Constraint Satisfaction Problems over Numeric Domains[*]

Manuel Bodirsky[1] and Marcello Mamino[2]

1 Institut für Algebra, TU Dresden, Dresden, Germany
manuel.bodirsky@tu-dresden.de

2 Institut für Algebra, TU Dresden, Dresden, Germany
marcello.mamino@tu-dresden.de

Abstract

We present a survey of complexity results for constraint satisfaction problems (CSPs) over the integers, the rationals, the reals, and the complex numbers. Examples of such problems are feasibility of linear programs, integer linear programming, the max-atoms problem, Hilbert's tenth problem, and many more. Our particular focus is to identify those CSPs that can be solved in polynomial time, and to distinguish them from CSPs that are NP-hard. A very helpful tool for obtaining complexity classifications in this context is the concept of a polymorphism from universal algebra.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Constraint satisfaction problems, Numerical domains

Digital Object Identifier 10.4230/DFU.Vol7.15301.79

1 Introduction

Many computational problems from many different research areas in theoretical computer science can be formulated as Constraint Satisfaction Problems (CSPs) where the variables might take values from an infinite domain. There is a considerable literature about the computational complexity of particular infinite domain CSPs, but there are only few systematic complexity results. Most of these results belong to two research directions. One is the development of the *universal-algebraic approach*, which has been so successful for studying the complexity of finite-domain constraint satisfaction problems. The other direction is to study constraint satisfaction problems over some of the most basic and well-known infinite domains, such as the numeric domains $\mathbb{Z}$ (the integers), $\mathbb{Q}$ (the rational numbers), $\mathbb{R}$ (the reals), or $\mathbb{C}$ (the complex numbers), and to focus on constraint relations that are first-order definable from usual addition and multiplication on those domains. In this way, many computational problems that are of fundamental importance in computer science and mathematics can be studied in the same framework. Several recent results are concerned with obtaining a systematic understanding of the computational complexity of such CSPs; and this survey article is devoted to presenting these recent results in a common context, and to highlight some of the common threads that are likely to be fruitful also in the future.

For a fixed structure Γ with finite relational signature τ, the constraint satisfaction problem CSP(Γ) is the problem of deciding whether a given finite conjunction of atomic

[*] The authors have received funding from the German Science Foundation (DFG, project number 622397) and from the European Research Council (ERC, grant agreement number 681988, CSP-Infinity).

The Constraint Satisfaction Problem: Complexity and Approximability. *Dagstuhl Follow-Ups*, Volume 7, ISBN 978-3-95977-003-3.
Editors: Andrei Krokhin and Stanislav Živný; pp. 79–111
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

τ-formulas is satisfiable in Γ. That is, for a given instance

$$R_1(\bar{x}_1) \wedge \cdots \wedge R_m(\bar{x}_m)$$

of this problem where $\bar{x}_1, \dots, \bar{x}_m$ are tuples of variables, we want to decide whether there exists an assignment of values to those variables such that all the conjuncts (the 'constraints') are satisfied. We often refer to Γ as the *template* of the CSP. The computational complexity of CSP(Γ) has been studied intensively when the template Γ has a finite domain; it is always in NP, and Feder and Vardi [35] conjectured that it is either in P or NP-complete; see Barto [6] for a short survey on the state of the art concerning progress towards proving the conjecture.

When the template Γ might have an infinite domain, there is no hope for a complete classification of the complexity of CSP(Γ) in general [16]: every computational problem is equivalent (under polynomial-time Turing reductions) to a problem of the form CSP(Γ). Indeed, even individual templates over numeric domains, usually as a consequence of their practical significance, are the focus of large branches of current mathematical research. This research fixes many points on the complexity map of CSPs. More recently, we have seen systematic results that provide complexity results for entire areas of interesting, albeit less expressive, templates in this map. In this introduction, and in our survey, we first discuss known concrete templates and then move on to more systematic classifications.

A famous example of a computational problem that can be formulated as a CSP over a numeric domain is Hilbert's 10th problem. It is CSP($\mathbb{Z}; R_+, R_*, R_{=1}$) where

- R_+ stands for the ternary addition relation $\{(x, y, z) \in \mathbb{Q}^3 \mid x + y = z\}$,
- R_* stands for the ternary multiplication relation $\{(x, y, z) \in \mathbb{Q}^3 \mid x * y = z\}$, and
- $R_{=1}$ stands for the unary relation $\{1\}$.

Matiyasevich [79], building upon results of Davis, Putnam, and Robinson, showed that this computational problem is undecidable (for a reference see [80]).

Another example, sitting at the opposite end of the complexity spectrum, is the feasibility problem for linear programs. It can be formulated as CSP(Γ_{lin}) for $\Gamma_{\text{lin}} := (\mathbb{Q}; \leq, R_+, R_{=1})$ where R_+ and $R_{=1}$ are defined as before, but over the rational numbers instead of the integers. Then CSP(Γ_{lin}) is easily seen to be polynomial-time equivalent to the feasibility problem for linear programs (see Section 2.3), which is a computational problem of outstanding theoretical and practical interest [94]. The complexity of this problem has been an open problem until Khachiyan's discovery that the ellipsoid method gives a polynomial-time algorithm [64] (see Section 3). It is natural to ask which relations can be added to Γ_{lin} so that the resulting expanded structure still has a CSP that can be solved in polynomial time; this is discussed in Section 4.1.

The choice of the domain of Γ might or might not have an impact on the computational complexity of CSP(Γ). For example, the structure $(\mathbb{Q}; \leq, R_+, R_{=1})$ has the same CSP as the structure $(\mathbb{R}; \leq, R_+, R_{=1})$ (where R_+ and $R_{=1}$ are defined as above but over $\mathbb{R}$ instead of $\mathbb{Q}$). On the other hand, if we consider the problem CSP($\mathbb{Z}; R_+, R_*, R_{=1}$) and replace the integers $\mathbb{Z}$ by the reals or the complex numbers, and adapt the interpretation of the relations R_+, R_*, and $R_{=1}$ correspondingly, the complexity of the CSP changes dramatically: CSP($\mathbb{R}; R_+, R_*, R_{=1}$) is equivalent to the existential theory of the reals, which is decidable (see Section 5). Even better complexity results are known for CSP($\mathbb{C}; R_+, R_*, R_{=1}$) (again, see Section 5). If we consider the structure $(\mathbb{Z}; \leq, R_+, R_{=1})$ instead of $(\mathbb{Q}; \leq, R_+, R_{=1})$ we obtain the famous NP-complete integer program feasibility problem.

Also over the integers there are many natural CSPs that can be solved in polynomial time. Well-known examples are

- linear Diophantine equation systems: here the constraints are of the form

  $$a_1x_1 + \cdots + a_kx_k = a_0$$

  for constants $a_0, a_1, \ldots, a_k \in \mathbb{Z}$. Such systems can be solved efficiently using linear algebra, via an appropriate implementation of an algorithm computing the Smith Normal form and a careful analysis of the size of the coefficients that appear during the computation [61, 29].
- Difference logic: here the constraints are of the form $x - y \leq c$ for $c \in \mathbb{Z}$. Such systems can be solved efficiently using shortest path computations (see, e.g., [32]).

So far, we have seen a powerful framework that captures many natural and important computational problems, but a systematic picture is missing. For some restricted settings, however, there is a complete classification of the polynomial-time tractable and the NP-hard cases. We present two such settings; more follows in the main body of the text. The first such setting is the class of structures with domain $\mathbb{Q}$ that are definable over $(\mathbb{Q}; <)$ with the order alone. This includes for example the structure $(\mathbb{Q}; \mathrm{Betw})$ where

$$\mathrm{Betw} := \{(x, y, z) \in \mathbb{Q}^3 \mid x < y < z \vee z < y < x\}$$

is the so-called *betweenness relation*. The problem $\mathrm{CSP}(\mathbb{Q}; \mathrm{Betw})$ is known as the *betweenness problem* in theoretical computer science and known to be NP-complete. Also the cyclic ordering problem, and CSPs from temporal reasoning such as the Point Algebra [101] and the Ord-Horn class [85] can be cast in this form.

In general, when Γ is a structure that has the same domain as some structure Δ, and all relations in Γ are first-order definable in Δ (we do allow equality in first-order formulas), we refer to Γ as a *first-order reduct* of Δ (we first consider the expansion of Δ by all first-order definable relations, and then take a reduct in the usual sense, that is, we drop some of the relations). Classifying the CSPs of first-order reducts of some structure Δ therefore corresponds to a *bottom-up* approach to classifying CSPs, since first-order reducts of a structure Δ are typically ‘simpler’ than Δ.

The main result of Bodirsky and Kára [20] states that $\mathrm{CSP}(\Gamma)$ is, for all first-order reducts Γ of $(\mathbb{Q}; <)$, either in P or NP-complete (Section 7). One of the central ideas of the classification in [20] is that the powerful *universal-algebraic approach* to constraint satisfaction, which has been developed for finite-domain CSPs, can be applied here, too. The reason for this is that first-order reducts of $(\mathbb{Q}; <)$ satisfy a strong model-theoretic condition, *ω-categoricity*, which can be seen as a finiteness condition via the characterization of Ryll-Nardzewski: a countably infinite structure Γ is *ω-categorical* if and only if the automorphism group of Γ has finitely many orbits of k-tuples, for all $k \geq 1$. For ω-categorical Γ, the complexity of $\mathrm{CSP}(\Gamma)$ is completely captured by the so-called *polymorphisms* of Γ (see Section 2.2); they are a generalization of the concept of an endomorphism to higher arities. The extension of the theory of finite-domain CSPs to (subclasses of) ω-categorical structures has advanced significantly [8, 9, 23] and is outside the scope of this survey.

Structures on numerical domains are typically *not* ω-categorical. Consider for example the structure $(\mathbb{Z}; \mathrm{Succ})$ for $\mathrm{Succ} = \{(x, y) \mid x = y + 1\}$, the integers with the successor relation. Its automorphism group has only one orbit, but infinitely many orbits of pairs, and is therefore not ω-categorical by the theorem of Ryll-Nardzewski mentioned above. The structure $(\mathbb{Z}; \mathrm{Succ})$ can be seen as one of the ‘simplest’ structures over a numerical domain and with finite signature that is not ω-categorical. Following again the bottom-up approach, we study the class of CSPs for first-order reducts of $(\mathbb{Z}; \mathrm{Succ})$. This class contains non-trivial CSPs that can be solved in polynomial time; let us mention for example

$$\mathrm{CSP}(\mathbb{Z}; \{(x, x, x + 1), (x, x + 1, x), (x, x + 1, x + 1) \mid x \in \mathbb{Z}\}) . \tag{1}$$

For first-order reducts Γ of $(\mathbb{Z}; \mathrm{Succ})$, we almost have a dichotomy: $\mathrm{CSP}(\Gamma)$ is in P, or NP-complete, or there exists a finite structure Γ' such that Γ and Γ' have the same CSP [24] (Section 8). Hence, the truth of the Feder-Vardi conjecture would imply that also the class of structures definable over the integers using the successor relation has a complexity dichotomy.

The border between polynomial-time tractable and NP-hard CSPs for first-order reducts of $(\mathbb{Z}; \mathrm{Succ})$ can again be described using polymorphisms, and polymorphisms also play an important role in the proof of the classification [24]. However, in order to work with polymorphisms even when the structure Γ is not ω-categorical, we might have to pass to a structure which has the same CSP as Γ, but a different domain. The point is that polymorphisms with certain properties might only exist when the structure is sufficiently saturated, in the model-theoretic sense. For example, instead of $(\mathbb{Z}; \mathrm{Succ})$, we would consider the structure $(\mathbb{Q}; \{(x, y) \mid x = y + 1\})$, which has the same CSP, but a richer set of (automorphisms and) polymorphisms. These phenomena are one of the reasons why the numeric domains $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$ should not be discussed in isolation; indeed, there are many fruitful cross-connections between classifications over these domains.

Certain polymorphisms are of particular importance over numeric domains. An important example for first-order reducts of $(\mathbb{Q}; +, *)$ and of $(\mathbb{R}; +, *)$ is the operation $(x, y) \mapsto (x+y)/2$; such a reduct Γ has this polymorphism if and only if all relations of Γ are convex. Convexity has recently become a very active topic also in real algebraic geometry [50, 66, 49, 48]. Many of the central questions that are relevant to CSPs are open.

Another remarkable polymorphism for numeric domains is $(x, y) \mapsto \max(x, y)$ (or, dually, $(x, y) \mapsto \min(x, y)$). If a finite structure Γ has max as a polymorphism, then $\mathrm{CSP}(\Gamma)$ is known to be in P [53, 54]. The same is true for first-order reducts of $(\mathbb{Q}; <)$, and for first-order reducts of $(\mathbb{Z}; \mathrm{Succ})$ (an example of such a reduct is the structure from (1)). But already for first-order reducts Γ over $(\mathbb{Q}; <, +, 1)$ it is an open problem whether having max as a polymorphisms implies that $\mathrm{CSP}(\Gamma)$ can be solved by a polynomial-time algorithm; such an algorithm would also imply polynomial-time tractability for many problems of open computational complexity in seemingly different areas of theoretical computer science: the model-checking problem of the propositional μ-calculus, solving mean payoff games and simple stochastic games; these connections are discussed in Section 6.

Researchers in model theory have obtained remarkable results in the classification of first-order reducts of structures like $(\mathbb{C}; +, *)$ or $(\mathbb{R}; +, *)$ up to first-order interdefinability. For instance, Marker and Pillay [76] prove that any first-order reduct Γ of $(\mathbb{C}; +, *)$ that contains $+$ is either a first-order reduct of an expansion of $(\mathbb{C}; +)$ by constants, or the complex multiplication $*$ has a first-order definition in Γ. Similar results are available for $(\mathbb{R}; +, *)$ instead of $(\mathbb{C}; +, *)$; see [86, 75, 77, 39]. In order to be applicable for complexity analysis for CSPs, we would need refinements of these results for *primitive positive definability* (see Section 2.1) instead of first-order definability.

Finally, we also discuss classes of constraint satisfaction problems over numeric domains that provably do *not* have a complexity dichotomy: this turns out to be the case for first-order reducts of $(\mathbb{Z}; +, *)$; a proof can be found in Section 9. We were unable to prove such a non-dichotomy result for first-order reducts of $(\mathbb{R}; +, *)$. On the other hand, we have seen that already classifying first-order reducts of $(\mathbb{Q}; 1, +, \leq)$ presents considerable challenges. We discuss in Section 10 what the next steps for the bottom-up approach to classifying CSPs on numerical domains might be.

2 Constraint Satisfaction Problems

We use standard notation and terminology from model theory; see e.g. [52]. For better readability we abuse notation and identify formulas with the relations they define; e.g., we write $(\mathbb{Q}; x > y^2)$ for the relational structure $(\mathbb{Q}; \{(x,y) \in \mathbb{Q}^2 \mid x > y^2\})$.

2.1 Primitive Positive Formulas

A first-order formula is *primitive positive* if it is of the form

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

where $\psi_1, \dots, \psi_n$ are atomic formulas; that is, no negation, disjunction, and universal quantification is allowed (but equality is allowed). The constraint satisfaction problem for a relational τ-structure Γ can then be rephrased as follows: given a primitive positive τ-sentence ϕ (i.e., a primitive positive formula without free variables, formed with relations from Γ), does ϕ hold in Γ?

The relevance of primitive positive formulas for the CSP comes from the following lemma, due to which works over finite and infinite structures alike [53]).

▶ **Lemma 1** (Jeavons–Cohen–Gyssens). *Let Γ be a relational structure, and R a relation that can be defined using a primitive positive formula over Γ . Then* $\mathrm{CSP}(\Gamma, R)$*, that is, the CSP for the expansion of Γ by the relation R, is log-space equivalent to* $\mathrm{CSP}(\Gamma)$*.*

The proof idea is to replace occurrences of R in an instance by their primitive positive definition, introducing new variables for the existentially quantified variables in these definitions (in optimization, even if not presented at this level of generality, this idea is commonly used, and the newly introduced variables are called *slack variables*).

2.2 Polymorphisms

Which relations are primitive positive definable over a given structure? This can be a difficult question, but we have seen in the previous section that it is an important question when we want to study the computational complexity of a CSP. A very important tool for answering this question are *polymorphisms*.

▶ **Definition 2.** A function $f \colon B^k \to B$ *preserves* a relation $R \subseteq B^m$ if for all

$$(a_1^1, \dots, a_1^m), \dots, (a_k^1, \dots, a_k^m) \in R$$

we have that $(f(a_1^1, \dots, a_k^1), \dots, f(a_1^m, \dots, a_k^m)) \in R$. When Γ is a relational structure then f is called a *polymorphism* of Γ if f preserves all relations of Γ.

In other words, f is a polymorphism of Γ if and only if f is a homomorphism from Γ^k to Γ. Unary polymorphisms are also called *endomorphisms*, and *automorphisms* are precisely the bijective endomorphisms whose inverse is also an endomorphism. It is well-known that the set of all automorphisms forms a *permutation group*, the set of all endomorphisms a *transformation monoid*, and the set of all polymorphisms a *function clone*, which are a central topic in universal algebra. The key fact that links polymorphisms with primitive positive definability is that when R is primitive positive definable in Γ , then R is preserved by the polymorphisms of Γ (for any structure Γ over any domain). This is useful in the contrapositive to show that something is not primitive positive definable over Γ : it suffices to exhibit a polymorphism of Γ that does not preserve R. This test gives a necessary and sufficient criterion for structures Γ over a finite domain [25], and also for many infinite structures Γ , e.g., when Γ is countably categorical (see Section 7). And even when Γ is not countably categorical, polymorphisms can be an important tool (see Section 8).

2.3 Infinite Signatures

We now come to an important issue that we hid so far, but which is an important aspect of CSPs, in particular of CSPs over numeric domains: the problem of encoding instances if the structure Γ has an *infinite signature*. An infinite signature is natural when we want to view for example the feasibility problem for linear programs as a CSP. There, the constraints in an instance of the CSP are of the form $a_1x_1 + \cdots + a_nx_n \geq a_0$, for some $a_0, a_1, \ldots, a_n \in \mathbb{Q}$. So we would consider the signature that contains a relation symbol for each of those relations; there is a countably infinite number of them.

Also over finite domains, many well-known computational problems call for CSP formulations with templates Γ having an infinite signature, for example:

- Horn-SAT: the signature contains for every $n \geq 0$ a symbol for the relations defined over $\{0, 1\}$ by the Boolean expression $\neg x_1 \vee \cdots \vee \neg x_n \vee x_0$ or by $\neg x_1 \vee \cdots \vee \neg x_n$.
- Linear equations over a finite field F: the signature contains for every $n \geq 0$ and $a_0, a_1, \ldots, a_n \in F$ a symbol for the n-ary relation defined by $a_1x_1 + \cdots + a_nx_n = a_0$.

However, when the signature of Γ is infinite, the computational complexity of CSP(Γ) depends on how the symbols of the signature of Γ are represented in the input instances to CSP(Γ). For finite structures Γ , the standard way to represent a relation symbol R from the signature of Γ is by an explicit list of tuples that are in the relation R. When the domain of Γ is infinite, this is typically no longer an option (already the basic relations $<$ over $\mathbb{Q}$ or Succ over $\mathbb{Z}$ contain infinitely many tuples, so they cannot be stored explicitly). Alternatively, we can use symbolic representations of relations; how precisely these representations might look like depends on the domain, and might also depend on the specific algorithmic result that we want to establish. We list a couple of options.

1. In general, when Γ is a first-order reduct of some structure Δ with a finite signature, we can represent a relation R of Γ by its first-order definition in Δ.
2. More specifically, for first-order reducts Γ of $(\mathbb{Q}; <)$ we can use the fact that $(\mathbb{Q}; <)$ has quantifier elimination (see, e.g., Hodges [52]), so we can represent a relation R from Γ by its quantifier-free definition over $(\mathbb{Q}; <)$ in disjunctive normal form (DNF).
3. Likewise, the structure $(\mathbb{Z}; s)$ has quantifier elimination, where s is the unary successor function, and again we might use quantifier-free formulas in DNF to represent the relations of first-order reducts of these structures. In this situation, we have to point out another subtlety, namely how to represent terms of the form $s^n(x) := s(s(\cdots s(x) \cdots))$: whether n is coded in unary or in binary can make the difference between an easy and a hard problem (see e.g. the problems from Definition 38 and Definition 40 in Section 6).
4. Finally, also for the structure $(\mathbb{Q}; <, +)$ we can use the fact that $(\mathbb{Q}; <, +)$ has quantifier elimination in the language that additionally contains a constant symbol for each rational number [36], so we can represent a relation R from Γ by its quantifier-free definition in disjunctive normal form (DNF). Here it is natural to assume that the constants for the rational numbers are represented in binary.

Option (1) listed above is not well-suited for obtaining polynomial-time results for CSPs, since already deciding whether a single relation, even when represented by a quantifier-free formula, is empty or not is NP-hard. For the representations given in (2)-(4), on the other hand, many interesting algorithms exist that solve the CSP for infinite-signature reducts in polynomial time. We would like to stress that when the signature of Γ is finite, these input representation issues do not arise: all the representations given above would give the same complexity results for CSP(Γ).

In some cases there is a different approach to deal with infinite signatures. Let Γ be a relational structure and let Γ' be the structure obtained from Γ by dropping some of the relations (i.e., Γ' is a reduct of Γ in the classical sense). We say that the relations of Γ' form a *basis* for Γ if all relations in Γ have a primitive positive definition over Γ'. In this case, and if the signature of Γ' is finite, then we can use primitive positive formulas over Γ' to represent the relations from Γ . The following lemma illustrates this approach (and justifies our presentation of linear program feasibility in the introduction); the proof is easy and can be found in [18].

▶ **Lemma 3.** *Every relation*

$$R := \{(x_1, \ldots, x_k) \in \mathbb{Q}^k \mid a_1x_1 + \cdots + a_kx_k \geq a_0\}$$

for $a_0, a_1, \ldots, a_k \in \mathbb{Q}$ *has a primitive positive definition over* $(\mathbb{Q}; \leq, R_+, R_{=1})$*. Moreover, we can find a primitive positive definition whose size is polynomial in the representation size of* $a_0, a_1, \ldots, a_k$ *when represented in binary.*

We would like to mention that for CSPs of templates with infinite signature it makes sense to consider the restricted version where only k variables are allowed in the input, for a fixed natural number k. The fact that k-variable integer linear program feasibility can be decided in polynomial time, for example, is a celebrated result of Lenstra [72].

For finite domains it is an open problem whether there are infinite constraint languages Γ such that CSP(Γ) is computationally hard if the relations in the constraints are represented explicitly, but where the computational hardness is not already witnessed by a finite subset of the relations of Γ; see the discussion in [26].

3 Linear Programming

Our journey through constraint satisfaction problems over numeric domains begins with the linear program feasibility problem. This problem will serve as a model, and also as a tool for further investigations.

▶ **Definition 4.** The problem LINEAR PROGRAM FEASIBILITY is the CSP of the structure with domain $\mathbb{R}$ and all the relations of the form

$$R^{\mathrm{LP}}_{a_1,\ldots,a_n,c} := \left\{x \in \mathbb{R}^n \;\middle|\; a_1x_1 + \cdots + a_nx_n \geq c\right\}$$

with $a_1, \ldots, a_n, c \in \mathbb{Q}$.

We assume, in our definition, that the variables range over the real numbers, but the coefficients must be rational. The restriction on the coefficients is justified by the need to manipulate them computationally. Nevertheless, one might want to abstract from the details of number representation. This might have practical reasons, because some applications rely on fixed precision floating point arithmetic implemented in hardware. Also, theoretically at least, one might pick the coefficients in a subset of the reals (or even of a non-Archimedean real-closed field) that is larger than the rational subfield, and yet admits an explicit representation. This approach leads to a model of computation in which (real) numbers are treated as black-box entities, and algorithms have access to an oracle that performs a certain set of basic operations on them. The complexity of algorithms is thus measured by the number of arithmetic operations and (crucially) order comparisons performed as a function of the amount of numerical inputs: this is the so-called Blum-Shub-Smale model of computation. An algorithm is said to be *strongly polynomial* if it is polynomial in both the Turing machine (or *bit*) model, and in the Blum-Shub-Smale model.

Linear programming is usually formulated as an optimization problem where the goal is to maximize a given linear function over the feasibility region. It is well known that linear programming is polynomial-time equivalent to the linear program feasibility problem (both in the Turing and in the Blum-Shub-Smale model). The groundbreaking application, due to Khachiyan, of the ellipsoid method provided the first polynomial-time algorithm for the linear program feasibility problem [64]. Other polynomial time algorithms addressing directly the optimization problem followed, notably Karmarkar's interior point *projective* method [62], and later barrier-function interior point methods (see for instance [103]). What all this techniques have in common is that they rely on infinite approximation procedures. To get from such methods a polynomial time decision procedure, one needs some a priori information of such nature, as to guarantee that the decision is determined by a degree of approximation obtainable in polynomial time. For example, Khachiyan's ellipsoid method needs a bound from below to the volume of the feasibility region (assuming that it has non-empty interior), and interior point methods require, essentially, to bound the representation size of the optimum (which is necessarily a rational number or $\pm\infty$). Thus neither of the known polynomial time algorithms could solve LINEAR PROGRAM FEASIBILITY on a non-Archimedean field, and none, in particular, is strongly polynomial.

The existence of a strongly polynomial algorithm for linear programming is, today, possibly the most important related problem. Dantzig's simplex method requires the complement of a *pivoting rule* to make a complete algorithm. Despite its practical effectiveness, no known pivoting rule has provably polynomial time, and most can be explicitly defeated [65, 55, 5, 43, 38, 37]. Sub-exponential randomized simplex algorithms have been constructed by Kalai [60] and Matoušek, Sharir, and Welzl [81]; see [47] for a recent improvement. If one restricts LINEAR PROGRAM FEASIBILITY to instances with specific properties, in some cases, strongly polynomial algorithms are known. By work of Tardos [97], this is the case if the coefficients a_i in Definition 4 are integral and bounded by a fixed constant. Megiddo describes a strongly polynomial algorithm for the case of two variables per inequality [82] and the same author gives a linear time combinatorial algorithm in fixed dimension [83] (i.e., for a fixed number of variables).

By formulating linear programming in the integer domain we obtain a well-known variation.

▶ **Definition 5.** The problem INTEGER LINEAR PROGRAM FEASIBILITY is the CSP of the structure with domain $\mathbb{Z}$ and all the relations of the form

$$R^{\text{ILP}}_{a_1,\dots,a_n,c} := \{x \in \mathbb{Z}^n \mid a_1x_1 + \cdots + a_nx_n \geq c\}$$

with $a_1, \dots, a_n, c \in \mathbb{Z}$.

As opposed to the real numbers formulation, INTEGER LINEAR PROGRAM FEASIBILITY is NP-complete. Indeed, even the special case in which the variables are restricted to the set $\{0, 1\}$ is among Karp's 21 problems [63]. We list a few notable polynomial-time restrictions. In the totally unimodular case (i.e., the instances are of the form $A\bar{x} = \bar{c}$ where the matrix A is totally unimodular), satisfiability in $\mathbb{R}$ implies satisfiability in $\mathbb{Z}$, hence INTEGER LINEAR PROGRAM FEASIBILITY can be solved in strongly polynomial time by Tardos's algorithm.

The fixed dimension case is solved in polynomial time by an algorithm of Lenstra [72]. By contrast to the real domain situation, the restriction to two variables per inequality is still NP-complete, by a result of Lagarias [71].

We will devote the next two sections to CSPs that expand linear programming. In particular, we will consider semilinear and algebraic constraints.

4 Semilinear Constraints

We say that a subset of $\mathbb{R}^n$ is *semilinear* if it can be defined by a finite Boolean combination of linear inequalities with integer coefficients. A relational structure with domain $\mathbb{R}$ is called *semilinear* if all its relations are. In particular, the template for LINEAR PROGRAM FEASIBILITY is semilinear. As for linear programming, in the semilinear context, the domains $\mathbb{R}$ and $\mathbb{Q}$ are interchangeable. Formally, we choose to state the results hereafter for $\mathbb{R}$.

4.1 Semilinear Expansions of Linear Programming

For semilinear expansions of linear programming, a P–NP-complete dichotomy has been proven by Bodirsky, Jonsson, and von Oertzen [18]. This dichotomy has then been extended by Jonsson and Thapper to all semilinear expansions of $(\mathbb{R}; +)$ in [59]. The dichotomy is based on the notion of essential convexity.

▶ **Definition 6.** A subset S of $\mathbb{R}^n$ is called *essentially convex* if for all $a, b \in S$ the straight line segment intersects the complement $\bar{S}$ of S in finitely many points.

▶ **Theorem 7** (Bodirsky–Jonsson–von Oertzen). *Let $R_1, \dots, R_n$ be semilinear relations. Then* $\mathrm{CSP}(\mathbb{R}; R_{=1}, R_+, \leq, R_1, \dots, R_n)$ *is in P if $R_1, \dots, R_n$ are essentially convex, and it is NP-complete otherwise.*

The hardness result in Theorem 7 follows from an even more general condition formulated in Lemma 30 in the next section. The algorithmic part is provided by the equivalence, for semilinear relations, of essential convexity and the class *Horn-DLR* proposed by Jonsson and Bäckström [57].

▶ **Definition 8.** A semilinear relation is called *Horn-DLR* (disjunctive linear relations) if it can be defined by a conjunction of clauses either of the form

$$p_1 \neq 0 \vee \cdots \vee p_n \neq 0$$

or of the form

$$p_1 \neq 0 \vee \cdots \vee p_n \neq 0 \vee p_0 \leq 0$$

where $p_0, \dots, p_n$ are linear terms with coefficients in $\mathbb{Z}$.

▶ **Lemma 9** (Bodirsky–Jonsson–von Oertzen). *A semilinear relation is essentially convex if and only if it is Horn-DLR.*

In turn, Horn-DLR constraints (represented by conjunctions of clauses as in Definition 8, with the coefficients expressed in binary) can be solved in polynomial time thanks to a resolution-like algorithm discovered by Jonsson and Bäckström [57] and independently by Koubarakis [69].

Convex semilinear relations can be characterised by a polymorphism. In fact, if $S \subset \mathbb{R}^n$ is semilinear then S is convex if and only if it is preserved by the midpoint function

$$(x, y) \mapsto \frac{x+y}{2}$$

(to see this, observe that if $p, q \in S$, then S contains a dense subset of the segment between p and q, so by being semilinear, S contains all but finitely many of the points of that segment). We mention that the same argument also works in an even more general setting, namely for *semialgebraic* relations, that will be introduced in Definition 18 in the next section. It would be desirable to also have a polymorphism characterisation of essentially convex semilinear relations. This is, unfortunately, not possible.

▶ **Observation 10.** *There is no function $f\colon \mathbb{R}^n \to \mathbb{R}$ such that a semilinear relation S is essentially convex if and only if f preserves S.*

Proof. For a contradiction, assume that such a function f exists. Without loss of generality, f depends on all its arguments. We will prove that f is injective, but first we see how to obtain a contradiction from this fact. Consider restriction of f to the diagonal

$$\begin{aligned} g\colon \mathbb{R} &\to \mathbb{R} \\ x &\mapsto f(x, x, \ldots, x) \end{aligned}$$

For each rational x, the singleton $\{x\}$ is clearly semilinear and essentially convex, therefore $g|_{\mathbb{Q}}$ is the identity. The order relation $\{(x, y) \mid x < y\}$ is a semilinear essentially convex set, therefore g is order preserving. It follows from the density of $\mathbb{Q}$ in $\mathbb{R}$ that g is the identity from $\mathbb{R}$ to $\mathbb{R}$. In particular, the image of g is precisely $\mathbb{R}$, so f can not be injective unless $n = 1$. In this case, $f = g$ is the identity and it preserves every set.

It remains to prove that f is injective. The relation $x = y \Rightarrow u = v$ is essentially convex, therefore it must be preserved by f. We claim that this implies the injectivity. Suppose that there are distinct $a, b \in \mathbb{R}^n$ such that $f(a) = f(b)$. We want to prove that f violates $x = y \Rightarrow u = v$. Pick i such that $a_i \neq b_i$. Since f depends on every argument, there are $c, d \in \mathbb{R}^n$ with $c_j = d_j$ for all $j \neq i$, and $c_i \neq d_i$ such that $f(c) \neq f(d)$. We claim that (a, b, c, d) witnesses that f violates $x = y \Rightarrow u = v$. In fact, for all $j \neq i$, we have $c_j = d_j$, and, a fortiori, $a_j = b_j \Rightarrow c_j = d_j$. Also $a_i = b_i \Rightarrow c_i = d_i$, because, by construction, the premise of the implication is false. We conclude that for all $j \in \{1, \ldots, n\}$ we have that $a_i = b_i$ implies $c_i = d_i$. However, $f(a) = f(b)$ and $f(c) \neq f(d)$. ◀

Nevertheless, one can see that essentially convex sets can be characterised by a polymorphism in a non-Archimedean extension on $\mathbb{Q}$. All totally ordered vector spaces over $\mathbb{Q}$ have the same semilinear CSP. In fact, the first-order theory of non-trivial totally ordered vector spaces over $\mathbb{Q}$ is complete (see for instance [99][Chapter 2, Remark 7.9]). Therefore we can replace $\mathbb{R}$ with such a non-Archimedean extension in the study of semilinear CSPs.

▶ Remark. Let $\mathbb{Q}[\epsilon]$ denote the $\mathbb{Q}$-vector space of all polynomials in one indeterminate ϵ with rational coefficients. Consider the order on $\mathbb{Q}[\epsilon]$ which is induced by viewing ϵ as a positive number that is smaller than all positive rational numbers; formally, we order the polynomials lexicographically with respect to their coefficients, starting with the constant term, then the coefficient of degree one, and so on in increasing order of degree. Semilinear and essentially convex subsets of $(\mathbb{Q}[\epsilon])^d$ are defined as for the reals. Now, let S be a semilinear relation, let ϕ be the Boolean combination of inequalities with integer coefficients that defines S, and let S' be the semilinear relation defined by ϕ on $\mathbb{Q}[\epsilon]$. Then the following are equivalent:

- S is essentially convex;
- S' is essentially convex;
- S' is preserved by the following function

$$\begin{aligned} &f\colon (\mathbb{Q}[\epsilon])^2 \to \mathbb{Q}[\epsilon] \\ &(\alpha(\epsilon), \beta(\epsilon)) \mapsto \gamma(\epsilon) := \left(\frac{1}{2} + \epsilon\right) \alpha(\epsilon^2) + \left(\frac{1}{2} - \epsilon\right) \beta(\epsilon^2)\,. \end{aligned}$$

Proof Sketch. If S is not essentially convex, then this is witnessed by rational points, and it is easy to see that S' cannot be preserved by f. Conversely, we prove that f preserves all essentially convex sets over $\mathbb{Q}[\epsilon]$. Observe that f is injective, because

$$\begin{aligned} &\alpha(\epsilon^2) = \left(\frac{1}{2} + \frac{1}{4\epsilon}\right) \gamma(\epsilon) + \left(\frac{1}{2} - \frac{1}{4\epsilon}\right) \gamma(-\epsilon) \\ \text{and}\quad &\beta(\epsilon^2) = \left(\frac{1}{2} - \frac{1}{4\epsilon}\right) \gamma(\epsilon) + \left(\frac{1}{2} + \frac{1}{4\epsilon}\right) \gamma(-\epsilon)\,. \end{aligned}$$

Therefore, using the syntactic characterization of essential convexity in [57] (which holds in all ordered $\mathbb{Q}$-vector spaces by the completeness mentioned above), we only need to prove that f preserves all the rational constants, $+$, and the positivity relation $x > 0$, and this is immediate. ◀

Jonsson and Thapper prove a dichotomy result for the larger class of all semilinear expansions of $(\mathbb{R}; +)$ [59]. These include all the CSPs covered by Theorem 7, but also contain templates that may not be able to express full linear programming. For instance, expanding linear programming by the non essentially convex relation $|x - y| > 1$ produces an NP-complete CSP, nevertheless $(\mathbb{R}; +, |x - y| > 1)$ is tractable. In its short form, Jonsson and Thapper's result reads as follows.

▶ **Theorem 11** (Jonsson–Thapper). *Let $R_1, \dots, R_n$ be semilinear relations. Then the problem* $\mathrm{CSP}(\mathbb{R}; R_+, R_1, \dots, R_n)$ *is either in P or NP-complete.*

Theorem 11 comes with explicit tractability conditions. Unfortunately, these conditions are too complex to fit comfortably into our survey. However, we can get a feeling of the insights by looking at the special case of semilinear expansions of $(\mathbb{R}; R_+, R_{=1})$.

▶ **Corollary 12.** *Let $R_1, \dots, R_n$ be semilinear relations. Then* $\mathrm{CSP}(\mathbb{R}; R_+, R_{=1}, R_1, \dots, R_n)$ *is in P if one of the following conditions is satisfied:*

1. *all unary relations primitively positively definable in* $(\mathbb{R}; R_+, R_{=1}, R_1, \dots, R_n)$ *are essentially convex;*
2. *all unary relations primitively positively definable in* $(\mathbb{R}; R_+, R_{=1}, R_1, \dots, R_n)$ *are either singletons or unbounded (i.e. not contained in an interval);*

otherwise, it is NP-complete.

Here the first case is dealt with by reduction to the essentially convex case, the second by a procedure, called *affine consistency*, that approximates from above the affine span of the feasibility region of the input constraints.

For every semilinear structure Γ with finite signature $\mathrm{CSP}(\Gamma)$ is in NP; this can be observed from a more general result which also holds for semilinear structures with an infinite signature. When the semilinear relations in Γ are represented by quantifier-free formulas where the constants in the polynomials are represented in binary, then we can check satisfiability of an instance non-deterministically by first guessing a disjunct for each disjunction in the formula, and then checking in polynomial time the satisfiability of the resulting feasibility problem obtained as the conjunction of the selected (strict or non-strict) linear inequalities, e.g. using the algorithm of Jonsson and Bäckström [57].

4.2 Semilinear Constraints over the Integers

Semilinear subsets of $\mathbb{Z}^n$ and semilinear structures with domain $\mathbb{Z}$ are defined analogously as above for $\mathbb{R}$. A special case of tractable semilinear constraints over the integers has been found by Jonsson and Lööw [58].

▶ **Definition 13.** A semilinear structure Γ over $\mathbb{R}$ is called *scalable* if for every relation R of Γ and for every $(x_1, \ldots, x_k) \in R$ there exists a positive $a \in \mathbb{R}$ such that $(bx_1, \ldots, bx_k) \in R$ for all $b > a$.

▶ **Definition 14.** A semilinear structure Γ with domain $\mathbb{R}$ has the *integer property* if all satisfiable instances of $\mathrm{CSP}(\Gamma)$ are satisfiable over $\mathbb{Z}$.

▶ **Theorem 15** (Jonsson–Lööw). *All scalable semilinear structures have the integer property.*

A partial converse to Theorem 15 is given by the following result of Jonsson and Thapper [59].

▶ **Theorem 16** (Jonsson–Thapper). *Let Γ be a semilinear structure such that $\mathrm{CSP}(\Gamma, R_+)$ has the integer property. Then Γ is scalable.*

The following observation shows that CSPs of semilinear structures over the integers in fixed dimension are in P, using Lenstra's algorithm.

▶ **Observation 17.** *For every fixed $d \in \mathbb{N}$ there is a polynomial time algorithm deciding the satisfiability over $\mathbb{Z}$ of Boolean combinations of formulas of the form $p(x_1, \ldots, x_n) \geq 0$ where p is a linear polynomial with variables $x_1, \ldots, x_n$ and integer coefficients that are represented in binary.*

Proof. For each input formula ϕ, our algorithm first extracts the list of the linear polynomials $p_1, \ldots, p_k$ appearing in ϕ. We say that two points $\bar{x}, \bar{y} \in \mathbb{R}^d$ have the same type if $\mathrm{sign}(p_i(\bar{x})) = \mathrm{sign}(p_i(\bar{y}))$ for all $i = 1, \ldots, k$, where

$$\mathrm{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}.$$

Clearly the truth value of $\phi(\bar{x})$ depends only on the type of $\bar{x}$.

It is easy to prove that there are points in $\mathbb{R}^d$ of at most $\tau_d(k)$ distinct types, where

$$\tau_d(k) = \sum_{i=0}^{d} 2^i \binom{k}{i}.$$

In fact, we claim that there are at most $\tau_d(k)$ types, and that this bound is tight when the hyperplanes $p_k = 0$ are in general position. By induction on k, we add the polynomials one by one. At the k-th step, we create at most $\tau_{d-1}(k-1)$ new types with $p_k = 0$, and exactly as many in general position. Now we count the $p_k \gtrless 0$ types. For each of the new $p_k = 0$ types, we see that its restriction to $p_1, \ldots, p_{k-1}$, that we had at the $(k-1)$-th step, splits into two new types corresponding to $p_k < 0$ and $p_k > 0$. The remaining part of the $(k-1)$-th step types is incompatible with either $p_k < 0$ or $p_k > 0$, so it does not produce any new non-empty type. Thus we get the recurrence $\tau_d(k) = 2\tau_{d-1}(k-1) + \tau_d(k-1)$, hence the formula.

In particular, $\tau_d(k)$ is polynomial in k, therefore we can compute a list of all these types in polynomial time by solving at most $\sum_{i=0}^{k-1} \tau_d(i)$ linear programs. Now, using Lenstra's algorithm, we can exclude those types that do not contain points in $\mathbb{Z}^d$, and finally we check the truth value of ϕ for each of the remaining ones. ◀

5 Algebraic Constraints

Adding zero sets of polynomials to the set of basic constraints will rapidly bring about intractability. In the integer domain, the general problem of the satisfiability of a single (multi-variate) polynomial equation, also known as Hilbert's 10th problem, is undecidable by the celebrated result of Davis, Matiyasevich, Putnam, and Robinson. In fact, they show that satisfiability of a 13 variable polynomial is undecidable over $\mathbb{N}$, later improved by Matiyasevich to 9 (see [78] and [56]). Using the 4 squares theorem, therefore, satisfiability of a single polynomial equation in $\mathbb{Z}$ is also undecidable for a fixed dimension (just replace each variable x_i with $y_{i,1}^2 + y_{i,2}^2 + y_{i,3}^2 + y_{i,4}^2$). We are not aware of any decidability result in dimension 2. It has been proven by Manders and Adleman [74] that deciding the satisfiability of a single equation of the form

$$ax^2 + bx = c$$

over $\mathbb{N}$ is NP-complete. The decidability of satisfiable polynomial equations over the rationals is a major open problem. For more details on Hilbert's tenth problem and its extension to $\mathbb{Q}$, we refer the reader to [80] and [95].

One might consider a reduct of integer arithmetic in which *only* multiplication, and no addition, is available. It is well known that this fragment, called Skolem arithmetic, has a decidable first-order theory. Exploratory work on CSPs of reducts of Skolem arithmetic has been published by Glaßer, Jonsson, Martin [42].

The natural relations to consider as basic constraints over the real numbers are the semialgebraic relations.

▶ **Definition 18.** A subset of $\mathbb{R}^n$ is called *semialgebraic* if it can be represented as a finite Boolean combination of *basic semialgebraic sets* of the form $\{\bar{x} \in \mathbb{R}^n \mid p(\bar{x}) \geq 0\}$ where p is a polynomial with integer coefficients.

Semialgebraic sets over $\mathbb{R}$ are a rich yet manageable class that forms the basis of real algebraic geometry. The computational treatment of semialgebraic geometry goes back to Tarski's decision procedure for the first-order theory of the reals [98].

▶ **Theorem 19** (Tarski)**.** *There is an effective quantifier elimination procedure for the first-order theory of the structure* $(\mathbb{R}; 0, 1, <, +, \times)$.

The complexity bound on Tarski's original procedure is very large: a tower of exponentials as high as the size of the input. Collins' cylindrical algebraic decomposition provides a method running in polynomial time for fixed dimension (fixed number of variables in the formula), and doubly exponential in the dimension [30]. This bound has been further improved to exponential in the dimension by Renegar [88, 89, 90]. Of special interest to us is the subproblem known as the existential theory of the reals.

▶ **Definition 20.** The problem EXISTENTIAL THEORY OF THE REALS is the CSP for the structure with domain $\mathbb{R}$ and all the relations of the form

$$R^{\text{ETR}}_{n,p} = \{\bar{x} \in \mathbb{R}^n \mid p(\bar{x}) = 0\}$$

where $p \in \mathbb{Z}[x_1, \ldots, x_n]$ is represented as a list of coefficients, in binary.

The best bound currently known on the complexity of EXISTENTIAL THEORY OF THE REALS is the following, obtained by Canny [28].

▶ **Theorem 21** (Canny). *The problem* EXISTENTIAL THEORY OF THE REALS *is in* PSPACE.

We chose basic relations of the form $p(x) = 0$. Adding to this set also all the relations of the form $p(x) \geq 0$ and $p(x) > 0$ would not increase the complexity of EXISTENTIAL THEORY OF THE REALS. In fact, the first is clearly equivalent to $\exists y \; p(x) - y^2 = 0$ and $x \neq 0$ is defined by $\exists y \; xy + 1 = 0$. More generally, it is easy to see that the semialgebraic relations are precisely those that have a primitive positive definition over the structure $(\mathbb{R}; (R^{\text{ETR}}_{n,p})_{n,p})$. As for linear programming, we chose to present the existential theory of the reals as an infinite signature CSP, it is however an easy exercise to find a finite basis for it. A non-trivial equivalence between the existential theory of the reals and the following CSP has been established by Schaefer and Štefankovič [92][Theorem 4.1]

▶ **Definition 22.** The problem STRICT INEQUALITIES is the CSP of the structure with domain $\mathbb{R}$ and the relations of the form

$$R^{\text{STRICT}}_{n,p} = \{\bar{x} \in \mathbb{R}^n \mid p(\bar{x}) > 0\}$$

where $p \in \mathbb{Z}[x_1, \ldots, x_n]$ is represented as a list of coefficients, in binary.

▶ **Theorem 23** (Schaefer–Štefankovič). *The problems* STRICT INEQUALITIES *and* EXISTENTIAL THEORY OF THE REALS *are polynomial time equivalent.*

Observe that the sets primitively positively definable in $(\mathbb{R}; (R^{\text{STRICT}}_{n,p})_{n,p})$ are necessarily open, therefore a strict subset of all semialgebraic sets. A number of other problems, many of a geometric flavour, have been proven to be equivalent to EXISTENTIAL THEORY OF THE REALS. We will mention two such results due to Kratochvíl and Matoušek [70], and Schaefer [91] respectively.

▶ **Definition 24.** The problem INTERSECTION GRAPH OF SEGMENTS is the CSP of the structure with domain $\mathbb{R}^4$ and the binary relations Int and its negation ¬Int, defined as follows. Let $S^{x_2,y_2}_{x_1,y_1}$ denote the straight line segment in $\mathbb{R}^2$ with endpoints (x_1, y_1) and (x_2, y_2).

$$\big((x_1, y_1, x_2, y_2)(x'_1, y'_1, x'_2, y'_2)\big) \in \text{Int} \;\Leftrightarrow\; S^{x_2,y_2}_{x_1,y_1} \cap S^{x'_2,y'_2}_{x'_1,y'_1} \neq \emptyset$$

▶ **Theorem 25** (Kratochvíl–Matoušek). *The problems* INTERSECTION GRAPH OF SEGMENTS *and* EXISTENTIAL THEORY OF THE REALS *are polynomial time equivalent.*

▶ **Definition 26.** The problem UNIT LENGTH LINKAGES is the CSP for the structure with domain $\mathbb{R}^2$ and the relation

$$R^{\text{ULL}} = \{\big((x, y), (x', y')\big) \mid (x - x')^2 + (y - y')^2 = 1\}$$

denoting that the Euclidean distance of (x, y) and (x', y') is one.

▶ **Theorem 27** (Schaefer). *The problems* UNIT LENGTH LINKAGES *and* EXISTENTIAL THEORY OF THE REALS *are polynomial time equivalent.*

The existential theory of the reals has a natural analogue in the complex field. The problem of deciding the satisfiability of a system of polynomial equations in $\mathbb{C}$ has been called the *Hilbert Nullstellensatz problem.*

▶ **Definition 28.** HILBERT NULLSTELLENSATZ PROBLEM is the CSP of the structure with domain $\mathbb{C}$ and the relations of the form

$$R^{\text{HN}}_{n,p} = \{\bar{x} \in \mathbb{C}^n \mid p(\bar{x}) = 0\}$$

where $p \in \mathbb{Z}[x_1, \ldots, x_n]$ is represented as a list of coefficients, in binary.

It is not hard to reduce HILBERT NULLSTELLENSATZ PROBLEM to EXISTENTIAL THEORY OF THE REALS, but a far lower complexity bound for HILBERT NULLSTELLENSATZ PROBLEM is known conditionally to the generalized Riemann Hypothesis, thanks to the following result of Koiran [67, 68].

▶ **Theorem 29** (Koiran). *Assuming the generalized Riemann Hypothesis,* HILBERT NULLSTELLENSATZ PROBLEM *is in the class Arthur-Merlin, hence, a fortiori, at the second level of the polynomial hierarchy (Π_2).*

For the Blum-Shub-Smale model of computation, HILBERT NULLSTELLENSATZ PROBLEM and EXISTENTIAL THEORY OF THE REALS are both NP-complete, for the complex and the real Turing machine model respectively, see [12].

In order for a semialgebraic expansion of linear programming to attain a complexity class below NP, it is necessary to restrict the template to essentially convex relations, because of the following result of Bodirsky, Jonsson, and von Oertzen [18][Lemma 3.5].

▶ **Lemma 30** (Bodirsky–Jonsson–von Oertzen). *Let R be a semialgebraic set which is not essentially convex, and suppose that the failure of essential convexity is witnessed by a segment with rational endpoints, then* $\text{CSP}(\mathbb{R}; R_+, R_{=1}, \leq, R)$ *is NP-hard.*

The technical assumption about rational endpoints is indeed necessary, see [18][Remark 3.6]. Even convexity is far from being a sufficient condition for tractability. One obstacle comes from the well-known sum of square roots problem.

▶ **Definition 31.** The problem SUM OF SQUARE ROOTS is the following computational task:
Input: $a_1, \ldots, a_n, b \in \mathbb{N}$
Output:

- YES if $b \leq \sqrt{a_1} + \cdots + \sqrt{a_n}$,
- NO otherwise.

It is a long standing open problem to characterise the complexity of this problem. The membership in NP of SUM OF SQUARE ROOTS is also open and would imply that the Euclidean travelling salesman problem is NP-complete [40]. In fact, SUM OF SQUARE ROOTS is not even known to be in the polynomial hierarchy, and the best upper bound today is the fourth level of the counting hierarchy [2]. Unfortunately, already very simple CSPs, for instance the expansion of the template for linear program feasibility by the relation $x^2 \leq y$, can simulate SUM OF SQUARE ROOTS.

▶ **Observation 32.** SUM OF SQUARE ROOTS *is polynomial time many-one reducible to*

$$\mathrm{CSP}(\mathbb{R}, R_+; R_{=1}, +, x^2 \leq y)\,.$$

Proof.

$$b \leq \sqrt{a_1} + \cdots + \sqrt{a_n} \;\Leftrightarrow\; \exists x_1, \ldots, x_n \in \mathbb{R} \begin{cases} b = x_1 + \cdots + x_n \\ x_1^2 \leq a_1 \\ \quad\vdots \\ x_n^2 \leq a_n \end{cases}$$

◀

▶ **Observation 33.** SUM OF SQUARE ROOTS *is polynomial time many-one reducible to*

$$\mathrm{CSP}(\mathbb{R}; R_+, R_{=1}, x^2 + y^2 \leq 1)\,.$$

Proof. It suffices to find a primitive positive definition for the relation $0 \leq x \leq \sqrt{k}$ of size $O(\log k)$ for all constant $k \in \mathbb{N}$. We use the following equivalence

$$x^2 \leq k \;\Leftrightarrow\; \exists a, b \in \mathbb{R} \begin{cases} a^2 + b^2 \leq 1 \\ (k+1)a = 2x \\ (k+1)b = k-1 \end{cases}$$

observing that the linear relations $(k+1)a = 2x$ and $(k+1)b = k-1$ admit a succinct primitive positive definition by iterated doubling. ◀

There is a connection between convex semialgebraic CSPs and semidefinite programming. A *spectrahedron* is a subset of $\mathbb{R}^d$ of the form

$$\big\{\bar{x} \in \mathbb{R}^d \;\big|\; x_1A_1 + \cdots + x_dA_d + B \text{ is positive semidefinite}\big\}$$

where $A_1, \ldots, A_d, B$ are real symmetric matrices of the same size. Semidefinite programming is the task of minimizing linear functions over spectrahedral domains. The relations that are primitively positively definable over spectrahedra are called *semidefinite representable*. Clearly all semidefinite representable relations are convex and semialgebraic. The converse has been conjectured by Helton and Nie [48]: they conjecture that every convex semialgebraic set is semidefinite representable. Helton and Nie's conjecture has been proven in dimensions two [93], and it remains one of the most important open problems in the field. The Helton-Nie conjecture implies that every CSP for a template with finitely many convex semialgebraic relations would be a special case of the semidefinite program feasibility problem. Not much is known about the complexity of this problem. By Ramana's duality [87] it is either in $\mathrm{NP} \cap \mathrm{coNP}$ or outside of $\mathrm{NP} \cup \mathrm{coNP}$. By results of Tarasov and Vyalyi [96] the evaluation of arithmetic circuits (PosSLP) is reducible to it, thus also SUM OF SQUARE ROOTS (for the reduction from SUM OF SQUARE ROOTS to PosSLP see [2]).

In fixed dimension, all semialgebraic CSPs are in P by the cylindrical decomposition algorithm [30]. However, if we except Boolean combinations of linear inequalities (with algebraic coefficients), we do not know any expansion of linear programming by a convex non-linear semialgebraic relation which has a CSP in P. The theory of primitive positive definability for reducts of the real field structure is not developed. For instance, we do not know whether the relation $x^6 \leq y$ is primitively positively definable over linear relations and $x^2 \leq y$.

6 Maximum as a Polymorphism

The maximum polymorphism describes interesting classes of semi-linear CSPs both in the rational and in the integer domain. To begin with, it has been proven by Jeavons and Cooper [54] that, for finite domains, if all the relations of a template are preserved by the maximum polymorphism (according to some total ordering of the domain) then the CSP is in P. The same authors observe that the CSP of all max-closed relations on a finite domain is maximally tractable, in other words adding any further constraints to it would make it NP-complete. Formally we have:

▶ **Theorem 34** (Jeavons–Cooper). *Any max-closed CSP instance containing c constraints each of them satisfied by at most t tuples can be solved in time $O(c^2t^2)$.*

▶ **Theorem 35** (Jeavons–Cooper). *Given a structure $(D;<)$ where $3 \leq |D| < \infty$ and $<$ is a total order on D, denote by $\mathcal{M}_{2,D}$ the (finite) set of all binary max-closed relations over D. Let R be any relation over D which is not max-closed. Then $\mathrm{CSP}(D;\mathcal{M}_{2,D},R)$ is NP-complete.*

Observe that the statement of Theorem 34 is uniform in the constraints, in the sense that the algorithm takes the constraints (represented in table form) as part of its input. Therefore, Theorem 34 is adapted to situations in which the relevant domain is not fixed. This algorithm is, indeed, the well-known *arc consistency* procedure. Generally speaking, algorithms of this variety keep, for each variable, a list of allowed values, and iterate over the constraints to check whether any of the allowed values can be excluded. Obviously the procedure converges to a fixed point after at most $n \cdot d$ iterations, where n denotes the size of the domain and d the number of variables of the input instance. If at any time some variable has no more allowed values, the instance is unsatisfiable. Otherwise, there might or there might not be a solution.

Jeavons and Cooper obtain Theorem 34 by showing that, despite providing in general only a one-sided test, arc consistency correctly decides all max-closed CSPs. Jeavons, Cohen, and Gyssens extended this result to finite domain CSPs having a semilattice polymorphism, defined as follows.

▶ **Definition 36.** The operation $f\colon D^2 \to D$ is called a *semilattice operation* if it is idempotent, commutative, and associative:

$$f(x,x) = x, \qquad f(x,y) = f(y,x), \qquad f(x,f(y,z)) = f(f(x,y),z).$$

▶ **Theorem 37** (Jeavons–Cohen–Gyssens). *A finite domain CSP whose relations are preserved by a semilattice operation is in P.*

To complete the picture for finite domain problems, let us point out that arc consistency naturally extends to a class of algorithms based on establishing *local consistency*. The finite domain CSPs that can be solved by this method have been recently characterized by Barto and Kozik [7].

Theorem 34 gives us a method to solve infinite domain CSPs through a technique termed *sampling* in [21, Definition 2.1]. We say that an infinite template Γ has a *sampling procedure* if for every instance I of $\mathrm{CSP}(\Gamma)$ we can construct in polynomial time a finite substructure Γ_I of Γ such that I is satisfiable if and only if I is satisfiable as an instance of $\mathrm{CSP}(\Gamma_I)$. Clearly, Theorem 34 coupled with a sampling procedure for Γ provides a polynomial-time decision procedure for $\mathrm{CSP}(\Gamma)$ whenever Γ is closed under maximum. This general method has provided polynomial and weakly polynomial time algorithms for several concrete CSPs in the past. Most notably the following results, published respectively by Hochbaum and Naor [51], and by Bezem, Nieuwenhuis, and Rodríguez-Carbonell [10].

▶ **Definition 38.** The problem MONOTONE TVPI INTEGER PROGRAMMING (for *two variables per inequality*) is the CSP of the structure with domain $\mathbb{Z}$ and all the relations of the form

$$R^{\text{mTVPI}}_{a,b,c} = \big\{(x,y) \in \mathbb{Z}^2 \;\big|\; ax - by \geq c\big\}$$

where $a, b \in \mathbb{N}$ and $c \in \mathbb{Z}$.

▶ **Theorem 39** (Hochbaum–Naor). *A solution to an instance of* MONOTONE TVPI INTEGER PROGRAMMING *taking values from $\{-N, \dots, N\}$ can be computed in time polynomial in N and the size of the instance.*

▶ **Definition 40.** The problem MAX-ATOMS is the CSP of the structure with domain $\mathbb{Z}$ and all the relations of the form

$$R^{\text{MA}}_{c} = \big\{(x,y,z) \in \mathbb{Z}^3 \;\big|\; \max(x,y) + c \geq z\big\} \qquad \text{for } c \in \mathbb{Z}.$$

▶ **Theorem 41** (Bezem–Nieuwenhuis–Rodríguez-Carbonell). *The problem* MAX-ATOMS *can be solved in weakly polynomial time.*

The proofs of these theorems rely on the observation that the respective signatures are max-closed (and the relations $R^{\text{mTVPI}}_{a,b,c}$ are indeed also min-closed). Broadly speaking, both results provide algorithms of the weakly polynomial kind: Theorem 41 explicitly, Theorem 39 because one can compute bounds $x_i^{\max}$ and $x_i^{\min}$ polynomial in the size of the instance [94][§17.1]. One might wonder whether these results can be improved to actually yield a polynomial algorithm. At least for Theorem 41 this is unfortunately not the case.

▶ **Theorem 42** (Lagarias). *The problem* MONOTONE TVPI INTEGER PROGRAMMING *is NP-complete.*

Lagarias reduces MONOTONE TVPI INTEGER PROGRAMMING to the problem WEAK PARTITION [71], which was proven NP-complete ultimately by reduction from KNAPSACK [100]. Theorems 39 and 42 together say that MONOTONE TVPI INTEGER PROGRAMMING is weakly NP-complete (assuming P $\neq$ NP). It can be observed, however, that being max-closed does not imply tractability in the integer domain even for finite signature CSPs.

▶ **Example 43.** CSP($\mathbb{Z}$; $x = 1$, $x = -1$, $x = 2y$, $x + y \geq z$) is NP-complete.

Proof. Our CSP is clearly in NP, because it is a sub-problem of INTEGER LINEAR PROGRAMMING. NP-hardness is proven through a reduction from MONOTONE TVPI INTEGER PROGRAMMING. The constraint $ax - by \geq c$ with $a, b \in \mathbb{N}$ is equivalent to $ax + (2^\beta - b)y - c \geq 2^\beta y$, where β is the smallest exponent such that $2^\beta \geq b$. This is, in turn, equivalent to the primitive positive formula

$$\exists\, z_1, z_2, z_3, z_4, z_5, z_6 \quad \begin{cases} ax \geq z_1 \\ (2^\beta - b)y \geq z_2 \\ z_1 + z_2 \geq z_3 \\ z_4 = \operatorname{sign}(-c) \\ |c|\, z_4 \geq z_5 \\ z_3 + z_5 \geq z_6 \\ z_6 = 2^\beta y \end{cases} \qquad .$$

All atomic relations in this formula are of the types $x+y \geq x$, $x = 2y$, 1 , -1 except $z_6 = 2^\beta y$ and those of the form $kv_1 \geq v_2$ for $k \geq 0$ and v_1, v_2 denoting variables. Relations of the first kind are easily expressed by chaining β constraints of the type $x = 2y$. To express $kv_1 \geq v_2$ we proceed recursively:

$$\begin{cases} 0 \geq v_2 \quad \Leftrightarrow \quad \exists t \;\; t = 2 * t \;\wedge\; t + t \geq v_2 & \text{for } k = 0 \\ kv_1 \geq v_2 \quad \Leftrightarrow \quad \exists t \;\; k'v_1 \geq t \;\wedge\; t + t \geq v_2 & \text{for } k = 2k' \text{ even} \\ kv_1 \geq v_2 \quad \Leftrightarrow \quad \exists t \;\; 2k''v_1 \geq t \;\wedge\; t + v_1 \geq v_2 & \text{for } k = 2k'' + 1 \text{ odd.} \end{cases}$$

◀

The concept of *primitive positive interpretation* formalises a powerful form of simulation of a CSP by another CSP, generalising the notion of primitive positive definability to templates that have different domains; for the technical definition see [14]. Since the existence of a semilattice polymorphism is preserved by primitive positive interpretations, the CSP of example 43 cannot interpret primitively positively any NP-complete finite domain CSP, yet it is NP-complete.

The situation changes for the MAX-ATOMS problem. First of all, one can formulate MAX-ATOMS also for the rational or the real domains, however, as it was observed already in [10], the real, rational, and integer formulations are polynomial-time equivalent. MAX-ATOMS is also unlikely to be NP-complete because it was proven in the same paper to be in NP ∩ coNP. The original proof of NP ∩ coNP membership constructs *small unsatisfiability certificates* using an appropriate proof system. The result, however, can be better understood through a connection with mean payoff games implicit in the work of Möhring, Skutella, and Stork on scheduling under and/or precedence constraints [84], and later discovered independently by Atserias and Maneva [4].

Mean payoff games are a class of so-called graph-games. The setup for a game $\mathcal{G}$ is a finite graph G whose edges (E_G) are labelled integer weights $\{w_e\}_{e \in E_G}$. The play takes place between two players, which we call Max and Min, taking turns at moving a token along the edges of G. The graph G has no sinks, and the vertices are divided into two subsets $V_G = V_G^{\max} \sqcup V_G^{\min}$, the first *belongs* to Max, the second to Min. The token is initially on some vertex v_0, and each player selects the next move when the token is on a vertex belonging to him. The value of a (infinite) play $v_0, v_1, \ldots$ is

$$\operatorname{val}(v_0, v_1, \ldots) = \liminf_{n \to \infty} \frac{1}{n+1} \sum_{i=0}^{n} w_{(v_i, v_i+1)} \,.$$

Player Max wants to maximize this value, Min wants to minimize it. The following theorem was proven by Eherenfeucht and Mycielski [34].

▶ **Theorem 44** (Ehrenfeucht–Mycielski). *Given a mean payoff game $\mathcal{G}$ and a starting vertex v_0 in the graph of $\mathcal{G}$, there is a value $\operatorname{val}(\mathcal{G}, v_0)$ and a pair of memoryless strategies σ and τ (not depending on v_0) for Max and Min respectively, such that Max can secure a value not smaller than $\operatorname{val}(\mathcal{G}, v_0)$ by playing according to σ, and Min can secure a value not larger than $\operatorname{val}(\mathcal{G}, v_0)$ by playing according to τ.*

Here *memoryless* means that at all times the move to play depends only on the current position of the token (and not, for instance, on the previous play or on a random choice).

▶ **Definition 45.** The problem MEAN PAYOFF GAMES is the following computational task:
Input: a mean payoff game $\mathcal{G}$.
Output:

- YES if $\text{val}(\mathcal{G}, v_0) \geq 0$ for all starting vertices v_0 in the graph of $\mathcal{G}$,
- NO otherwise.

It turns out that MAX-ATOMS and MEAN PAYOFF GAMES are equivalent in the following precise sense [84].

▶ **Theorem 46** (Möhring–Skutella–Stork). *The problems* MAX-ATOMS *and* MEAN PAYOFF GAMES *are polynomial-time many-one reducible to each other.*

The NP ∩ coNP membership of MEAN PAYOFF GAMES was first observed by Zwick and Paterson [104][Theorem 4.2], and it is essentially a consequence of the symmetric nature of the game. In fact, although Definition 45 is not symmetric in its form, standard arguments (see for instance [102][Propositions 2.7, 2.8]) prove that that the decision problem MEAN PAYOFF GAMES is polynomial-time Turing equivalent to solving mean payoff games (i.e., computing a pair of optimal memoryless strategies with the property from Theorem 44).

Theorem 46 is actually a re-discovery of the concept of *potential transformation*, which has been known for games since the work of Gurvich, Karzanov, and Khachiyan [45]. The set of conditions that a potential transformation must satisfy in order to witness $\text{val}(\mathcal{G}, v_0) \geq 0$ for all v_0 is, indeed, precisely the corresponding MAX-ATOMS instance in Atserias and Maneva's reduction.

The problem MAX-ATOMS has been, in turn, proven equivalent to a number of other well-known computational tasks, most notably scheduling under and-or precedence constraints, and solving two-sided systems of max-plus linear equations. The problem TWO SIDED MAX-PLUS LINEAR SYSTEMS is the problem of deciding whether a given system of equalities of the form

$$\max(a_1 + x_1, \ldots, a_m + x_m) = \max(b_1 + y_1, \ldots, b_n + y_n)$$

with $a_1, \ldots, a_m, b_1, \ldots, b_n \in \mathbb{Z}$ has a solution. As for MAX-ATOMS, it is immaterial whether we are looking for a solution over $\mathbb{Z}$, $\mathbb{Q}$, or $\mathbb{R}$.

▶ **Theorem 47** (Bezem–Nieuwenhuis). *The problems* MAX-ATOMS *and* TWO SIDED MAX-PLUS LINEAR SYSTEMS *are polynomial-time many-one reducible to each other.*

Max-plus algebra studies the algebraic properties of the *tropical semiring*, which can be defined as the structure $\mathbb{R} \cup \{-\infty\}$ with the standard ring operations $+$ and $\times$ replaced by max and $+$ respectively. Its study is motivated by applications to scheduling and routing problems, as well as more abstract connections to algebraic geometry. TWO-SIDED MAX-PLUS LINEAR SYSTEMS are precisely the analogue in the tropical semiring of standard systems of linear equations.

The best known algorithms for this cluster of equivalent problems come apparently from the game perspective. They are weakly polynomial, for instance Zwick and Paterson's [104], or sub-exponential (randomized), for instance Halman's [46], or both, as Björklund and Vorobyov's [11]. Other algorithms are known to converge quickly in practice (similarly to the simplex algorithm for linear programming), for example Gurvich, Karzanov, and Khachiyan's procedure [45] and its variants.

The problem of the existence of a polynomial-time solution for MEAN PAYOFF GAMES, or equivalently MAX-ATOMS, has been described as presenting now *exactly the same challenge* *as linear programming did before 1979* [102]. This description is grounded essentially on observations of an algorithmic nature. For instance, the two sub-exponential algorithms cited above are based precisely on adapting the sub-exponential method of Matoušek Sharir and Welzl for combinatorial linear programming [81]. A formal connection is presented by Allamigeon, Benchimol, Gaubert, and Joswig [1]: if there is a *combinatorial* pivoting rule for the simplex method yielding polynomial time convergence, then mean payoff games are in P.

We obtain an interesting development considering the following formulation of max-atoms over the real numbers, which we know to be equivalent to the problem over $\mathbb{Z}$.

▶ **Definition 48.** The problem MAX-ATOMS OVER $\mathbb{R}$ is defined as the CSP of $\mathbb{R}$ with all the relations of the form

$$R_c^{\mathrm{MA}\mathbb{R}} = \left\{(x,y,z) \in \mathbb{R}^3 \;\middle|\; \max(x,y) + c \geq z\right\} \quad \text{for } c \in \mathbb{Q}.$$

Obviously the constraints $R_c^{\mathrm{MA}\mathbb{R}}$ are semilinear, closed under maximum, and it is also apparent that these constraints are preserved by all translations $t_k(x) := x + k$ for $k \in \mathbb{Q}$. The class of max-closed translation-invariant semilinear sets has been studied in the context of tropical geometry, where these sets have been called *tropically convex* [33] (in this field, it is customary to consider the dual situation, using min instead of max). Formally, tropically convex sets are the tropical analogue of the convex cones in classical geometry. In this sense, the CSP for tropically convex relations is the notional analogue of LINEAR PROGRAM FEASIBILITY for tropical geometry. One can observe that not all tropically convex relations arise as feasibility regions of instances of MAX-ATOMS over $\mathbb{R}$. An example of such a relation is the relation defined by $x \leq (y+z)/2$, and more generally the relations of the form $\{(x,y,z) \mid (a+b)x \leq ay + bz\}$ for $a, b \neq 0$.

Bodirsky and Mamino [22] give a finite basis for tropically convex semilinear relations.

▶ **Theorem 49** (Bodirsky–Mamino). *A subset of $\mathbb{R}^n$ is semilinear, max-closed, and translation-invariant if and only if it is primitively positively definable over*

$$(\mathbb{R}; x = 1, x = -1, <, 2x \leq y + z\,,\; x \leq y \vee x \leq z)\,.$$

The CSP for tropically convex semilinear relations is equivalent to solving an extension of mean payoff games called *stochastic mean payoff games* [22]. These are defined similarly to the deterministic case, except that the set of vertices is partitioned in three components $V_G = V_G^{\max} \sqcup V_G^{\min} \sqcup V_G^{\mathrm{rand}}$; when the token is on a vertex in the new component V_G^{rand}, the next move is selected uniformly at random. The goal for Max (resp. Min) is to maximize (resp. minimize) the expected value of the play.

▶ **Definition 50.** The problem TROPICALLY CONVEX CONSTRAINTS is

$$\mathrm{CSP}(\mathbb{R}; x = 1, x = -1, <,\; 2x \leq y + z,\; x \leq y \vee x \leq z)\,.$$

▶ **Theorem 51** (Bodirsky–Mamino). *The problem* TROPICALLY CONVEX CONSTRAINTS *is polynomial-time Turing equivalent to solving stochastic mean payoff games, thus in NP $\cap$ coNP.*

Stochastic mean payoff games are a generalization of mean payoff games, and they have the same computational complexity as Condon's *simple stochastic games* [3]. There are many connections between mean payoff games, stochastic mean payoff games, and simple stochastic games: they all admit optimal memoryless strategies [41, 73, 31], they all are in NP $\cap$ coNP, and similar algorithmic approaches are used to solve them [46]. On the other hand, TROPICALLY CONVEX CONSTRAINTS is in some respects a problem more robust than MAX-ATOMS. For instance, TROPICALLY CONVEX CONSTRAINTS is a finite language CSP. On the contrary, MAX-ATOMS does not admit a finite basis, and any finite signature restriction of MAX-ATOMS is immediately in P by sampling.

Max-atoms does not have a polymorphism definition, i.e., the feasibility regions of MAX-ATOMS OVER $\mathbb{R}$ cannot be singled out among all the semilinear sets by means of a polymorphism condition. The class of semilinear relations preserved by all polymorphisms of the template for MAX-ATOMS over $\mathbb{R}$ turns out to be precisely the class of tropically convex semilinear relations [22].

The NP$\cap$coNP membership of TROPICALLY CONVEX CONSTRAINTS can be illustrated, for the special case of closed constraints (i.e., positive Boolean combinations of weak inequalities), by a duality statement.

Let $\mathcal{O}_n$ be the class of functions mapping $\left(\mathbb{Q}\cup\{+\infty\}\right)^n$ to $\mathbb{Q}\cup\{+\infty\}$ of either of the following forms

$$
\begin{aligned}
&(x_1 \dots x_n) \mapsto \max(x_{j_1} + k_1 \dots x_{j_m} + k_m)\\
&(x_1 \dots x_n) \mapsto \min(x_{j_1} + k_1 \dots x_{j_m} + k_m)\\
&(x_1 \dots x_n) \mapsto \frac{\alpha_1 x_{j_1} + \cdots + \alpha_m x_{j_m}}{\alpha_1 + \cdots + \alpha_m} + k
\end{aligned}
$$

where $k, k_i \in \mathbb{Q}$ and $\alpha_i \in \mathbb{Q}^{>0}$.

For any given vector of operators $\bar{o} \in \mathcal{O}_n^n$ consider the following satisfiability problems: the *primal* $P(\bar{o})$ and the *dual* $D(\bar{o})$

$$
P(\bar{o})\colon \begin{cases} \bar{x} \in \mathbb{Q}^n \\ \bar{x} < \bar{o}(\bar{x}) \end{cases} \qquad D(\bar{o})\colon \begin{cases} \bar{y} \in \left(\mathbb{Q} \cup \{+\infty\}\right)^n \setminus \{+\infty\}^n \\ \bar{y} \geq \bar{o}(\bar{y}) \end{cases}
$$

where $<$ and $\geq$ are component-wise.

▶ **Theorem 52** (Bodirsky–Mamino)**.** *For any $\bar{o} \in \mathcal{O}_n^n$ one and only one of the problems $P(\bar{o})$ and $D(\bar{o})$ is satisfiable.*

The special case of Theorem 52 for max-atoms has been observed constraints already in [44].

7 Reducts of the Order of the Rationals

If we want to classify the CSP for first-order reducts Γ of an infinite structure Δ, the simplest structure to start with is the structure Δ with the empty signature. An example of such a reduct Γ is $(\mathbb{Q}; \neq, x = y \Rightarrow u = v)$; its CSP can be solved in polynomial time. On the other hand, the CSP for $(\mathbb{Q}; x = y \neq z \vee x \neq y = z)$ is NP-complete. The complexity of the CSP for such reducts has been classified: a reduct Γ of a countably infinite structure without any structure either has a constant polymorphism or a binary injective polymorphism, and CSP(Γ) is in P, or all polymorphisms of Γ can be written as an injection composed with projections, in which case CSP(Γ) is NP-complete [19]. Instead of giving further details of this fundamental result, we immediately step to a larger class that is of particular relevance for numeric domains, namely to reducts Γ of $(\mathbb{Q}; <)$. This class contains many interesting CSPs (some of which have been mentioned in the introduction), and classifying the complexity of CSP(Γ) is considerably more difficult.

We first state an older result, which provides a pre-classification of reducts of $(\mathbb{Q}; <)$ that plays an important role in the proof of the complexity classification. Let Γ be a first-order reduct of $(\mathbb{Q};<)$ with finite relational signature. Cameron [27] proved that $\mathrm{Aut}(\Gamma)$ equals one out of the following five groups:

- $\mathrm{Aut}(\mathbb{Q};<)$;
- $\mathrm{Aut}(\mathbb{Q};\mathrm{Betw})$ where $\mathrm{Betw} = \{(x,y,z) \in \mathbb{Q}^3 \mid x < y < z \vee z < y < x\}$ is the betweenness relation that we have already seen in the introduction;
- $\mathrm{Aut}(\mathbb{Q};\mathrm{Cycl})$ where $\mathrm{Cycl} = \{(x,y,z) \in \mathbb{Q}^3 \mid x < y < z \vee y < z < x \vee z < y < x\}$ is the so-called *cyclic order relation*;
- $\mathrm{Aut}(\mathbb{Q};\mathrm{Sep})$ where Sep is the relation that contains all $(x,y,u,v) \in \mathbb{Q}^4$ such that the sets $[x,y] \setminus [u,v]$, $[x,y] \cap [u,v]$, and $[u,v] \setminus [x,y]$ are non-empty, where $[p,q]$ denotes the smallest interval that contains p and q;
- $\mathrm{Aut}(\mathbb{Q};=)$, the symmetric group on $\mathbb{Q}$ consisting of *all* permutations of $\mathbb{Q}$.

Classifications of the automorphism groups of reducts Γ are too coarse for obtaining complexity results for $\mathrm{CSP}(\Gamma)$; we need to look at polymorphisms. Here are some binary operations over $\mathbb{Q}$ that are of particular importance when classifying the complexity of $\mathrm{CSP}(\Gamma)$ for reducts Γ of $(\mathbb{Q};<)$. The first one is the maximum operation, $(x,y) \mapsto \max(x,y)$, which we have already seen in the introduction. The following operations are more difficult to describe; for illustrations, see Figure 1.

- Let $\mathrm{mx}\colon \mathbb{Q}^2 \to \mathbb{Q}$ be any operation that satisfies

  $$\begin{aligned}\mathrm{mx}(x,y) < \mathrm{mx}(x',y') \Leftrightarrow\ & \min(x,y) < \min(x',y') \\ & \vee (\min(x,y) = \min(x',y') = x' = y' < \max(x,y)).\end{aligned}$$

  The relations $\leq$ and $\neq$ are not preserved by mx; however, mx preserves for example the relation $X := \{(x,y,z) \in \mathbb{Q}^3 \mid x = y < z \vee x = z < y \vee y = z < x\}$.
- Let $\mathrm{mi}\colon \mathbb{Q}^2 \to \mathbb{Q}$ be any operation that satisfies

  $$\begin{aligned}\mathrm{mi}(x,y) < \mathrm{mi}(x',y') \Leftrightarrow\ & \min(x,y) < \min(x',y') \\ & \vee (\min(x,y) = \min(x',y') = x < x').\end{aligned}$$

  Relations preserved by mi are for instance $\leq$, $\neq$, and the relation of arity four given by $x \neq u \vee y < u \vee z \leq u$; a syntactic description of all the relations with a first-order definition over $(\mathbb{Q};<)$ that are preserved by mi, due to Michał Wrona, can be found in [14].
- Let $\mathrm{ll}\colon \mathbb{Q}^2 \to \mathbb{Q}$ be any operation that satisfies

  $$\begin{aligned}\mathrm{ll}(x,y) < \mathrm{ll}(x',y') \Leftrightarrow\ & \min(x,y) < \min(x',y') \\ & \vee (\min(x,y) = \min(x',y') \wedge x < x') \\ & \vee (\min(x,y) = \min(x',y') = x < y').\end{aligned}$$

  (Here we slightly deviate from the terminology Bodirsky and Kára [20]; the operation ll that we present here has a simpler behaviour that relates more clearly to the operations mx and mi, but the smallest polymorphism clone of a reduct of $(\mathbb{Q};<)$ that contains it is the same as for the operation described in [20].) The relation ll preserves $\neq$, $\leq$, but also the relations defined by $x = y \Rightarrow u = v$ and $x > \min(y,z)$ that we have encountered before.

In all cases, it can be shown that such functions exist.

Each of these operations f has a *dual* f^*, defined by $f^*(x,y) := -f(-x,-y)$; for example, the minimum operation is the dual of the maximum operation. Finally, there is the constant

2	0	1	2
1	0	1	1
0	0	0	0
min	0	1	2

2	0	2	4
0	0	3	2
1	1	0	0
mx	0	1	2

2	2	5	6
1	2	3	4
0	0	1	1
mi	0	1	2

2	4	7	8
1	3	5	6
0	0	1	2
ll	0	1	2

Figure 1 Illustration for the operations min, mx, mi, and ll (in this order).

function whose image has cardinality one; clearly, having a constant polymorphism implies that Γ has the same CSP as a one-element structure, and hence CSP(Γ) is in P. But also if Γ has one of the other operations described above as a polymorphism, or one of their duals, then CSP(Γ) can be solved in polynomial time. We mention that the polynomial-time algorithms can also treat reducts Γ with an infinite signature when the relation symbols are represented by quantifier-free formulas in disjunctive normal form (recall that $(\mathbb{Q}; <)$ has quantifier elimination). We can now state the classification result (from [20]; also see [14]).

▶ **Theorem 53** (Bodirsky–Kára). *Let Γ be a reduct of $(\mathbb{Q}; <)$ with finite relational signature. Then Γ has* ll, min, mx, mi, *one of their duals, or a constant operation as polymorphism, and* CSP(Γ) *is in P, or* CSP(Γ) *is NP-hard.*

It is natural to ask which semilinear relations can be added to the tractable cases of this theorem so that the CSP of the resulting expanded structure is still tractable. The polymorphism max and its algorithmic relevance in this context has already been discussed earlier in Section 6. On the other hand, the operations ll, mx, and mi can be adapted to preserve additional relations that are definable in $(\mathbb{Q}; <, +, 1)$, and we quickly reach CSPs of open computational complexity; some will be listed in Section 10.

8 Reducts of the Successor Relation over the Integers

The structure $(\mathbb{Z}; \mathrm{Succ})$ of the integers with the successor relation is among the simplest structures that is not ω-categorical, and it is a reduct of most of the interesting structures over the integers, such as $(\mathbb{Z}; <)$ or $(\mathbb{Z}; +, 1)$. Hence, the class of reducts of these two more expressive structures includes the reducts of $(\mathbb{Z}; \mathrm{Succ})$, and following the bottom-up approach mentioned in the introduction, we study the CSPs of reducts of $(\mathbb{Z}; \mathrm{Succ})$ first.

Moreover, the structure $(\mathbb{Z}; \mathrm{Succ})$ has the same CSP as the structure $(\mathbb{Q}; x = y + 1)$, and reducts of $(\mathbb{Z}; \mathrm{Succ})$ have the same CSP as the corresponding reducts of $(\mathbb{Q}; x = y + 1)$. Hence, even when we want to classify the complexity of the CSP for reducts of $(\mathbb{Q}; +, 1)$ we have to classify the complexity of CSPs for reducts of $(\mathbb{Z}; \mathrm{Succ})$.

Let Γ be a first-order reduct of $(\mathbb{Z}; \mathrm{Succ})$ with finite relational signature. We want to describe the border between those Γ whose CSP can be solved in polynomial time and those whose CSP is NP-hard, because we believe that the shape of this description could be paradigmatic for complexity classifications of larger classes of CSPs over numeric domains. In order to state the classification, we introduce certain binary operations.

▶ **Definition 54.** For $d \in \mathbb{Z}$, $d \geq 1$, the *d-modular max* is the binary operation $\max_d \colon \mathbb{Z}^2 \to \mathbb{Z}$ defined by

$$\max_d(x, y) := \begin{cases} \max(x, y) & \text{if } x \equiv y \text{ modulo } d \\ x & \text{otherwise.} \end{cases}$$

If Γ has a d-modular max as a polymorphism, for some $d \geq 1$, then CSP(Γ) can be solved in polynomial time [15]. In a nutshell, the idea is that the situation for $d > 1$ can be reduced to the situation for $d = 1$. For $d = 1$, we obtain the regular max operation, and we can solve CSP(Γ) using the sampling technique that has been described in Section 6.

We now describe another family of reducts of $(\mathbb{Z}; \mathrm{Succ})$ whose CSP can be solved in polynomial time. Note that the structure $(\mathbb{Q}; \{(x,y) \mid x = y+1\})$ is isomorphic to $(\mathbb{Q}; \{(x,y) \mid x = y+1\})^2$; let si be any isomorphism. It can be shown that a relation with a first-order definition in $(\mathbb{Q}; \{(x,y) \mid x = y+1\})$ is preserved by i if and only if it has a Horn definition over $(\mathbb{Q}; s)$ where s is the successor *function*. With this description, it is not difficult to come up with an algorithm for reducts of $(\mathbb{Z}; \mathrm{Succ})$ with i as a polymorphism.

We would like to state that all other first-order reducts Γ of $(\mathbb{Z}; \mathrm{Succ})$ with finite relational signature have an NP-hard CSP; an in fact, this is true when Succ has a primitive positive definition in Γ. But otherwise, unfortunately, life is not as simple as that. It might be that Γ has the same CSP as some other reduct Γ′ of $(\mathbb{Z}; \mathrm{Succ})$, and that Γ′ has max as a polymorphism. Allowing such changes in the choice of the template greatly helps in classification projects. When Γ is ω-categorical, then there is a good theory for finding the (up to isomorphism unique) 'nicest' template to work with; we have to refer to [13, 17] for a discussion. The main point is that in these nicer templates, many of the basic relations are primitive positive definable. For reducts Γ of $(\mathbb{Z}; \mathrm{Succ})$, we can replace Γ by some structure Γ′ with the same CSP such that Succ is primitive positive definable in Γ′, unless Γ is of a 'degenerate' form; see Theorem 55 for a formal statement. Unlike the ω-categorical situation, we do not have an a-priori justification for this phenomenon. With this perspective, we can now phrase the complete classification statement from [24].

▶ **Theorem 55.** *Let Γ be a reduct of $(\mathbb{Z}; \mathrm{Succ})$ with finite signature. Then there exists a structure Δ such that CSP(Δ) equals CSP(Γ) and one of the following cases applies.*

1. *Δ has a finite domain, and Feder and Vardi conjectured that CSP(Δ) is in P or NP-complete.*
2. *Δ is a reduct of $(\mathbb{Q}; =)$, and CSP(Δ) is either in P or NP-complete by Theorem 53.*
3. *Δ is a reduct of $(\mathbb{Z}; \mathrm{Succ})$ and Succ is primitive positive definable in Δ. In this case, if Δ has a d-modular* max *or a d-modular* min *polymorphism, then CSP(Δ) is in P; otherwise, CSP(Δ) is NP-complete.*

9 The Unclassifiable

In this section we make essential use of the following theorem, which is due to Davis, Matiyasevich, Putnam, and Robinson.

▶ **Theorem 56** (See e.g. [80]). *A subset of $\mathbb{Z}$ is recursively enumerable if and only if it has a primitive positive definition in $(\mathbb{Z}; *, +, 1)$, the integers with addition and multiplication.*

▶ **Theorem 57.** *For every recursively enumerable problem $\mathcal{P}$ there exists a reduct Γ of $(\mathbb{Z}; *, +, 1)$ with finite relational signature such that CSP(Γ) is polynomial-time Turing equivalent to $\mathcal{P}$.*

Proof. Code $\mathcal{P}$ as a set L of natural numbers, viewing the binary encodings of natural numbers as bit strings. More precisely, $s \in \mathcal{P}$ if and only if the number represented in binary by the string $1s$ is in L. That is, we append the symbol 1 at the front so that for instance $00 \in \mathcal{P}$ and $01 \in \mathcal{P}$ correspond to different numbers in L. Now consider the structure $\Gamma := (\mathbb{Z}; S, D, L', N)$ where

- S is the binary relation defined by

$$S(x,y) \Leftrightarrow \big((y = x + 1 \wedge x \geq 0) \vee (x = y = -1)\big);$$

- D is the binary relation defined by

$$D(x,y) \Leftrightarrow \big((y = 2x \wedge x \geq 0) \vee (x = y = -1)\big);$$

- $L' := L \cup \{-1\}$;
- $N := \{0\}$.

Clearly, if $\mathcal{P}$ is recursively enumerable, then L and L' are recursively enumerable, too.

We have to verify that CSP(Γ) is polynomial time equivalent to $\mathcal{P}$. We first show that there is a polynomial-time reduction from $\mathcal{P}$ to CSP(Γ). View an instance of $\mathcal{P}$ as a number $n \geq 0$ as above, and let $\eta(x)$ be a primitive positive definition for $x = n$ in Γ. It is possible to find such a definition in polynomial time by repeatedly doubling ($y = x + x$) and incrementing ($y = x + 1$) the value 0 (this also follows from the more general Lemma 3). It is clear that n codes a yes-instance of $\mathcal{P}$ if and only if $\exists x(\eta(x) \wedge L'(x))$ is true in Γ.

To reduce CSP(Γ) to $\mathcal{P}$, we present a polynomial-time algorithm for CSP(Γ) that uses an oracle for $\mathcal{P}$ (so our reduction will be a polynomial-time Turing reduction). Let ϕ be an instance of CSP(Γ), and let H be the undirected graph whose vertices are the variables W of ϕ, and which has an edge between x and y if ϕ contains the constraint $S(x,y)$ or the constraint $D(x,y)$. Compute the connected components of H. If a connected component does not contain x with a constraint $N(x)$ in ϕ, then we can set all variables of that component to -1 and satisfy all constraints involving those variables.

Otherwise, suppose that we have a component C that does contain x_0 with a constraint $N(x_0)$. Observe that by connectivity, if there exists a solution, then all variables in C must take non-negative value. Consider the following linear system: for each constraint of the form $S(x,y)$ for $x,y \in C$ we add $y = x + 1$ and $x \geq 0$ to the system, and for each constraint of the form $D(x,y)$ for $x,y \in D$ we add $z = 2x$ and $x \geq 0$. Subject to $x_0 = 0$ this system has either one or no solution. As we have discussed earlier, one can check in polynomial time whether a linear system with 2 variables per constraint has an integer solution, and if there is no solution, the algorithm rejects. Otherwise, the algorithm assigns to each variable $x \in C$ its unique integer value, and if ϕ contains a constraint $L'(x)$, we call the oracle for $\mathcal{P}$ with the binary encoding of this value. If any of those oracle calls has a negative result, reject. Otherwise, we have found an assignment that satisfies all constraints, and accept. ◀

The universal-algebraic approach fails badly when it comes to analyzing the computational complexity of CSP(Γ) for the structure Γ from the proof of Theorem 57: the semi-lattice operation $(x,y) \mapsto \max(x,y)$ preserves Γ for all structures Γ considered in the previous proof, and from that we cannot draw any consequences for the computational complexity of CSP(Γ).

10 Next Steps for . . .

Constraint Satisfaction Problems over a numeric domain, such as $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, for a fixed set of constraints that are first-order definable using addition and multiplication, provide an extremely rich class of natural and fundamental computational problems. A great variety of problems from the literature can be expressed in this way. Many of these problems have an open computational complexity status, and belong to the central topics of research in their respective fields.

The universal-algebraic approach, which has originally been developed for finite-domain constraint satisfaction, will most likely only be of limited help for studying the complexity of computational complexity of the famously open problems. However, this approach is useful for for relating these problems, for obtaining classification results, and for identifying new polynomial-time restrictions.

We state a list of concrete open problems, which we structure according to the numeric domain.

10.1 . . . the Integers

There is the obvious goal: classify the complexity of $\mathrm{CSP}(\Gamma)$ for first-order reducts Γ of $(\mathbb{Z}; <, +, 1)$. This being a very ambitious goal, we propose substeps and concrete relevant questions.

1. Classify the complexity of first-order reducts of $(\mathbb{Z}; \mathrm{Succ}, 0)$.
2. Classify the complexity of first-order reducts of $(\mathbb{Z}; +)$.
3. What is the complexity of $\mathrm{CSP}(\mathbb{Z}; \leq, \mathrm{Succ}, x = 2y)$?

10.2 . . . the Rationals

Again, there is the obvious goal: classify the complexity of $\mathrm{CSP}(\Gamma)$ for first-order reducts Γ of $(\mathbb{Q}; <, +, 1)$. A complete answer would involve the solution of long-standing open problems from the literature, e.g., the complexity of MEAN PAYOFF GAMES.

4. What is the complexity of $\mathrm{CSP}(\Gamma)$ for first-order reducts Γ of $(\mathbb{Q}; <, +, 1)$ with max as a polymorphism? Already containment in $\mathrm{NP} \cap \mathrm{coNP}$ is unclear.
5. What is the complexity of $\mathrm{CSP}(\mathbb{Q}; X, R_{\mathrm{Succ}})$ where $X := \{(x, y, z) \in \mathbb{Q}^3 \mid x = y < z \vee x = z < y \vee y = z < x\}$ is the relation from the introduction?
6. Classify the complexity of $\mathrm{CSP}(\Gamma)$ for first-order reducts Γ of $(\mathbb{Q}; +)$.

10.3 . . . the Reals

Is it possible to classify the complexity of $\mathrm{CSP}(\Gamma)$ for first-order reducts Γ of $(\mathbb{R}; +, *)$? This goal is at least as difficult as the initial goal from Section 10.2, since 1 and $<$ are first-order definable in $(\mathbb{R}; +, *)$, and since every reduct of $(\mathbb{R}; +, <, 1)$ has the same CSP as the corresponding reduct of $(\mathbb{Q}; +, <, 1)$.

7. What is the computational complexity of the problem $\mathrm{CSP}(\mathbb{R}, R_+; R_{=1}, +, x^2 \leq y)$ from Observation 32?
8. What is the computational complexity of the problem $\mathrm{CSP}(\mathbb{R}; R_+, R_{=1}, x^2 + y^2 \leq 1)$ from Observation 33?

The classification for first-order reducts of $(\mathbb{R}; +, *)$ is probably strictly more difficult, because a complete complexity classification might provide the solution to further famous computational problems of open complexity, e.g., the sums-of-square-roots problem or the feasibility problem for semidefinite programs.

10.4 . . . and the Complex Numbers

Also over the complex numbers, many fundamental questions are open. The obvious general question is whether we can classify $\mathrm{CSP}(\Gamma)$ for all first-order reducts of $(\mathbb{C}; +, *)$. Clearly, if Γ is a first-order reduct of $(\mathbb{R}^2; \leq, +, 1, i)$ (with the usual identification of $\mathbb{C}$ and $\mathbb{R}^2$), then $\mathrm{CSP}(\Gamma)$ can be reduced to a linear program feasibility problem. On the other hand, for characterizing the reducts Γ with an NP-hard CSP, it would be interesting to have a primitive positive version of the theorem of Marker and Pillay mentioned in the introduction. More precisely, we are interested in the following mathematical question.

9. Let $R \subseteq \mathbb{C}^n$ be the relation defined over $(\mathbb{C}; +, *)$ by $p(x_1, \dots, x_n) = 0$ for some non-linear polynomial p. Is there a primitive positive definition of complex multiplication in $(\mathbb{C}; +, 1, R)$?

Acknowledgement. The authors would like to thank the referee for the many comments that helped to improve the readability.

References

1 Xavier Allamigeon, Pascal Benchimol, Stéphane Gaubert, and Michael Joswig. Combinatorial simplex algorithms can solve mean payoff games. *SIAM J. Optim.*, 24(4):2096–2117, 2014.

2 Eric Allender, Peter Bürgisser, Johan Kjeldgaard-Pedersen, and Peter Bro Miltersen. On the complexity of numerical analysis. *SIAM J. Comput.*, 38(5):1987–2006, 2008/09. doi:10.1137/070697926.

3 Daniel Andersson and Peter Bro Miltersen. The complexity of solving stochastic games on graphs. In *Algorithms and Computation, 20th International Symposium, ISAAC 2009, Honolulu, Hawaii, USA, December 16-18, 2009. Proceedings*, pages 112–121, 2009.

4 A. Atserias and E. Maneva. Mean-payoff games and the max-atom problem, 2009. URL: https://www.cs.upc.edu/~atserias/papers/mean-payoff-games-max-atom/mpma.pdf.

5 D. Avis and V. Chvátal. Notes on bland's pivoting rule. In *Polyhedral Combinatorics*, pages 24–34. Springer Berlin Heidelberg, 1978.

6 L. Barto. The constraint satisfaction problem and universal algebra. *The Bulletin of Symbolic Logic*, 21(3):319–337, 2015.

7 Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *Journal of the ACM*, 61(1):3:1–3:19, 2014.

8 Libor Barto, Jakub Opršal, and Michael Pinsker. The wonderland of reflections. Preprint arXiv:1510.04521, 2015.

9 Libor Barto and Michael Pinsker. The algebraic dichotomy conjecture for infinite domain constraint satisfaction problems. In *Proceedings of LICS'16*, pages 615–622, 2016. Preprint arXiv:1602.04353.

10 Marc Bezem, Robert Nieuwenhuis, and Enric Rodríguez-Carbonell. The max-atom problem and its relevance. In *Proceedings of the International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, pages 47–61, 2008.

11 Henrik Björklund and Sergei Vorobyov. Combinatorial structure and randomized subexponential algorithms for infinite games. *Theoretical Computer Science*, 349(3):347–360, 2005.

12 Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. *Complexity and real computation.* Springer-Verlag, New York, 1998. With a foreword by Richard M. Karp.

13 Manuel Bodirsky. Cores of countably categorical structures. *Logical Methods in Computer Science*, 3(1):1–16, 2007.

14 Manuel Bodirsky. Complexity classification in infinite-domain constraint satisfaction. Mémoire d'habilitation à diriger des recherches, Université Diderot – Paris 7. Available at arXiv:1201.0856, 2012.

15 Manuel Bodirsky, Víctor Dalmau, Barnaby Martin, Antoine Mottet, and Michael Pinsker. Distance constraint satisfaction problems. *Information and Computation*, 247:87–105, 2016.

16 Manuel Bodirsky and Martin Grohe. Non-dichotomies in constraint satisfaction complexity. In Luca Aceto, Ivan Damgard, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Proceedings of the International Colloquium*

on Automata, Languages and Programming (ICALP), Lecture Notes in Computer Science, pages 184–196. Springer Verlag, July 2008.

17 Manuel Bodirsky, Martin Hils, and Barnaby Martin. On the scope of the universal-algebraic approach to constraint satisfaction. *Logical Methods in Computer Science (LMCS)*, 8(3:13), 2012. An extended abstract that announced some of the results appeared in the proceedings of Logic in Computer Science (LICS'10).

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

19 Manuel Bodirsky and Jan Kára. The complexity of equality constraint languages. *Theory of Computing Systems*, 3(2):136–158, 2008. A conference version appeared in the proceedings of Computer Science Russia (CSR'06).

20 Manuel Bodirsky and Jan Kára. The complexity of temporal constraint satisfaction problems. *Journal of the ACM*, 57(2):1–41, 2009. An extended abstract appeared in the Proceedings of the Symposium on Theory of Computing (STOC).

21 Manuel Bodirsky, Dugald Macpherson, and Johan Thapper. Constraint satisfaction tractability from semi-lattice operations on infinite sets. *Transaction of Computational Logic (ACM-TOCL)*, 14(4):1–30, 2013.

22 Manuel Bodirsky and Marcello Mamino. Max-closed semilinear constraints. In *Proceedings of CSR*, pages 88–101, 2016.

23 Manuel Bodirsky and Antoine Mottet. Reducts of finitely bounded homogeneous structures, and lifting tractability from finite-domain constraint satisfaction. In *Proceedings of LICS'16*, pages 623–632, 2016. Preprint available under ArXiv:1601.04520.

24 Manuel Bodirsky, Antoine Mottet, and Barnaby Martin. Constraint satisfaction problems over the integers with successor. In *Proceedings of ICALP*, pages 256–267, 2015. ArXiv:1503.08572.

25 V. G. Bodnarčuk, L. A. Kalužnin, V. N. Kotov, and B. A. Romov. Galois theory for Post algebras, part I and II. *Cybernetics*, 5:243–539, 1969.

26 Andrei A. Bulatov and Peter Jeavons. Algebraic structures in combinatorial problems. Technical report MATH-AL-4-2001, Technische Universität Dresden, 2001.

27 Peter J. Cameron. Transitivity of permutation groups on unordered sets. *Mathematische Zeitschrift*, 148:127–139, 1976.

28 John Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing (STOC)*, pages 460–467, New York, NY, USA, 1988. ACM.

29 T.-W. J. Chou and G. E. Collins. Algorithms for the solution of systems of linear Diophantine equations. *SIAM Journal on Computing*, 11:687–708, 1982.

30 George E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In *Automata Theory and Formal Languages (2nd GI Conference Kaiserslautern)*, pages 134–183. Springer Berlin, Heidelberg, 1975.

31 Anne Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, 1992. doi:10.1016/0890-5401(92)90048-K.

32 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT Press, 2009. Third edition.

33 Mike Develin and Bernd Sturmfels. Tropical convexity. *Documenta Mathematica*, 9:1–27, 2004.

34 Andrzej Ehrenfeucht and Jan Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8(2):109–113, 1979.

35 Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: a study through Datalog and group theory. *SIAM Journal on Computing*, 28:57–104, 1999.

36 Jeanne Ferrante and Charles Rackoff. A decision procedure for the first order theory of real addition with order. *SIAM Journal on Computing*, 4(1):69–76, 1975.

37 Oliver Friedmann. A subexponential lower bound for zadeh's pivoting rule for solving linear programs and games. In *Integer Programming and Combinatoral Optimization: 15th International Conference (IPCO, New York)*, pages 192–206. Springer Berlin Heidelberg, 2011.

38 Oliver Friedmann, Thomas Dueholm Hansen, and Uri Zwick. Subexponential lower bounds for randomized pivoting rules for the simplex algorithm. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing (STOC)*, pages 283–292, New York, NY, USA, 2011. ACM. doi:10.1145/1993636.1993675.

39 Joseph A. Gallian. *Contemporary Abstract Algebra.* Brooks/Core, 2006.

40 M. R. Garey, R. L. Graham, and D. S. Johnson. Some NP-complete geometric problems. In *Eighth Annual ACM Symposium on Theory of Computing (Hershey, Pa., 1976)*, pages 10–22. Assoc. Comput. Mach., New York, 1976.

41 D. Gillette. Stochastic games with zero probabilities. *Contributions to the Theory of Games*, 3:179–187, 1957.

42 Christian Glaßer, Peter Jonsson, and Barnaby Martin. Circuit satisfiability and constraint satisfaction around skolem arithmetic. In *Pursuit of the Universal – 12th Conference on Computability in Europe, CiE 2016, Paris, France, June 27 – July 1, 2016, Proceedings*, pages 323–332, 2016.

43 Donald Goldfarb and William Y. Sit. Worst case behavior of the steepest edge simplex method. *Discrete Applied Mathematics*, 1(4):277–285, 1979. doi:10.1016/0166-218X(79)90004-0.

44 Dima Grigoriev and Vladimir V. Podolskii. Tropical effective primary and dual Nullstellensätze. In *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, pages 379–391, 2015.

45 V. A. Gurvich, A. V. Karzanov, and L. G. Khachiyan. Cyclic games and finding minimax mean cycles in digraphs. *Zh. Vychisl. Mat. i Mat. Fiz.*, 28(9):1407–1417, 1439, 1988.

46 Nir Halman. Simple stochastic games, parity games, mean payoff games and discounted payoff games are all LP-type problems. *Algorithmica*, 49(1):37–50, 2007. doi:10.1007/s00453-007-0175-3.

47 Thomas Dueholm Hansen and Uri Zwick. An improved version of the random-facet pivoting rule for the simplex algorithm. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 209–218, 2015.

48 J. William Helton and Jiawang Nie. Sufficient and necessary conditions for semidefinite representability of convex hulls and sets. *SIAM Journal on Optimization*, 20(2):759–791, 2009.

49 J. William Helton and Jiawang Nie. Semidefinite representation of convex sets. *Math. Program.*, 122(1):21–64, 2010.

50 J. William Helton and Victor Vinnikov. Linear matrix inequality representation of sets. *Communications on Pure and Applied Mathematics*, 60(5), 2007.

51 Dorit S. Hochbaum and Joseph Naor. Simple and fast algorithms for linear and integer programs with two variables per inequality. *SIAM Journal on Computing*, 23(6):1179–1192, 1994.

52 Wilfrid Hodges. *Model theory.* Cambridge University Press, 1993.

53 Peter Jeavons, David Cohen, and Marc Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, 1997.

54 P. G. Jeavons and M. C. Cooper. Tractable constraints on ordered domains. *Artificial Intelligence*, 79(2):327–339, 1995.

55 R. G. Jeroslow. The simplex algorithm with the pivot rule of maximizing criterion improvement. *Discrete Mathematics*, 4(4):367–377, 1973.

56 J. P. Jones. Universal diophantine equation. *Journal of Symbolic Logic*, 47(3):549–571, 1982.

57 Peter Jonsson and Christer Bäckström. A unifying approach to temporal constraint reasoning. *Artificial Intelligence*, 102(1):143–155, 1998.

58 Peter Jonsson and Tomas Lööw. Computation complexity of linear constraints over the integers. *Artificial Intelligence*, 195:44–62, 2013.

59 Peter Jonsson and Johan Thapper. Constraint satisfaction and semilinear expansions of addition over the rationals and the reals. *Journal of Computer and System Sciences*, 82(5):912–928, 2016. `doi:10.1016/j.jcss.2016.03.002`.

60 Gil Kalai. A subexponential randomized simplex algorithm (extended abstract). In *Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing (STOC)*, pages 475–482, New York, NY, USA, 1992. ACM. `doi:10.1145/129712.129759`.

61 Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comput.*, 8(4):499–507, 1979.

62 N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984. `doi:10.1007/BF02579150`.

63 Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations*, pages 85–103, Boston, MA, 1972. Springer US.

64 L. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademii Nauk SSSR*, 244:1093–1097, 1979.

65 Victor Klee and George J. Minty. How good is the simplex algorithm? In *Inequalities, III (Proc. Third Sympos., Univ. California, Los Angeles, Calif., 1969; dedicated to the memory of Theodore S. Motzkin)*, pages 159–175. Academic Press, New York, 1972.

66 Igor Klep. An exact duality theory for semidefinite programming based on sums of squares. *Mathematics of Operations Research*, 38:569–590, 2013.

67 Pascal Koiran. Hilbert's Nullstellensatz is in the polynomial hierarchy. *J. Complexity*, 12(4):273–286, 1996.

68 Pascal Koiran. Hilbert's Nullstellensatz is in the polynomial hierarchy. Technical Report DIMACS Technical report 96-27, Center for Discrete Mathematics and Theoretical Computer Science, July 1996.

69 M. Koubarakis. Tractable disjunctions of linear constraints: Basic results and applications to temporal reasoning. *Theoretical Computer Science*, 266:311–339, 2001.

70 J. Kratochvil and J. Matousek. Intersection graphs of segments. *Journal of Combinatorial Theory, Series B*, 62(2):289–315, 1994. `doi:10.1006/jctb.1994.1071`.

71 J. C. Lagarias. The computational complexity of simultaneous Diophantine approximation problems. *SIAM Journal on Computing*, 14(1):196–209, 1985.

72 H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

73 Thomas M. Liggett and Steven A. Lippman. Stochastic games with perfect information and time average payoff. *SIAM Review*, 11(4):604–607, 1969.

74 Kenneth L. Manders and Leonard Adleman. NP-complete decision problems for binary quadratics. *Journal of Computer and System Sciences*, 16(2):168–184, 1978.

75 David Marker, Yaacov Peterzil, and Anand Pillay. Additive reducts of real closed fields. *Journal of Symbolic Logic*, 57(1):109–117, 1992.

76 David Marker and Anand Pillay. Reducts of $(C, +, *)$ which contain $+$. *J. Symb. Log.*, 55(3):1243–1251, 1990.

77 Gary A. Martin. Definability in reducts of algebraically closed fields. *Journal of Symbolic Logic*, 53(1):188–199, 1988.

78 Ju. V. Matijasevic. Some purely mathematical results inspired by mathematical logic. In *Foundations of Mathematics and Computability Theory*, pages 121–127. Reidel, Dordrecht, 1977.

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

80 Yuri V. Matiyasevich. *Hilbert's Tenth Problem.* MIT Press, Cambridge, Massachusetts, 1993.

81 J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4):498–516, 1996. doi:10.1007/BF01940877.

82 Nimrod Megiddo. Towards a genuinely polynomial algorithm for linear programming. *SIAM Journal on Computing*, 12(2):347–353, 1983.

83 Nimrod Megiddo. Linear programming in linear time when the dimension is fixed. *J. ACM*, 31(1):114–127, January 1984. doi:10.1145/2422.322418.

84 Rolf H. Möhring, Martin Skutella, and Frederik Stork. Scheduling with and/or precedence constraints. *SIAM Journal on Computing*, 33(2):393–415, 2004.

85 Bernhard Nebel and Hans-Jürgen Bürckert. Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. *Journal of the ACM*, 42(1):43–66, 1995.

86 Ya'acov Peterzil. Reducts of some structures over the reals. *J. Symb. Log.*, 58(3):955–966, 1993.

87 Motakuri V. Ramana. An exact duality theory for semidefinite programming and its complexity implications. *Mathematical Programming*, 77:129–162, 1997.

88 James Renegar. On the computational complexity and geometry of the first-order theory of the reals. Part I: Introduction. Preliminaries. The geometry of semi-algebraic sets. The decision problem for the existential theory of the reals. *Journal of Symbolic Computation*, 13(3):255–299, 1992. doi:10.1016/S0747-7171(10)80003-3.

89 James Renegar. On the computational complexity and geometry of the first-order theory of the reals. Part II: The general decision problem. Preliminaries for quantifier elimination. *Journal of Symbolic Computation*, 13(3):301–327, 1992. doi:10.1016/S0747-7171(10)80004-5.

90 James Renegar. On the computational complexity and geometry of the first-order theory of the reals. Part III: Quantifier elimination. *Journal of Symbolic Computation*, 13(3):329–352, 1992. doi:10.1016/S0747-7171(10)80005-7.

91 Marcus Schaefer. Realizability of graphs and linkages. In *Thirty Essays on Geometric Graph Theory*, pages 461–482. Springer New York, 2013.

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

93 Claus Scheiderer. Semidefinite representation for convex hulls of real algebraic curves, 2012. Preprint, arXiv:1208.3865.

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

95 Alexandra Shlapentokh. *Hilbert's tenth problem: Diophantine classes and extensions to global fields*, volume 7 of *New Mathematical Monographs.* Cambridge University Press, 2007.

96 Sergey P. Tarasov and Mikhail N. Vyalyi. Semidefinite programming and arithmetic circuit evaluation. *Discrete Applied Mathematics*, 156(11):2070–2078, 2008. In Memory of Leonid Khachiyan (1952–2005).

97 Éva Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, 34(2):250–256, 1986. URL: `http://www.jstor.org/stable/170819`.

98 Alfred Tarski. A decision method for elementary algebra and geometry: Prepared for publication with the assistance of J. C. C. McKinsey. Technical report, RAND Corporation, Santa Monica, CA, 1951.

99 Lou van den Dries. *Tame topology and o-minimal structures*, volume 248 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1998. `doi:10.1017/CBO9780511525919`.

100 Peter van Emde Boas. Another NP-complete partition problem and the complexity of computing short vectors in a lattice. Technical Report Technical Report 81-04, Mathematische Instituut, University of Amsterdam, 1981.

101 Marc Vilain, Henry Kautz, and Peter van Beek. Constraint propagation algorithms for temporal reasoning: A revised report. *Reading in Qualitative Reasoning about Physical Systems*, pages 373–381, 1989.

102 Sergei Vorobyov. Cyclic games and linear programming. *Discrete Applied Mathematics*, 156(11):2195–2231, 2008. In Memory of Leonid Khachiyan (1952–2005). `doi:10.1016/j.dam.2008.04.012`.

103 Margaret H. Wright. The interior-point revolution in optimization: history, recent developments, and lasting consequences. *Bull. Amer. Math. Soc.*, 42:39–56, 2005.

104 Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theor. Comput. Sci.*, 158(1&2):343–359, 1996.