i_1}, \dots, x_{i_k})$ over variables V , $i = 1, \dots, m$. Let $Z = \bigcap_{i=1}^m \hat{R}_i$ denote the solution space of the instance. Let A^* denote the value of A when the algorithm terminates. It is clear that Z is contained in A throughout the execution of the algorithm. Therefore, $\text{aff}(Z) = \text{aff}(Z \cap A^*)$ so it suffices to show that $\text{aff}(Z \cap A^*) = A^*$. We will show that $\text{aff}(\bigcap_{i=1}^j \hat{R}_i \cap A^*) = A^*$ for all $j = 1, \dots, m$. When the algorithm terminates, we have $\text{aff}(\hat{R}_i \cap A^*) = A^*$ for every $i = 1, \dots, m$. In particular, the claim holds for $j = 1$. Now assume that the claim holds for $j - 1$. Then, $P = \bigcap_{i=1}^{j-1} \hat{R}_i \cap A^*$ and $Q = \hat{R}_j \cap A^*$ satisfy the requirements of Lemma 7 with $\text{aff}(P) = \text{aff}(Q) = A^*$. Therefore, we can use this lemma to conclude that $\text{aff}(\bigcap_{i=1}^j \hat{R}_i \cap A^*) = \text{aff}(P \cap Q) = A^*$.

Finally, we show that the algorithm can be implemented to run in polynomial time. The call to Algorithm 2 in the inner loop is carried out at most mn times, where $n = |V|$. To represent \hat{R}_i we use the inequalities in the representation of R_i and add $\mathcal{O}(n)$ additional coefficients with value 0 for the variables in $\{x_1, \dots, x_n\} \setminus \{x_{i_1}, \dots, x_{i_k}\}$. The size of \hat{R}_i is therefore at most $\text{size}(R_i) + \mathcal{O}(n)$, so the size of A never exceeds $\mathcal{O}(mn(\text{size}(R) + n))$, where R is a relation with maximal representation size. Therefore, each call to Algorithm 2 takes polynomial time and consequently, the entire algorithm runs in polynomial time. \square

3.2. Essential convexity

We will now identify another family of polynomial-time solvable semilinear CSPs. This time, we base our result on essentially convex semilinear constraint languages (Theorem 3). We extend this result to the situation where we are only guaranteed that all unary relations that are pp-definable in the language are essentially convex. The idea is that even if we do not have the constant relation $\{1\}$ to help us identify excluded intervals, we are still able to see excluded full-dimensional holes. We follow up this by showing that we can remove certain lower-dimensional holes and thus recover an equivalent essentially convex constraint language. We remind the reader that the dimension of a set is defined with respect to its affine hull, as in Section 3.1.

For $x, y \in \mathbb{Q}^k$, we let $\|x\|$ denote the euclidean norm of x , and $\text{dist}(x, y) = \|x - y\|$ the euclidean distance between x and y .

Lemma 9. *Let $U \in \{1\} + \mathcal{I}(c)$ for some $0 < c < 1$ and assume that $R \in SL_{\mathbb{Q}}[\mathbb{Q}]$ is a semilinear relation such that every unary relation in $\langle \{R_+, U, R\} \rangle$ is essentially convex. Then, R can be defined by a formula $\varphi_0 \wedge \neg\varphi_1 \wedge \dots \wedge \neg\varphi_k$, where φ_0 defines a convex semilinear set, and $\varphi_1, \dots, \varphi_k$ are conjunctions over $LI_{\mathbb{Q}}[\mathbb{Q}]$ that define convex sets of dimensions strictly lower than the dimension of the set defined by φ_0 .*

PROOF. Let $\text{conv}(R)$ denote the convex hull of R and let d denote its dimension. The set $\text{conv}(R)$ is semilinear (see, for instance, Stengle et al. [14]). Let φ_0 be a formula for $\text{conv}(R)$ and let $\varphi_1 \vee \dots \vee \varphi_k$ be a formula for $\text{conv}(R) \setminus R$ on quantifier-free DNF over $LI_{\mathbb{Q}}[\mathbb{Q}]$. It remains to show that for each i , the dimension of the convex set S_i defined by φ_i is smaller than d . To prove this, we show that for every point p in S_i , and every $\varepsilon > 0$, there exists a point x in R such that $\text{dist}(p, x) < \varepsilon$. Since every d -dimensional convex set contains a small d -dimensional open ball around every point in its interior, it follows from this that none of the sets S_i can be d -dimensional.

Carathéodory's theorem (cf. Schrijver [13, Section 7.7]) states that for every $p \in \text{conv}(R)$, we can find $m + 1 \leq d + 1$ affinely independent points, $x_0, \dots, x_m \in R$, such that p lies in $B = \text{conv}(\{x_0, \dots, x_m\})$. By induction over m , we show that for every point $b \in B$, and every $\varepsilon > 0$, there is a point $z \in R$ such that $\text{dist}(b, z) < \varepsilon$. For $m = 0$, this statement follows trivially as each x_j was chosen from R . Now assume that $0 < m \leq d$ and that the statement holds for all $0 \leq m' < m$. By the induction hypothesis, the statement holds for the set $A = \text{conv}(\{x_0, \dots, x_{m-1}\})$. Every $b \in B$ can be written as $b = y_b \cdot a + (1 - y_b) \cdot x_m$ for some $a \in A$ and $0 \leq y_b \leq 1$. Let $a' \in R$ be

a point that is at distance at most $\varepsilon/2$ from a and let $b' = y_b \cdot a' + (1 - y_b) \cdot x_m$. Then,

$$\begin{aligned} \text{dist}(b, b') &= \|(y_b \cdot a + (1 - y_b) \cdot x_m) - (y_b \cdot a' + (1 - y_b) \cdot x_m)\| \\ &\leq y_b \|a - a'\| \\ &\leq \varepsilon/2. \end{aligned}$$

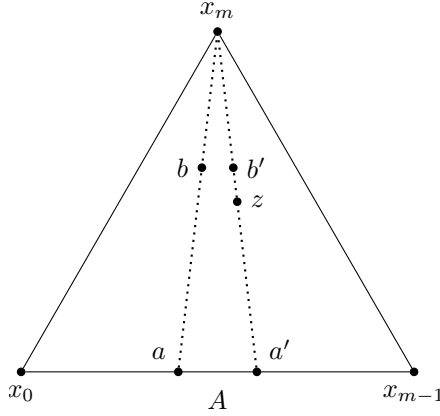


Figure 1: An illustration of the entities involved in the induction step.

Let $\delta > 0$ be a small constant to be fixed later. By Lemma 5, there exists a unary relation $U_\delta \in \{1\} + \mathcal{I}(\delta) \cap \langle \{R_+, U\} \rangle$. Consider the following relation:

$$T(y) \equiv \exists t, z. U_\delta(t) \wedge R(z) \wedge z = y \cdot a' + (t - y) \cdot x_m.$$

Since $U_\delta \in \langle \{R_+, U\} \rangle$, we also have $T \in \langle \{R_+, U, R\} \rangle$. By assumption, T does not exclude an interval, so there exists a y'_b such that $T(y'_b)$ and $|y'_b - y_b| < \delta$. Then, by the definition of T , there exists a $t \in (1 - \delta, 1 + \delta)$ and a point $z \in R$ such that:

$$\begin{aligned} \text{dist}(b', z) &= \|(y_b \cdot a' + (1 - y_b) \cdot x_m) - (y'_b \cdot a' + (t - y'_b) \cdot x_m)\| \\ &= \|(y_b - y'_b) \cdot a' + (1 - t) \cdot x_m + (y'_b - y_b) \cdot x_m\| \\ &\leq \|(y_b - y'_b) \cdot a'\| + \|(1 - t) \cdot x_m\| + \|(y'_b - y_b) \cdot x_m\| \\ &\leq (|y_b - y'_b| + |1 - t| + |y'_b - y_b|) \max\{\|a'\|, \|x_m\|\} \\ &< 3\delta C, \end{aligned}$$

where $C := \max\{\|a'\|, \|x_m\|, 1\}$ is a constant for a fixed B , and the first inequality follows from the triangle inequality.

The claim now follows for the point b by taking $\delta = (\varepsilon/2) \cdot (3C)^{-1}$ since $\text{dist}(b, z) \leq \text{dist}(b, b') + \text{dist}(b', z) < \varepsilon$. \square

Theorem 5. *Let $\{R_+\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language. Assume that there exists a unary relation $U \in \{1\} + \mathcal{I}(c) \cap \langle \Gamma \rangle$, for some $0 < c < 1$, and that every unary relation in $\langle \Gamma \rangle$ is essentially convex. Then, $CSP(\Gamma)$ is equivalent to $CSP(\Gamma')$ for an essentially convex constraint language $\Gamma' \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$.*

PROOF. If Γ is essentially convex, then there is nothing to prove. Assume therefore that Γ is not essentially convex. By Lemma 9, each $R \in \Gamma$ can be defined by a formula $\varphi_0 \wedge \neg\varphi_1 \wedge \dots \wedge \neg\varphi_k$, where $\varphi_0, \varphi_1, \dots, \varphi_k$ are conjunctions over $LI_{\mathbb{Q}}[\mathbb{Q}]$, and $\varphi_1, \dots, \varphi_k$ define sets whose affine hulls are of dimensions strictly lower than that of the set defined by φ_0 . Assume additionally that the formulas are numbered so that the affine hulls of the sets defined by $\varphi_1, \dots, \varphi_m$ do not contain $(0, \dots, 0)$ and that the affine hulls of the sets defined by $\varphi_{m+1}, \dots, \varphi_k$ do contain $(0, \dots, 0)$. Define R' by

$$\varphi \wedge \neg\varphi'_1 \wedge \dots \wedge \neg\varphi'_m \wedge \neg\varphi_{m+1} \wedge \dots \wedge \neg\varphi_k,$$

where φ'_i defines the affine hull of the set defined by φ_i . Then, the constraint language $\Gamma' = \{R' \mid R \in \Gamma\}$ is essentially convex since witnesses of an excluded interval only occur inside an affine subspace not containing $(0, \dots, 0)$; otherwise we could use such a witness to pp-define a unary relation excluding an interval.

Let Φ be an instance of $CSP(\Gamma)$ over the variables $V = \{x_1, \dots, x_n\}$ and assume $\Phi \equiv \exists x_1, \dots, x_n. \psi$ where ψ is quantifier-free. Construct an instance Φ' of $CSP(\Gamma')$ by replacing each occurrence of a relation R in Φ by R' . Clearly, if Φ' is satisfiable, then so is Φ . Conversely, let $s \in \mathbb{Q}^V$ be a solution to Φ and assume that Φ' is not satisfiable. Let L be the line in \mathbb{Q}^V through $(0, \dots, 0)$ and s and let U be the unary relation $\mathcal{L}_{\psi, (0, \dots, 0), s}$. Note that an equation defining the line through $(0, \dots, 0)$ and s is homogeneous. Since $R_+ \in \Gamma$, it follows from Lemma 3 that $U \in \langle \Gamma \rangle$. The instance Φ has at least the solution s which corresponds to the point $1 \in U$. In fact, every point in U corresponds to a solution to Φ on the line L that is not a solution to Φ' since, by assumption, Φ' is not satisfiable.

Fix a constraint $R(x_1, \dots, x_l)$ in Φ and consider the points on the line L that satisfy this constraint but not $R'(x_1, \dots, x_l)$. These are the points $p \in \mathbb{Q}^V$ on L for which $(p(x_1), \dots, p(x_l))$ satisfies $(\varphi'_1 \vee \dots \vee \varphi'_m) \wedge \neg(\varphi_1 \vee \dots \vee \varphi_m)$. For each $1 \leq i \leq m$, φ'_i satisfies at most one point on L since otherwise the affine hull of the relation defined by φ_i would contain $(0, \dots, 0)$. Hence, each constraint $R(x_1, \dots, x_l)$ in Φ can account for at most a finite number of points in U , so U is finite.

Assume first that $|U| > 1$. Then, U is not essentially convex which contradicts the assumption that every unary relation in $\langle \Gamma \rangle$ is essentially convex. Assume instead that $U = \{1\}$, where the single point in U corresponds

to the solution s . Recall that Γ is not essentially convex. Let $R \in \Gamma$ be a k -ary relation that is not essentially convex and let $p, q \in \mathbb{Q}^k$ witness this. The relation $\mathcal{L}_{R,p,q} \in \langle \Gamma \rangle$ since $\{1\} \in \langle \Gamma \rangle$. Then, $\mathcal{L}_{R,p,q} \in \langle \Gamma \rangle$ is unary and not essentially convex which leads to a contradiction. It follows that if Φ is satisfiable, then so is Φ' . \square

4. NP-hardness

We now derive a unified condition for all hard CSPs classified in this article. It is based on a polynomial-time reduction from the NP-hard problem Not-All-Equal 3SAT [15], i.e. the problem $\text{CSP}(\{R_{\text{NAE}}\})$ where $R_{\text{NAE}} = \{-1, 1\}^3 \setminus \{(-1, -1, -1), (1, 1, 1)\}$. The proof is divided into three different lemmas. First, we present a reduction from Not-All-Equal 3SAT to a simple semilinear CSP. We then show that having a BNU T that is bounded away from 0 allows us to pp-define unary relations that are, in a certain sense, close to being either the relation $\{1\}$ or $\{-1, 1\}$. In the final step, we combine these two results and show that having a BNU T that excludes an interval and that is bounded away from 0 is a sufficient condition for $\text{CSP}(\{R_+, T\})$ to be NP-hard.

Lemma 10. *Let $T \in \{-1, 1\} + \mathcal{I}(\frac{1}{2})$. Then, $\text{CSP}(\{R_+, T\})$ is NP-hard.*

PROOF. The proof is by a polynomial time reduction from $\text{CSP}(\{R_{\text{NAE}}\})$. Let Φ denote an arbitrary instance of $\text{CSP}(\{R_{\text{NAE}}\})$. Construct an instance Φ' of $\text{CSP}(\{R_+, T\})$ as follows. Impose the constraint $T(v)$ on each variable. For each constraint $R_{\text{NAE}}(x, y, z)$ in Φ , introduce the constraints $x + y + z + w = 0$ and $T(w)$, where w is a fresh variable.

Assume that Φ has a solution. Consider a constraint $R_{\text{NAE}}(x, y, z)$ in Φ . If two of the variables are assigned the value 1, then the equation $x + y + z + w = 0$ is satisfied by choosing $w = -1$. If two of the variables are assigned the value -1 , then the equation $x + y + z + w = 0$ is satisfied by choosing $w = 1$. Hence, Φ' is satisfiable.

Assume that Φ' has a solution s' . Then, Φ has a solution s defined by $s(x) = 1$ if $s'(x) > 0$ and $s(x) = -1$ if $s'(x) < 0$. Assume to the contrary that $s(x) = s(y) = s(z) = 1$ for some variables with a constraint $R_{\text{NAE}}(x, y, z)$. Consider the equation $x + y + z + w = 0$ in Φ' . By the assumption on T , we have $s'(x) + s'(y) + s'(z) > \frac{3}{2}$, and hence $s'(w) < -\frac{3}{2}$. But this is a contradiction as the constraint $T(w)$ is also in Φ' . We can similarly rule out the case $s(x) = s(y) = s(z) = -1$. This proves that s is a solution to Φ . \square

Lemma 11. *Let $T \neq \emptyset$ be a bounded unary relation such that $T \cap (-\varepsilon, \varepsilon) = \emptyset$, for some $\varepsilon > 0$. Then, either $\langle R_+, T \rangle$ contains a unary relation $U_\delta \in \{1\} + \mathcal{I}(\delta)$*

for every $\delta > 0$; or $\langle R_+, T \rangle$ contains a unary relation $U_\delta \in \{-1, 1\} + \mathcal{I}(\delta)$, for every $\delta > 0$.

PROOF. If $T \cap -T \neq \emptyset$, then the result follows from Lemma 6. Otherwise, by Lemma 2, there exists a constant $c^+ > 0$ such that the set $T^+ = \{x \in T \mid |x| \geq c^+\}$ is non-empty and contains points that are either all positive or all negative. Similarly, there exists a constant $c^- > 0$ such that $T^- = \{x \in T \mid |x| \leq c^-\}$ is non-empty and contains points that are either all positive or all negative. Let $a \in T^+$ and $b \in T^-$. Assume that both sets contain positive points only or that both sets contain negative points only. Then, the result follows using Lemma 5 with the relation $U = a^{-1} \cdot T \cap b^{-1} \cdot T$ (or $-U$ if the points of U are negative). The case when the one set contains positive points and the other contains negative points is handled similarly using the relation $U' = a^{-1} \cdot T \cap b^{-1} \cdot (-T)$. \square

Lemma 12. *Let T be a BNU such that $T \cap (-\varepsilon, \varepsilon) = \emptyset$, for some $\varepsilon > 0$, and U be a unary relation that excludes an interval. Then, $\text{CSP}(\{R_+, T, U\})$ is NP-hard.*

PROOF. We show that $\langle R_+, T, U \rangle$ contains a unary relation $\{-1, 1\} + \mathcal{I}(\frac{1}{2})$. The result then follows from Lemma 10. If already $\langle R_+, T \rangle$ contains such a relation, then we are done. Otherwise, by Lemma 11, $\langle R_+, T \rangle$ contains a unary relation $U_\delta \in \{1\} + \mathcal{I}(\delta)$, for every $\delta > 0$. Since U excludes an interval, there are points $p, q \in U$ and $0 < \delta_1 < \delta_2 < 1$ such that $p + (q - p)y \notin U$ whenever $\delta_1 \leq y \leq \delta_2$. Furthermore, p and q can be chosen so that $\delta_1 < 1/2 < \delta_2$, and by scaling U , we may assume that $|q - p| = 2$. Let $m = (p + q)/2$. Note that $U \cap (m - \varepsilon', m + \varepsilon') = \emptyset$, for some $\varepsilon' > 0$. Similarly, possibly by first scaling T , let $p', q' \in T$ be distinct points with $|q' - p'| = 2$ and let $m' = (p' + q')/2$.

Now, define the following unary relations:

$$\begin{aligned} T_0(x) &\equiv \exists y \exists z . U_\delta(y) \wedge U(z) \wedge z = x \cdot (q - p)/2 + y \cdot m \\ T_\infty(x) &\equiv \exists y' \exists z' . U_\delta(y') \wedge T(z') \wedge z' = x \cdot (q' - p')/2 + y' \cdot m'. \end{aligned}$$

The relations T_0 and T_∞ are roughly translations of U and T , where the constant relation $\{1\}$ has been approximated by the relation U_δ . Since $1 \in U_\delta$, we have $\{-1, 1\} \subseteq T_0, T_\infty$. Hence, if δ is chosen small enough, then the relation $T_0 \cap T_\infty \in \langle R_+, T, U \rangle$ will satisfy the conditions of Lemma 6. This finishes the proof. \square

5. Semilinear expansions of $\{R_+\}$

In this section, we prove our main result: Theorem 2. We divide the proof into two parts. Consider the following two properties:

- (P₀) There is a unary relation U in $\langle \Gamma \rangle$ that contains a positive point and satisfies $U \cap (0, \varepsilon) = \emptyset$ for some $\varepsilon > 0$.
- (P_∞) There is a unary relation U in $\langle \Gamma \rangle$ that contains a positive point and satisfies $U \cap (M, \infty) = \emptyset$ for some $M < \infty$.

In the first part of the proof (Section 5.1), we consider constraint languages that simultaneously satisfy the properties (P₀) and (P_∞). In the second part (Section 5.2), we consider constraint languages that violate at least one of them. In both parts, we give a detailed description of the boundary between easy and hard problems. By combining Theorem 6 and Theorem 8, we establish Theorem 2.

In addition to the two algorithmic results in Sections 3.1 and 3.2, there is also a trivial source of tractability. A relation is *0-valid* if it contains the tuple $(0, \dots, 0)$ and a constraint language is *0-valid* if every relation in it is 0-valid. Every instance of a CSP over a 0-valid constraint language admits the solution that assigns 0 to every variable.

When we consider constraint languages that are not 0-valid, the following lemma shows that there is always a pp-definable unary relation that is not 0-valid.

Lemma 13. *Let $\{R_+\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language. If Γ is not 0-valid, then $\langle \Gamma \rangle$ contains a non-empty unary relation that is not 0-valid.*

PROOF. By assumption, Γ contains some k -ary relation R that is not 0-valid, and by our definition of a constraint language, R is non-empty. Let $t \in R$ be a tuple that contains the largest possible number m of zeroes. Assume for simplicity that the first m entries of t equals 0. Consider the following unary relation in $\langle \Gamma \rangle$.

$$U = \{x \in \mathbb{Q} \mid \exists y_{m+1} \dots y_{k-1} . R(0, 0, \dots, 0, y_{m+1}, \dots, y_{k-1}, x)\}$$

The relation U is non-empty and not 0-valid. □

5.1. The case (P₀) and (P_∞)

The following theorem covers the case when the constraint language satisfies both of the properties (P₀) and (P_∞). As a corollary, we obtain a complete classification for semilinear constraint languages containing $\{R_+, \{1\}\}$. The latter result is interesting in itself and it will also be used in Section 5.2 and Section 6.

Theorem 6. *Let $\{R_+\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language that satisfies (P₀) and (P_∞). The problem $CSP(\Gamma)$ is in P if*

- Γ is 0-valid (trivially);
- $\langle \Gamma \rangle$ does not contain a BNU (by establishing affine consistency); or
- all unary relations in $\langle \Gamma \rangle$ are essentially convex (by a reduction to an essentially convex constraint language).

Otherwise, $\text{CSP}(\Gamma)$ is NP-hard.

PROOF. Let \mathcal{U} be the set of all bounded, non-empty unary relations U in $\langle \Gamma \rangle$ such that $U \cap (-\varepsilon, \varepsilon) = \emptyset$ for some $\varepsilon > 0$. Assume that Γ is not 0-valid. First, we show that \mathcal{U} is non-empty. By Lemma 13, $\langle \Gamma \rangle$ contains a non-empty unary relation that is not 0-valid. Scale this relation so that it contains 1 and call the resulting relation U' . Let $U_0 \in \langle \Gamma \rangle$ be a unary relation witnessing (P_0) and let $U_\infty \in \langle \Gamma \rangle$ be a unary relation witnessing (P_∞) . Scale U_0 and U_∞ so that some positive point from each coincides with 1 and let $T = U' \cap U_0 \cap U_\infty$. If T does not contain a negative point, then $T \in \mathcal{U}$. Otherwise, T contains a negative point b . It follows that $T \cap b \cdot T \in \mathcal{U}$. Hence, the set \mathcal{U} is non-empty.

Assume that $\langle \Gamma \rangle$ does not contain a BNU. Then, neither does \mathcal{U} and hence \mathcal{U} contains only constants. It follows by Theorem 4 that establishing affine consistency solves $\text{CSP}(\Gamma)$.

Otherwise, \mathcal{U} contains a BNU. If all unary relations of $\langle \Gamma \rangle$ are essentially convex, then by Lemma 11 and Theorem 5, $\text{CSP}(\Gamma)$ is equivalent to $\text{CSP}(\Gamma')$ for an essentially convex constraint language Γ' . Tractability follows from Theorem 3.

Finally, if \mathcal{U} contains a BNU and $\langle \Gamma \rangle$ contains a unary relation that excludes an interval, then NP-hardness follows from Lemma 12. \square

Corollary 1. *Let $\{R_+, \{1\}\} \subseteq \Gamma \subseteq \text{SL}_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language. The problem $\text{CSP}(\Gamma)$ is in P if $\langle \Gamma \rangle$ does not contain a BNU or if Γ is essentially convex. Otherwise, $\text{CSP}(\Gamma)$ is NP-hard.*

PROOF. If $\langle \Gamma \rangle$ does not contain a BNU, then tractability follows from Theorem 4. If all relations in Γ are essentially convex, then tractability follows from Theorem 3.

Otherwise, $\langle \Gamma \rangle$ contains a BNU and Γ contains a relation R that is not essentially convex. Let $p, q \in R$ be witnesses to this, and note that $\mathcal{L}_{R,p,q}$ is a unary relation that is not essentially convex and that $\mathcal{L}_{R,p,q} \in \langle \Gamma \rangle$, since $\{R_+, \{1\}\} \subseteq \Gamma$. Since $\{1\} \in \Gamma$ is not 0-valid, NP-hardness then follows from Theorem 6. \square

5.2. The case $\neg(P_0)$ or $\neg(P_\infty)$

Let $\{R_+\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language such that either (P_0) or (P_∞) is violated. In this section, we show that Γ can be replaced by an equivalent constraint language of a restricted type. Let $HSL_{\mathbb{Q}}[\mathbb{Q}]$ denote the set of relations that are finite unions of homogeneous linear sets. We will call such relations *homogeneous* semilinear relations. We remind the reader that we can always pp-define the relations $\{0\}$ and $M = \{(x, -x) \mid x \in \mathbb{Q}\}$ in Γ : $x = 0 \Leftrightarrow R_+(x, x, x)$ and $(x, y) \in M \Leftrightarrow R_+(x, y, 0) \Leftrightarrow \exists z. R_+(x, y, z) \wedge R_+(z, z, z)$. Hence, we can freely use the constant 0 and negation in forthcoming pp-definitions.

From now on, let $\mathbb{Q}_+ = \{a \in \mathbb{Q} \mid a > 0\}$, $\mathbb{Q}_- = \{a \in \mathbb{Q} \mid a < 0\}$, and $\mathbb{Q}_{\neq 0} = \mathbb{Q}_- \cup \mathbb{Q}_+ = \mathbb{Q} \setminus \{0\}$. For a relation $R \in SL_{\mathbb{Q}}[\mathbb{Q}]$, define $\text{cone}(R) = \{\lambda \cdot x \mid \lambda \in \mathbb{Q}_+, x \in R\}$ to be the *cone* over R . For a constraint language $\Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$, let $\text{cone}(\Gamma) = \{\text{cone}(R) \mid R \in \Gamma\}$. Note that, for $\{R_+\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$, we have $\text{cone}(\Gamma) \subseteq HSL_{\mathbb{Q}}[\mathbb{Q}]$, and since $\text{cone}(R_+) = R_+$, we also have $R_+ \in \text{cone}(\Gamma)$.

For an assignment $s : V \rightarrow \mathbb{Q}$ and a rational $c \in \mathbb{Q}$, let $c \cdot s$ denote the assignment $x \mapsto c \cdot s(x)$.

Theorem 7. *Let $\{R_+\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language such that either (P_0) or (P_∞) is violated. Then, $\text{CSP}(\Gamma)$ is equivalent to $\text{CSP}(\text{cone}(\Gamma))$.*

PROOF. Assume that Γ does not satisfy (P_0) . The proof for the case when Γ does not satisfy (P_∞) follows similarly.

Let R be a relation in Γ and let $\varphi = \varphi_1 \vee \dots \vee \varphi_k$ be a quantifier-free DNF formula for R , where each formula φ_j is a conjunction of strict and non-strict inequalities. Remove every disjunct φ_j that contains a non-homogeneous inequality which is not satisfied by the $(0, \dots, 0)$ -tuple. Let S be the relation defined by the resulting formula $\varphi' = \varphi'_1 \vee \dots \vee \varphi'_{k'}$. Since Γ does not satisfy (P_0) , it follows that for every point x in $R \setminus S$, there is a point x' in S that lies on the open line segment between $(0, \dots, 0)$ and x . Therefore, $\text{cone}(S) = \text{cone}(R)$. Next, for each j , let S_j be the relation defined by φ'_j . Remove every non-homogeneous inequality from φ'_j , let φ''_j be the resulting formula and let T_j be the relation defined by φ''_j . Clearly, $\text{cone}(S_j) \subseteq \text{cone}(T_j)$. Let $\lambda \cdot x$ be a point in $\text{cone}(T_j)$ with $\lambda \in \mathbb{Q}_+$ and $x \in T_j$. Since every non-homogeneous inequality in φ'_j is satisfied by the $(0, \dots, 0)$ -tuple, it follows that they are satisfied by every point in a small ball B centred at $(0, \dots, 0)$. Let x' be a point in B on the line segment between $(0, \dots, 0)$ and x and note that every homogeneous inequality in φ'_j satisfies x and therefore also x' . It follows that x' is in S_j so x and $\lambda \cdot x$ are in $\text{cone}(S_j)$, which shows that $\text{cone}(T_j) \subseteq \text{cone}(S_j)$. Let $\varphi'' = \varphi''_1 \vee \dots \vee \varphi''_{k''}$ and let T be

the relation defined by φ'' . Then, $\text{cone}(R) = \text{cone}(T)$ and $\text{cone}(T) = T$ since φ'' only contains homogeneous inequalities. Therefore, φ'' defines $\text{cone}(R)$, so $\text{cone}(R) \in \text{HSL}_{\mathbb{Q}}[\mathbb{Q}]$ and $\text{cone}(\Gamma) \subseteq \text{HSL}_{\mathbb{Q}}[\mathbb{Q}]$.

For the equivalence of $\text{CSP}(\Gamma)$ and $\text{CSP}(\text{cone}(\Gamma))$, arbitrarily choose an instance Φ of $\text{CSP}(\Gamma)$. Construct an instance Φ' of $\text{CSP}(\text{cone}(\Gamma))$ by replacing each occurrence of a relation R in Φ by $\text{cone}(R)$. Every solution to Φ is also a solution to Φ' . It remains to show that if Φ' has a solution, then so does Φ .

Let $s : \text{Vars}(\Phi') \rightarrow \mathbb{Q}$ be a solution to Φ' . If $s \equiv 0$, then it follows immediately that s is a solution to Φ since, for every $R \in \Gamma$, $(0, \dots, 0) \in \text{cone}(R)$ if and only if $(0, \dots, 0) \in R$. Assume therefore that $s \not\equiv 0$. For every constraint $R_i(x_{i_1}, \dots, x_{i_k})$ of Φ , $(s(x_{i_1}), \dots, s(x_{i_k})) \in \text{cone}(R_i)$ holds. By the construction of $\text{cone}(R_i)$, this implies that $r \cdot (s(x_{i_1}), \dots, s(x_{i_k})) \in R_i$, for some $r > 0$. Define the unary relation $U \in \langle \Gamma \rangle$ by the pp-formula:

$$\psi(y) \equiv \exists z_1, \dots, z_k. z_1 = y \cdot s(x_{i_1}) \wedge \dots \wedge z_k = y \cdot s(x_{i_k}) \wedge R_i(z_1, \dots, z_k).$$

Now $r \in U$, so by the assumption on Γ and using Lemma 2, it follows that $(0, \varepsilon_i) \subseteq U$, for some $\varepsilon_i > 0$, and hence that $y \cdot (s(x_{i_1}), \dots, s(x_{i_k})) \in R_i$, for all $y \in (0, \varepsilon_i)$. Let $\varepsilon = \min_i \varepsilon_i$. Then $(\varepsilon/2) \cdot s$ is a solution to Φ . \square

By Theorem 7, it is thus sufficient to determine the computational complexity of $\text{CSP}(\Gamma)$ for $\{R_+\} \subseteq \Gamma \subseteq \text{HSL}_{\mathbb{Q}}[\mathbb{Q}]$.

Given a relation $R \subseteq \mathbb{Q}^k$, we say that a function $e : \mathbb{Q} \rightarrow \mathbb{Q}$ is an *endomorphism* of R if for every tuple $(a_1, \dots, a_k) \in R$, the tuple $(e(a_1), \dots, e(a_k)) \in R$. One may equivalently view an endomorphism as a homomorphism from R to R . We extend this notion to constraint languages $\Gamma = \{R_1, \dots, R_n\}$: a function $e : \mathbb{Q} \rightarrow \mathbb{Q}$ is an endomorphism of Γ if e is an endomorphism of R_i , $1 \leq i \leq n$.

Lemma 14. *Let $a > 0$ be a rational number. Every $R \in \text{HSL}_{\mathbb{Q}}[\mathbb{Q}]$ has the endomorphism $e(x) = a \cdot x$.*

PROOF. We know that R can be written as $R = \bigcup_{i=1}^m H_i$ where H_i , $1 \leq i \leq m$, is defined by a (finite) system of homogeneous linear (strict or non-strict) inequalities. Consider an inequality $\sum_{i=1}^n c_i \cdot x_i \geq 0$ in such a system. We immediately see that

$$\sum_{i=1}^n c_i \cdot x_i \geq 0 \Leftrightarrow a \cdot \sum_{i=1}^n c_i \cdot x_i \geq 0 \Leftrightarrow \sum_{i=1}^n a \cdot c_i \cdot x_i \geq 0 \Leftrightarrow \sum_{i=1}^n c_i \cdot e(x_i) \geq 0.$$

This equivalence also holds if we consider strict inequalities. Therefore, each H_i , $1 \leq i \leq m$, has the endomorphism e .

Now, arbitrarily choose a tuple $t = (t_1, \dots, t_k) \in R$ and assume that $t \in H_i$. It follows that $(e(t_1), \dots, e(t_k)) \in H_i \subseteq R$, so the function e is an endomorphism of R . \square

A direct consequence of Lemma 14 is the following: if an instance Φ of $\text{CSP}(\text{HSL}_{\mathbb{Q}}[\mathbb{Q}])$ has a solution s , then $a \cdot s$ is a solution for every rational number $a > 0$.

The complexity classification of constraint languages that violate either (P_0) or (P_{∞}) , in Theorem 8, follows from two intermediate results which we now present in Lemma 15 and Lemma 16.

Lemma 15. *Let Γ be a subset of $\text{HSL}_{\mathbb{Q}}[\mathbb{Q}]$ and let U be a unary relation in $\langle \Gamma \rangle$. If U contains an element $p > 0$, then $\mathbb{Q}_+ \subseteq U$. If U contains an element $p < 0$, then $\mathbb{Q}_- \subseteq U$.*

PROOF. Let $q \in \mathbb{Q}$ be any element with the same sign as p . By Lemma 14, $e(x) = (q/p) \cdot x$ is an endomorphism of U . Since $p \in U$, it follows that $q = e(p) \in U$. \square

Lemma 16. *Let $\{R_+\} \subseteq \Gamma \subseteq \text{HSL}_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language. Either*

- Γ is 0-valid; or
- $\text{CSP}(\Gamma)$ is polynomial-time equivalent to $\text{CSP}(\Gamma \cup \{\{1\}\})$.

PROOF. Assume that Γ is not 0-valid. By Lemma 13, $\langle \Gamma \rangle$ contains a non-empty unary relation that is not 0-valid. The lemma follows by considering three different cases.

Case 1. $\langle \Gamma \rangle$ contains a non-empty unary relation U such that $0 \notin U$ and $U \subseteq \mathbb{Q}_+$. By Lemma 15, $\mathbb{Q}_+ \subseteq U$ so $U = \mathbb{Q}_+$. We claim that $\text{CSP}(\Gamma \cup \{\{1\}, \mathbb{Q}_+\})$ is polynomial-time equivalent to $\text{CSP}(\Gamma \cup \{\mathbb{Q}_+\})$. The polynomial-time reduction from right to left is trivial. To show the other direction, let Φ be an arbitrary instance of $\text{CSP}(\Gamma \cup \{\{1\}, \mathbb{Q}_+\})$. Assume without loss of generality that the relation $\{1\}$ appears in exactly one constraint $\{1\}(x)$. Construct Φ' by replacing this constraint with $\mathbb{Q}_+(x)$.

If Φ' has no solution, then Φ has no solution. Suppose instead that Φ' has the solution s . Then we know that $s(x) > 0$. Choose $a \in \mathbb{Q}$ such that $a \cdot s(x) = 1$. By Lemma 14, the function $a \cdot s$ is then a solution to Φ .

Case 2. $\langle \Gamma \rangle$ contains a non-empty unary relation U such that $0 \notin U$ and $U \subseteq \mathbb{Q}_-$. By Lemma 15, $\mathbb{Q}_- \subseteq U$ so $U = \mathbb{Q}_-$. We can now pp-define \mathbb{Q}_+ since $x > 0 \Leftrightarrow -x < 0$ and go back to Case 1.

Case 3. $\langle \Gamma \rangle$ contains a non-empty unary relation U such that $0 \notin U$ and no unary relation $U' \in \langle \Gamma \rangle$ equals \mathbb{Q}_+ or \mathbb{Q}_- . Lemma 15 implies that $U = \mathbb{Q}_- \cup \mathbb{Q}_+$.

We claim that $\text{CSP}(\Gamma)$ is polynomial-time equivalent to $\text{CSP}(\Gamma \cup \{\{1\}\})$. The reduction from left to right is trivial. To show the other direction, let $\Phi \equiv \exists x_1, \dots, x_m. \varphi(x_1, \dots, x_m)$ be an arbitrary instance of $\text{CSP}(\Gamma \cup \{\{1\}, U\})$, where φ is quantifier-free, and assume without loss of generality that the relation $\{1\}$ appears in exactly one constraint $\{1\}(x_m)$. Construct Φ' by replacing this constraint with $\mathbb{Q}_{\neq 0}(x_m)$.

If Φ' has no solution, then Φ has no solution. Suppose instead that Φ' has a solution. Assume first that every solution assigns a negative number to the variable x_m . Then we can pp-define a unary relation $T \subseteq \mathbb{Q}_-$ by

$$T(x_m) \equiv \exists x_1, \dots, x_{m-1}. \varphi(x_1, \dots, x_m)$$

and this contradicts our initial assumptions. Thus, there is a solution s such that $s(x_m) > 0$. Choose $a \in \mathbb{Q}$ such that $a \cdot s(x) = 1$. By Lemma 14, the function $a \cdot s$ is a solution to Φ . \square

Theorem 8. *Let $\{R_+\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language that violates (P_0) and/or (P_{∞}) . The problem $\text{CSP}(\Gamma)$ is in P if*

- Γ is 0-valid;
- $\langle \text{cone}(\Gamma) \cup \{\{1\}\} \rangle$ does not contain a BNU; or
- $\text{cone}(\Gamma)$ is essentially convex.

Otherwise, $\text{CSP}(\Gamma)$ is NP-hard.

PROOF. By Theorem 7, $\text{CSP}(\Gamma)$ is equivalent to $\text{CSP}(\text{cone}(\Gamma))$. By Lemma 16, $\text{CSP}(\text{cone}(\Gamma))$ is either trivially in P , if it is 0-valid, or $\text{CSP}(\text{cone}(\Gamma))$ is polynomial-time equivalent to $\text{CSP}(\text{cone}(\Gamma) \cup \{\{1\}\})$. In the latter case, the result follows from Corollary 1. \square

6. Optimisation

In this section, we study the optimisation problem where the objective is to maximise a linear function over the solution set of a semilinear CSP. For an arbitrary constraint language $\Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$, we formally define the problem $\text{Opt}(\Gamma)$ as follows.

Problem: $\text{Opt}(\Gamma)$

Input: A $\text{CSP}(\Gamma)$ -instance Φ and a vector $c \in \mathbb{Q}^{\text{Vars}(\Phi)}$.

Output: One of the following four answers.

- ‘unbounded’ if for every $K \in \mathbb{Q}$, there exists a solution x such that $c^T x \geq K$.
- ‘optimum: K ’ if there exists a $K \in \mathbb{Q}$ and a solution x such that $c^T x = K$, but there is no solution x' such that $c^T x' > K$.
- ‘optimum is arbitrarily close to K ’ if there exists a $K \in \mathbb{Q}$ such that there is no solution x satisfying $c^T x \geq K$, but for every $K' < K$ there is a solution x' with $c^T x' \geq K'$.
- ‘unsatisfiable’ if there is no solution.

By Lemma 3, the problem $\text{Opt}(\{R_+, \leq, \{1\}\})$ is polynomial-time equivalent to linear programming. Bodirsky et al. [6] have shown that for semilinear constraint languages containing $\{R_+, \leq, \{1\}\}$, the problem $\text{CSP}(\Gamma)$ is polynomial-time solvable (NP-hard) if and only if the problem $\text{Opt}(\Gamma)$ is polynomial-time solvable (NP-hard) (cf. Theorem 1).

In Theorem 10, we show that, for semilinear constraint languages containing $\{R_+, \{1\}\}$, the complexity of the decision problem and of the optimisation problem is similarly related. We first prove an analogue of Theorem 4 for the optimisation problem.

Theorem 9. *Let $\{R_+, \{1\}\} \subseteq \Gamma \subseteq \text{SL}_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language. If there is no BNU in $\langle \Gamma \rangle$, then $\text{Opt}(\Gamma)$ can be solved in polynomial time.*

PROOF. Let Φ be an instance of $\text{CSP}(\Gamma)$, let $V = \text{Vars}(\Phi) = \{x_1, \dots, x_m\}$, and let $c \in \mathbb{Q}^V$ be a vector. Assume $\Phi \equiv \exists x_1, \dots, x_m. \varphi$ where φ is quantifier-free. Algorithm 1 in Section 3 finds the affine hull A of the set of satisfying assignments to Φ in polynomial time. If $A = \emptyset$, then we answer ‘unsatisfiable’.

Otherwise, the affine hull A is represented by a set of inequalities, each with representation size that is polynomial in the input size. Therefore, we can solve the system $z_1, z_2 \in A$, $c^T(z_1 - z_2) > 0$, in polynomial time. Assume that this system has a solution. Let $k = \dim(A) + 1$ and let y_1, \dots, y_k be affinely independent satisfying assignments to Φ . Then, we can write $z_1 = \sum_{i=1}^k a_{1i} y_i$ and $z_2 = \sum_{i=1}^k a_{2i} y_i$ with $\sum_{i=1}^k a_{1i} = \sum_{i=1}^k a_{2i} = 1$. Since

$$c^T(z_1 - z_2) = \sum_{i=1}^k a_{1i} c^T y_i - \sum_{i=1}^k a_{2i} c^T y_i > 0,$$

we must have $c^T y_i \neq c^T y_j$ for some $1 \leq i, j \leq k$. Let $U = \mathcal{L}_{R_\varphi, y_i, y_j} \in \langle \Gamma \rangle$, where $R_\varphi = \{(x_1, \dots, x_n) \in \mathbb{Q}^V \mid \varphi(x_1, \dots, x_n) \text{ is true in } \Gamma\}$ and for each $a \in U$, let $y_a \in \mathbb{Q}^V$ denote the corresponding point on the line through y_i and y_j . Fix an arbitrary constant $K \in \mathbb{Q}$. Since there is no BNU in $\langle \Gamma \rangle$, it follows from Lemma 4(1) that U is unbounded in both directions. Since $c^T y_a$ is linear and non-constant, it attains arbitrarily large values, and since U is unbounded in both directions, there is a point $a \in U$ such that $y_a \in R_\varphi$ and $c^T y_a > K$. We can therefore answer ‘unbounded’.

Otherwise, $c^T(z_1 - z_2) = 0$ for all $z_1, z_2 \in A$, so $c^T z$ is constant for $z \in A$. Since A is the affine hull of the set of satisfying assignments to Φ , $c^T z = c^T z'$ for every $z \in A$ and every satisfying assignment z' to Φ . In polynomial time, we can find a $z \in A$ with polynomial representation size. It then suffices to evaluate $K = c^T z$ and answer ‘optimum: K ’. \square

Theorem 10. *Let $\{R_+, \{1\}\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language. The problem $\text{Opt}(\Gamma)$ is polynomial-time solvable if $\langle \Gamma \rangle$ does not contain a BNU or if Γ is essentially convex. Otherwise, $\text{Opt}(\Gamma)$ is NP-hard.*

PROOF. The polynomial-time solvable cases follow from Theorem 1 and Theorem 9. The hardness follows from Corollary 1. \square

A comparison between Theorem 10 and Corollary 1 shows that, for a semilinear constraint language Γ containing $\{R_+, \{1\}\}$, $\text{CSP}(\Gamma)$ is polynomial-time solvable (NP-hard) if and only if $\text{Opt}(\Gamma)$ is polynomial-time solvable (NP-hard). The following example shows that this tight relationship between the complexity of a constraint satisfaction problem and its corresponding optimisation problem cannot be further extended to the class of all semilinear constraint languages containing the relation R_+ .

Example 2. *Let $R = \{(0, 0, 0, 0)\} \cup \{(x, y, z, 1) \mid (x, y, z) \in R_{\text{NAE}}\}$ (cf. Section 4). Then, $\Gamma = \{R, R_+\}$ is semilinear, 0-valid, and $\mathcal{L}_{R, (0,0,0,0), (0,1,1,1)} = \{0, 1\}$ is a unary relation in $\langle \Gamma \rangle$ and hence, Γ satisfies both (P_0) and (P_∞) . Let Φ be an arbitrary instance of $\text{CSP}(\{R_{\text{NAE}}\})$. Construct an instance Φ' of $\text{Opt}(\Gamma)$ by introducing an auxiliary variable w , and for each constraint $R_{\text{NAE}}(x, y, z)$ in Φ , introduce a constraint $R(x, y, z, w)$ in Φ' . Finally, let the vector $c \in \mathbb{Q}^{\text{Vars}(\Phi')}$ be defined by $c_w = 1$ and $c_x = 0$ for all other variables x . Then, the instance Φ has a solution if and only if an optimal solution of Φ' has value 1. We conclude that $\text{CSP}(\Gamma)$ is polynomial-time solvable (since Γ is 0-valid), but that $\text{Opt}(\Gamma)$ is NP-hard.*

7. Integer solutions

In this section, we study the problem of finding *integer* solutions to CSPs defined by semilinear relations. We consider two different approaches: (1) allowing an additional unary constraint that forces a chosen variable to take an integral value, and (2) identifying constraint languages which guarantee the existence of integer solutions.

The reader should note that in the first approach we *do not* consider semilinear relations defined over the integers. Instead, we consider ways of checking whether a given problem instance has a solution where some variables are assigned integral values. Some of the problems in the second approach can be seen as semilinear CSPs over the integers, but our methods do not lend themselves to a systematic study of such problem. See Bodirsky et al. [16] for a recent approach to such a systematic study.

7.1. The relation \mathbb{Z}

The unary relation \mathbb{Z} can be used to ensure that a variable is given an integral value. By Lemma 2, this relation is not semilinear over \mathbb{Q} , so the constraint languages that we classify in the next theorem are formally not semilinear.

Theorem 11. *Let $\{R_+\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language that satisfies (P_{∞}) . The problem $CSP(\Gamma \cup \{\mathbb{Z}\})$ is in P if*

- Γ is 0-valid; or
- $\langle \Gamma \rangle$ does not contain a BNU.

Otherwise, $CSP(\Gamma \cup \{\mathbb{Z}\})$ is NP-hard.

PROOF. If Γ is 0-valid, then $\Gamma \cup \{\mathbb{Z}\}$ is 0-valid, so every instance admits the solution $(0, 0, \dots, 0)$.

Otherwise, assume first that $\langle \Gamma \rangle$ does not contain a BNU. Suppose that Γ does not satisfy (P_0) and let U be a unary relation witnessing that Γ satisfies (P_{∞}) . Then U contains a positive point, so $(0, \varepsilon_1) \subseteq U$ for some $\varepsilon_1 > 0$. Since U is not a BNU, it follows that it must contain negative points. But then, $(-\varepsilon_2, 0) \subseteq U$ for some $\varepsilon_2 > 0$, so $U \cap -U$ is a BNU, contradicting the assumption. Therefore Γ must satisfy (P_0) . By an argument on the set of all bounded, non-empty unary relations in $\langle \Gamma \rangle$ similar to that used in the proof of Theorem 6, it follows that $\langle \Gamma \rangle$ contains the relation $\{1\}$.

Let Φ be an arbitrary instance of $CSP(\Gamma \cup \{\mathbb{Z}\})$, let $I \subseteq \text{Vars}(\Phi)$ be the set of variables that are constrained by \mathbb{Z} in Φ , and let Φ' be the instance of

CSP(Γ) obtained from Φ by removing all \mathbb{Z} -constraints. Let S be the set of satisfying assignments to Φ' . By running Algorithm 1, we obtain a system of inequalities that defines the affine hull A of the satisfying assignments S . We now substitute each such inequality for an equality. The resulting system of linear equalities still defines A . Let $A' = \{\pi_I(x) \mid x \in A\}$, where $\pi_I(x)$ is the projection of x to the coordinates given by the variables in I . We can compute a system of linear equations for A' in polynomial time by first computing a parameter form for A , removing the coordinates not corresponding to I , and then computing the equivalent system of linear equations. This can be in polynomial time by being careful with the representation sizes of the intermediary results (cf. Schrijver [13, Section 3]). We then solve the resulting system of linear equations for an integer solution in polynomial time (cf. Schrijver [13, Corollary 5.3]). If no such solution exists, then Φ is unsatisfiable. Otherwise, the integer points in A' are given by $L = \{c_0 + \sum_{i=1}^k \lambda_i c_i \mid \lambda_1, \dots, \lambda_k \in \mathbb{Z}\}$, for some linearly independent vectors $c_0, \dots, c_k \in \mathbb{Z}^I$, where $k = \dim(A')$. The vectors c_i can be found explicitly in polynomial time, but since we are only interested in showing that there exists a satisfying assignment to Φ , it suffices that L has the aforementioned form.

For $p \in A'$ and constant $\varepsilon > 0$, define $B(p, \varepsilon) = \{x \in A' \mid \|p - x\| < \varepsilon\}$. Let $S' = \{\pi_I(x) \mid x \in S\}$ and note that $S' \in \langle \Gamma \rangle$. Then, by assumption, $\langle R_+, \{1\}, S' \rangle \subseteq \langle \Gamma \rangle$ does not contain a BNU. Furthermore, S' is semilinear, so we can write $S' = S'_1 \cup \dots \cup S'_l$ as a union of linear sets. Since $\text{aff}(S') = \text{aff}(A') = A'$, Lemma 7 is applicable with $S' = S'_1 \cup \dots \cup S'_l$ and A' (which is itself linear). It follows that S' contains a linear set $S'_j \subseteq \mathbb{Q}^I$ such that $\text{aff}(S'_j) = A'$. Let $p \in S'_j$ and $\varepsilon > 0$ be such that $B(p, \varepsilon) \subseteq S'_j \subseteq S'$. We claim that there exist distinct $q_1, q_2 \in L$ such that the line through q_1 and q_2 intersects $B(p, \varepsilon)$ in an open line segment. Let $U = \mathcal{L}_{S', q_1, q_2}$. Since $\langle \Gamma \rangle$ does not contain a BNU, it follows that $(M, \infty) \subseteq U$ for some $M < \infty$. Therefore, $q' = q_1 + t(q_2 - q_1) \in S'$ for a large enough integer t . Hence, there exists a point $q \in S$ such that $\pi_I(q) = q'$, so Φ is satisfiable.

To prove the claim, let $B = B(p, \varepsilon)$ and let $q_1 \in L \setminus B$. Consider the cone $C = \{q_1 + t(x - q_1) \mid x \in B, t \geq 0\}$ and note that C contains $B' := \{q_1 + \delta\varepsilon^{-1}(x - q_1) \mid x \in B\} = B(q_1 + \delta\varepsilon^{-1}(p - q_1), \delta)$. For a large enough positive constant δ , the set $B' \cap L$ is non-empty. Let $q_2 \in B' \cap L \subseteq C$. Then, the line through q_1 and q_2 intersects B in an open line segment.

Finally, assume that $\langle \Gamma \rangle$ contains a BNU U . We may assume that U is not 0-valid: By Lemma 13, $\langle \Gamma \rangle$ contains a non-empty unary relation T that is not 0-valid. Let $c \in \mathbb{Q}$ be a non-zero constant such that $U \cap c \cdot T \neq \emptyset$. If $U \cap c \cdot T$ contains more than one element, then it is a BNU that is not 0-valid. Otherwise, $U \cap c \cdot T$ is a constant unary relation, so $\langle \Gamma \rangle$ contains $\{1\}$. In this case, for a large enough constant $c \in \mathbb{Q}$, the relation $U + c \in \langle \Gamma \rangle$ is a BNU

that is not 0-valid.

Let $r_1, r_2 \in U$ be two distinct points and let $c \in \mathbb{Q}$ be a non-zero constant such that $c \cdot r_1, c \cdot r_2 \in \mathbb{Z}$. Then, $U' = c \cdot U \cap \mathbb{Z}$ is a BNU that excludes an interval and $U' \cap (-1, 1) = \emptyset$. NP-hardness of $\{R_+, U'\} \subseteq \langle \Gamma \cup \{\mathbb{Z}\} \rangle$ therefore follows from Lemma 12. \square

7.2. The integer property

In this section, we will determine those semilinear constraint languages containing R_+ for which knowing that there is a solution guarantees that there is an integer solution. We make the following definition.

Definition 1. *Let Γ be a constraint language over \mathbb{Q} . We say that Γ has the integer property if every instance of $\text{CSP}(\Gamma)$ has a solution if and only if it has an integer solution.*

The integer property can be used to infer tractability of certain semilinear constraint languages over \mathbb{Z} . In particular, if Γ is a semilinear constraint language over \mathbb{Q} that satisfies the integer property, then $\text{CSP}(\Gamma)$ and $\text{CSP}(\Gamma|_{\mathbb{Z}})$ are equivalent. To see that $\Gamma|_{\mathbb{Z}}$ is a semilinear constraint language over \mathbb{Z} , take an arbitrary $R \in \Gamma$ and let φ be a quantifier-free definition of R over $LI_{\mathbb{Q}}[\mathbb{Z}]$. Then, φ is also a quantifier-free definition of $R|_{\mathbb{Z}}$ over $LI_{\mathbb{Z}}[\mathbb{Z}]$.

The following lemma shows that the integer property is preserved under pp-definitions.

Lemma 17. *Let Γ be a constraint language over \mathbb{Q} . If Γ has the integer property, then so does $\langle \Gamma \rangle$.*

PROOF. Let Ψ be an CSP-instance with relations R_1, \dots, R_k from $\langle \Gamma \rangle$, let $\varphi_1, \dots, \varphi_k$ be pp-definitions of R_1, \dots, R_k in Γ , and let Ψ' be the CSP(Γ)-instance obtained from Ψ by replacing each relation R_i by the quantifier-free part of φ_i , and adding existential quantifiers for all auxiliary variables. If Ψ has a rational solution, then Ψ' has a rational solution, so Ψ' has an integer solution. Note that the restriction of any solution of Ψ' to $\text{Vars}(\Psi)$ is a solution to Ψ . Therefore, the restriction of an integer solution of Ψ' to $\text{Vars}(\Psi)$ is an integer solution to Ψ , which proves the lemma. \square

Let Γ denote a semilinear constraint language that contains R_+ . Observe that if $\{1\} \in \langle \Gamma \rangle$, then $\text{CSP}(\Gamma)$ cannot have the integer property since the following CSP($R_+ \cup \{\{1\}\}$)-instance has the unique solution $x = \frac{1}{2}, y = 1$:

$$\exists x, y. x + x = y \wedge \{1\}y.$$

Definition 2. Let Γ be a constraint language over \mathbb{Q} . We say that Γ is scalable if the following holds: for each $R \in \Gamma$ and for each $x = (x_1, \dots, x_k) \in R$, there exists a positive constant A such that $(ax_1, \dots, ax_k) \in R$, for all $a \geq A$.

Clearly, scalable constraint languages cannot contain any unary constant relation $\{c\}$ except when $c = 0$. Note that if Γ has endomorphisms $e(x) = a \cdot x$ for all rational $a > A > 0$, then Γ is indeed scalable. Inferring the existence of endomorphisms from the scalability property is, in general, not straightforward or even possible. The scalability property was originally defined slightly differently [9] but it is easy to verify that the two definitions coincide.

The following result completely characterises the semilinear constraint languages that contain R_+ and have the integer property.

Theorem 12. Let $\{R_+\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language that is not 0-valid. Then, the following are equivalent:

1. Γ has the integer property.
2. every non-empty unary relation in $\langle \Gamma \rangle$ is either $\{0\}$ or unbounded.
3. Γ does not satisfy (P_{∞}) .
4. Γ is scalable.

PROOF. (1) \Rightarrow (2). We show $\neg(2) \Rightarrow \neg(1)$. Suppose that $T_1 \neq \{0\}$ is a non-empty bounded unary relation in $\langle \Gamma \rangle$. By Lemma 13, there is a non-empty unary relation T_2 in $\langle \Gamma \rangle$ that is not 0-valid. Therefore, for some $c \in \mathbb{Q}$, the unary relation $U = T_1 \cap c \cdot T_2$ in $\langle \Gamma \rangle$ is non-empty, bounded, and not 0-valid. Let $k = 1 + \lceil \max(|\sup U|, |\inf U|) \rceil$. Consider the CSP instance

$$\exists x, y . U(x) \wedge k \cdot y = x,$$

and note that it has a solution: arbitrarily choose $x \in U$ and let $y = x/k$. However, it cannot have any integer solution since $0 \notin U$ and k was chosen such that $k > |x|$. Both U and the equation $k \cdot y = x$ are pp-definable in Γ , so the claim follows from Lemma 17.

(2) \Rightarrow (3). We show $\neg(3) \Rightarrow \neg(2)$. Assume that there exists a unary relation U in $\langle \Gamma \rangle$ containing a positive point and $(M, \infty) \cap U = \emptyset$, for some $M < \infty$. If U is bounded, then $\neg(2)$ follows immediately. Otherwise, by Lemma 2, there exists some $M' < \infty$ such that $(M', \infty) \cap U = \emptyset$ and $(-\infty, -M') \subseteq U$. By Lemma 4(2), there exists a non-empty bounded unary relation in $\langle \{R_+, U\} \rangle$ and, consequently, there exists such a relation in $\langle \Gamma \rangle$.

(3) \Rightarrow (4). We show $\neg(4) \Rightarrow \neg(3)$. Arbitrarily choose an n -ary relation $R \in \Gamma$ such that R is not scalable. Arbitrarily choose a tuple $p = (p_1, \dots, p_n) \in R$ that witnesses that R is not scalable, i.e., the set $Y = \{y \geq 1 \mid y \cdot p \notin R\}$ is unbounded. Consider the set $U = \{a \in \mathbb{Q} \mid a \cdot p \in R\}$ and note that U is pp-definable in $\{R, R_+\}$ by Lemma 3:

$$U(x) \equiv \exists y_1, \dots, y_n \cdot y_1 = x \cdot p_1 \wedge \dots \wedge y_n = x \cdot p_n \wedge R(y_1, \dots, y_n).$$

Note that $1 \in U$ so U contains a positive point. Furthermore, since Y is unbounded, it follows from Lemma 2 that $(M, \infty) \subseteq Y$ for some $M < \infty$. Hence, by definition of U , we have $(M, \infty) \cap U = \emptyset$, so Γ satisfies (P_∞) .

(4) \Rightarrow (1). This implication is not difficult to deduce from the proof of Lemma 6 in [9]. We include an argument here for completeness. Assume that Γ is scalable and let Φ be an arbitrary instance of $\text{CSP}(\Gamma)$ with a solution x . Let R_1, \dots, R_m be an enumeration of the atoms of Φ that contain a relation symbol from Γ . Since Γ is scalable, it follows that there exists a constant A_i such that ax satisfies $R_i(x_{i_1}, \dots, x_{i_k})$ for all $a \geq A_i$. Let $A = \max\{A_1, \dots, A_m\}$. Then, ax satisfies all atoms (including the equalities) of Φ , for all $a \geq A$. Therefore, if a is chosen to be a large enough common multiple of the denominators in x , then ax is an integral solution to Φ . \square

As an immediate application of Theorem 12 we give the complement to Theorem 11 in the case when Γ violates (P_∞) .

Corollary 2. *Let $\{R_+\} \subseteq \Gamma \subseteq \text{SL}_{\mathbb{Q}}[\mathbb{Q}]$ be a constraint language that violates (P_∞) . The problem $\text{CSP}(\Gamma \cup \{\mathbb{Z}\})$ is in P if*

- Γ is 0-valid;
- $\langle \text{cone}(\Gamma) \cup \{\{1\}\} \rangle$ does not contain a BNU; or
- $\text{cone}(\Gamma)$ is essentially convex.

Otherwise, $\text{CSP}(\Gamma \cup \{\mathbb{Z}\})$ is NP-hard.

PROOF. If Γ is 0-valid, then $\Gamma \cup \{\mathbb{Z}\}$ is 0-valid, and hence in P. Otherwise, Theorem 12 implies that Γ has the integer property. Therefore, every instance of $\text{CSP}(\Gamma)$ has a solution if and only if it has an integer solution. It follows that $\text{CSP}(\Gamma \cup \{\mathbb{Z}\})$ is polynomial-time equivalent to $\text{CSP}(\Gamma)$. Since Γ violates (P_∞) , the result follows from Theorem 8. \square

8. Discussion

8.1. Generalisations

A natural goal, following the proof of Theorem 2, would be to determine the complexity of $\text{CSP}(\Gamma)$ for an arbitrary semilinear constraint language Γ , i.e., when Γ does not necessarily contain R_+ . Below we indicate a few such attempts and the difficulties that accompany them.

Consider Corollary 1. Our main result, Theorem 2, generalises this by removing the assumption that $\{1\}$ is in Γ . A natural question is then what happens if we instead remove the assumption that the addition relation needs to be in Γ . To this end, let SL^1 denote the set of semilinear constraint languages such that $\{\{1\}\} \subseteq \Gamma$ and $\{R_+\} \not\subseteq \langle \Gamma \rangle$. A straightforward modification of the construction in Section 6.3 of Jonsson and Löw [9] gives the following: for every constraint language Γ' over a finite domain, there exists a $\Gamma \in SL^1$ such that $\text{CSP}(\Gamma')$ and $\text{CSP}(\Gamma)$ are polynomial-time equivalent problems. Hence, a complete classification would give us a complete classification of finite-domain CSPs, and such a classification is a major open question within the CSP community [3, 4, 17]. We also observe that for every *temporal constraint language* (i.e., languages that are first-order definable in $\{<\}$ over the rationals), there exists a $\Gamma \in SL^1$ such that $\text{CSP}(\Gamma')$ and $\text{CSP}(\Gamma)$ are polynomial-time equivalent problems. This follows from the fact that every temporal constraint language Γ' admits a polynomial-time reduction from $\text{CSP}(\Gamma' \cup \{\{1\}\})$ to $\text{CSP}(\Gamma')$: simply equate all variables appearing in $\{1\}$ -constraints and note that any solution can be translated into a solution such that this variable is assigned the value 1. The complexity of temporal constraint languages is fully determined [18] and the polynomial-time solvable cases fall into nine different categories. The proof is complex and it is based on the universal-algebraic approach for studying CSPs. We conclude that a complete classification of the languages in SL^1 will require advanced techniques and will have to be conditioned on the classification of finite-domains CSPs.

A smaller first step towards removing R_+ from Corollary 1 would be to only slightly relax the addition relation. Consider the *affine* addition relation $A_+ = \{(a, b, c, d) \in \mathbb{Q}^4 \mid a - b + c = d\}$. This relation can be viewed as a ‘relaxed’ variant of R_+ since A_+ can be pp-defined in $\{R_+\}$ but not the other way round. Let Γ be a constraint language such that $\{A_+, \{c\}\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$ for some $c \in \mathbb{Q}$. It is not hard to reduce the complexity classification for such constraint language to that of Theorem 2:

Given a relation $R \subseteq \mathbb{Q}^k$ and a rational number $c \in \mathbb{Q}$, let $R+c$ denote the relation $\{(x_1 + c, \dots, x_k + c) \mid (x_1, \dots, x_k) \in R\}$. For instance, $A_+ + c = A_+$.

Similarly, we define $\Gamma + c = \{R + c \mid R \in \Gamma\}$ for constraint languages Γ . Note that $\text{CSP}(\Gamma)$ and $\text{CSP}(\Gamma + c)$ are polynomial-time equivalent problems.

Arbitrarily choose a constraint language $\{A_+, \{c\}\} \subseteq \Gamma \subseteq \text{SL}_{\mathbb{Q}}[\mathbb{Q}]$ and let $\Gamma' = \Gamma + (-c)$. The problem $\text{CSP}(\Gamma')$ is polynomial-time equivalent with $\text{CSP}(\Gamma)$, $A_+ \in \Gamma'$ and $\{0\} \in \Gamma'$. The fact that $A_+ \in \Gamma'$ and $\{0\} \in \Gamma'$ implies that $R_+ \in \langle \Gamma' \rangle$ since $R_+(x, y, z)$ can be pp-defined by

$$\exists w . \{0\}(w) \wedge A_+(x, w, y, z).$$

Consequently, $\text{CSP}(\Gamma')$ and $\text{CSP}(\Gamma' \cup \{R_+\})$ are polynomial-time equivalent problems. We conclude that $\text{CSP}(\Gamma)$ is either in P or NP-complete by Theorem 2.

An interesting way forward would be to classify the complexity of $\text{CSP}(\Gamma)$ for all $\{A_+\} \subseteq \Gamma \subseteq \text{SL}_{\mathbb{Q}}[\mathbb{Q}]$. Such a result would be a substantial generalisation of the results in Section 4 of Bodirsky et al. [7]. Here, we see no obvious obstacles as in the case above for $\Gamma \in \text{SL}^1$.

8.2. The metaproblem

Theorem 2 shows that for every constraint language $\{R_+\} \subseteq \Gamma \subseteq \text{SL}_{\mathbb{Q}}[\mathbb{Q}]$, the problem $\text{CSP}(\Gamma)$ is either in P or NP-complete. This makes the following computational problem (sometimes referred to as a *metaproblem* in the literature) relevant: Given a constraint language $\{R_+\} \subseteq \Gamma \subseteq \text{SL}_{\mathbb{Q}}[\mathbb{Q}]$, is $\text{CSP}(\Gamma)$ in P or NP-complete?

We do not know the complexity of this problem and, in fact, it is not clear whether it is decidable or not. Interesting methods for tackling similar questions have been identified by, for instance, Bodirsky et al. [19] and Dumortier et al. [20, 21]. Bodirsky et al. analyse the decidability of abstract properties of constraint languages such as whether certain relations are pp-definable or not. Their results are based on a number of different techniques from model theory, universal algebra, Ramsey theory, and topological dynamics. Dumortier et al. [20, 21] show that it is decidable whether a given first-order formula using the binary functions $*$ and $+$, and the binary relation \leq over \mathbb{R} with parameters from \mathbb{Q} defines a semilinear relation. These results indicate that there are non-obvious properties of semilinear relations that may be relevant for proving (un)decidability of the metaproblem.

Acknowledgements

The authors thank Manuel Bodirsky for suggesting the relation A_+ as a relaxation of R_+ (cf. Section 8.1) and the anonymous reviewers for their numerous useful comments and suggestions.

References

- [1] J. Ferrante, C. Rackoff, A decision procedure for the first order theory of real addition with order, *SIAM J. Comput.* 4 (1) (1975) 69–76.
- [2] A. Bulatov, P. Jeavons, A. Krokhin, Classifying the computational complexity of constraints using finite algebras, *SIAM J. Comput.* 34 (3) (2005) 720–742.
- [3] T. Feder, M. Y. Vardi, Monotone monadic SNP and constraint satisfaction, in: *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC-1993)*, 1993, pp. 612–622.
- [4] T. Feder, M. Y. Vardi, The computational structure of monotone monadic SNP and constraint satisfaction: a study through datalog and group theory, *SIAM J. Comput.* 28 (1) (1998) 57–104.
- [5] M. Bodirsky, M. Grohe, Non-dichotomies in constraint satisfaction complexity, in: *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP-2008)*, 2008, pp. 184–196.
- [6] M. Bodirsky, P. Jonsson, T. von Oertzen, Essential convexity and complexity of semi-algebraic constraints, *Log. Methods Comput. Sci.* 8 (4) (2012).
- [7] M. Bodirsky, P. Jonsson, T. von Oertzen, Horn versus full first-order: Complexity dichotomies in algebraic constraint satisfaction, *J. Logic Comput.* 22 (3) (2012) 643–660.
- [8] D. Marker, *Model Theory: An Introduction*, Springer, 2002.
- [9] P. Jonsson, T. Lööw, Computational complexity of linear constraints over the integers, *Artif. Intell.* 195 (2013) 44–62.
- [10] P. Jonsson, J. Thapper, Affine consistency and the complexity of semi-linear constraints, in: *Proceedings of 39th International Symposium on Mathematical Foundations of Computer Science (MFCS-2014)*, 2014, pp. 420–431.
- [11] P. Jeavons, On the algebraic structure of combinatorial problems, *Theoret. Comput. Sci.* 200 (1–2) (1998) 185–204.
- [12] W. Hodges, *A Shorter Model Theory*, Cambridge University Press, New York, 1997.

- [13] A. Schrijver, Theory of linear and integer programming, John Wiley & Sons, 1986.
- [14] G. Stengle, J. McEnerney, R. Robson, Convex polarities over ordered fields, *J. Pure Appl. Algebra* 214 (4) (2010) 370–379.
- [15] T. J. Schaefer, The complexity of satisfiability problems, in: Proceedings of the 10th ACM Symposium on Theory of Computing (STOC-1978), 1978, pp. 216–226.
- [16] M. Bodirsky, B. Martin, A. Mottet, Constraint satisfaction problems over the integers with successor, in: Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP-2015) Part I, 2015, pp. 256–267.
- [17] P. Hell, J. Nešetřil, Colouring, constraint satisfaction, and complexity, *Comput. Sci. Rev.* 2 (3) (2008) 143–163.
- [18] M. Bodirsky, J. Kára, The complexity of temporal constraint satisfaction problems, *J. ACM* 57 (2).
- [19] M. Bodirsky, M. Pinsker, T. Tsankov, Decidability of definability, *J. Symb. Log.* 78 (4) (2013) 1036–1054.
- [20] F. Dumortier, M. Gyssens, L. Vandeurzen, D. V. Gucht, On the decidability of semilinearity for semialgebraic sets and its implications for spatial databases, *J. Comput. System Sci.* 58 (3) (1999) 535–571.
- [21] F. Dumortier, M. Gyssens, L. Vandeurzen, D. V. Gucht, On the decidability of semilinearity for semialgebraic sets and its implications for spatial databases – CORRIGENDUM, *J. Comput. System Sci.* 59 (3) (1999) 557–562.