



HAL
open science

Affine consistency and the complexity of semilinear constraints

Peter Jonsson, Johan Thapper

► **To cite this version:**

Peter Jonsson, Johan Thapper. Affine consistency and the complexity of semilinear constraints. 39th International Symposium on Mathematical Foundations of Computer Science (MFCS-2014), 2014, Budapest, Hungary. hal-01762331

HAL Id: hal-01762331

<https://hal.science/hal-01762331v1>

Submitted on 9 Apr 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.

Affine consistency and the complexity of semilinear constraints

Peter Jonsson^{1*} and Johan Thapper²

¹ Department of Computer and Information Science, Linköping University, Sweden
`peter.jonsson@liu.se`

² LIGM, Université Paris-Est Marne-la-Vallée, France
`thapper@u-pem.fr`

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, just to mention a few examples. We concentrate on relations over the reals and rational numbers. Under this assumption, the computational complexity of the constraint satisfaction problem (CSP) is known for all finite sets Γ of semilinear relations containing the relations $R_+ = \{(x, y, z) \mid x + y = z\}$, \leq , and $\{1\}$. These problems correspond to extensions of LP feasibility. We generalise this result as follows. We introduce an algorithm, based on computing affine hulls, which solves a new class of semilinear CSPs in polynomial time. This allows us to fully determine the complexity of $\text{CSP}(\Gamma)$ for semilinear Γ containing R_+ and satisfying two auxiliary conditions. Our result covers all semilinear Γ such that $\{R_+, \{1\}\} \subseteq \Gamma$. We continue by studying the more general case when Γ contains R_+ but violates either of the two auxiliary conditions. We show that each such problem is equivalent to a problem in which the relations are finite unions of homogeneous linear sets and we present evidence that determining the complexity of these problems may be highly non-trivial.

1 Introduction

Let $X = \mathbb{Q}$ or $X = \mathbb{R}$. 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. Alternatively, R is semilinear if it is first-order definable in $\{R_+, \leq, \{1\}\}$ where $R_+ = \{(x, y, z) \in X^3 \mid x + y = z\}$. 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). Here, we are given a set of variables that take their values from a (finite or infinite) domain and a set of constraints (e.g. relations) that constrain the values different variables can take, and the question is whether the variables can be assigned values

* Partially supported by the Swedish Research Council (VR) under grant 621-2012-3239.

such that all constraints are satisfied or not. CSPs are often parameterized by a finite set Γ of allowed relations, known as a *constraint language*. All constraints in the input of $\text{CSP}(\Gamma)$ must be members of Γ . This way of parameterizing 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 constraints. The complexity of finite-domain CSPs has been studied for a long time and a powerful algebraic toolkit has gradually formed [6]. Much of this work has been devoted to the *Feder-Vardi conjecture* [7], i.e., 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: every computational problem is polynomial-time equivalent to an infinite-domain CSP [1]. Obtaining a full understanding of their computational complexity is thus extremely difficult and we have to contain ourselves to studying restricted cases. The main motivation behind this paper is the following result by Bodirsky et al. [2].

Theorem 1. *If Γ is a finite set of semilinear relations over \mathbb{R} or \mathbb{Q} that contains R_+ , \leq , and $\{1\}$, then $\text{CSP}(\Gamma)$ is either polynomial-time solvable or NP-complete.*

Characterizing the polynomial-time solvable cases is fairly simple: we say that a relation $R \subseteq \mathbb{R}^k$ 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 . If Γ contains essentially convex relations only, then $\text{CSP}(\Gamma)$ is in P by exploiting linear programming, and the problem is NP-complete otherwise. One may suspect that there are semilinear constraint languages Γ such that $\text{CSP}(\Gamma) \in P$ but Γ is not essentially convex. This is indeed true and we identify two such cases.

In the first case, we encounter polynomial-time solvable classes that, informally speaking, contain 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 this kind of highly non-convex relations. Thus, we introduce (in Section 3.1) a new algorithm based on computing affine hulls.

In the second case, we have polynomial-time solvable classes where the relations look essentially convex when viewed from the origin. That is, any points p and q that witnesses that the relations are not essentially convex lie on a line that does not pass through the origin. We show (in Section 3.2) that we can remove the holes witnessed by p and q and find an equivalent language that is essentially convex, and thereby solve the problem.

Combining these algorithmic results with certain hardness results (that are collected in Section 4) yields a dichotomy: if a semilinear constraint language Γ contains R_+ and satisfies two additional properties (P_0) and (P_∞), then $\text{CSP}(\Gamma)$ is either polynomial-time solvable or NP-hard. Actually, $\text{CSP}(\Gamma)$ is always in NP for a semilinear constraint language Γ , cf. Theorem 5.2 in Bodirsky et al. [2].

This result immediately generalizes Theorem 1 since it implies that semilinear constraint languages that contain R_+ and $\{1\}$ (but not necessarily \leq) exhibit a dichotomy. These results and their proofs together with formal definitions of the properties (P_0) and (P_∞) can be found in Section 5.

A natural goal at this point would be to determine the complexity when one or more of the side conditions are not met. In Section 6, we prove that if $\{R_+\} \subseteq \Gamma$ does not satisfy (P_0) and/or (P_∞) , then $\text{CSP}(\Gamma)$ is equivalent to a problem $\text{CSP}(\Gamma')$ where Γ' contains *homogeneous* semilinear relations only. By a homogeneous semilinear relation, we mean that it can be defined in terms of homogeneous inequalities. How hard it is to determine the complexity of $\text{CSP}(\Gamma')$ is difficult to say and, consequently, we discuss this issue in some detail.

2 Preliminaries

2.1 Constraint satisfaction problems

Let $\Gamma = \{R_1, \dots, R_n\}$ be a finite set of finitary relations over some domain D (which will usually be infinite). We refer to Γ as a *constraint language*. 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 ψ_i are formulas of the form $x = y$ 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*. The conjuncts in a pp-formula Φ are also called the *constraints* of Φ . The *constraint satisfaction problem for Γ* ($\text{CSP}(\Gamma)$ for short) is the computational problem to decide whether a given primitive positive sentence Φ is true in Γ .

Definition 1. *The problem $\text{CSP}(\Gamma)$ is tractable (or polynomial-time solvable) if for every finite $\Gamma' \subseteq \Gamma$, $\text{CSP}(\Gamma')$ is solvable in polynomial time. We say that $\text{CSP}(\Gamma)$ is NP-hard if $\text{CSP}(\Gamma')$ is NP-hard for some finite $\Gamma' \subseteq \Gamma$.*

A relation $R(x_1, \dots, x_k)$ is *pp-definable from Γ* if there exists a quantifier-free 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 simple but important result explains the importance of primitive positive definability for the constraint satisfaction problem. We will use it extensively in the sequel without making explicit references to it.

Lemma 1. *Let Γ be a constraint language and $\Gamma' = \Gamma \cup \{R\}$ where $R \in \langle \Gamma \rangle$. Then $\text{CSP}(\Gamma)$ is polynomial-time equivalent to $\text{CSP}(\Gamma')$.*

2.2 Semilinear relations

The domain, X , of every relation in this paper will either be the set of reals, \mathbb{R} , or the set of rationals, \mathbb{Q} . 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. Let $LE_X[Y]$ denote the set of linear equalities over X with coefficients in Y and $LI_X[Y]$ denote 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*. The set of *semilinear sets*, $SL_X[Y]$, 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{R} and \mathbb{Q} , respectively.

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

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 .*

Due to the alternative definition of a semilinear relation as a relation that is first-order definable in $\{R_+, \leq, \{1\}\}$, the set $SL_X[Y]$ is closed under pp-definitions. Consequently, Lemma 2 is applicable to all relations discussed in this paper.

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 can exclusively concentrate on relations from $SL_{\mathbb{Q}}[\mathbb{Q}]$ in the sequel. Let $\Gamma \subseteq SL_{\mathbb{R}}[\mathbb{Q}]$ and let I be an instance of $\text{CSP}(\Gamma)$. Construct an instance I' of $\text{CSP}(\Gamma|_{\mathbb{Q}})$ by replacing each occurrence of R in I by $R|_{\mathbb{Q}}$. If I' has a solution, then I has a solution since $R|_{\mathbb{Q}} \subseteq R$ for each $R \in \Gamma$. If I has a solution, then it has a rational solution by Lemma 3.7 in Bodirsky et al. [2] so I' has a solution, too.

Lemma 3 (Lemma 4.3 in Bodirsky et al. [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 $r = 0$.*

It follows that $LE_{\mathbb{Q}}[\mathbb{Q}] \subseteq \langle \{R_+, \{1\}\} \rangle$ and $LI_{\mathbb{Q}}[\mathbb{Q}] \subseteq \langle \{R_+, \leq, \{1\}\} \rangle$.

2.3 Unary semilinear relations

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

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

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 $\langle LE_{\mathbb{Q}}[\mathbb{Q}] \cup \{R\} \rangle$ and that $\mathcal{L}_{R,a,b}$ is pp-definable in $\{R_+, \{1\}, R\}$ by Lemma 3.

A k -ary relation R is *bounded* if there exists an $a \in \mathbb{Q}$ such that $R \subseteq [-a, a]^k$. If $k = 1$, then we say that R is *unbounded in one direction* if there exists $a \in \mathbb{Q}$ such that exactly one of the following hold: for all $b \leq a$, there exists a $c \leq b$ such that $c \in R$; or for all $b \geq a$, there exists a $c \geq b$ such that $c \in R$.

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, $\langle \{R_+, \{1\}, U\} \rangle$ contains a bnu.*

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 $|x - y| < \delta$.

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

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 $\delta > 0$.*

2.4 Essential convexity

Let R be a k -ary relation over \mathbb{Q}^k . 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 *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 . 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 . 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 p and q *witness* that R is not essentially convex. A semilinear relation is essentially convex if and only if it does not exclude an interval.

Theorem 2 (Theorem 5.1 and 5.4 in Bodirsky et al. [2]). *If Γ is an essentially convex semilinear constraint language, then $\text{CSP}(\Gamma)$ is tractable.*

3 Tractability

In this section, we present our two main sources of tractability. In Section 3.1, we introduce 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 show that the algorithm in Theorem 2 has a wider applicability.

3.1 Affine consistency

For a subset $X \subseteq \mathbb{Q}^n$, let $\text{aff}(X)$ denote the *affine hull of X in \mathbb{Q}^n* : $\text{aff}(X) = \{\sum_{i=1}^k c_i x_i \mid k \geq 1, c_i \in \mathbb{Q}, x_i \in X, \sum_{i=1}^k c_i = 1\}$. An *affine subspace* is a subset $X \subseteq \mathbb{Q}^n$ for which $\text{aff}(X) = X$. The points $p_1, \dots, p_k \in \mathbb{Q}^n$ are said to be *affinely independent* if $a_1 p_1 + \dots + a_k p_k = 0$ with $a_1 + \dots + a_k = 0$ implies $a_1 = \dots = a_k = 0$. The dimension, $\dim(X)$, of a set $X \subseteq \mathbb{Q}^n$ is defined to be one less than the maximum number of affinely independent points in X .

We define a notion of consistency for sets of semilinear constraints which we call affine consistency. Let V be a finite set of variables. 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 an affine subspace $A \subseteq \mathbb{Q}^V$ 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 establishes affine consistency for a set of constraints 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 how for such constraint languages, the algorithm can be implemented to run in polynomial time.

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

Algorithm 1: Affine consistency

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

Output: “yes” if the resulting affine subspace is non-empty, “no” otherwise

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

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 $B(c) = \{y \in \mathbb{Q}^n \mid \alpha \cdot y \notin [-c, c]\}$.

Express the line L_i^p as $\{y \in \mathbb{Q}^n \mid y = a \cdot x + b, x \in \mathbb{Q}\}$, for some $a, b \in \mathbb{Q}^n$. Define a unary relation T by the formula $\varphi(x) \equiv \exists y \in \mathbb{Q}^n. P(y) \wedge y = a \cdot x + b$. Note that $T \in \langle LE_{\mathbb{Q}}[\mathbb{Q}] \cup \{P\} \rangle$. Since T contains p_0 and p_i , it follows that T is not a constant and hence unbounded. By Lemma 4, T is unbounded in both directions. By Lemma 2, $B(c_i^p) \cap L_i^p \subseteq T \subseteq P$, 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 $c = \max\{c'\} \cup \{c_i^p, c_j^q \mid 1 \leq i, j \leq d\}$. This ensures that for any $x > c$, $H(x) \cap P$ intersects the lines L_i^p in d affinely independent points, and that $H(x) \cap Q$ intersects the lines L_j^q in d affinely independent points.

We now have $\text{aff}(H(x) \cap P) = \text{aff}(H(x) \cap Q) = A'(x)$ with $\dim(A'(x)) = \dim(A) - 1$, for every $x > c$. By induction on $H(x) \cap P = (H(x) \cap P_1) \cup \dots \cup (H(x) \cap P_k)$ and $H(x) \cap Q = (H(x) \cap Q_1) \cup \dots \cup (H(x) \cap Q_l)$, it follows that $\text{aff}(H(x) \cap P_{i(x)} \cap Q_{j(x)}) = A'(x)$ for some $i(x)$ and $j(x)$. This holds for all $x > c$, hence 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 $A'(x_1), A'(x_2) \subseteq \text{aff}(P_{i'} \cap Q_{j'})$, $A'(x_1) \cap A'(x_2) = \emptyset$, 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 $A' \subset \text{aff}(P_{i'} \cap Q_{j'}) \subseteq A$, and $\dim(A'(x_1)) = \dim(A) - 1$. Therefore we have the equality $\text{aff}(P_{i'} \cap Q_{j'}) = A$. The lemma follows. \square

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 $U \subseteq \mathbb{Q}^n$ be an affine subspace. Algorithm 2 computes a set*

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

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

Output: A set of inequalities defining $\text{aff}(R \cap U)$.

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

of linear inequalities S defining $\text{aff}(R \cap U)$ in time polynomial in $\text{size}(R) + \text{size}(U)$ and with $\text{size}(S) \leq \text{size}(R) + \text{size}(U)$.

Proof. Let $R = R_1 \cup \dots \cup R_k$ be the representation of R as the union of (convex) linear sets R_i . By Lemma 7, there exists an i such that $\text{aff}(R \cap U) = \text{aff}(R_i \cap U)$ and since $\text{aff}(R_j \cap U) \subseteq \text{aff}(R \cap U)$ for all j , the algorithm will find such an i on line 1 by simply comparing the dimensions of these sets. If $\text{aff}(R \cap U) = \text{aff}(R_i \cap U) = \emptyset$, 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 (see for example [11, 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 $ineq \in I \cup J$ is an inequality that cannot be removed without increasing the dimension of the affine hull, then it is clear that $ineq$ 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, U_S , and $U_S = \text{aff}(U_S) = \text{aff}(R_i \cap U) = \text{aff}(R \cap U)$.

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 [11, 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 $\text{size}(R) + \text{size}(U)$. Finally, the set S is a subset of $I \cup J$, so $\text{size}(U_S) \leq \text{size}(R) + \text{size}(U)$. \square

Theorem 3. Let $\{R_+, \{1\}\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$. If there is no bnu in $\langle \Gamma \rangle$, then Algorithm 1 solves $\text{CSP}(\Gamma)$ in polynomial time.

Proof. Assume that each relation $R \in \Gamma$ is given as $R = R_1 \cup \dots \cup R_k$, where R_i is a (convex) linear set for each i . First, we show that the algorithm terminates with U 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.

It is clear that Z is contained in U throughout the execution of the algorithm. Therefore, $\text{aff}(Z) = \text{aff}(Z \cap U)$ so it suffices to show that $\text{aff}(Z \cap U) = U$. We will show that $\text{aff}(\bigcap_{i=1}^j \hat{R}_i \cap U) = U$ for all $j = 1, \dots, m$. When the algorithm terminates, we have $\text{aff}(\hat{R}_i \cap U) = U$ for every $i = 1, \dots, m$. In particular, the claim holds for $j = 1$. Now assume that the claim holds for $j - 1$. Then, $P = \bigcap_{i=1}^{j-1} \hat{R}_i \cap U$ and $Q = \hat{R}_j \cap U$ satisfy the requirements of Lemma 7 with $\text{aff}(P) = \text{aff}(Q) = U$. Therefore, we can use this lemma to conclude that $\text{aff}(\bigcap_{i=1}^j \hat{R}_i \cap U) = \text{aff}(P \cap Q) = U$.

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 $n = |V|$. The size of \hat{R} is at most $\text{size}(R) + \log n$, so the size of U never exceeds $\mathcal{O}(mn(\text{size}(R) + \log n))$, 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

The dimension of a set is defined with respect to its affine hull, as in Section 3.1. We give a result on the structure of certain not necessarily essentially convex relations. It is based on the intuition 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 this up by showing that we can remove certain lower-dimensional holes and thus recover an equivalent essentially convex CSP.

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 $\varphi_0, \dots, \varphi_k$ are conjunctions over $LI_{\mathbb{Q}}[\mathbb{Q}]$, and $\varphi_1, \dots, \varphi_k$ define convex sets of dimensions strictly lower than the set defined by φ_0 .*

Theorem 4. *Let $U \in \{1\} + \mathcal{I}(c)$ for some $0 < c < 1$ and assume that $\{R_+, U\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$ is a constraint language such that every unary relation in $\langle \Gamma \rangle$ is essentially convex. Then, $\text{CSP}(\Gamma)$ is equivalent to $\text{CSP}(\Gamma')$ 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 conjunction 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 I be an arbitrary instance of $\text{CSP}(I)$ over the variables V and construct an instance I' of $\text{CSP}(I')$ by replacing each occurrence of a relation R in I by R' . Clearly, if I' is satisfiable, then so is I . Conversely, let $s \in \mathbb{Q}^V$ be a solution to I and assume that I' 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 $\mathcal{L}_{I,s,(0,\dots,0)} \in \langle I' \rangle$. All tuples in U correspond to solutions of I that are not solutions to I' .

Fix a constraint $R(x_1, \dots, x_l)$ in I and consider the points in U 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 in I can account for at most a finite number of points in U , so U is finite.

There are two cases to consider: (1) U contains more than one point and therefore excludes an interval; or (2) U is the constant $\{1\}$. Since I is assumed not to be essentially convex, the relation U can then be used to pp-define a unary relation that is not essentially convex. In either case, there is a contradiction to the assumption that every unary relation in $\langle I' \rangle$ is essentially convex. It follows that I' must be satisfiable. \square

4 NP-hardness

We now prove the necessary hardness result. It is based on the following simple reduction from the NP-hard problem Not-All-Equal 3SAT [10]. We then show that having a unary 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. This is a unified hardness condition for all CSPs classified in this paper.

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

For a rational c , and a unary relation U , let $c \cdot U = \{c \cdot x \mid x \in U\} \in \langle \{R_+, U\} \rangle$.

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 $T \cap (m - \varepsilon', m + \varepsilon') = \emptyset$, 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 unary relation $T_0(x) \equiv \exists y \exists z. U_\delta(y) \wedge z = x - y \cdot m \wedge U(z)$, and the unary relation $T_\infty(x) \equiv \exists y' \exists z'. U_\delta(y') \wedge z' = x - y' \cdot m' \wedge T(z')$. 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 Expansions of $\{R_+\}$

We now study the class of semilinear constraint languages containing R_+ and having the properties (P_0) and (P_∞) . These properties are defined as follows.

- (P_0) 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 $M < \infty$.

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. We now state our main result and get a complete classification for semilinear constraint languages containing R_+ and $\{1\}$ as an immediate corollary.

Theorem 5. *Let $\{R_+\} \subseteq \Gamma \subseteq \text{SL}_{\mathbb{Q}}[\mathbb{Q}]$ be a finite constraint language that satisfies (P_0) and (P_∞) .*

- *If Γ is 0-valid, then $\text{CSP}(\Gamma)$ is trivially tractable;*
- *otherwise, if Γ does not contain a bnu, then $\text{CSP}(\Gamma)$ is tractable by establishing affine consistency;*
- *otherwise, if all unary relations in $\langle \Gamma \rangle$ are essentially convex, then $\text{CSP}(\Gamma)$ is tractable via 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 and let R be a relation in Γ such that $(0, \dots, 0) \notin R$ and $a \in R$. Then, the relation

$\mathcal{L}_{R,a,(0,\dots,0)} \in \langle \{R_+, R\} \rangle$ is unary, does not contain 0 but does contain 1. 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 $T = \mathcal{L}_{R,a,(0,\dots,0)} \cap U_0 \cap U_\infty$. 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 3 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 4, $\text{CSP}(\Gamma)$ is equivalent to $\text{CSP}(\Gamma')$ for an essentially convex constraint language Γ' . Tractability follows from Theorem 2.

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

Theorem 6. *Let $\{R_+, \{1\}\} \subseteq \Gamma \subseteq SL_{\mathbb{Q}}[\mathbb{Q}]$ be a finite constraint language. If $\langle \Gamma \rangle$ contains a bnu and $\langle \Gamma \rangle$ contains a relation that is not essentially convex, then $\text{CSP}(\Gamma)$ is NP-hard. Otherwise, $\text{CSP}(\Gamma)$ is tractable.*

6 Discussion and future work

We have determined the complexity of $\text{CSP}(\Gamma)$ for all finite semilinear Γ containing R_+ and satisfying (P_0) and (P_∞) . Clearly, one would like to obtain a full classification for semilinear constraint languages without side conditions but this appears to be an extremely hard task. One may instead study cases when (P_0) and (P_∞) hold (and $\{R_+\} \not\subseteq \Gamma$) or when only $\{R_+\} \subseteq \Gamma$ is required (and (P_0) or (P_∞) do not hold). We will discuss these two possibilities below.

For simplicity, let SL^1 denote the set of semilinear constraint languages such that $\{\{1\}\} \subseteq \Gamma$ and $\{R_+\} \not\subseteq \Gamma$. The languages in SL^1 satisfy both (P_0) and (P_∞) . A straightforward modification of the construction in Sec. 6.3 of [9] gives the following: for every finite 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 [7, 8]. We also observe that for every finite *temporal constraint language* (i.e., languages that are first-order definable in $(\mathbb{Q}; <)$), 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 [4] 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.

Let us now consider the set SL^+ of all semilinear constraint languages containing R_+ and not satisfying (P_0) or (P_∞) . If either (P_0) or (P_∞) is violated,

then we can show that the languages in SL^+ are of a particular restricted type. Let $HSL_{\mathbb{Q}}[\mathbb{Q}]$ denote the set of *homogeneous* semilinear relations, i.e., relations $R \subseteq \mathbb{Q}^n$ that are finite unions of homogeneous linear sets.

Theorem 7. *Arbitrarily choose $\Gamma \in SL^+$. $CSP(\Gamma)$ is equivalent to $CSP(\Gamma')$ for a finite constraint language $\{R_+\} \subseteq \Gamma' \subseteq HSL_{\mathbb{Q}}[\mathbb{Q}]$.*

It seems like a difficult task to classify the complexity of subsets of $HSL_{\mathbb{Q}}[\mathbb{Q}]$ since this would imply the previously mentioned classification of temporal constraint problems and also the constraint problems studied by Bodirsky et al. [3]. It is also closely connected with CSPs over domains of size 3 as demonstrated by the following proposition.

Proposition 1. *Let A be a finite constraint language over the domain $\{-1, 0, 1\}$. There is a $\Gamma \subseteq HSL_{\mathbb{Q}}[\mathbb{Q}]$ such that $CSP(A)$ and $CSP(\Gamma)$ are polynomial-time equivalent.*

The complexity of CSPs on three-element domains is fully determined [5] and the lengthy proof is based on machinery from universal algebra. One has to note, though, that we may not need to completely classify the complexity of $HSL_{\mathbb{Q}}[\mathbb{Q}]$ in order to classify the complexity of SL^+ : we have the additional condition that R_+ is a member of the languages under consideration. It is plausible that this would simplify the task.

References

1. M. Bodirsky and M. Grohe. Non-dichotomies in constraint satisfaction complexity. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP-2008)*, pages 184–196, 2008.
2. M. Bodirsky, P. Jonsson, and T. von Oertzen. Essential convexity and complexity of semi-algebraic constraints. *Logical Methods in Computer Science*, 8(4), 2012.
3. M. Bodirsky, P. Jonsson, and T. von Oertzen. Horn versus full first-order: Complexity dichotomies in algebraic constraint satisfaction. *J. Log. Comput.*, 22(3):643–660, 2012.
4. M. Bodirsky and J. Kára. The complexity of temporal constraint satisfaction problems. *J. ACM*, 57(2), 2010.
5. A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM*, 53(1):66–120, 2006.
6. A. Bulatov, P. Jeavons, and A. Krokhin. Classifying the computational complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742, 2005.
7. T. Feder and M. Y. Vardi. Monotone monadic SNP and constraint satisfaction. In *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC-1993)*, pages 612–622, 1993.
8. P. Hell and J. Nešetřil. Colouring, constraint satisfaction, and complexity. *Computer Science Review*, 2(3):143–163, 2008.
9. P. Jonsson and T. Löw. Computational complexity of linear constraints over the integers. *Artif. Intell.*, 195:44–62, 2013.
10. T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th ACM Symposium on Theory of Computing (STOC-1978)*, pages 216–226, 1978.
11. A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1986.