An Algebraic Approach to Multi-sorted Constraints

Andrei A. Bulatov and Peter Jeavons
Computing Laboratory, University of Oxford, UK
{Andrei.Bulatov,Peter.Jeavons}@comlab.ox.ac.uk

Abstract. Most previous theoretical study of the complexity of the constraint satisfaction problem has considered a simplified version of the problem in which all variables have the same domain. We show here that this apparently minor simplification can in fact change the complexity of the problem, and hence mask the existence of certain tractable constraint types. In this paper we describe a new algebraic framework which allows us to deal more precisely with problems where different variables may have different domains. Using this new framework we are able to identify new tractable classes of constraints, by combining algorithms devised for the simplified, single domain, problem. We also systematically develop an algebraic structural theory for the general problem, and show that this theory can be used to generalise earlier results about the complexity of certain constraint types.

1 Introduction

There is a striking difference between theoretical studies of the complexity of the Constraint Satisfaction Problem (CSP), and more applied work on this problem: in most theoretical studies constraint satisfaction problems are assumed to have the same domain for all variables (we will call such problems *one-sorted*), while in practice the different variables of a CSP often have different domains (we will call problems of this type *multi-sorted*). This apparently minor simplification can have serious consequences for the analysis of the complexity of different forms of constraint; it can in fact mask the difference between tractability and NP-completeness for certain problems.

Example 1. Consider the following relation of arity 5 containing 17 tuples (shown vertically):

$$\varrho = \begin{pmatrix} 3&3&1&1&1&1&3&3&3&3&1&1&3&3&1&1&3 \\ 1&3&0&2&1&3&0&2&1&3&0&2&1&3&1&3&3 \\ 2&2&2&2&0&0&0&0&2&2&2&2&2&2&0&0&2 \\ c&c&c&c&c&c&c&c&c&c&c&c&a&a&a&a&a \\ b&b&b&b&b&b&b&b&a&a&a&a&b&b&b&b&a \end{pmatrix}$$

If we consider this relation in the usual way as a one-sorted relation over the domain $\{0,1,2,3,a,b,c\}$, then it does not fall into any of the many known (one-sorted) tractable classes[1] [2, 3, 7, 9, 13, 18, 19, 22, 25, 26].

[1] This was established by using the program *Polyanna* [?] available from http://www.comlab.ox.ac.uk/oucl/research/areas/constraints/software/

However, if we consider this relation as a *multi-sorted* relation, over the two separate domains $\{0, 1, 2, 3\}$ and $\{a, b, c\}$ (in the sense defined below), then it can easily be shown to be tractable, using the results obtained in this paper (see Example 8).

This paper is part of a general investigation into how the complexity of the constraint satisfaction problem varies with the forms of constraints which are allowed. Considerable progress has been made in this investigation over the past few years. For example, a complete characterisation of tractable constraint types is now known for both 2-element domains [24] and 3-element domains [1]. In addition, a number of novel efficient algorithms have been developed for solving particular types of CSPs [2, 10, 13, 19].

However, almost all previous work on complexity has focused on the one-sorted CSP; the first goal of this paper is to develop an approach which allows us to study the complexity of multi-sorted constraint satisfaction problems, where different variables have different domains. Using this approach, we show that many of the known algorithms for the one-sorted case can be combined, and hence applied to much broader classes of constraint satisfaction problems.

The second goal of the paper is to further develop the strong links between the study of complexity of the CSP and the mathematical study of finite algebras. These links were introduced and developed for the one-sorted case in [18, 17, 6]. In the multi-sorted case, these links allow one to use even more effectively the powerful mathematical theory developed for classifying the structure of finite algebras [21, 15]. For example, early versions of the results given here have already been successfully applied to obtain new tractable classes [3, 2]. Moreover, the results described in this paper are heavily used in proving a dichotomy theorem for constraint satisfaction problems over a 3-element domain [1].

As an example of the flexibility of the multi-sorted approach developed here, we consider the analysis of constraints which restrict the domain of each individual variable in their scope. By using the link with finite algebras described below, we are able to generalize and strengthen the dichotomy result of [12] classifying all constraints which restrict the domain of each variable to two possible values.

2 The Multi-Sorted Constraint Satisfaction Problem

The central notion in the mathematical study of constraints and constraint satisfaction problems is the notion of a *relation*. In this paper we will allow *multi-sorted* relations, that is, relations over an arbitrary collection of sets. These are defined as follows.

Definition 1. *For any collection of sets $\mathcal{A} = \{A_i \mid i \in I\}$, and any list of indices $(i_1, i_2, \ldots, i_m) \in I^m$, a subset ϱ of $A_{i_1} \times A_{i_2} \times \cdots \times A_{i_m}$, together with the list $(i_1, i_2, \ldots, i_m)$, will be called a* multi-sorted relation *over $\mathcal{A}$ with arity m and* signature $(i_1, i_2, \ldots, i_m)$. *For any such relation ϱ, the signature of ϱ will be denoted $\sigma(\varrho)$.*

In the special case where $\mathcal{A}$ contains only a single set A, we shall refer to a relation over $\mathcal{A}$ as a *one-sorted* relation over the set A.

Example 2. Let $\mathcal{A} = \{A_1, A_2\}$, where $A_1 = \{0, 1, 2, 3\}$ and $A_2 = \{a, b, c\}$.

The relation ϱ, defined in Example 1, can be viewed as a multi-sorted relation over $\mathcal{A}$ with arity 5 and signature $(1, 1, 1, 2, 2)$.

Given any set of multi-sorted relations, we can define a corresponding class of multi-sorted constraint satisfaction problems, in the following way.

Definition 2. *Let Γ be a set of multi-sorted relations over a collection of sets $\mathcal{A} = \{A_i \mid i \in I\}$. The* multi-sorted constraint satisfaction problem over Γ*, denoted* $\mathrm{MCSP}(\Gamma)$*, is defined to be the decision problem with*

INSTANCE: *A quadruple $(V; \mathcal{A}; \delta; \mathcal{C})$ where*
- *V is a set of* variables*;*
- *δ is a mapping from V to I, called the* domain function*;*
- *$\mathcal{C}$ is a set of* constraints*, where each constraint $C \in \mathcal{C}$ is a pair $\langle s, \varrho \rangle$, such that*
  - *$s = (v_1, \ldots, v_{m_C})$ is a tuple of variables of length m_C, called the* constraint scope*;*
  - *ϱ is an element of Γ with arity m_C and signature $(\delta(v_1), \ldots, \delta(v_{m_C}))$, called the* constraint relation*.*

QUESTION: *Does there exist a* solution*, i.e., a function φ, from V to $\bigcup_{A \in \mathcal{A}} A$, such that, for each variable $v \in V$, $\varphi(v) \in A_{\delta(v)}$, and for each constraint $\langle s, \varrho \rangle \in \mathcal{C}$, with $s = (v_1, \ldots, v_m)$, the tuple $(\varphi(v_1), \ldots, \varphi(v_m))$ belongs to ϱ?*

Example 3. Constraint satisfaction problems in which each variable has a distinct set of possible values frequently arise in the study of *databases*, although a rather different vocabulary is normally used, as in the following definition.

Definition 3. *A* relational database *is a finite collection of* tables*. A table consists of a* scheme *and an* instance*:*

A scheme is a finite set of attributes*, where each attribute has an associated set of possible values, referred to as a* domain*.*
An instance is a finite set of rows*, where each row is a mapping that associates with each attribute of the scheme a value in its domain.*

A standard problem arising in connection with relational databases is the CONJUNCTIVE QUERY EVALUATION problem [20, 27]. In this problem we are asked if a *conjunctive query* to a relational database, that is, a query of the form $\exists x_1 \ldots \exists x_k (\varrho_1 \wedge \ldots \wedge \varrho_n)$ where $\varrho_1, \ldots, \varrho_n$ are atomic formulas, has a solution. An instance of a multi-sorted constraint satisfaction problem corresponds to a conjunctive query over a relational database by a simple translation of terms: 'attributes' have to be replaced with 'variables', 'tables' with 'constraint relations', 'scheme' with 'signature', 'instance' with 'constraint relation', and 'rows' with 'tuples'. Hence a conjunctive query is equivalent to a multi-sorted CSP instance whose variables are the variables of the query. For each atomic formula ϱ_i in the query, there is a constraint C such that the scope of C is the list of variables of ϱ_i and the constraint relation of C is the set of models of ϱ_i.

In the special case where Γ is a set of one-sorted relations over a single set A, we shall use the notation $\mathrm{CSP}(\Gamma)$, and refer to this as a *one-sorted* problem class. An instance of $\mathrm{CSP}(\Gamma)$ can be specified by a triple $(V; A; \mathcal{C})$.

Example 4. Consider the GRAPH q-COLORABILITY problem. An instance of this problem consists of a graph G, and the question is whether the vertices of G can be labelled with q colours so that adjacent vertices are assigned different colours.

Each instance G of GRAPH q-COLORABILITY corresponds to an instance $\mathcal{P}_G$ of $\mathrm{CSP}(\{\neq_A\})$, where A is a q-element set and $\neq_A \;= \{(a,b) \in A^2 \mid a \neq b\}$. The variables of $\mathcal{P}_G$ are the vertices of the graph G, and for each edge $\{v,w\}$ of G, there is a constraint $(\{v,w\}, \neq_A)$ in $\mathcal{P}_G$.

In the remainder of the paper we shall be concerned with distinguishing between those sets of relations which give rise to *tractable* problems (i.e., problems for which there exists a polynomial-time solution algorithm) and those which do not. In order to be able to classify *infinite*, as well as finite, sets of relations, we define the notion of a tractable set of relations in a way that depends on finite subsets only.

Definition 4. *A set of multi-sorted relations, Γ, is said to be* tractable*, if* $\mathrm{MCSP}(\Gamma')$ *is tractable for each finite subset $\Gamma' \subseteq \Gamma$.*

A set of multi-sorted relations, Γ, is said to be NP-complete*, if* $\mathrm{MCSP}(\Gamma')$ *is NP-complete for some finite subset $\Gamma' \subseteq \Gamma$.*

It might be tempting to assume that the complexity of a set of multi-sorted relations could be determined by considering each of the domains involved separately; in other words, by separating the relations into a number of one-sorted relations, and analysing the complexity of each of these. However, in general this simple approach does not work, as the next example demonstrates.

Example 5. Consider the sets $A_1 = \{0,1\}$ and $A_2 = \{a,b,c\}$, and the relations

$$\varrho_1 = \begin{pmatrix} 1\,0\,0 \\ a\;b\;c \end{pmatrix}, \qquad \varrho_2 = \begin{pmatrix} 0\,1\,0 \\ a\;b\;c \end{pmatrix}, \qquad \varrho_3 = \begin{pmatrix} 0\,0\,1 \\ a\;b\;c \end{pmatrix}$$

over $\{A_1, A_2\}$, each with signature $(1,2)$.

If we divide each of these relations into two separate one-sorted relations, then we obtain just the unary relations $\{0,1\}$ and $\{a,b,c\}$ over the sets A_1 and A_2 respectively. Each of these unary relations individually is clearly tractable.

However, by establishing a reduction from the NP-complete problem ONE-IN-THREE [14], it can be shown that the set of multi-sorted relations $\Gamma = \{\varrho_1, \varrho_2, \varrho_3\}$ is NP-complete.

To obtain the reduction we note that the ONE-IN-THREE problem may be expressed as $\mathrm{CSP}(\{\varrho\})$ where $\varrho = \{(1,0,0),(0,1,0),(0,0,1)\}$. Given any problem instance $\mathcal{P} = (V; \{0,1\}; \{C_1, \ldots, C_q\}) \in \mathrm{CSP}(\{\varrho\})$, we introduce q auxiliary variables $v_1, \ldots, v_q$, distinct from the variables in V, and set $\mathcal{P}' = (V \cup \{v_1, \ldots, v_q\}; \{A_1, A_2\}; \delta; \{C_1^1, C_1^2, C_1^3, C_2^1, \ldots, C_q^3\})$, where $\delta(v) = 1$ if $v \in V$, $\delta(v) = 2$ otherwise, and for each $C_i = \langle (w_1, w_2, w_3), \varrho \rangle$, the new constraints are $C_i^1 = \langle (w_1, v_i), \varrho_1 \rangle$, $C_i^2 = \langle (w_2, v_i), \varrho_2 \rangle$, and $C_i^3 = \langle (w_3, v_i), \varrho_3 \rangle$. It is easy to check that $\mathcal{P}$ and $\mathcal{P}'$ are equivalent.

The next example indicates that a set of constraints which is NP-complete when viewed as one-sorted, can become tractable when viewed as multi-sorted: the tractability is due to the signatures of the relations rather than the tuples they contain.

Example 6. Let A_1 and A_2 be two distinct supersets of a set A, and let Γ be the set containing the single binary disequality relation $\neq_A$ as defined in Example 4, but now considered as a multi-sorted relation over $\{A_1, A_2\}$ with signature $(1, 2)$.

Because of the signature, this constraint can only be imposed between two variables when one of them has domain A_1 and the other has domain A_2. Hence, in this case MCSP(Γ) corresponds to the problem of colouring a *bipartite graph* with $|A|$ colours, which is clearly tractable.

It is often desirable to convert a multi-sorted constraint satisfaction problem into a one-sorted problem. The most straightforward way to do this for a given multi-sorted problem instance $(V; \mathcal{A}; \delta; \mathcal{C})$, is to take $B = \bigcup_{A \in \mathcal{A}} A$, and replace each constraint relation with a one-sorted relation over B containing exactly the same tuples.

However, applying this procedure to the disequality relation in Example 6 gives the usual disequality relation over A, which for $|A| > 2$ is NP-complete (see Example 4). Hence, this straightforward conversion method does *not* necessarily preserve the tractability of Γ. To ensure that we do preserve the tractability of Γ, we shall make use of a more sophisticated conversion technique, based on the following definition. Note that in this definition, and throughout the paper, the ith component of a tuple $\overline{a}$ is denoted by $\overline{a}[i]$.

Definition 5. *For any m-ary relation ϱ over $\{A_1, \ldots, A_n\}$ with signature $\sigma(\varrho) = (i_1, \ldots, i_m)$, let $A = A_1 \times A_2 \times \cdots \times A_n$ and define the one-sorted m-ary relation $\chi(\varrho)$ over A as follows:*

$$\chi(\varrho) = \{(\overline{a}_1, \overline{a}_2, \ldots, \overline{a}_m) \in A^m \mid (\overline{a}_1[i_1], \ldots, \overline{a}_m[i_m]) \in \varrho\}.$$

For any set of relations Γ, the set $\{\chi(\varrho) \mid \varrho \in \Gamma\}$ will be denoted $\chi(\Gamma)$.

Note that for any one-sorted relation ϱ, we have $\chi(\varrho) = \varrho$.

Example 7. Let ϱ be the binary disequality relation $\neq_A$ over A_1, A_2 with signature $(1, 2)$, as in Example 6. In this case $\chi(\varrho)$ is the relation consisting of all pairs $((a, a'), (b, b')) \in (A_1 \times A_2) \times (A_1 \times A_2)$ such that $a, b' \in A$ and $a \neq b'$.

Proposition 1. *Let Γ be a set of multi-sorted relations over the finite sets $A_1, \ldots, A_n$. The set Γ is tractable if and only if the corresponding set of one-sorted relations $\chi(\Gamma)$ is tractable.*

Proof: Let $\mathcal{P} = (V; \{A_1, \ldots, A_n\}; \delta; \mathcal{C})$ be an instance of MCSP(Γ) where $\mathcal{C} = \{C_1, C_2, \ldots, C_q\}$ and each $C_i = \langle s_i, \varrho_i \rangle$.

Consider the one-sorted instance $\mathcal{P}' = (V; A; \mathcal{C}')$ where $A = A_1 \times A_2 \times \cdots \times A_n$, $\mathcal{C}' = \{C'_1, C'_2, \ldots, C'_q\}$, and $C'_i = \langle s_i, \chi(\varrho_i) \rangle$. Note that every solution to $\mathcal{P}$ can be used to construct a solution for $\mathcal{P}'$ by extending the value assigned to each variable (arbitrarily) to a tuple over A. Conversely, every solution to $\mathcal{P}'$

can be used to obtain a solution for $\mathcal{P}$ by projecting the tuple over A assigned to each variable v onto the co-ordinate given by $\delta(v)$. Hence we have a reduction from $\mathrm{MCSP}(\Gamma)$ to $\mathrm{CSP}(\chi(\Gamma))$.

Furthermore, every tuple in every constraint relation of $\mathcal{P}$ is replaced by a fixed number of tuples (depending only on the cardinalities of the sets $A_1, \ldots, A_n$ and on the arities of the constraint relations), to obtain the corresponding constraint relation in $\mathcal{P}'$. Hence, for any finite subset of Γ the reduction can be carried out in linear time.

The same arguments can be applied in the reverse direction to obtain a polynomial-time reduction from $\mathrm{CSP}(\chi(\Gamma))$ to $\mathrm{MCSP}(\Gamma)$. □

3 Polymorphisms and Tractability Results

In earlier papers [18, 17, 6] it has been shown that in the one-sorted case the complexity of $\mathrm{CSP}(\Gamma)$ is determined by certain algebraic properties of the relations in Γ, known as *polymorphisms*. In this section, we first state some of these earlier results on the complexity of the one-sorted case, and then show how they can be extended to the multi-sorted case.

Definition 6. *Let ϱ be a one-sorted relation over a set A, with arity m.*

The operation $f : A^k \to A$ is said to be a polymorphism *of the relation ϱ if, for any tuples $(a_{11}, \ldots, a_{m1}), \ldots, (a_{1k}, \ldots, a_{mk}) \in \varrho$ the tuple $(f(a_{11}, \ldots, a_{1k}), \ldots, f(a_{m1}, \ldots, a_{mk}))$ also belongs to ϱ.*

For any given set of one-sorted relations Γ, the set of all those operations which are polymorphisms of *every* relation in Γ is denoted $\mathrm{Pol}(\Gamma)$.

Theorem 1 ([18, 16]). *Let Γ, Γ_0 be sets of one-sorted relations over a finite set A. If Γ_0 is finite, and $\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\Gamma_0)$, then there is a polynomial time reduction from $\mathrm{CSP}(\Gamma_0)$ to $\mathrm{CSP}(\Gamma)$.*

Furthermore, certain simple forms of polymorphism have been shown to be sufficient to ensure tractability of the associated one-sorted relations [18, 17].

Definition 7.
- *An operation f is called a* constant *operation if there is some fixed $c \in A$ such that $f(x_1, \ldots, x_n) = c$ for all $x_1, \ldots, x_n$.*
- *A binary operation f is called a* semilattice *operation*[2] *if it satisfies the following three identities: $f(x, f(y, z)) = f(f(x, y), z)$; $f(x, y) = f(y, x)$; $f(x, x) = x$.*
- *An n-ary operation f is called a* near-unanimity *operation if $f(y, x, \ldots, x) = f(x, y, x, \ldots, x) = \cdots = f(x, \ldots, x, y) = x$ for all $x, y \in A$.*
- *A ternary operation f is called an* affine *operation if $f(x, y, z) = x - y + z$ for all $x, y, z \in A$, where $(A, +, -)$ is an Abelian group.*

Proposition 2 ([18, 17]). *Let Γ be a set of one-sorted relations over a finite set A.*

If $\mathrm{Pol}(\Gamma)$ contains either a constant operation, or a semilattice operation, or a near-unanimity operation, or an affine operation, then $\mathrm{CSP}(\Gamma)$ is tractable.

[2] Note that in some earlier papers [18, 16] the term *ACI operation* is used

To extend the above results to the multi-sorted case, we need to define a suitable extension of the notion of a polymorphism. As we have shown in the previous section (see Example 5), we cannot simply separate out different domains and consider polymorphisms on each one separately; we must ensure that all of the domains are treated in a co-ordinated way. In the following definition, this is achieved by defining different *interpretations* for the same operation applied to different sets.

Definition 8. *Let $\mathcal{A}$ be a collection of sets. An n-ary* multi-sorted operation t *on $\mathcal{A}$ is defined by a collection of* interpretations $\{t^A \mid A \in \mathcal{A}\}$*, where each t^A is an n-ary operation on the corresponding set A. The multi-sorted operation t on $\mathcal{A}$ is said to be a* polymorphism *of a multi-sorted relation ϱ over $\mathcal{A}$ with signature $(\delta(1), \ldots, \delta(m))$ if, for any $(a_{11}, \ldots, a_{m1}), \ldots, (a_{1n}, \ldots, a_{mn}) \in \varrho$, we have*

$$t\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} = \begin{pmatrix} t^{A_{\delta(1)}}(a_{11}, \ldots, a_{1n}) \\ \vdots \\ t^{A_{\delta(m)}}(a_{m1}, \ldots, a_{mn}) \end{pmatrix} \in \varrho.$$

For any given set of multi-sorted relations Γ, the set of all those multi-sorted operations which are polymorphisms of *every* relation in Γ is denoted $\mathrm{MPol}(\Gamma)$.

The next theorem is the main result of this section. It establishes the remarkable fact that the known one-sorted tractable classes listed in Proposition 2 can be combined in almost arbitrary ways to obtain new multi-sorted tractable classes.

Note that a multi-sorted operation, t, is said to be *idempotent* if all of its interpretations t^A satisfy the identity $t^A(x, x, \ldots, x) = x$.

Theorem 2. *Let Γ be a set of multi-sorted relations over a collection of finite sets $\mathcal{A} = \{A_1, \ldots, A_n\}$.*

If, for each $A_i \in \mathcal{A}$, $\mathrm{MPol}(\Gamma)$ contains a multi-sorted operation f_i such that

- *$f_i^{A_i}$ is a constant operation; or*
- *$f_i^{A_i}$ is a semilattice operation; or*
- *$f_i^{A_i}$ is a near-unanimity operation; or*
- *f_i is idempotent and $f_i^{A_i}$ is an affine operation,*

then $\mathrm{MCSP}(\Gamma)$ is tractable.

The proof of Theorem 2 is set out in Propositions 3, 4, and 5.

Before giving these proofs, we need to give a precise definition for the notion of k-consistency, which is widely used in the study of constraint satisfaction problems, but is unfortunately defined in the literature in a number of slightly different ways.

For an n-ary relation ϱ and a set of indices $I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$, the relation $\mathrm{pr}_I \varrho = \{(a_{i_1}, \ldots, a_{i_k}) \mid (a_1, \ldots, a_n) \in \varrho\}$ is called the *projection* of ϱ onto I. (It will sometimes be convenient to abuse this notation by using the variables from a constraint scope as indices of the corresponding constraint relation.)

Definition 9. *Let $\mathcal{P} = (V; \mathcal{A}; \delta; \mathcal{C})$ be an instance of a multi-sorted constraint satisfaction problem. For any subset W of V, the* subproblem of $\mathcal{P}$ generated by W, *denoted $\mathcal{P}|_W$, is defined to be the problem instance $(W; \mathcal{A}; \delta|_W; \mathcal{C}')$, where the constraints $\mathcal{C}'$ are obtained from the constraints of $\mathcal{P}$ as follows: for each constraint $\langle s, \varrho\rangle \in \mathcal{C}$ of $\mathcal{P}$, choose $s' = s \cap W$ to be a list of those elements of W occurring in s, and set $\langle s', \mathrm{pr}_{s'} \varrho\rangle$ as a constraint of $\mathcal{P}|_W$.*

Definition 10. *For any $k \geq 2$, a constraint satisfaction problem $\mathcal{P}$ is said to be* k-consistent *if for any subset W containing $k-1$ variables, and any variable v, any solution to $\mathcal{P}|_W$ can be extended to a solution to $\mathcal{P}|_{W\cup\{v\}}$.*

If $\mathcal{P}$ is i-consistent for $1 \leq i \leq k$, then it is said to be strong k-consistent.

Any constraint satisfaction problem instance $\mathcal{P}$ can be modified to obtain a k-consistent problem instance $\mathcal{P}'$ without changing the set of solutions, by solving all subproblems involving k variables, and then imposing additional constraints on all subsets of $k-1$ variables that allow only these solutions. This procedure is called 'establishing k-consistency', and $\mathcal{P}'$ is said to be the k-consistent instance associated with $\mathcal{P}$ (see [8] for the one-sorted case).

Definition 11. *A class* **C** *of constraint satisfaction problems is said to be of* essential width k *if any problem instance $\mathcal{P}$ from* **C** *has a solution if and only if the k-consistent problem associated with $\mathcal{P}$ contains no empty constraint.*

Note that Feder and Vardi [13] introduced a very similar notion of width, which they characterised in terms of Datalog programs.

Every class of problems with finite essential width is tractable, because, for any fixed k, establishing k-consistency takes polynomial time, and recognising the presence of empty constraints can be carried out in linear time.

Proposition 3. *Let Γ be a set of multi-sorted relations over $\mathcal{A} = \{A_1, \ldots, A_n\}$. If, for each $A_i \in \mathcal{A}$,* $\mathrm{MPol}(\Gamma)$ *contains a multi-sorted operation f_i such that $f_i^{A_i}$ is either a semilattice operation or a near-unanimity operation, then* $\mathrm{MCSP}(\Gamma)$ *has finite essential width, and is, therefore, tractable.*

A proof of Proposition 3 can be derived from the results of [13, 17, 18].

The next proposition deals with the most involved part of the proof of Theorem 2, because it exploits the subtle interaction between *affine* operations, and the operations considered in Proposition 3, above. To state this proposition, we first need to define certain sets of multi-sorted relations which can be associated with a given set of multi-sorted relations.

Definition 12. *Let Γ be a set of multi-sorted relations over a collection of sets $\mathcal{A}$.*

- *The set of all multi-sorted relations over $\mathcal{A}$ which have the same multi-sorted polymorphisms as Γ will be denoted $\langle\Gamma\rangle$.*
- *For any subset $\mathcal{B}$ of $\mathcal{A}$, the set of all multi-sorted relations in Γ which are multi-sorted relations over $\mathcal{B}$ will be denoted $\Gamma_{\mathcal{B}}$.*

Proposition 4. *Let Γ be a set of multi-sorted relations over $\mathcal{A} = \{A_1, \dots, A_n\}$. If* $\mathrm{MCSP}(\langle\Gamma\rangle_{A_{l+1},\dots,A_n})$ *is of finite essential width, and for each $A_i \in \mathcal{A}$ with $i \leq l$,* $\mathrm{MPol}(\Gamma)$ *contains an idempotent multi-sorted operation g_i such that $g_i^{A_i}$ is affine, then* $\mathrm{MCSP}(\Gamma)$ *is tractable.*

Proof: (sketch) Due to space restrictions, we can only give a very brief outline here of a polynomial-time algorithm for $\mathrm{MCSP}(\Gamma)$.

Let $\mathcal{P}$ be any instance of $\mathrm{MCSP}(\Gamma)$. The variables of $\mathcal{P}$ can be split into two parts, those with domains in $A_1, \dots, A_l$ (the "affine" part), and those with domains in $A_{l+1}, \dots, A_n$ (the "finite width" part).

Consider first the affine part. The conditions of Proposition 4 mean, in particular, that, for $i \leq l$, $g_i^{A_i}(x,y,z) = x -_i y +_i z$ where $+_i, -_i$ are the operations of an Abelian group on the base set A_i. It can be shown that in this case $\mathrm{MPol}(\Gamma)$ also contains a single ternary idempotent operation d such that $d^{A_i}(x,y,z) = g_i^{A_i}(x,y,z)$ for $i = 1, 2, \dots, l$. It follows that $\mathrm{Pol}(\chi(\langle\Gamma\rangle_{A_1,\dots,A_l}))$ contains an affine operation, and hence, any problem instance from $\mathrm{CSP}(\chi(\langle\Gamma\rangle_{A_1,\dots,A_l}))$ is solvable in polynomial time, by an algorithm similar to Gaussian elimination. The same is true for the class $\mathrm{MCSP}(\langle\Gamma\rangle_{A_1,\dots,A_l})$. Moreover, this solution algorithm can be modified to efficiently compute a *basis* of the solution space; in other words, to find a representation for the complete set of solutions which is polynomial in the size of the problem.

Now consider the finite width part. If we restrict $\mathcal{P}$ to these variables, then we need only consider subproblems of size at most k, where k is the essential width of $\mathrm{CSP}(\langle\Gamma\rangle_{A_{l+1},\dots,A_n})$. For each solution to such a bounded-size subproblem we can efficiently compute a basis for the possible extensions of that solution to the affine part, as described above. Combining these basis sets, we can find a basis for the complete set of possible solutions to the affine part, and check if it is empty. $\square$

Proposition 5. *Let Γ be a set of multi-sorted relations over $\mathcal{A} = \{A_1, \dots, A_n\}$. If* $\mathrm{MCSP}(\langle\Gamma\rangle_{A_{l+1},\dots,A_n})$ *is tractable, and for each $A_i \in \mathcal{A}$ with $i \leq l$,* $\mathrm{MPol}(\Gamma)$ *contains a multi-sorted operation g_i such that $g_i^{A_i}$ is constant, then* $\mathrm{MCSP}(\Gamma)$ *is tractable.*

Proof: Assume that $\mathrm{MPol}(\Gamma)$ contains a multi-sorted operation g_i, for $i = 1, 2, \dots, l$, such that $g_i^{A_i}$ is constant. Note that in this case the unary operation $g_i'(x) = g_i(x, \dots, x)$ also belongs to $\mathrm{MPol}(\Gamma)$. Hence, the operation $g(x) = g_1'(\dots g_l'(x) \dots)$ belongs to $\mathrm{MPol}(\Gamma)$ and is constant for all $A_1, \dots, A_l$. Denote the constant value of g^{A_i} by c_i.

Take a problem instance $\mathcal{P} = (V; \{A_1, \dots, A_n\}; \delta; \mathcal{C})$ of $\mathrm{MCSP}(\Gamma)$, and set $W = \{v \in V \mid \delta(v) > l\}$. Then the instance $\mathcal{P}|_W$ belongs to $\mathrm{MCSP}(\langle\Gamma\rangle_{A_{l+1},\dots,A_n})$, and therefore is tractable, by assumption. Solve $\mathcal{P}|_W$. If it has no solution then neither does $\mathcal{P}$. Otherwise, let φ be a solution. We claim that the mapping $\psi\colon V \to A_1 \cup \dots \cup A_n$ defined as follows

$$\psi(v) = \begin{cases} c_{\delta(v)}, & \text{if } \delta(v) \leq l, \\ g^{A_{\delta(v)}}(\varphi(v)), & \text{otherwise} \end{cases}$$

is a solution to $\mathcal{P}$.

To establish this, note that for any constraint $C = \langle s, \varrho \rangle \in \mathcal{C}$ there is a tuple $\overline{a} \in \varrho$ such that $\overline{a}[v] = \varphi(v)$ for $v \in s \cap W$. Setting $\overline{b} = g(\overline{a}) \in \varrho$ we get $\overline{b}[v] = c_{\delta(v)} = \psi(v)$ if $\delta(v) \leq l$, and $\overline{b}[v] = g^{A_{\delta(v)}}(\varphi(v)) = \psi(v)$ otherwise. Thus ψ satisfies every constraint in $\mathcal{C}$, and hence is a solution to $\mathcal{P}$. $\square$

Example 8. Recall the relation ϱ over the sets $A_1 = \{0, 1, 2, 3\}$ and $A_2 = \{a, b, c\}$, defined in Example 1. We can now prove that $\{\varrho\}$ is tractable. To see this, it is sufficient to check that ϱ has two multi-sorted polymorphisms $t(x, y, z)$ and $g(x, y)$, where

- t^{A_1} is the affine operation of the group $\mathbb{Z}_4$, and t^{A_2} is the (ternary) maximum operation on A_2, with respect to the order $a < b < c$ (which is idempotent).
- $g^{A_1}(x, y) = y$, and g^{A_2} is the (binary) maximum operation on A_2, with respect to the order $a < b < c$ (which is a semilattice operation).

Hence we can apply Theorem 2, and conclude that $\mathrm{MCSP}(\{\varrho\})$ is tractable.

4 From Polymorphisms to Algebras

Polymorphisms provide a powerful tool for studying the constraint satisfaction problem. However, as was observed in [6], we get an even more powerful tool if we consider the set of polymorphisms along with the set on which they are defined.

Definition 13. *An* algebra *is an ordered pair $(A; F)$, where A is a nonempty set and F is a family of finitary operations on A. The set A is called the* universe *(or the* base set*), and the operations from F are called* basic. *An algebra is said to be* finite *if its universe is finite.*

The advantage of working explicitly with algebras, rather than just sets of polymorphisms, is that one can exploit the well-developed mathematical theory of algebras. This idea was pursued for the one-sorted case in [6]. Here we briefly summarise the relevant aspects of the one-sorted case, and then consider how the link with algebras can be extended to the multi-sorted case.

First we observe that we can associate any set Γ of one-sorted relations with a corresponding algebra $\mathbb{A}_\Gamma = (A; \operatorname{Pol} \Gamma)$. In the reverse direction, we can start with an algebra and obtain a corresponding set of one-sorted relations, in the following way. Given any set of operations, C, on a set A, the set of all relations over A for which all operations from C are polymorphisms is denoted $\operatorname{Inv}(C)$. Hence, given any algebra $(A; F)$, we can define an associated set of one-sorted relations $\operatorname{Inv}(F)$. We will say that an algebra $(A; F)$ is *tractable* if the associated set of one-sorted relations $\operatorname{Inv}(F)$ is tractable.

Using the definitions above, and Theorem 1, we can translate questions about the tractability of a set of one-sorted relations into questions about the tractability of the corresponding algebra.

Corollary 1. *Let Γ be a set of one-sorted relations over a finite set A. The set Γ is tractable if and only if the corresponding algebra $\mathbb{A}_\Gamma$ is tractable.*

It follows from Corollary 1 that all maximal tractable sets of one-sorted relations can be defined by specifying a suitable algebra. Of course, we may need very many operations to define an arbitrary algebra. However, in all known cases, including those listed in Proposition 2 and in papers [1–3, 5, 10, 11], we need very few operations to ensure that an algebra is tractable. Hence the algebraic theory we have sketched here allows a very concise description of all known maximal tractable sets of one-sorted relations, even though each maximal tractable set contains infinitely many relations.

One simple way to extend these ideas to the multi-sorted case is to combine Corollary 1 with Proposition 1, as follows.

Corollary 2. *Let Γ be a set of multi-sorted relations over the collection of finite sets $\{A_1, \ldots, A_n\}$. The set Γ is tractable if and only if the corresponding algebra $\mathbb{A}_{\chi(\Gamma)}$ is tractable.*

However, this simple extension is rather unsatisfactory, because of the potentially large size of the universe of $\mathbb{A}_{\chi(\Gamma)}$. For example, if we have 10 distinct domains each with 20 elements then $\mathbb{A}_{\chi(\Gamma)}$ is an algebra with 20^{10} elements. In the remainder of this section we are going to show that, for any set of multi-sorted relations over $\{A_1, \ldots, A_n\}$, we can define a collection of algebras $\{\mathbb{A}_1, \ldots, \mathbb{A}_n\}$, where the universe of each $\mathbb{A}_i$ is the set A_i, and the complexity of the multi-sorted problem is determined by this collection of smaller algebras. Using these results for the case just mentioned, we may deal with 10 individual 20-element algebras, which is much more convenient, and allows a more complete analysis of the algebraic structure.

To develop this novel theory concerning the tractability of *collections* of algebras we need to introduce a little more standard algebraic terminology. First we note that algebras can be grouped into families which share the same set of basic operations.

Definition 14. *The collection of algebras $\boldsymbol{A} = \{(A_i; F^{A_i}) \mid i \in I\}$ is said to be a collection of* similar *algebras if there exists some fixed set F of multi-sorted operations over the sets $\{A_i \mid i \in I\}$, such that each set of basic operations F^{A_i} is the set of interpretations of the functions in F on the set A_i. The set F is called the set of basic operations of the collection $\boldsymbol{A}$.*

Definition 15. *For any collection of similar algebras $\boldsymbol{A} = \{(A_i; F^{A_i}) \mid i \in I\}$ with basic operations F, we define* $\mathrm{MInv}(\boldsymbol{A})$ *to be the set Γ of all multi-sorted relations over the sets $\{A_i \mid i \in I\}$ such that* $\mathrm{MPol}(\Gamma) = F$.

A collection of algebras $\mathbf{A}$ will be called tractable if the set of multi-sorted relations $\mathrm{MInv}(\mathbf{A})$ is tractable. Similarly, $\mathbf{A}$ will be called NP-complete if $\mathrm{MInv}(\mathbf{A})$ is NP-complete.

The following theorem shows that, for any set Γ of multi-sorted relations over a finite collection of finite sets, $\mathcal{A}$, there exists a finite algebra $\mathbb{A}$, and a collection $\mathbf{A}$ of similar algebras whose universes are the members of $\mathcal{A}$, such that Γ, $\mathbb{A}$, and $\mathbf{A}$ are all tractable or intractable simultaneously.

Definition 16. *Let $\mathbb{A}_1$ and $\mathbb{A}_2$ be similar algebras with universes A_1, A_2 and basic operations F. A mapping $\varphi\colon A_1 \to A_2$ is called a* homomorphism *from $\mathbb{A}_1$ to $\mathbb{A}_2$ if $\varphi f^{\mathbb{A}_1}(a_1, \ldots, a_k) = f^{\mathbb{A}_2}(\varphi(a_1), \ldots, \varphi(a_k))$ for all $f \in F$ and all $a_1, \ldots, a_k \in A_1$, where k is the arity of f.*

If the map φ is surjective, then $\mathbb{A}_2$ is called a homomorphic image *of $\mathbb{A}_1$.*

Theorem 3. *Let Γ be a set of multi-sorted relations over the finite sets $\{A_1, \ldots, A_n\}$. The following are equivalent:*

(a) Γ is tractable;
(b) $\mathbb{A}_{\chi(\widehat{\Gamma})}$ is tractable, where $\widehat{\Gamma} = \Gamma \cup \{=_{A_1}, \ldots, =_{A_n}\}$, and each $=_{A_i}$ is the binary equality relation on A_i;
(c) $\{\mathbb{A}_1, \ldots, \mathbb{A}_n\}$ is tractable, where each $\mathbb{A}_i$ is the image of $\mathbb{A}_{\chi(\widehat{\Gamma})}$ under the homomorphism φ_i given by $\varphi_i(\overline{a}) = \overline{a}[i]$.

A corresponding equivalence also holds when "tractable" is replaced by "NP-complete".

Proof: Omitted, see [4]. □

5 Applications to One-Sorted Problems

As well as providing a sound framework for the analysis of the complexity of multi-sorted constraint satisfaction problems, the results developed in this paper have some surprising applications to the one-sorted case. (For example, see [1].)

We will complete this paper by describing an application of the results presented above to the analysis of the complexity of one-sorted constraint satisfaction problems in which the constraints limit each variable to at most two possible values (which may be different for different variables). In other words, we consider sets of one-sorted relations relations Γ such that for each $\varrho \in \Gamma$, each unary projection $\mathrm{pr}_i\varrho$ contains at most two elements. This problem was previously considered in [12], which established a dichotomy theorem for the complexity of such sets of relations: they are either tractable or NP-complete.

We generalize this result by showing that for slightly more general sets of relations Γ, the one-sorted problem class $\mathrm{CSP}(\Gamma)$ is polynomial-time equivalent to a certain multi-sorted constraint satisfaction problem, and then applying Theorem 2 to give a complete characterisation of the tractable cases.

Definition 17. *Let Γ be a set of one-sorted relations.*

For any relation $\varrho \in \Gamma$, the set $\Delta(\varrho)$ is defined to be the set of all unary projections of ϱ, and the set $\Delta(\Gamma)$ is defined to be the union of the sets $\Delta(\varrho)$ over all non-unary $\varrho \in \Gamma$.

Definition 18. *Let Γ be a set of one-sorted relations where $\Delta(\Gamma) = \{A_1, \ldots, A_k\}$.*

For any relation $\varrho \in \Gamma$ with arity n, the set Γ_ϱ is defined to be the set of all multi-sorted relations over $\Delta(\Gamma)$ of the form $\varrho \cap (A_{i_1} \times \ldots \times A_{i_n})$ with signature $(i_1, \ldots, i_n)$, for all possible choices of $i_1, \ldots, i_n$.

The set Γ^+ is defined to be the union of the sets Γ_ϱ over all non-unary $\varrho \in \Gamma$.

Proposition 6. *Let Γ be a finite set of one-sorted relations.*

The one-sorted problem $\mathrm{CSP}(\Gamma)$ *is polynomial-time equivalent to the multi-sorted problem* $\mathrm{MCSP}(\Gamma^+)$.

Proof: First we reduce $\mathrm{CSP}(\Gamma)$ to $\mathrm{MCSP}(\Gamma^+)$. Consider any problem instance $\mathcal{P} = (V; A; \mathcal{C}) \in \mathrm{CSP}(\Gamma)$. Let U be the set of variables constrained by unary constraints only. For any $v \in U$, if the intersection of the unary constraints imposed on v is empty, then $\mathcal{P}$ has no solution. Otherwise, we can assign v with an arbitrary value from this intersection. Hence, we may assume that $U = \varnothing$.

Let $\Delta(\Gamma) = \{A_1, A_2, \ldots, A_k\}$. For each $v \in V$, let $\langle s_v, \varrho_v \rangle \in \mathcal{C}$ be a non-unary constraint whose scope contains v. The set $\mathrm{pr}_v \varrho_v$ is equal to some element $A_i \in \Delta(\Gamma)$; set $\delta(v)$ equal to i. It is not hard to see that $\mathcal{P}$ is equivalent to the multi-sorted problem $\mathcal{P}' = (V; \{A_1, \ldots, A_k\}; \delta; \mathcal{C}')$ where $\mathcal{C}'$ contains the constraint $\langle s, \varrho \cap \prod_{v \in s} A_{\delta(v)} \rangle$ for each $\langle s, \varrho \rangle \in \mathcal{C}$.

To show the converse reduction, we notice that Γ^+ can be viewed as a set of one-sorted relations over the set A. Moreover, every multi-sorted problem instance in $\mathrm{MCSP}(\Gamma^+)$ can be viewed as a one-sorted instance. Hence, we have a trivial reduction from $\mathrm{MCSP}(\Gamma^+)$ to $\mathrm{CSP}(\Gamma^+)$. By using well-known basic properties of polymorphisms (see, for example, [23]), any $f \in \mathrm{Pol}(\Gamma)$ is also a polymorphism of all unary relations A_i, all Cartesian products of the form $A_{i_1} \times \ldots \times A_{i_l}$, and all relations of the form $\varrho \cap (A_{i_1} \times \ldots \times A_{i_l})$, for any $\varrho \in \Gamma$. Hence, $\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\Gamma^+)$, so $\mathrm{CSP}(\Gamma^+)$ is polynomial-time reducible to $\mathrm{CSP}(\Gamma)$, by Theorem 1. □

Corollary 3. *Let Γ be a set of one-sorted relations over a finite set, such that $|A_i| \leq 2$ for each $A_i \in \Delta(\Gamma)$.*

If, for each $A_i \in \Delta(\Gamma)$ there is an operation $f_i \in \mathrm{MPol}(\Gamma^+)$ such that $f_i^{A_i}$ is either a constant operation, or a semilattice operation, or a near-unanimity operation, or an affine operation, then $\mathrm{CSP}(\Gamma)$ *is tractable. Otherwise, it is NP-complete.*

Proof: Let $\mathbb{A}_1, \ldots, \mathbb{A}_k$ be the algebras corresponding to Γ^+ as defined in Theorem 3. If the conditions of the corollary hold, then $\Gamma^+ \subseteq \mathrm{MInv}(\mathbb{A}_1, \ldots, \mathbb{A}_k)$ is tractable, by Theorem 2. (The requirement in Theorem 2 for affine operations to be idempotent can be shown to be unnecessary when all sets contain at most 2 elements, by a careful examination of the possible cases.)

Conversely, if for a certain A_i there is no operation with the required properties then, by the dichotomy theorem for one-sorted constraints on a two-element set [24, 6], $\mathrm{MInv}(\{\mathbb{A}_i\}) \subseteq \mathrm{MInv}(\{\mathbb{A}_1, \ldots, \mathbb{A}_k\})$ is NP-complete. Hence, by Theorem 3, Γ^+ is also NP-complete, and therefore Γ is NP-complete, by Proposition 6. □

6 Conclusion

Practical constraint satisfaction problems often involve different domains for different variables. We have shown in this paper that analyzing the complexity of such problems by simply assuming that all these domains are subsets of some single large domain can give a misleading picture of their complexity. To remedy this problem, we have presented a set of algebraic tools that allow one to deal more accurately with such multi-sorted problems.

Using these tools we have been able to show that certain existing polynomial-time algorithms for the one-sorted case can be combined, to give a more powerful polynomial-time algorithm, solving much wider classes of multi-sorted problems.

Finally, we have further investigated the significant link between the study of the constraint satisfaction problem and the study of finite algebras, by extending this link to the multi-sorted case. Elements of this extended algebraic machinery have already proved to be essential tools in the study of the complexity of the conventional one-sorted constraint satisfaction problem. We therefore believe that the theory developed here will lead to a deeper understanding of the structure of both one-sorted and multi-sorted constraint satisfaction problems.

References

1. A.A. Bulatov. A dichotomy theorem for constraints on a three-element set. In *Proceedings 43rd IEEE Symposium on Foundations of Computer Science, FOCS'02*, pages 649–658, Vancouver, Canada, November 2002.
2. A.A. Bulatov. Mal'tsev constraints are tractable. Technical Report PRG-RR-02-05, Computing Laboratory, University of Oxford, Oxford, UK, 2002.
3. A.A. Bulatov and P.G. Jeavons. Tractable constraints closed under a binary operation. Technical Report PRG-TR-12-00, Computing Laboratory, University of Oxford, Oxford, UK, 2000.
4. A.A. Bulatov and P.G. Jeavons. Algebraic approach to multi-sorted constraints. Technical Report PRG-RR-01-18, Computing Laboratory, University of Oxford, Oxford, UK, 2001.
5. A.A. Bulatov, P.G. Jeavons, and A.A. Krokhin. The complexity of maximal constraint languages. In *Proceedings of the 33rd Annual ACM Simposium on Theory of Computing*, pages 667–674, Hersonissos, Crete, Greece, July 2001. ACM Press.
6. A.A. Bulatov, A.A. Krokhin, and P.G. Jeavons. Constraint satisfaction problems and finite algebras. In *Proceedings of 27th International Colloquium on Automata, Languages and Programming—ICALP'00*, volume 1853 of *Lecture Notes in Computer Science*, pages 272–282. Springer-Verlag, 2000.
7. D.A. Cohen, P.G. Jeavons, P. Jonsson, and M. Koubarakis. Building tractable disjunctive constraints. *Journal of the ACM*, 47:826–853, 2000.
8. M.C. Cooper. An optimal k-consistency algorithm. *Artificial Intelligence*, 41:89–95, 1989.
9. M.C. Cooper, D.A. Cohen, and P.G. Jeavons. Characterising tractable constraints. *Artificial Intelligence*, 65:347–361, 1994.
10. V. Dalmau. A new tractable class of constraint satisfaction problems. In *Proceedings 6th International Symposium on Artificial Intelligence and Mathematics*, 2000.

11. V. Dalmau and J. Pearson. Set functions and width 1 problems. In *Proceedings 5th International Conference on Constraint Programming, CP'99*, volume 1713 of *Lecture Notes in Computer Science*, pages 159–173. Springer-Verlag, 1999.
12. T. Feder. Classification of homomorphisms to oriented cycles and of k-partite satisfiability. *SIAM J. of Discrete Math.*, 14(4):471–480, 2001.
13. T. Feder and M.Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM Journal of Computing*, 28:57–104, 1998.
14. M. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, CA., 1979.
15. D. Hobby and R.N. McKenzie. *The Structure of Finite Algebras*, volume 76 of *Contemporary Mathematics*. American Mathematical Society, Providence, R.I., 1988.
16. P.G. Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.
17. P.G. Jeavons, D.A. Cohen, and M.C. Cooper. Constraints, consistency and closure. *Artificial Intelligence*, 101(1-2):251–265, 1998.
18. P.G. Jeavons, D.A. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44:527–548, 1997.
19. L. Kirousis. Fast parallel constraint satisfaction. *Artificial Intelligence*, 64:147–160, 1993.
20. Ph.G. Kolaitis and M.Y. Vardi. Conjunctive-query containment and constraint satisfaction. *J. Comput. Syst. Sci.*, 61:302–332, 2000.
21. R.N. McKenzie, G.F. McNulty, and W.F. Taylor. *Algebras, Lattices and Varieties*, volume I. Wadsworth and Brooks, California, 1987.
22. U. Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences*, 7:95–132, 1974.
23. R. Pöschel and L.A. Kalužnin. *Funktionen- und Relationenalgebren*. DVW, Berlin, 1979.
24. T.J. Schaefer. The complexity of satisfiability problems. In *Proceedings 10th ACM Symposium on Theory of Computing (STOC'78)*, pages 216–226, 1978.
25. P. van Beek and R. Dechter. On the minimality and decomposability of row-convex constraint networks. *Journal of the ACM*, 42:543–561, 1995.
26. P. van Hentenryck, Y. Deville, and C-M. Teng. A generic arc-consistency algorithm and its specializations. *Artificial Intelligence*, 57:291–321, 1992.
27. M.Y. Vardi. Constraint satisfaction and database theory: a tutorial. In *Proceedings of 19th ACM Symposium on Priciples of Database Systems (PODS'00)*, 2000.