Quantified Constraints in Twenty Seventeen

Barnaby Martin

School of Engineering and Computing Sciences, University of Durham, UK
barnaby.d.martin@durham.ac.uk

Abstract

I present a survey of recent advances in the algorithmic and computational complexity theory of non-Boolean Quantified Constraint Satisfaction Problems, incorporating some more modern research directions.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Quantified constraints, Constraint satisfaction problems, Computational complexity, Parameterized complexity, Universal algebra

Digital Object Identifier 10.4230/DFU.Vol7.15301.327

1 Introduction

The *Quantified Constraint Satisfaction Problem* (QCSP) might be thought of as the dissolute younger brother of its better-studied restriction, the *Constraint Satisfaction Problem* (CSP). The CSP has been called *Königsproblem*[1] as it sits at the interface between Combinatorics, Logic and Universal Algebra. The QCSP is a logical generalisation of the CSP whose combinatorial definition is ugly. Similarly, although the algebraic theory of the QCSP is useful, its algebraic objects are a bit unwieldy since surjective operations are not closed under composition. CSPs are ubiquitous in Computer Science, especially when one considers them in their infinite-domain generality, while QCSPs can not nearly claim to be so important in applications. This is no doubt due to modelling difficulties induced by the universal quantifier in the absence of disjunction, together with the lack of ability to relativise (guard) the universal quantifiers. The variant of QCSP in which both quantifiers may be relativised (for the CSP this is called the *list* or *conservative* case) has in fact been fully classified [7]. The only remaining QCSPs for which such relativisation is not desirable are the Boolean QCSPs. These are long since classified and Quantified Satisfiability itself, better-known as QBF (Quantified Boolean Formulas) – which is indeed useful – may be considered its own research area, and is therefore left out of the scope of this survey (see [75]).

In my papers on QCSP, I have often cited its use in planning [93] and modelling non-monotonic reasoning [56], but inspection of both these papers reveals this is just the Boolean case of QBF. There are various claims for the usefulness of non-Boolean QCSPs but specific explanations are sparse and even the examples given for QCSPs often involve more than just conjunction in the quantifier-free part (Example 1 in [89] is of this form). Thus, what is left of the true non-Boolean QCSP is a problem I believe to be mostly of interest to theorists, especially those who are not afraid of getting their hands dirty. For this is the lot of the researcher into complexity of QCSPs! However, in this quagmire there is still beauty to be found and interesting structured classifications are known often mixing curiously techniques combinatorial and algebraic. Indeed, the complexity researcher can even draw succour from

[1] King of problems, rather than problem of kings.

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. 327–346
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

the spurious claim that QCSPs are more important than CSPs since a classification for the former embeds that of the latter.

1.1 Previous Surveys

I know of two previous articles on general QCSPs that might be considered as surveys, both written by Hubie Chen [32, 36]. Neither is marketed as as a survey, [32] serves as an introduction to the topic and [36] is a more reflective piece from the viewpoint of the author. In this survey, I will try to be more comprehensive, within the scope, and at least in regard of recent work. However, this article is not intended to be introductory, and will not even contain definitions for all the concepts introduced.

After a section of background, this survey will have three principal sections, corresponding to what I see as the three largest research themes into non-Boolean QCSPs in the past decade. In Section 4, I will survey the state-of-the-art in classical complexity classifications for QCSP while in Section 5, I will discuss the recent work done in the area of their parameterized complexities. In Section 6, I will look at new algorithms for QCSPs and proof theories for their evaluation.

2 Preliminaries

In this article we tend toward the logical definitions for constraint satisfaction. A *constraint language* may therefore be seen as a relational first-order structure (possibly with constants). Relational structures are denoted $\mathcal{B}$ with domain B of cardinality $|B|$. Infinity might appear in two guises, either in the domain size or number of relations in the signature. When there is an infinite number of relations there is an issue as to how they are encoded, so we prefer to avoid this within computational problems. When we describe a constraint language as finite, we mean both in the domain size and the number of relations. We sometimes talk of a constraint language or structure (or CSP) *with constants* where we assume that all elements of the domain are named by their own constant.

The logic associated with CSP, the fragment of first-order containing just $\exists$, $\wedge$ and $=$, is usually known as *primitive positive* (pp) logic. The generalisation to QCSP involves the restoration of $\forall$ to primitive positive logic and appears in the literature, like the devil, under a myriad of names. In older mathematical logic texts it is known as *positive Horn* [74] while more modern works term it *quantified constraint formulas* [33], *quantified conjunctive-positive* [41], *quantified conjunctive* [20, 37], *conjunctive positive* [28] and even *few* [32] (this last is from *forall-exists-wedge*). The foundational [18] even leaves the logic unnamed. Since *positive conjunctive* is not among these names, it is the designation this article will use.

For a constraint language $\mathcal{B}$, the problem QCSP($\mathcal{B}$) takes as input a sentence ϕ of positive conjunctive logic and returns a yes-instance precisely when ϕ is true on $\mathcal{B}$, denoted $\mathcal{B} \models \phi$. The problem CSP($\mathcal{B}$) is defined similarly but for primitive positive logic. We sometimes also refer to the *(Q)CSP of* $\mathcal{B}$ as a disingenuous shorthand for (Q)CSP($\mathcal{B}$).

Note that, in fullest generality, the (so-called *uniform*) QCSP takes as input a pair $(\phi, \mathcal{B})$, where we may imagine finite $\mathcal{B}$ to be given by listing domain and tuples in relations, and asks whether $\mathcal{B} \models \phi$. The (non-uniform) problems QCSP($\mathcal{B}$) are examples of *right-hand* restrictions of QCSP, where *left-hand* restrictions involve limiting the form of the positive conjunctive sentence that may be input.

A quantifier block (sequence of quantified variables) is in Π_{2k} when it begins with universally quantified variables and alternates between the two quantifiers no more than

$2k-1$ times thereafter. Π_{2k}-CSP($\mathcal{B}$) is the restriction of QCSP($\mathcal{B}$) in which the positive conjunctive input is restricted to be prenex with quantifier block Π_{2k}.

A *homomorphism* from a structure $\mathcal{A}$ to a structure $\mathcal{B}$ in the same signature is a function f from A to B such that for each k-ary relation R, if $R(x_1,\ldots,x_k)$ holds in $\mathcal{A}$, then $R(f(x_1),\ldots,f(x_k))$ holds in $\mathcal{B}$. An *endomorphism* of $\mathcal{B}$ is a homomorphism from $\mathcal{B}$ to itself. A structure is a *core* when all of its endomorphisms are automorphisms. The CSP is well-known to be equivalent to the homomorphism problem, that given finite input $\mathcal{A}$ and $\mathcal{B}$, asks whether there is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$? In this guise, it is apparent whence the names left- and right-hand restrictions, seen in the previous paragraph, arose. Let us dwell a little on what makes the CSP equivalent to the homomorphism problem, i.e. how we translate between primitive positive sentences and relational structures. This is through the juxtaposition of *canonical query* and *canonical database* [77], where one turns a prenex primitive positive sentence to a relational structure by mapping variables v_i to elements v_i and positive atoms $R(v_1,\ldots,v_k)$ in the conjunction to tuples $(v_1,\ldots,v_k)$ in a relation R.

An *algebra* is a first-order functional structure. Algebras are denoted $\mathbb{A}$ with domain A. If R is an m-ary relation over B and f is a k-ary operation on B, then we say f *preserves* R if, for any $(x_1^1,\ldots,x_1^m),\ldots,(x_k^1,\ldots,x_k^m)\in R$ we have also $(f(x_1^1,\ldots,x_k^1),\ldots,f(x_1^m,\ldots,x_k^m))\in R$. When f preserves R we also say that R is *invariant* under f and f is a *polymorphism* of R. A constraint language $\mathcal{B}$ is preserved by f if all of its relations are. Note that the unary polymorphisms of $\mathcal{B}$ are precisely its endomorphisms (whence the term polymorphism). Let Pol($\mathcal{B}$) be the set of polymorphisms of $\mathcal{B}$ and let Inv($\mathbb{A}$) be the set of relations on A which are invariant under (each of) the operations of $\mathbb{A}$. Pol($\mathcal{B}$) is an object known in Universal Algebra as a *clone*, which is a set of operations containing all projections and closed under composition (superposition). I will conflate sets of operations over the same domain and algebras just as I do sets of relations over the same domain and constraint languages (relational structures). Indeed, the only technical difference between such objects is the movement away from an ordered signature, which is not something we will ever need. Let $\langle\mathbb{A}\rangle$ be the clone generated by (the set of operations) of $\mathbb{A}$ and let s($\langle\mathbb{A}\rangle$) be this clone restricted to its surjective operations. Finally, let sPol($\mathcal{B}$) be the set of surjective polymorphisms of $\mathcal{B}$ and let $\langle\mathcal{B}\rangle_{\mathrm{pp}}$ and $\langle\mathcal{B}\rangle_{\mathrm{pc}}$ be the class of relations definable on $\mathcal{B}$ in primitive positive and positive conjunctive logic, respectively.

The algebraic approach to CSP and QCSP is based on the following observations, each producing a *Galois Correspondence*. Let $\mathcal{B}$ be a constraint language and $\mathbb{A}$ an algebra, over the same finite domain. Then,

$$\begin{array}{llcll} \mathrm{Inv}(\mathrm{Pol}(\mathcal{B}))=\langle\mathcal{B}\rangle_{\mathrm{pp}} & [63, 14] & \& & \mathrm{Inv}(\mathrm{sPol}(\mathcal{B}))=\langle\mathcal{B}\rangle_{\mathrm{pc}} & [18] \\ \mathrm{Pol}(\mathrm{Inv}(\mathbb{A}))=\langle\mathbb{A}\rangle & [92] & \& & \mathrm{sPol}(\mathrm{Inv}(\mathbb{A}))=\mathrm{s}(\langle\mathbb{A}\rangle). & \end{array}$$

We consider a Galois Correspondence to be an order-inverting isomorphism between two lattices. This is exactly in line with the original correspondence of Evariste Galois, except that we will allow our lattices to be infinite. The order-inverting isomorphisms are given by Inv and Pol, for the CSP above left, between clones and sets of relations closed under primitive positive definability. For the QCSP above right, they are given by Inv and sPol, between the surjective reducts of clones and sets of relations closed under positive conjunctive definability (the second part of this correspondence, appearing on the bottom row above, is seldom noted in the literature). It is stated in [33] that a careful reading of the proof from [18] shows that conjunctive positive definability in $\langle\mathcal{B}\rangle_{\mathrm{pc}}$ may even be replaced with its Π_2 fragment. Thus, on finite structures, positive conjunctive logic collapses to its Π_2 fragment.

The consequence of the Galois Correspondence is that whenever $\mathrm{Pol}(\mathcal{B})\subseteq\mathrm{Pol}(\mathcal{B}')$ there is a logspace reduction from CSP($\mathcal{B}'$) to CSP($\mathcal{B}$). Note that $\mathcal{B}$ and $\mathcal{B}'$ must share the same domain. Similarly, when $\mathrm{sPol}(\mathcal{B}) \subseteq \mathrm{sPol}(\mathcal{B}')$ there is a logspace reduction from $\mathrm{QCSP}(\mathcal{B}')$ to $\mathrm{QCSP}(\mathcal{B})$. Thus, the *(surjective) polymorphisms control the complexity* of these problems.

I assume a basic familiarity with the modern theory of Computational Complexity. For more details on complexity classes, I refer the reader to [91] for (classical) Complexity Theory and [54] for Parameterized Complexity. Further, I will carelessly use *tractable* as a synonym for polynomially solvable. For finite $\mathcal{B}$, we note that the problems $\mathrm{CSP}(\mathcal{B})$, Π_{2k}-$\mathrm{CSP}(\mathcal{B})$ and $\mathrm{QCSP}(\mathcal{B})$ are in the complexity classes NP, Π_{2k}^{P} and Pspace, respectively.

2.1 More Algebra

Let $[n] := \{1, \ldots, n\}$. An operation f is called *idempotent* if, for each x, $f(x, \ldots, x) = x$. It is a *majority* if it is ternary and satisfies $f(x, x, y) = f(x, y, x) = f(y, x, x) = x$. It is *Mal'tsev* if it is ternary and satisfies $f(x, x, y) = f(y, x, x) = y$. On a totally ordered domain, the binary operations *min* and *max* return the minimum and maximum of their two arguments, respectively. A *semilattice* operation is a binary operation that is associative, commutative and idempotent. Both *min* and *max* are semilattice operations. For 2-*semilattice*, one relaxes associativity to the weaker condition $f(f(x, x), y) = f(x, f(x, y))$. A *unit* for a semilattice operation f is an element i so that $f(x, i) = x$ for all x. A k-ary operation f is called a *set operation* if $f(x_1, \ldots, x_k) = f(y_1, \ldots, y_k)$ whenever $\{x_1, \ldots, x_k\} = \{y_1, \ldots, y_k\}$. An algebra is called *idempotent trivial* if all of its idempotent operations are projections. A constraint language is called idempotent trivial if its polymorphism clone is.

Majority operations can be generalised to k-ary *near-unanimity* operations, which satisfy $f(x, \ldots, x, y) = f(x, \ldots, y, x) = \ldots = f(y, \ldots, x, x) = x$. This definition can now be relaxed for *weak near-unanimity* operations which are idempotent operations that only satisfy $f(x, \ldots, x, y) = f(x, \ldots, y, x) = \ldots = f(y, \ldots, x, x)$. Finally, a k-ary operation t is *Taylor* if it satisfies a system of identities $t(x_i^1, \ldots, t_i^k) = t(y_i^1, \ldots, y_i^k)$, for $i \in [k]$, in the variables x and y, where $x_i^i = x$ and $y_i^i = y$. One can see that weak near-unanimities are examples of Taylor operations.

For a finite-domain algebra $\mathbb{A}$ we associate a function $f_{\mathbb{A}} : \mathbb{N} \to \mathbb{N}$, giving the cardinality of the minimal generating sets of the sequence $\mathbb{A}, \mathbb{A}^2, \mathbb{A}^3, \ldots$ as $f(1), f(2), f(3), \ldots$, respectively. We may say $\mathbb{A}$ has the g-GP if $f(m) \leq g(m)$ for all m. The question then arises as to the growth rate of f and specifically regarding the behaviours constant, logarithmic, linear, polynomial and exponential. Wiegold proved in [98] that if $\mathbb{A}$ is a finite semigroup then $f_{\mathbb{A}}$ is either linear or exponential, with the former prevailing precisely when $\mathbb{A}$ is a monoid. This dichotomy classification may be seen as a gap theorem because no growth rates intermediate between linear and exponential may occur. We say $\mathbb{A}$ enjoys the *polynomially generated powers* property (PGP) if there exists a polynomial p so that $f_{\mathbb{A}} = O(p)$ and the *exponentially generated powers* property (EGP) if there exists a constant b so that $f_{\mathbb{A}} = \Omega(g)$ where $g(i) = b^i$.

For a finite-domain, idempotent algebra $\mathbb{A}$, k-*collapsibility* may be seen as a special form of the PGP in which the generating set for $\mathbb{A}^m$ is constituted of all tuples $(x_1, \ldots, x_m)$ in which at least $m - k$ of these elements are equal. k-*switchability* may be seen as another special form of the PGP in which the generating set for $\mathbb{A}^m$ is constituted of all tuples $(x_1, \ldots, x_m)$ in which there exists $a_i < \ldots < a_{k'}$, for $k' \leq k$, so that

$$(x_1, \ldots, x_m) = (x_1, \ldots, x_{a_1}, x_{a_1+1}, \ldots, x_{a_2}, x_{a_2+1}, \ldots, \ldots, x_{a'_k}, x_{a'_k+1}, \ldots, x_m),$$

where $x_1 = \ldots = x_{a_1-1}$, $x_{a_1} = \ldots = x_{a_2-1}$, $\ldots$, $x_{a_{k'}} = \ldots = x_{a_m}$. Thus, $a_1, a_2, \ldots, a_{k'}$ are the indices where the tuple switches value. Note that these are not the original definitions [33, 35] but they are provably equivalent [27]. We say that $\mathbb{A}$ is collapsible (switchable) if there exists k such that it is k-collapsible (k-switchable). For any finite algebra, k-collapsibility implies k-switchability and for any 2-element algebra, k-switchability implies k-collapsibility. Chen originally introduced switchability because he found a 3-element algebra that enjoyed the PGP but was not collapsible [35].

An alternative algebraic formulation of the CSP and QCSP has the form, for the latter, of QCSP($\mathbb{A}$) for some algebra $\mathbb{A}$. One might imagine a restriction here to surjective operations but it seems tractability may be found beyond this. The input to this problem is of the form $(\phi, \mathcal{B})$ where $\mathcal{B}$ is a finite constraint language which is invariant under $\mathbb{A}$. Note that $\mathcal{B}$ and $\mathbb{A}$ share the same domain and for fixed, finite $\mathbb{A}$, the problem to determine if input $\mathcal{B}$ is invariant under $\mathbb{A}$ is in polynomial time.

2.2 Various Digraphs

A *digraph* is a structure with a single binary relation E; if this is symmetric then it is further a *graph.* A *(self-)loop* on a vertex x is an instance of $(x, x) \in E$. A graph without self-loops is termed *irreflexive* and a graph with loops on every vertex is termed *reflexive*. Sometimes we term a graph *partially reflexive* to emphasise we are somewhere inbetween.

A *clique* is an irreflexive graph where all distinct vertices are adjacent. A *k-partite* graph is one whose vertices may be partitioned into k classes where there are no edges between vertices in the same class. If all other edges are present one refers to a *complete* k-partite (or *multipartite*) graph. A graph is a *tree* if it is connected and contains no cycles, and a *forest* is the disjoint union of trees. A *pseudotree* is a connected graph with at most one cycle, and a *pseudoforest* is the disjoint union of pseudotrees.

A *semicomplete* digraph is an irreflexive graph so that for distinct x, y at least one of $(x, y), (y, x) \in E$. A *tournament* further satisfies that precisely one of $(x, y), (y, x) \in E$. A *local tournament* satisfies, for every x and distinct y and z, such that either both $(x, y), (x, z) \in E$ or both $(y, x), (z, x) \in E$, that there be precisely one of the edges (y, z) or (z, y) in E. A *source* (respectively, *sink*) in a digraph is a vertex with in-degree (respectively, out-degree) zero. A digraph is *smooth* if it has neither source nor sink.

2.3 The Modern Study of CSPs

The foundational paper for studying the complexity of CSPs came in 1978 from Thomas Schaefer [95] in which he proved a P versus NP-complete dichotomy for Boolean CSPs. The classification has six tractable classes which we will give below alongside their associated polymorphism. The characterisation with polymorphisms appeared first in [70] which is the foundational paper for the algebraic approach to CSPs.

0-valid (constant 0)	Horn (min)	bijunctive (majority)
1-valid (constant 1)	dual Horn (max)	affine (Mal'tsev)

Note that there are unique operations on the Boolean domain that are majority and Mal'tsev, respectively. Obviously, this applies also to min and max once the order $0 < 1$ is assumed. The outstanding conjecture in the area of finite-domain CSPs was later formulated by Feder and Vardi in [58] and is that these are all either in P or are NP-complete, which is surprising given these CSPs appear to form a large microcosm of NP, and NP itself is thought unlikely to have this dichotomy property since the work of [79]. It seems Feder and Vardi tried very hard to reproduce an argument à la Ladner, in a logic that can express all finite-domain CSPs, yet failed. The original Feder-Vardi conjecture did not specify where the boundary between P and NP-complete should be, but this has now been concretely conjectured in the algebraic language [21]. This conjecture remains unsettled, although dichotomy is now known on substantial classes, e.g. structures with domains of size ≤ 3 [95, 22] and smooth digraphs [68, 5].[2]

The conjectured complexity delineation in [21] was that a finite-domain constraint language $\mathcal{B}$ that is expanded with all constants should be so that its CSP is NP-complete precisely if $\mathrm{Pol}(\mathcal{B})$ has a G-set as a factor. Remember that all polymorphisms of a constraint language expanded with its constants are idempotent. We will not define what it is for a G-set to be a factor since there are various more modern specifications that are more user-friendly for our purposes. For example, the condition for tractability is equivalent to possessing a Taylor operation [81] or a weak near-unanimity operation [84]. We note that the backward direction to this conjecture is known to be true [21].

3 Background

Schaefer announced a dichotomy theorem for Boolean QCSP in the same paper as he proved the dichotomy for Boolean CSP [95]. The proof was omitted and the result in any case could ostensibly only apply to the situation with constants (0 and 1) allowed in instances. The resolution of the Pspace-hard cases in the case where constants do not appear was finally given much later, independently in [47] and [48]. Schaefer was also quite vague about how to extend the polynomial algorithms for subclasses of Boolean CSP to the same subclasses of Boolean QCSP, and articles fleshing out these algorithms continued for a number of years: [72] (Horn and dual-Horn cases), [1] (bijunctive case).

The situation for non-Boolean QCSPs seems to have been taken up largely only in the new millennium with two communities working on the problem, one more applied, with a background in Constraint Programming [16, 15, 64, 60] and one more theoretical with a background in Algorithms and Complexity (and often Universal Algebra) [18, 33]. Both communities are united in their attempts to take established algorithmic methods for CSPs, such as *local consistency* and *linear equations* and adapt them for QCSPs. I belong to the theoretical community, and the main thrust of this survey will be in this direction.

Following Schaefer, and in line with the like program for CSPs, the bulk of the research into complexity of QCSPs in the theoretical community has been in the non-uniform, right-hand framework in which one parameterises the problem by the constraint language. We already noted that such a complexity taxonomy for QCSPs embeds the similar one for CSPs. This is because $\mathrm{CSP}(\mathcal{B})$ and $\mathrm{QCSP}(\mathcal{B} \uplus 1)$ are polynomially equivalent for all $\mathcal{B}$, where $\mathcal{B} \uplus 1$ denotes the disjoint union of $\mathcal{B}$ with an isolated element. The algebraic approach to QCSP was pioneered in [17] whose expanded journal version (also with another author) appeared much later as [18]. This early work gave the Inv-sPol Galois Correspondence already mentioned and provided uniform explanations of QCSP tractability for classes of problems based on the presence of certain surjective (even idempotent) polymorphisms. This fruitful approach was continued in [33] where the key idea of *collapsibility* was introduced. Collapsibility was originally introduced as a (relational) property of constraint languages but was later explored as a property of idempotent algebras [33]. When a constraint language $\mathcal{B}$, expanded with all constants, is collapsible then the evaluations of instances ϕ of $\mathrm{QCSP}(\mathcal{B})$ may be reduced to a polynomial system of instances of $\mathrm{CSP}(\mathcal{B})$, and so the maximal complexity of the problem is

[2] Petar Markovic announced a proof of dichotomy for 4-element domains in 2011. The argument has since passed several years in confinement for refinement. It is in preparation as this goes to press. Very recently, Dmitriy Zhuk has announced a proof for the 5-element case (at AAA 91, February 2016, Brno).

reduced to NP [33]. If, further CSP($\mathcal{B}$) is in P, then this yields polynomial solvability for the QCSP.

Another manner of extension of ideas from the CSP to the QCSP arises in [39] where the notion of *establishing strong k-consistency*[3] is generalised for the QCSP. Establishing strong k-consistency is a well-known procedure for solving various CSPs, dating back to at least [53, 97] and is now known to be an algorithm for an even larger class of problems since its relationship with a certain pebble game was uncovered in [78]. It will be convenient to assume that our constraint language is closed under projections of its relations and further that our input ϕ is closed under projections of its aliquot atoms (thus, for example, if it contains an atom $R(v_1, \ldots, v_k)$ then it also contains atoms corresponding to each $\exists v_i R(v_1, \ldots, v_k)$). We say that ϕ is *k-consistent* (with respect to $\mathcal{B}$) if, for every assignment α of $k-1$ variables $x_1, \ldots, x_{k-1}$ from ϕ to elements from B that satisfy the conjuncts of ϕ that involve no other variables than $x_1, \ldots, x_{k-1}$, and for every variable x_k from ϕ, the assignment α can be extended to x_k such that all conjuncts of ϕ that involve no other variables than $x_1, \ldots, x_k$ are satisfied on $\mathcal{B}$ by the extension of α. We say that ϕ is *strongly k-consistent* if ϕ is j-consistent for all j with $2 \leq j \leq k$. We would say that ϕ is *globally consistent* if ϕ is k-consistent for all $k > 0$.

When strong k-consistency implies global consistency, then we have that establishing strong k-consistency will be an algorithm for the CSP. Further, establishing strong-k-consistency is an algorithm for the CSP for a large class of constraint languages including those preserved by near-unanimity [71] and set [50] polymorphisms.

In [39], a suitable pebble game is found for the variant of establishing k-consistency associated with the QCSP and this gives positive algorithms for left-hand restrictions where a generalisation of treewidth comes into play (we will return to this line of enquiry later). Returning to the land of dexterity, establishing k-consistency is shown to be a polynomial algorithm for QCSP($\mathcal{B}$) when $\mathcal{B}$ is preserved by a near-unanimity operation. Something stronger than establishing [strong] k-consistency is termed establishing *default* k-consistency [39] and this is shown to give a polynomial algorithm for QCSP($\mathcal{B}$) when $\mathcal{B}$ is preserved by an idempotent set operation f that has an additional property termed *unique minimal coherent set* with respect to f [39, 31]. As it happens, both of these classes of polymorphism bestow collapsibility and so the polynomial algorithm is implied from [33], together with the corresponding works on the CSP [50]. However, the direct algorithm explained through the pebble game gives a simple and unifying description of these tractabilities.

3.1 Constants and Idempotency

The complexity classification problem for finite-domain CSPs is greatly facilitated by the fact that one may assume that all constants are primitive positive definable, up to isomorphism. This is due to the key notion of cores (recall these are structures for which all endomorphisms are automorphisms). Every finite-domain constraint language is homomorphically equivalent to a constraint language that is a core and in this core all constants are primitive positive definable up to isomorphism. In the algebraic language this corresponds to the assumption, without loss of generality, that all polymorphisms are idempotent. A robust notion of core for positive conjunctive logic and QCSP is not exactly known. The putative notion Q-core of [42] has been successfully deployed in complexity classifications [85, 83], as we shall see later; however, many of its properties (even "uniqueness") are not yet clear. Thus, it is not known

[3] The authors of [39] call this establishing k-consistency but I align with, inter alia, [78].

if the QCSP algebraic classification may be reduced, from sets of surjective operations, to idempotent clones. Certainly, a constraint language, agreeing on all positive conjunctive sentences, with constants up to isomorphism positive conjunctive definable, is not always possible [42]. As a result of this, a certain amount of the literature on QCSP classifications is based on the idempotent assumption and consequently only deals with constraint languages already expanded with constants. This is true for many of the papers in which algebra plays a central role, for example, [33, 27]. Even in the foundational [18], all of the polymorphisms giving tractability are idempotent (if the idempotent reduct of sPol($\mathcal{B}$) is tractable for QCSP then clearly so is sPol($\mathcal{B}$) itself). Thus, although the results of [18] apply where sPol($\mathcal{B}$) need not contain only idempotent operations, the work itself may be said to focus on the idempotent. Several works involving combinatorial results operate in the wilderness outside of idempotency [86, 85, 83] but the only totally algebraic paper in this *terra incognita* (sic!) is [44]. In this last work, the object of study is QCSP($\mathbb{A}$) where $\mathbb{A}$ is a monoid. It follows from collapsibility that this problem is always in NP and the authors give a P versus NP-complete dichotomy with the condition for tractability being that $\mathbb{A}$ is a *block group* that is generated by its *regular* elements. The condition for tractability of CSP($\mathbb{A}$), even in the more general situation in which $\mathbb{A}$ is a semigroup, is just $\mathbb{A}$ being a block group [23]. The really interesting thing about the QCSP classification here is an explanation of tractability through a binary monoid operation f that may not be idempotent. Curiously, the tractability comes from a term operation generated by f that may not even be surjective (though f itself is). However, when this term operation is restricted to the so-called *idempotents* of $\mathbb{A}$ it is surjective (indeed, idempotent).

4 Classical Complexity

The modern sport for complexity classifications for QCSP was apparent already in [18]. In this paper the new algebraic methods were leveraged to obtain the first trichotomy for QCSP. A binary relation is a *graph of a permutation* if it is the set of pairs $\{(i, \pi(i)) : i \in [n]\}$ for some permutation π. Let $\mathcal{B}$ be a constraint language on $[n]$ that contains all ($n!$) graphs of permutations. Then QCSP($\mathcal{B}$) is either in P, is NP-complete or is Pspace-complete (Theorem 7.4 in [18] deals with $n \geq 3$, the remainder come from Schaefer [95]). The polynomial cases possess either a majority or Mal'tsev polymorphism. For NP membership of the NP-complete cases, an early form of "collapsibility" appears as Proposition 7.1. Then, for the remaining cases of NP-hardness, the classification for CSP from [49] is cited.

The next QCSP classification seems to have come in [30]. 2-semilattice polymorphisms r are sufficient to make the CSP tractable [24] but for QCSP(Inv(r)), Chen observed two types of behaviours, namely being in P and being co-NP-hard [30]. The separating criterion has to do with the number of strongly connected minimal components in the graph given by $E(x, y)$ iff $r(a, b) = a$. When this number of minimal components is one (it is unique) the problem is in P, otherwise it is co-NP-hard. The situation for semilattice operations s is more fully understood. QCSP(Inv(s)) is in P, if f has a unit, else QCSP(Inv(s)) is Pspace-complete (the Pspace-hardness in this case seems to be due to Bulatov in [18] and did not appear in any previous version of that paper).

A largely combinatorial approach to QCSP complexity was followed in the works [86, 85, 83], deriving various classifications, although the algebraic method was used for tractability arguments in the latter two. These papers were an amusing diversion for their authors but do not necessarily shed much light on how one might argue for complexity classifications in general. The graphs considered in these papers are all instances of partially reflexive pseudoforests, for which a complexity classification for the Retraction problem is known. The Retraction problem may be seen as the CSP in which all constants are available (the CSP classification for partially reflexive pseudoforests without constants is nearly trivial).

In [85] a dichotomy is given for QCSP($\mathcal{H}$) where $\mathcal{H}$ is a partially reflexive forest. In the case of partially reflexive forests, this is between NL and NP-hard, whereas for partially reflexive paths it is between NL and Pspace-complete. A key idea here is *loop-connectedness*, which asks whether the subgraph induced by the self-loops is connected. This plays an important role in the classification of the Retraction problem for (partially reflexive) pseudoforests in [59]. In [85] it is noted that loop-connectivity of a partially reflexive tree $\mathcal{H}$ is a sufficient criterion for NL membership of QCSP($\mathcal{H}$), and this is witnessed by a majority polymorphism (thus even when $\mathcal{H}$ is expanded with constants the problem remains in NL). However, it is not a necessary condition and a simple example is furnished by the undirected path on five vertices, $\mathcal{P}_{10100}$, with the middle vertex as well as one end a loop. This is clearly not loop-connected, yet QCSP($\mathcal{P}_{10100}$) is in NL. Indeed, QCSP($\mathcal{P}_{10100}$) coincides with QCSP($\mathcal{P}_{100}$), where $\mathcal{P}_{100}$ is the undirected path on three vertices with a loop at one end. Clearly, $\mathcal{P}_{100}$ is loop-connected, and so the result follows. It is here where the notion of Q-core explored in [42] is useful, because the Q-core of $\mathcal{P}_{10100}$ is $\mathcal{P}_{100}$. Indeed, when $\mathcal{H}$ is a partially reflexive tree that is a Q-core all of the known classifications for the QCSP (from [86, 85, 83]) are consistent, in the sense of P versus NP-hard, with the classification for the Retraction problem in [59]. Note that the jump from pseudotrees to pseudoforests creates a disconnected graph which causes a collapse in QCSP complexity to NP; and even further to NL, in the case that the graph has a loop or is bipartite.

The QCSP complexity classification for irreflexive pseudotrees is given in [86] (one can easily infer the result for irreflexive pseudoforests). Finally, the classification for partially reflexive cycles is given in [83]. An interesting observation appearing in this classification is the identification of a QCSP tractable graph $\mathcal{C}_{1110}$, the undirected 4-cycle with three self-loops, without a majority polymorphism. Indeed, $\mathcal{C}_{1110}$ is even a Q-core. This is surprising because all tractable cases from the classification for partially reflexive forests possess these majorities in their Q-core.

In [27], a QCSP classification is given for the case of partially reflexive paths expanded with constants. Here, the distinction between NL and NP-hard perfectly follows the classification of [59], but now the case NP-complete becomes possible (some NL cases become NP-complete, for example, this is the case for $\mathcal{P}_{10100}$ expanded with constants). Thus, we reach here a classification which is a trichotomy between NL, NP-complete and Pspace-complete, where we had a dichotomy without constants between NL and Pspace-complete. Note that once we expand with constants we have a constraint language that is both a core and a Q-core.

In [51], a more advanced marriage of combinatorics and algebra was attempted, and for the first-time this mixed approach gave quite sophisticated algebraically proven lower bounds. Semicomplete digraphs are cores so one may assume without affecting QCSP complexity that they are expanded with constants naming their vertices and that their polymorphisms are all idempotent. The complexity classification for QCSP($\mathcal{H}$), where $\mathcal{H}$ is a semicomplete digraph, given in [51] is rooted in the long proof that smooth semicompletes with more than one cycle are idempotent trivial. It follows that QCSP($\mathcal{H}$) is Pspace-complete in this case [18]. Note that the only smooth semicompletes with no more than one cycle are the directed 2- and 3-cycles. It turns out QCSP($\mathcal{H}$) is also Pspace-complete when $\mathcal{H}$ is a smooth semicomplete with more than one cycle with a sequence of sinks added (respectively, sources added). The remainder of the classification is simple, if $\mathcal{H}$ has both a source and a sink, then QCSP($\mathcal{H}$) is in NP; and if $\mathcal{H}$ is the directed 2- or 3-cycle with a sequence of sinks (respectively, sources) added, then QCSP($\mathcal{H}$) is in NL. The NP-complete cases (source and sink plus more than one cycle) can be inferred from the classification for the CSP [4].

4.1 Bounded Alternation

An interesting restriction of the QCSP addresses the case where one allows only bounded alternation of the quantifiers in a (prenex) input instance. An interesting application for the Π_2-CSP is given in Section 3 of [6] (but note the universal quantifiers are relativised). The situation for Boolean constraint languages parallels the QCSP, with the problems Π_{2k}-CSP($\mathcal{B}$) being Π_{2k}^{P}-complete precisely when QCSP($\mathcal{B}$) is Pspace-complete [69].

Bounded alternation reappeared in the theoretical study of the QCSP in the remarkable paper [34] where it was noted that for certain constraint languages $\mathcal{B}$, Π_{2k}-CSP($\mathcal{B}$) is in co-NP, for each k. Indeed, an example of co-NP-completeness may be given by some constraint languages invariant under a semilattice operation without unit [34, 30]. In fact, if f is a semilattice operation, then either Π_{2k}-CSP(Inv(f)) is in P, if f has a unit, or otherwise Π_{2k}-CSP(Inv(f)) is co-NP-complete. Let us recall the dichotomy for the QCSP in this situation, already mentioned, that QCSP(Inv(f)) is in P, if f has a unit, else QCSP(Inv(f)) is Pspace-complete [18]. So, it seems for bounded alternation QCSP, there are three complexity regimes above P: NP-complete, co-NP-complete and maximal complexity (Π_{2k}-complete), where for the QCSP there are only NP-complete and Pspace-complete.

Chen proved in [35] that a computationally effective form of PGP is sufficient to place Π_{2k}-CSP($\mathbb{A}$) in NP. Subsequently, Zhuk has proved that switchability is the only type of PGP for finite algebras [101], thus all forms of PGP associated with finite-domain Π_{2k}-CSP and QCSP are computationally effective.

4.2 Counting Quantifiers

Quite recently has appeared, in the context of the CSP, the study of counting quantifiers of the form $\exists^{\geq j}$ [87]. These quantifiers allow one to assert the existence of at least j elements such that the ensuing property holds, so on a structure $\mathcal{B}$ with domain of size $|B|$, the quantifiers $\exists^{\geq 1}$ and $\exists^{\geq |B|}$ are precisely $\exists$ and $\forall$, respectively. Thus one can study variants of CSP($\mathcal{B}$) in which the input sentence to be evaluated on $\mathcal{B}$ remains positive conjunctive in its quantifier-free part, but is quantified by various counting quantifiers from some non-empty set. For $X \subseteq \{1, \ldots, |B|\}$, $X \neq \emptyset$, the X-CSP($\mathcal{B}$) takes as input a sentence given by a conjunction of atoms quantified by quantifiers of the form $\exists^{\geq j}$ for $j \in X$. It then asks whether this sentence is true on $\mathcal{B}$. In this fashion, the X-CSP may be seen as a natural generalisation of the CSP and QCSP.

In [87] a panoply of classifications is given for X-CSP($\mathcal{B}$), mostly for the situation where $\mathcal{B}$ is some kind of graph. In general, as with the QCSP, complexities of the form P, NP-complete and Pspace-complete are readily available and classifications are either trichotomies or dichotomies between P and NP-hard. Interestingly, when one has access to all the quantifiers $\exists^{\geq 1}, \ldots, \exists^{\geq |B|}$ the intermediate complexity NP-complete seems to disappear (at least I do not know a case of it). The problems X-CSP($\mathcal{H}$) are completely characterised, for any $X \subseteq [|H|]$, when $\mathcal{H}$ is an undirected clique or cycle, into the classes P, NP-complete and Pspace-complete. Then the problem $\{1, 2\}$-CSP($\mathcal{H}$) is considered where $\mathcal{H}$ is an undirected graph, something of a companion to the theorem of Hell and Nešetřil that these CSPs are in P if $\mathcal{H}$ is bipartite and are NP-complete otherwise. The authors show that $\{1, 2\}$-CSP($\mathcal{H}$) is in P if $\mathcal{H}$ is a forest or a bipartite graph with a 4-cycle, and is NP-hard otherwise. For bipartite graphs $\mathcal{H}$ that are neither forest nor contain a 4-cycle it is even shown that $\{1, 2\}$-CSP($\mathcal{H}$) is Pspace-complete.

Finally, a trichotomy theorem is shown for $\{1,2\}$-CSP($\mathcal{H}$) when $\mathcal{H}$ is a complete multipartite graph, with such problems being either in L, NP-complete or Pspace-complete.

The algebraic method, so potent in understanding the complexity of CSPs and QCSPs has recently been tailored to counting quantifiers [25]. The algebraic objects may be seen as a kind of *expanding* polymorphism. Call an operation $f : B^k \to B$ j*-expanding* if, for all $X_1, \ldots, X_k \subseteq B$ such that $|X_1| = \ldots = |X_k| = j$, we have $|f(X_1, \ldots, X_k)| \geq j$. This condition at $j = 1$ is trivial (it says that f is a function) and at $j = |B|$ asserts surjectivity. For $X \subseteq \{1, \ldots, |B|\}$, we say that f is X*-expanding* if it is j-expanding for all $j \in X$. Now, the relations that are X-pp-definable over $\mathcal{B}$ are exactly those that are preserved by the X-expanding polymorphisms of $\mathcal{B}$. In the case of $\{1\}$-pp and $\{1, |B|\}$-pp, this includes the Galois Correspondences we have already met.

Applying this algebraic theory, as was done with the QCSP, à la [18], would be splendid but a number of the arguments seem to fail for expanding polymorphisms (though Mal'tsev seems to go through).

4.3 Infinite Domains

The study of QCSP($\mathcal{B}$), for infinite-domain $\mathcal{B}$, is not as advanced as the like program for CSPs, and was pioneered in [8]. The authors gave an L, NP-complete, co-NP-hard trichotomy for *equality languages*, which are those constraint languages that admit a first-order definition in $(\mathbb{Q}; =)$.

In the modern systematic study of infinite-domain CSPs, the first classification following equality languages was that for *temporal languages* [10], that admit a first-order definition in $(\mathbb{Q}; <)$, and it seems the major thrust in infinite-domain QCSPs today is in this class. Note that $(\mathbb{Q}; =)$ and $(\mathbb{Q}; <)$ both admit quantifier elimination and all first-order definable relations are already quantifier-free definable, say in conjunctive normal form (CNF). For the classification of [8] a key role is played by *negative* and *positive* languages. The latter do not permit negation of any form while the former is broadly the opposite, allowing only disequalities in CNFs except for singleton clauses (conjuncts) of equality. When $\mathcal{B}$ is an equality language, QCSP($\mathcal{B}$) is in L if $\mathcal{B}$ is negative, QCSP($\mathcal{B}$) is NP-complete if $\mathcal{B}$ positive but not negative[4], and co-NP-hard otherwise.

In the papers [29, 28], a classification for QCSPs is given for the positive temporal languages, that is with a positive definition in $(\mathbb{Q}; \leq)$, where the authors show that these QCSPs are either in L, NL-complete, P-complete, NP-complete or Pspace-complete. Each of these cases is both algebraically and syntactically characterised. In the history of temporal CSPs, the fragment *Ord-Horn* played a key role [90]. In [46] a subclass known as *Guarded Ord-Horn* is established as tractable for QCSP. The significance of this class is noted in [100] where it is shown that Guarded Ord-Horn languages are the only tractable case within the *dually-closed* (if language $\mathcal{B}$ over numeric domain pp-defines k-ary R then it also pp-defines $\{(-x_1, \ldots, -x_k) : (x_1, \ldots, x_k) \in R\}$) Ord-Horn languages. That is, when $\mathcal{B}$ is a dually-closed temporal language that is not Guarded Ord-Horn, then QCSP($\mathcal{B}$) is co-NP-hard [100].

In [9] it is shown that temporal constraint languages with polymorphism *min* (also *max*) and *mx* (also its dual) have a tractable QCSP. I will leave the polymorphism *mx* undefined but suffice it to say that it plays an important role in the temporal CSP classification [11].

Counting quantifiers have also been taken to the domain of the equality languages in [88] with various classifications given. Here it is appropriate to also consider quantifiers of the

[4] A relation is both negative and positive when it is just a conjunction of equalities.

form $\forall^{>j}$, meaning that the associated binding holds on all but at most j elements of the domain.

5 Parameterized Complexity

There has been a plethora of papers in recent years devoted to the parameterized complexity of model-checking classes of structures in various fragments of first-order logic. QCSP($\mathcal{B}$) is nothing other than the model-checking problem for positive conjunctive logic on the singleton class $\{\mathcal{B}\}$ and so it is unsurprising that this new line of research impinges upon its study.

The paper [20] considers the positive conjunctive model-checking problem for *posets* where the sentence and poset are both input but where the poset is restricted to come from a certain class. A preliminary result shows a concrete, four-element poset for which the QCSP is NP-hard (most likely one could prove this is Pspace-complete). The main result then is that model-checking positive conjunctive logic on bounded width posets is FPT, where the parameter is the sentence size. In fact, they prove a stronger result where the class of posets is unrestricted, that is model-checking positive conjunctive logic on posets is FPT, where the parameter is the sentence size plus the poset width. The paper is a companion to one proving a result similar to the first but for existential logic [19]. This line of enquiry was pursued by other authors culminating in the demonstration that model-checking first-order logic on posets is FPT when the parameter is the sentence size plus the poset width [62]. Parameterized intractability also features in [20], where the model-checking problem for positive conjunctive logic on posets of bounded depth and bounded cover-degree is shown to be co-W[2]-hard.

Another research direction has to do with left-hand restrictions, where the class from which the input sentence comes is restricted. In this area results usually may appear in both classical and parameterized flavours since they appear as gaps between P and not FPT (assuming W[1] = FPT). The first outstanding result in this area is due to Grohe [66], in which it is noted, when relational arity is bounded, that if the restricted class of primitive positive sentences does not possess *bounded tree-width*, the model-checking problem for this class is not FPT, unless W[1] = FPT. In this line of research the parameter is always the size of the sentence. The converse, that bounded-treewidth yields tractability for this model-checking problem was already known [61], so this result gives a typical type of classification theorem, based on a complexity-theoretic assumption. It is not immediately apparent what the graph-theoretic property of bounded-treewidth means for a class of sentences, yet the translation between primitive positive sentences and relational structures through canonical query and canonical database has been discussed. It is similarly possible to consider such measures for other classes of arity-bounded first-order formulas, and this line of thought was pursued by other authors. Bounded treewidth alone does not guarantee tractability for the model-checking problem for positive conjunctive logic unless both the relational arity and constraint language size are bound [65].

A precursor of Grohe's result was furnished in [67] where only classes of sentences satisfying a certain closure property (broadly speaking that of isomorphism, when the sentence is viewed as a graph) were considered. This situation was generalised for positive conjunctive logic in [40] and then to full first-order logic in [38]. The key notion in these works, and the closure condition alluded to, is *graphical closure*. A class of sentences is graphically closed if it satisfies two types of closure, the first syntactic and the second graphical. The syntactic closure deals with types of logical equivalence including de Morgans laws, associativity, commutativity and distributivity, together with some rules when a bound variable does not a appear free in a literal in a conjunct or disjunct. The graphical closure deals with substitutions of relation symbols in atoms, such that if, e.g., E and F are ternary relation symbols in the signature, then any sentence containing the atoms $E(x, y, x)$ may have this substituted by $F(x, y, x)$ (di-graphical might be more appropriate here, since the order and multiplicity of x and y matter). The main result of [38] uses a relative of tree-width to discern for which graphically closed sets of sentences Φ of bounded arity model-checking first-order logic is FPT. Specifically, if Φ satisfies a condition known as *bounded thickness* then the model-checking problem is in P and therefore FPT; otherwise it is not (assuming FPT $\neq W[1]$). The restriction of bounded thickness to positive conjunctive sentences is *elimination width*, for which the gap between P and not FPT was proved in [40]. The polynomial result of that paper subsumes the earlier polynomial result of [39], where the notion of treewidth was differently generalised for positive conjunctive logic.

Finally, a true analog of Grohe's Theorem, for positive conjunctive logic, not making the assumption of graphical closure, has been given in [43].

A further investigation into parameterized complexity comes in the very recent [57] where a new parameter *prefix pathwidth* is introduced for QBF. Atserias and Oliva [2] had previously shown that, in contrast to SAT, many of the well-known decompositional parameters (such as treewidth and pathwidth) do not reduce the complexity of QBF. The main reason for this appears to be a blindness of these parameters towards the quantifier dependencies between variables of a QBF formula. Prefix pathwidth mitigates some of these difficulties and it is proved that QBF is FPT with respect to this parameter (and the width of the dependency poset). The result directly applies also to QCSP with any bounded domain size and hence has been eligble for inclusion in this survey.

6 Proof Theory and Evaluation

The canonical proof system for propositional formulas in CNF is *Resolution* [52, 94] in which one tries to prove a system of clauses is contradictory. This, therefore, gives the proof theory for SAT. Resolution has been extended for QBF to the popular system of *Q-Resolution* [76]. Q-Resolution, being a system for QBF, is outside the scope of this survey, but in [37] Chen identifies two of its weaknesses as being the restriction to the Boolean domain as well as requiring the input sentence to be in prenex form. In his *QCSP proof system* (glorious in its anonymity) he overcomes these shortcomings. Egly [55] had previously proposed a proof system for non-prenex QBF, but Chen's system appears to be the first to allow for the possibility of a non-Boolean domain. The width notion of this proof system is associated with the notion of *k-judge-consistency* which implies an earlier notion of consistency which Chen and Dalmau used to demonstrate algorithmic tractability in [39].

Recall the positive conjunctive sentence width notion arising from [38] (cf. Section 5) was elimination width. We now designate the *Q-width* of a positive conjunctive sentence to be the maximum of its elimination width and any arity of a relation appearing within. Chen's work [37] has an algorithmic side-effect, giving a simple generic polynomial method for deciding the (uniform) instances $(\phi, \mathcal{B})$ of QCSP where we assume ϕ has Q-width bounded by some constant k. Of course the polynomiality of this has long since been known but relies on such non-trivial-but-tractable devices as the computing of tree decompositions. The positive algorithmic result from [37] extends that from the earlier [39] in two important ways: firstly, there is no prenex assumption; and, secondly, the notion of Q-width is more general than the generalised treewidth. We also already mentioned that k-judge consistency implies the earlier notion of consistency, which can be seen as a further generality of [37] over [39], though this allows the condition in the former to activate the algorithm in the latter. It might also be said that in [37] a bird's eye view of the landscape is obtained through a marriage between the proof system and its associated algorithm.

7 Future Prospects

In [36], Chen made a number of natural conjectures regarding QCSPs, typically concerning idempotent algebras $\mathbb{A}$. My favourite speculated that QCSP($\mathbb{A}$) should be in NP if $\mathbb{A}$ has the PGP and is otherwise Pspace-complete (see Conjectures 5 and 6 in [36]). Dmitriy Zhuk has settled the backward direction by proving that the only form of PGP for finite-domain algebras is switchability [101]. I suspect there is some tight relationship between NP-membership for the QCSP, and PGP in the associated algebra, which will soon find expression.

A structure is *ω-categorical* if it is up to isomorphism the only countably infinite model of its first-order theory. The Galois Correspondence Inv-Pol is known to hold for ω-categorical constraint languages [12] and has been instrumental in a number of recent CSP classifications (e.g. [11, 13]). For the operational side one needs to insist the clones satisfy a certain topological (local) closure [96]. It is not known whether $\mathrm{Inv}(\mathrm{sPol}(\mathcal{B})) = \langle \mathcal{B} \rangle_{\mathrm{pc}}$ for ω-categorical $\mathcal{B}$. The best general result in this direction involves the *periomorphisms* of [45] which work on periodic elements in $\mathcal{B}^\omega$, which have the form $(b_1, \ldots, b_k)^\omega$. The periodic elements induce a countable substructure $\mathcal{B}^{\mathrm{per}}$ in $\mathcal{B}^\omega$ and a periomorphism is a homomorphism from this structure to $\mathcal{B}$. The fact that $\mathcal{B}^{\mathrm{per}}$ is countable permits the use of a standard back-and-forth argument whence it is shown that if a relation is invariant under the periomorphisms of ω-categorical $\mathcal{B}$, then indeed it is positive conjunctive definable. The correspondence $\mathrm{Inv}(\mathrm{sPol}(\mathcal{B})) = \langle \mathcal{B} \rangle_{\mathrm{pc}}$ is known, a posteriori, when $\mathcal{B}$ is an equality language [8]. For languages first-order definable in $(\mathbb{Q}; <)$ it is still in general open.

Meanwhile, let us leave the unfinished classification for temporal QCSPs to ponder that for equality languages. In the conference version of [8] the trichotomy was announced to be between L, NP-complete and Pspace-complete but the greater lower bound was reduced to co-NP-hard in the journal version. The culprit is the erroneous supporting Theorem 4.1, for which one can construct a counterexample, and from which the missing Galois Correspondence of the previous paragraph would have followed (Theorem 4.1 holds for infinite direct products). The major open question from the journal version is whether QCSP($\mathbb{Q}; x = y \rightarrow y = z$), known to be co-NP-hard, is in fact Pspace-complete. Were this to be Pspace-complete, it would complete the promotion of all the outstanding co-NP-hard cases to Pspace-complete. However, were it to be in co-NP, for example, there would remain additional work to be done, not to mention that the trichotomy would become a tetrachotomy, since many cases, including QCSP($\mathbb{Q}; x = y \rightarrow u = v$), are known to be Pspace-complete.

It would be interesting to unite the results of [86, 85, 83] into a QCSP classification for partially reflexive pseudoforests with the classes likely to be NL, NP-complete and Pspace-complete. Even a partial classification into NL and NP-hard might require a patience and diligence that could remain unrewarded by the result. For partially reflexive pseudoforests that are Q-cores, most likely the NL/ NP-hard boundary follows that for Retraction [59].

A number of open questions arise regarding CSP with counting quantifiers and the most interesting relate to potential applications of the new algebraic theory. A more combinatorial question is as to the precise complexity of $\{2\}$-CSP($\mathcal{K}_4$), where $\mathcal{K}_4$ is the 4-clique, which is known to be in P but not in L or NL. This question is interesting as it is the only case in the classification of X-CSP($\mathcal{H}$) in [87], where $\mathcal{H}$ is an undirected clique or cycle, that is known to be in P but not L.

The question of the idempotent remains a thorn in our side. For every finite $\mathcal{B}$, is there a finite $\mathcal{C}$ so that, say, QCSP($\mathcal{B}$) and QCSP($\mathcal{C}$) are polynomially equivalent, and all constants are positive conjunctive definable in $\mathcal{C}$ up to isomorphism? We know this is not true if we strengthen polynomial equivalence to positive conjunctive equivalent [42].

The QCSP program initiated in [51] continues a fascinating combinatorial-algebraic program itself well-established, e.g. in [80, 26, 3]. Bandelt has classified both which reflexive and which bipartite graphs admit a majority polymorphism (see [3]). Indeed, the distinction between majority and not is established for partially reflexive trees in [85]. In a similar vein, in [26] it is established precisely which digraphs have a Mal'tsev polymorphism. Curiously, Malt'sev digraphs also have a majority [73]. The business of [51] is more in line with several investigations of Benoit Larose into idempotent triviality (see [80]). Other recent work has focused on whether certain constraint languages for which we know the CSP classification follow also the conjectured algebraic classification (which indeed they do). MacGillivray and Swarts [82] prove that the (irreflexive) locally semicomplete digraphs whose CSPs are tractable are exactly those that admit a weak near-unanimity polymorphism, and Wires has proved [99] that the partially reflexive tournaments whose CSP with constants is tractable are exactly those that admit a Taylor polymorphism. Recall that a finite idempotent algebra generates a weak near-unanimity operation iff it generates a Taylor operation. It would be fun to establish which irreflexive locally semicomplete and partially reflexive tournaments are idempotent trivial, with a view to leveraging this knowledge towards a QCSP classification, of the kind in [51].

The outstanding question left open in parameterized complexity in the region of QCSP, though somewhat superseding it, is to unify the works [38] add [43]. That is, to give a theorem à la Grohe, as the latter, for full first-order logic.

Acknowledgements. I am grateful for helpful discussions with a number of people and especially corrections to an early draft from Hubie Chen and Michał Wrona. I am also grateful for corrections from an anonymous referee appointed by the editors.

References

1 Bengt Aspvall, Michael F. Plass, and Robert Endre Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Inf. Process. Lett.*, 8(3):121–123, 1979.

2 Albert Atserias and Sergi Oliva. Bounded-width QBF is PSPACE-complete. *J. Comput. Syst. Sci.*, 80(7):1415–1429, 2014.

3 Hans-Jürgen Bandelt. Graphs with edge-preserving majority functions. *Discrete Mathematics*, 103(1):1–5, 1992.

4 Jørgen Bang-Jensen, Pavol Hell, and Gary MacGillivray. The complexity of colouring by semicomplete digraphs. *SIAM J. Discrete Math.*, 1(3):281–298, 1988.

5 Libor Barto, Marcin Kozik, and Todd Niven. The CSP dichotomy holds for digraphs with no sources and no sinks (a positive answer to a conjecture of Bang-Jensen and Hell). *SIAM Journal on Computing*, 38(5):1782–1802, 2009.

6 Frédéric Benhamou and Frédéric Goualard. *Principles and Practice of Constraint Programming – CP 2000: 6th International Conference, CP 2000 Singapore, September 18–21, 2000 Proceedings*, chapter Universally Quantified Interval Constraints, pages 67–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

7 Manuel Bodirsky and Hubie Chen. Relatively quantified constraint satisfaction. *Constraints*, 14(1):3–15, 2009.

8 Manuel Bodirsky and Hubie Chen. Quantified equality constraints. *SIAM J. Comput.*, 39(8):3682–3699, 2010. Conference version appeared at LICS 2007.

9 Manuel Bodirsky, Hubie Chen, and Michal Wrona. Tractability of quantified temporal constraints to the max. *IJAC*, 24(8):1141–1156, 2014.

10 Manuel Bodirsky and Jan Kára. The complexity of temporal constraint satisfaction problems. In *Proceedings of STOC'08*, pages 29–38, 2008. Accepted for publication in J. ACM.

11 Manuel Bodirsky and Jan Kára. The complexity of temporal constraint satisfaction problems. *J. ACM*, 57(2), 2010.

12 Manuel Bodirsky and Jaroslav Nešetřil. Constraint satisfaction with countable homogeneous templates. *Journal of Logic and Computation*, 16(3):359–373, 2006.

13 Manuel Bodirsky and Michael Pinsker. Schaefer's theorem for graphs. *Journal of the ACM*, 62(3):Article no. 19, 1–52, 2015. A conference version appeared in the Proceedings of STOC 2011, pages 655–664.

14 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.

15 Lucas Bordeaux, Marco Cadoli, and Toni Mancini. Generalizing consistency and other constraint properties to quantified constraints. *ACM Trans. Comput. Log.*, 10(3), 2009. Extended abstract appeared at AAAI 2005 titled CSP properties for for quantified constraints: Definitions and complexity.

16 Lucas Bordeaux and Eric Monfroy. Beyond NP: arc-consistency for quantified constraints. In *Principles and Practice of Constraint Programming – CP 2002, 8th International Conference, CP 2002, Ithaca, NY, USA, September 9-13, 2002, Proceedings*, pages 371–386, 2002.

17 F. Börner, A. Krokhin, A. Bulatov, and P. Jeavons. Quantified constraints and surjective polymorphisms. Technical Report PRG-RR-02-11, Oxford University, 2002. Conference version appeared at CSL 2003 titled: Quantified Constraints – Algorithms and Complexity.

18 Ferdinand Börner, Andrei A. Bulatov, Hubie Chen, Peter Jeavons, and Andrei A. Krokhin. The complexity of constraint satisfaction games and qcsp. *Inf. Comput.*, 207(9):923–944, 2009.

19 Simone Bova, Robert Ganian, and Stefan Szeider. Model checking existential logic on partially ordered sets. In *Joint Meeting of the 23rd EACSL Annual Conf. on Computer Science Logic (CSL) and the 29th Annual ACM/IEEE Symp. on Logic in Computer Science (LICS), CSL-LICS'14, Vienna, Austria, July 14 – 18, 2014*, pages 21:1–21:10, 2014.

20 Simone Bova, Robert Ganian, and Stefan Szeider. Quantified conjunctive queries on partially ordered sets. *Theor. Comput. Sci.*, 618:72–84, 2016. Extended abstract appeared at IPEC 2004.

21 A. Bulatov, A. Krokhin, and P. G. Jeavons. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34:720–742, 2005.

22 Andrei Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM*, 53(1):66–120, 2006.

23 Andrei Bulatov, Peter Jeavons, and Mikhail Volkov. Finite semigroups imposing tractable constraints. In *Proceedings of the School on Algorithmic Aspects of the Theory of Semigroups and its Applications, Coimbra, Portugal, 2001*, pages 313–329. World Scientific, Singapore, 2002.

24 Andrei A. Bulatov. Combinatorial problems raised from 2-semilattices. *Journal of Algebra*, 298(2):321–339, 2006.

25 Andrei A. Bulatov and Amir Hedayaty. Galois correspondence for counting quantifiers. *Multiple-Valued Logic and Soft Computing*, 24(5-6):405–424, 2015. First version appeared on arxiv in 2012.

26 Catarina Carvalho, László Egri, Marcel Jackson, and Todd Niven. On maltsev digraphs. *Electr. J. Comb.*, 22(1):P1.47, 2015. Extended abstract appeared at CSR 2011.

27 Catarina Carvalho, Florent R. Madelaine, and Barnaby Martin. From complexity to algebra and back: Digraph classes, collapsibility, and the PGP. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 462–474, 2015.

28 Witold Charatonik and Michal Wrona. Quantified positive temporal constraints. In *Computer Science Logic, 22nd International Workshop, CSL 2008, 17th Annual Conference of the EACSL, Bertinoro, Italy, September 16-19, 2008. Proceedings*, pages 94–108, 2008.

29 Witold Charatonik and Michal Wrona. Tractable quantified constraint satisfaction problems over positive temporal templates. In *Logic for Programming, Artificial Intelligence, and Reasoning, 15th International Conference, LPAR 2008, Doha, Qatar, November 22-27, 2008. Proceedings*, pages 543–557, 2008.

30 Hubie Chen. Quantified constraint satisfaction and 2-semilattice polymorphisms. In *Principles and Practice of Constraint Programming – CP 2004, 10th International Conference, CP 2004, Toronto, Canada, September 27 – October 1, 2004, Proceedings*, pages 168–181, 2004.

31 Hubie Chen. Quantified constraint satisfaction, maximal constraint languages, and symmetric polymorphisms. In *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science, Stuttgart, Germany, February 24-26, 2005, Proceedings*, pages 315–326, 2005.

32 Hubie Chen. A rendezvous of logic, complexity, and algebra. *SIGACT News*, 2006.

33 Hubie Chen. The complexity of quantified constraint satisfaction: Collapsibility, sink algebras, and the three-element case. *SIAM J. Comput.*, 37(5):1674–1701, 2008. Incorporates extended abstract titled *Collapsibility and Consistency in Quantified Constraint Satisfaction* presented at AAAI 2004.

34 Hubie Chen. Existentially restricted quantified constraint satisfaction. *Inf. Comput.*, 207(3):369–388, 2009.

35 Hubie Chen. Quantified constraint satisfaction and the polynomially generated powers property. *Algebra universalis*, 65(3):213–241, 2011. An extended abstract appeared in ICALP B 2008.

36 Hubie Chen. Meditations on quantified constraint satisfaction. In *Logic and Program Semantics – Essays Dedicated to Dexter Kozen on the Occasion of His 60th Birthday*, pages 35–49, 2012.

37 Hubie Chen. Beyond q-resolution and prenex form: A proof system for quantified constraint satisfaction. *Logical Methods in Computer Science*, 10(4), 2014.

38 Hubie Chen. The tractability frontier of graph-like first-order query sets. In *Joint Meeting of the 23rd EACSL Annual Conf. on Computer Science Logic (CSL) and the 29th Annual ACM/IEEE Symp. on Logic in Computer Science (LICS), CSL-LICS'14, Vienna, Austria, July 14-18, 2014*, pages 31:1–31:9, 2014.

39 Hubie Chen and Víctor Dalmau. From pebble games to tractability: An ambidextrous consistency algorithm for quantified constraint satisfaction. In *Computer Science Logic, 19th International Workshop, CSL 2005, 14th Annual Conference of the EACSL, Oxford, UK, August 22-25, 2005, Proceedings*, pages 232–247, 2005.

40 Hubie Chen and Víctor Dalmau. Decomposing quantified conjunctive (or disjunctive) formulas. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*, pages 205–214, 2012.

41 Hubie Chen, Florent Madelaine, and Barnaby Martin. Quantified constraints and containment problems. In *23rd Annual IEEE Symposium on Logic in Computer Science*, pages 317–328, 2008.

42 Hubie Chen, Florent R. Madelaine, and Barnaby Martin. Quantified constraints and containment problems. *Logical Methods in Computer Science*, 11(3), 2015. Extended abstract appeared at LICS 2008. This journal version incorporates principal part of CP 2012 *Containment, Equivalence and Coreness from CSP to QCSP and Beyond.*

43 Hubie Chen and Dániel Marx. Block-sorted quantified conjunctive queries. In *Automata, Languages, and Programming – 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*, pages 125–136, 2013.

44 Hubie Chen and Peter Mayr. Quantified constraint satisfaction on monoids, 2016.

45 Hubie Chen and Moritz Müller. An algebraic preservation theorem for aleph-zero categorical quantified constraint satisfaction. *Logical Methods in Computer Science*, 9(1), 2012. An extended abstract appeared at LICS 2012.

46 Hubie Chen and Michal Wrona. Guarded ord-horn: A tractable fragment of quantified constraint satisfaction. In *19th International Symp. on Temporal Representation and Reasoning, TIME 2012, Leicester, United Kingdom, September 12-14, 2012*, pages 99–106, 2012.

47 Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems.* SIAM Monographs on Discrete Mathematics and Applications 7, 2001.

48 Victor Dalmau. Some dichotomy theorems on constant-free quantified boolean formulas. Technical Report LSI-97-43-R. Departament LSI (UPC), 1997.

49 Víctor Dalmau. A new tractable class of constraint satisfaction problems. *Ann. Math. Artif. Intell.*, 44(1-2):61–85, 2005.

50 Víctor Dalmau and Justin Pearson. Closure functions and width 1 problems. In *Principles and Practice of Constraint Programming – CP'99, 5th International Conference, Alexandria, Virginia, USA, October 11-14, 1999, Proceedings*, pages 159–173, 1999.

51 Petar Dapic, Petar Markovic, and Barnaby Martin. QCSP on semicomplete digraphs. In *Automata, Languages, and Programming – 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 847–858, 2014.

52 Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, July 1960.

53 Rina Dechter. From local to global consistency. *Artif. Intell.*, 55(1):87–108, 1992.

54 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013.

55 Uwe Egly. On sequent systems and resolution for QBFs. In *Theory and Applications of Satisfiability Testing – SAT 2012 – 15th International Conference, Trento, Italy, June 17-20, 2012. Proceedings*, pages 100–113, 2012.

56 Uwe Egly, Thomas Eiter, Hans Tompits, and Stefan Woltran. Solving advanced reasoning tasks using quantified boolean formulas. In *Proc. 17th Nat. Conf. on Artificial Intelligence and 12th Conf. on Innovative Applications of Artificial Intelligence*, pages 417–422. AAAI Press/ The MIT Press, 2000.

57 Eduard Eiben, Robert Ganian, and Sebastian Ordyniak. Using decomposition-parameters for qbf: Mind the prefix! In *AAAI 2016*, 2016.

58 T. Feder and M. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing*, 28:57–104, 1999.

59 Tomás Feder, Pavol Hell, Peter Jonsson, Andrei A. Krokhin, and Gustav Nordh. Retractions to pseudoforests. *SIAM J. Discrete Math.*, 24(1):101–112, 2010.

60 Alex Ferguson and Barry O'Sullivan. Relaxations and explanations for quantified constraint satisfaction problems. In *Principles and Practice of Constraint Programming – CP 2006, 12th International Conf., CP 2006, Nantes, France, September 25-29, 2006, Proceedings*, pages 690–694, 2006.

61 E. C. Freuder. Complexity of k-tree structured constraint satisfaction problems. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 4–9, 1990.

62 Jakub Gajarský, Petr Hlinený, Daniel Lokshtanov, Jan Obdrzálek, Sebastian Ordyniak, M. S. Ramanujan, and Saket Saurabh. FO model checking on posets of bounded width. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 963–974, 2015.

63 D. Geiger. Closed systems of functions and predicates. *Pacific Journal of Mathematics*, 27:95–100, 1968.

64 Ian P. Gent, Peter Nightingale, Andrew G. D. Rowley, and Kostas Stergiou. Solving quantified constraint satisfaction problems. *Artif. Intell.*, 172(6-7):738–771, 2008. This is an extended version of conference papers Encoding Quantified CSPs as Quantified Boolean Formulae (ECAI 2004) and QCSP-Solve: A Solver for Quantified Constraint Satisfaction Problems (IJCAI 2005).

65 Georg Gottlob, Gianluigi Greco, and Francesco Scarcello. The complexity of quantified constraint satisfaction problems under structural restrictions. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*, pages 150–155, 2005.

66 Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM*, 54(1), 2007. Extended abstract appeared at FOCS 2003.

67 Martin Grohe, Thomas Schwentick, and Luc Segoufin. When is the evaluation of conjunctive queries tractable? In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 657–666, 2001.

68 P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B*, 48:92–110, 1990.

69 Edith Hemaspaandra. Dichotomy theorems for alternation-bounded quantified boolean formulas. *CoRR*, cs.CC/0406006, 2004.

70 P. G. Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.

71 Peter Jeavons, David Cohen, and Martin Cooper. Constraints, consistency and closure. *AI*, 101(1-2):251–265, 1998.

72 Marek Karpinski, Hans Kleine Büning, and Peter H. Schmitt. On the computational complexity of quantified horn clauses. In *CSL'87, 1st Workshop on Computer Science Logic, Karlsruhe, Germany, October 12-16, 1987, Proceedings*, pages 129–137, 1987.

73 Alexandr Kazda. Maltsev digraphs have a majority polymorphism. *Eur. J. Comb.*, 32(3):390–397, 2011.

74 Jerome Keisler. Reduced products and Horn classes. *Trans. Amer. Math. Soc.*, 117:307–328, 1965.

75 Hans Kleine Büning and Uwe Bubeck. Theory of quantified boolean formulas. In *Handbook of Satisfiability*, pages 735–760. 2009.

76 Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified boolean formulas. *Inf. Comput.*, 117(1):12–18, 1995.

77 Ph. G. Kolaitis and M. Y. Vardi. *Finite Model Theory and Its Applications (Texts in Theoretical Computer Science. An EATCS Series)*, chapter A logical Approach to Constraint Satisfaction. Springer-Verlag New York, Inc., 2005.

78 Ph. G. Kolaitis and M. Y. Vardi. A game-theoretic approach to constraint satisfaction. In *Proceedings of the 17th National Conference on AI*, pages 175–181, 2000.

79 Richard E. Ladner. On the structure of polynomial time reducibility. *J.ACM*, 22(1):155–171, 1975.

80 Benoit Larose. Taylor operations on finite reflexive structures. *International Journal of Mathematics and Computer Science*, 1:1–21, 2006.

81 Benoit Larose and László Zádori. Taylor terms, constraint satisfaction and the complexity of polynomial equations over finite algebras. *IJAC*, 16(3):563–582, 2006.

82 Gary MacGillivray and Jacobus Swarts. Weak near-unanimity functions and digraph homomorphism problems. *Theor. Comput. Sci.*, 477:32–47, 2013.

83 Florent R. Madelaine and Barnaby Martin. QCSP on partially reflexive cycles – the wavy line of tractability. In *Computer Science – Theory and Applications – 8th International Computer Science Symposium in Russia, CSR 2013, Ekaterinburg, Russia, June 25-29, 2013. Proceedings*, pages 322–333, 2013.

84 Miklós Maróti and Ralph McKenzie. Existence theorems for weakly symmetric operations. *Algebra universalis*, 59(3):463–489, 2008.

85 Barnaby Martin. QCSP on partially reflexive forests. In *Principles and Practice of Constraint Programming – 17th International Conference, CP 2011*, 2011.

86 Barnaby Martin and Florent Madelaine. Towards a trichotomy for quantified H-coloring. In *2nd Conf. on Computatibility in Europe, LNCS 3988*, pages 342–352, 2006.

87 Barnaby Martin, Florent R. Madelaine, and Juraj Stacho. Constraint satisfaction with counting quantifiers. *SIAM J. Discrete Math.*, 29(2):1065–1113, 2015. This unifies and expands conference papers from CSR 2012 and 2014.

88 Barnaby Martin, András Pongrácz, and Michal Wrona. The complexity of counting quantifiers on equality languages. In *Pursuit of the Universal – 12th Conf. on Computability in Europe, CiE 2016, Paris, France, 2016, Proceedings*, pages 333–342, 2016.

89 Deepak Mehta, Barry O'Sullivan, and Luis Quesada. Extending the notion of preferred explanations for quantified constraint satisfaction problems. In *Theoretical Aspects of Computing – ICTAC 2015 – 12th International Colloquium Cali, Colombia, October 29-31, 2015, Proceedings*, pages 309–327, 2015.

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

91 Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

92 R. Pöschel and L.,A. Kalužnin. *Funktionen- und Relationenalgebren*. Deutscher Verlag der Wissenschaften, 1979.

93 Jussi Rintanen. Constructing conditional plans by a theorem-prover. *Journal of Artificial Intelligence Research*, 10:323–352, 1999.

94 J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, January 1965.

95 T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of STOC'78*, pages 216–226, 1978.

96 Á. Szendrei. *Clones in universal algebra*. Séminaire de mathématiques supérieures. Presses de l'Université de Montréal, 1986.

97 Peter van Beek. On the minimality and decomposability of constraint networks. In *Proceedings of the 10th National Conference on Artificial Intelligence. San Jose, CA, July 12-16, 1992.*, pages 447–452, 1992.

98 James Wiegold. Growth sequences of finite semigroups. *Journal of the Australian Mathematical Society (Series A)*, 43:16–20, 8 1987. Communicated by H. Lausch.

99 Alexander Wires. Dichotomy for finite tournaments of mixed-type. *Discrete Mathematics*, 338(12):2523–2538, 2015.

100 Michal Wrona. Tractability frontier for dually-closed ord-horn quantified constraint satisfaction problems. In *Mathematical Foundations of Computer Science 2014 – 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*, pages 535–546, 2014.

101 D. Zhuk. The Size of Generating Sets of Powers. *ArXiv e-prints*, April 2015.